An International Journal of Optimization and Control:
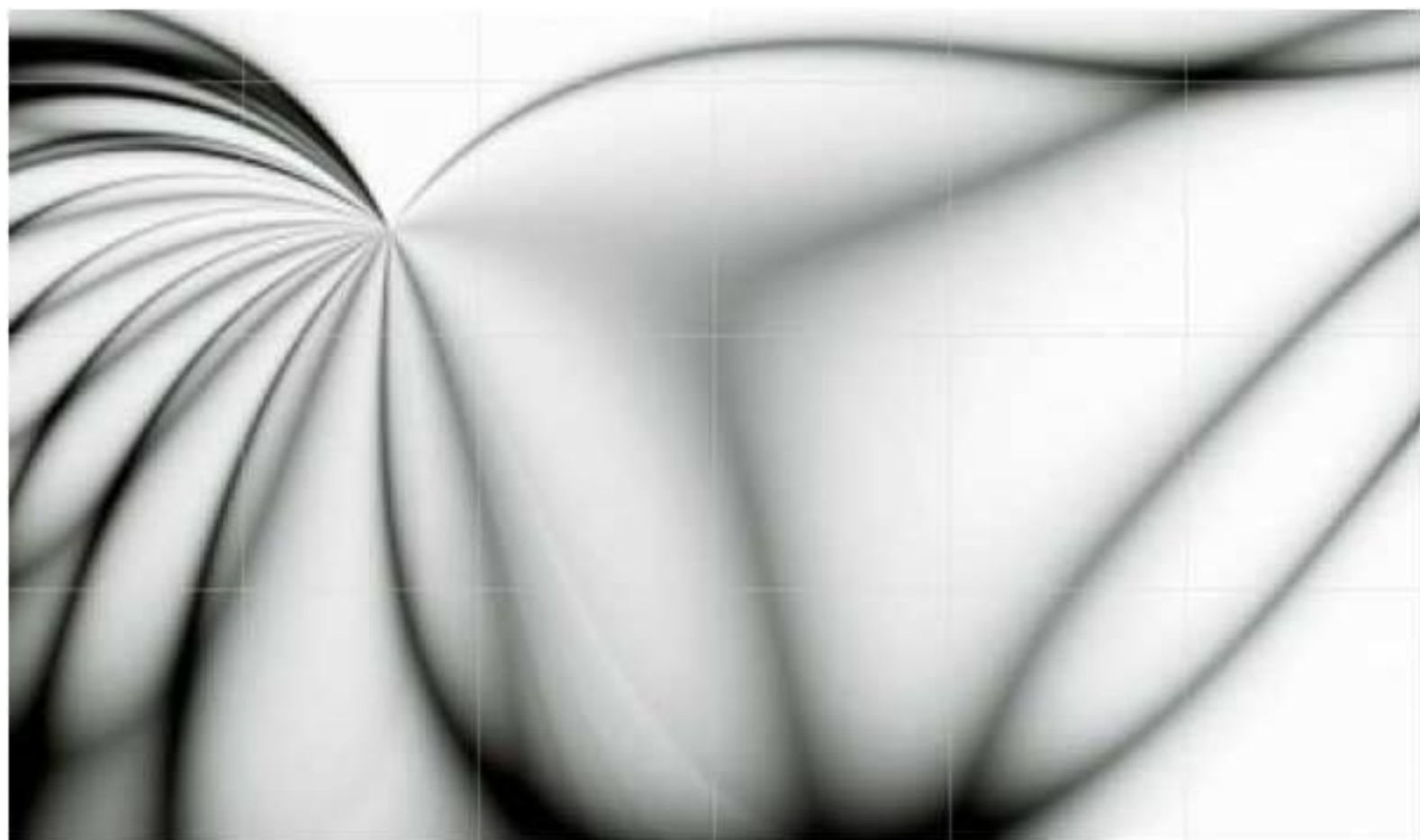
Theories & Applications

An International Journal of Optimization and Control:
Theories & Applications
Volume: 10   Number: 1
January 2020

CONTENTS

RESEARCH ARTICLE

On the new wave behavior of the Magneto-Electro-Elastic(MEE) circular rod longitudinal wave equation

Onur Alp İlhan[*a], Hasan Bulut[b], Tukur A. Sulaiman[b] and Haci Mehmet Baskonus[c]

[a] Erciyes Univeristy, Faculty of Education, Melikgazi-Kayseri, Turkey
[b] Firat University, Department of Mathematics, Elazığ, Turkey
[c] Harran University, Faculty of education, Department of Mathematics, Sanliurfa, Turkey
oailhan@erciyes.edu.tr, hbulut@firat.edu.tr, mtukur74@yahoo.com, hmbaskonus@gmail.com

ARTICLE INFO	ABSTRACT
	The analytical solution of the longitudinal wave equation in the MEE circular rod is analyzed by the powerful sine-Gordon expansion method. Sine - Gordon expansion is based on the well-known wave transformation and sine - Gordon equation. In the longitudinal wave equation in mathematical physics, the transverse Poisson MEE circular rod is caused by the dispersion. Some solutions with complex, hyperbolic and trigonometric functions have been successfully implemented. Numerical simulations of all solutions are given by selecting the appropriate parameter values. The physical meaning of the analytical solution explaining some practical physical problems is given.

1. Introduction

Innovative analytical new solutions for non-linear evolution equations (NEEs) has very important role in area of non-linear physics. Non-linear evolution equations are often used to state complex models that appear in different areas of non-linear science, such as biological sciences, quantum mechanics, and plasma physics. Recently, different analytical techniques have been invested to search new types of solutions. NLEs such as the new general algebra method [1], the $\tan(\frac{F(\xi)}{2})$-expansion method [2], the extended tanh method [3], the jacobi elliptic function method [4], the homogeneous balance method [5], the generalized Kudryashov method [6], the generalized (G'/G) method [7], the extended homoclinic test function method [8], the improved Bernoulli sub-equation function method [9], the improved exp $(-\Phi(\xi))$-expansion function method [10] and so on. In general, many more analytical techniques have been designed and used in obtaining analytical solutions of different NLEs [11–22]. Authors of

[23–28] obtained new lump and interaction for some of models in which arise in applied sciences. Moreover, Manafian and co-authors [29, 30] used the analytical methods for getting to exact solutions.

The powerful sine-Gordon expansion method (SGEM) [31, 32] was used to find some new solution methods to the longitudinal wave equation of the magneto-electro-elastic (MEE) circular rod [33] in this study. The longitudinal wave equation of the MEE circular rod is developed by [33], the longitudinal wave equation is a dispersion equation caused by the transverse Poisson's effect in MEE circular rod, developed from [34];

$$u_{tt} - q^2 u_{xx} - \left(\frac{q}{2}u^2 + pu_{tt}\right)_{xx} = 0, \qquad (1)$$

where p is the dispersion parameter and q is the linear longitudinal wave velocity of the MEE circular rod which depend on material properties and rod geometry [34]. Different analytical methods have been put in place to find solutions

---

*Corresponding Author

to the longitudinal wave equation in magneto-electro-elastic MEE circular rod, like the improved (G'/G)-expansion method [35], the functional variable method [36], the ansatz method [37], etc.

2. The SGEM

The general cases of SGEM was given in this section,

Take into account the following sine-Gordon equation [38], [39]:

$$u_{xx} - u_{tt} = n^2 sin(u). \tag{2}$$

where $u = u(x, t)$ and $n \in \mathbb{R} \setminus \{0\}$.

Using the wave transformation $u = u(x, t) = U(\beta)$, $\beta = \alpha(x - kt)$ on Eq. (2), following non-linear ordinary differential equation (NODE) was gotten as:

$$U'' = \frac{n^2}{\alpha^2(1 - k^2)} sin(U), \tag{3}$$

as $U = U(\beta)$, the amplitude of the traveling wave is β and k is the speed of the traveling wave. To integrate the equation (3), we get the following equation:

$$\left[\left(\frac{U}{2}\right)'\right]^2 = \frac{n^2}{\alpha^2(1 - k^2)} sin^2\left(\frac{U}{2}\right) + Q, \tag{4}$$

as the integral constant is Q .

Set $Q = 0$, $\phi(\beta) = \frac{U}{2}$ and $b^2 = \frac{n^2}{\alpha^2(1-k^2)}$ in Eq. (4), gives:

$$\phi' = bsin(\phi), \tag{5}$$

inserting $b = 1$ into Eq. (5), produces:

$$\phi' = sin(\phi), \tag{6}$$

simplifying Eq. (6), creates the following two important equations;

$$sin(\phi) = sin(\phi(\beta)) = \left.\frac{2de^\beta}{d^2e^{2\beta} + 1}\right|_{d=1} = sech(\beta), \tag{7}$$

$$cos(\phi) = cos(\phi(\beta)) = \left.\frac{d^2e^{2\beta} - 1}{d^2e^{2\beta} + 1}\right|_{d=1} = tanh(\beta), \tag{8}$$

as the integral constant is d .

For the given non-linear partial differential equation Eq. (9);

$$P(u, \ uu_x, \ u^2u_t, \ \ldots), \tag{9}$$

its solution in the form as;

$$U(\beta) = \sum_{i=1}^m tanh^{i-1}(\beta)\big[B_i sech(\beta) + A_i tanh(\beta)\big] + A_0. \tag{10}$$

Equation (10) may be given according to Eq. (7) and (8) as;

$$U(\phi) = \sum_{i=1}^m cos^{i-1}(\phi)\big[B_i sin(\phi) + A_i cos(\phi)\big] + A_0. \tag{11}$$

m is determined by balancing the highest power non-linear term and the highest derivative in the transformed NODE. Taking each summation of the coefficients of $sin^i(w)cos^j(w)$, $0 \leq i, j \leq m$ to be zero, produces a set of equations. This set of equation is solved with the symbolic computational computational software, yields the values of the coefficients A_i, B_i, μ and c. Eventually, inserting the produced values of these coefficients into Eq. (10) accompanied by the value of m, gives the fresh travelling wave solutions to Eq. (9).

3. Applications

The SGEM is used in searching the fresh solutions to Eq. (1) in this section. Considering Eq. (1), the following NODE was gotten by using the wave transformation; $u = U(\beta)$, $\beta = \mu(-kt + x)$;

$$2pk^2\mu^2U'' - 2(k^2 - c_0^2)U + c_0^2U^2 = 0, \tag{12}$$

p is non-zero constant and we get $m = 2$ by balancing U'' and U^2 in Eq. (12).

Using Eq. (11) together with the value $m = 2$, we get the following equation;

$$\begin{aligned} U(\phi) = \ &B_1 sin(\phi) + A_1 cos(\phi) + B_2 cos(\phi)sin(\phi) \\ &+ A_2 cos^2(\phi) + A_0, \end{aligned} \tag{13}$$

differentiating Eq. (13) twice, we get:

$$U''(\phi) = B_1 cos^2(\phi)sin(\phi) - B_1 sin^3(\phi)$$
$$- 2A_1 sin^2(\phi)cos(\phi) + B_2 cos^3(\phi)sin(\phi)$$
$$- 5B_2 sin^3(\phi)cos(\phi) - 4A_2 cos^2(\phi)sin^2(\phi)$$
$$+ 2A_2 sin^4(\phi),$$

$$(14)$$

Setting Eq. (13) and (15) to Eq. (12), generating trigonometric equations. After replacing the trigonometric constants in the trigonometric equation, a set of algebraic equations is collected by setting each sum of the coefficients of the trigonometric functions of the same power to zero. The set of equations is solved with assistance of symbolic mathematical softwares; to get coefficient values for different cases. We insert coefficient values for each case into the Eq. (10) with a value of $m = 2$, this gives us a new solution Eq. (1).

Case-1:

$$A_0 = 4(1 + \frac{k^2}{q^2}), A_1 = 0, B_1 = 0, A_2 = -6(1 - \frac{k^2}{q^2}),$$

$$B_2 = -6i + \frac{6ik^2}{q^2}, p = \frac{1}{k^2\mu^2}(k^2 - q^2).$$

Case-2:

$$A_0 = 4 - \frac{4}{1 + p\mu^2}, A_1 = 0, B_1 = 0,$$

$$A_2 = -6 + \frac{6}{1 + p\mu^2}, B_2 = \frac{6p\mu^2(p\mu^2 - 1)}{p^2\mu^4 - 1}i,$$

$$q = -k\sqrt{1 + p\mu^2}.$$

Case-3:

$$A_0 = -6 + \frac{6k^2}{q^2}, A_1 = 0, B_1 = 0, A_2 = 6 - \frac{6k^2}{q^2},$$

$$B_2 = 6i(1 - \frac{k^2}{q^2}), \mu = -\frac{1}{k\sqrt{p}}\sqrt{(k^2 - q^2)}.$$

Case-4:

$$A_0 = 1 + \frac{k^2}{q^2}, A_1 = 0, B_1 = 0, A_2 = -3(1 - \frac{k^2}{q^2}),$$

$$B_2 = 0, p = \frac{k^2 - q^2}{4k^2\mu^2}.$$

Case-5:

$$A_0 = 1 - \frac{1}{4p\mu^2 + 1}, A_1 = 0, B_1 = 0,$$

$$A_2 = -3 + \frac{3}{4p\mu^2 + 1}, B_2 = 0, q = k\sqrt{4p\mu^2 + 1}.$$

Case-6:

$$A_0 = 1 - \frac{k^2}{q^2}, A_1 = 0, B_1 = 0, A_2 = 3(\frac{k^2}{q^2} - 1),$$

$$B_2 = 0, \mu = \frac{1}{2k\sqrt{p}}\sqrt{(k^2 - q^2)}i.$$

Solutions:

(1). The following solution is gotten by with case 1;

$$u_1(x,t) = \frac{6(k^2 - q^2)}{q^2}(1 + i\ sech[\mu(x - kt)]$$
$$\times tanh[\mu(x - kt)] - tanh[(-kt + x)\mu]^2)$$

$$(15)$$
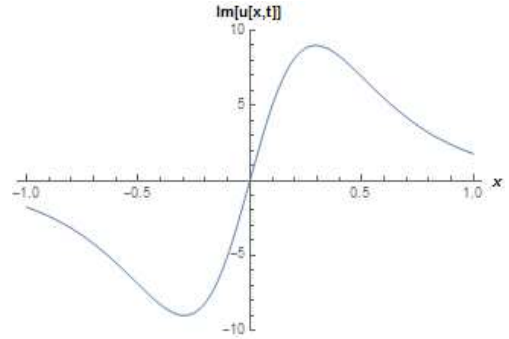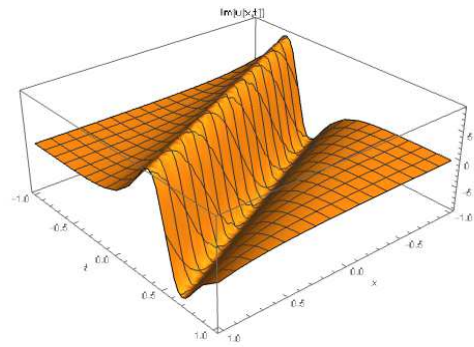
Figure 1. The 3D shape for the imaginary part of Eq. (15) with the values $k = 2$, $c_0 = 1$, $\mu = 3$, $-3 < x < 3$, $-5 < t < 5$ and $t = 0$ for the graphic of 2D.
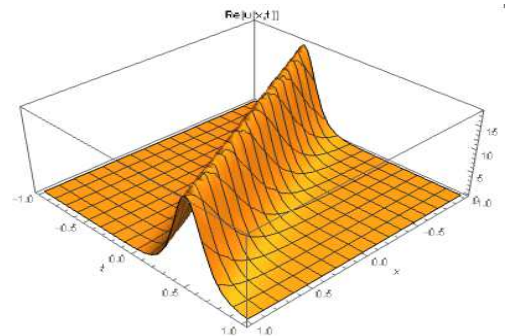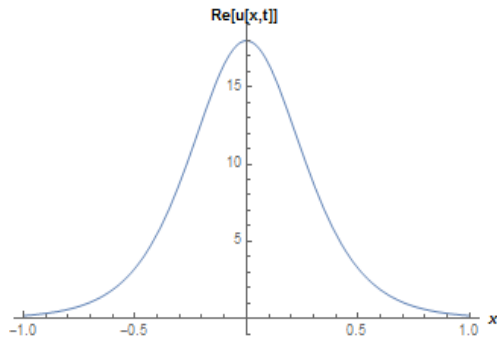
Figure 2. The 3D shape for the real part of Eq. (15) with the values $k = 2$, $c_0 = 1$, $\mu = 3$, $-3 < x < 3$, $-5 < t < 5$ and $t = 0$ for the graphic of 2D.
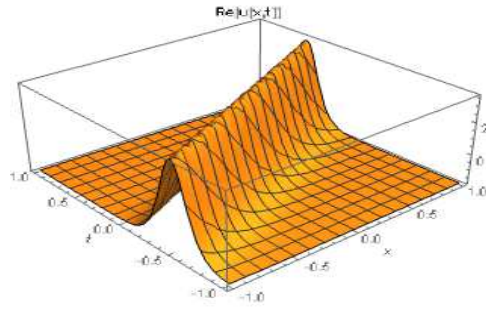
(2). The following solution is gotten by with case 2;

$$u_2(x,t) = 4 - \frac{4}{1 + p\mu^2} + (6ip\mu^2(-1 + p\mu^2)).$$
$$\cdot \frac{\sec h[(-kt + x)\mu]\tanh[-kt + x)\mu]}{-1 + p^2\mu^4}$$
$$+ (-6 + \frac{6}{1 + p^2\mu^2})\tanh[(-kt + x)\mu]^2.$$
(16)
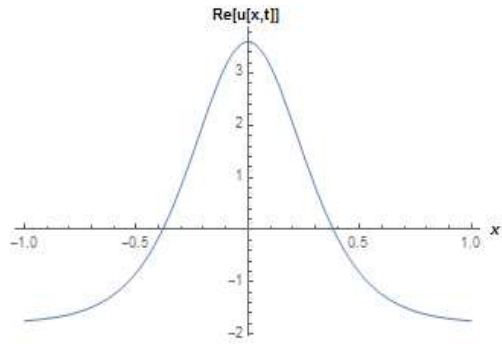
Figure 3. The 2D and 3D shape for the imaginary and real part of Eq. (16) with the values $k = 2$, $p = 1$, $\mu = 3$, $-5 < x < 8$, $0 < t < 2$ and $t = 0$ for the graphics of 2D.
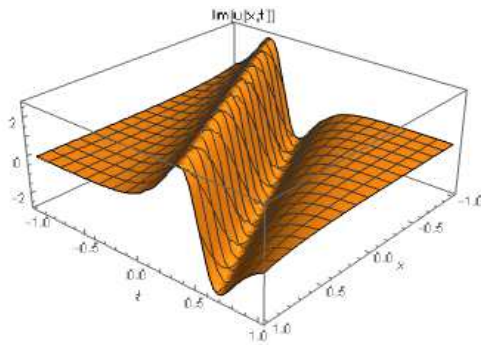
(3). The following solution is gotten by with case 3;

$$u_3(x,t) = \frac{6}{q^2}(q^2 - k^2)(-1 - i \ \sec h[\frac{1}{k\sqrt{p}}(\sqrt{k^2 - q^2})$$
$$\times (x - kt)]\tanh[\frac{1}{k\sqrt{p}}(\sqrt{k^2 - q^2})(x - kt)] +$$
$$+ \tanh[\frac{1}{k\sqrt{p}}(\sqrt{k^2 - q^2})(x - kt)]^2).$$
(17)

Figure 5. The 2D and 3D shape for the Eq. (18) with the values $k = 0.005$, $\mu = 3$, $c_0 = 1$, $-1 < x < 1$, $0 < t < 2$ and $t = 0$ for the graphic of 2D.

(5). The following solution is gotten by with case 5;

$$u_5(x,t) = \frac{4p\mu^2}{1 + 4p\mu^2}(1 - 3\tanh[\mu(x - kt)]^2) \quad (19)$$
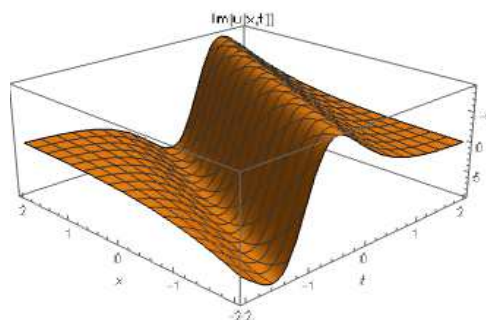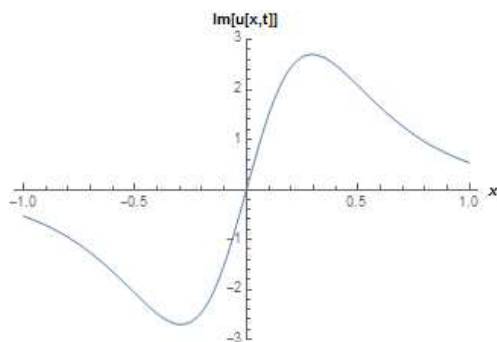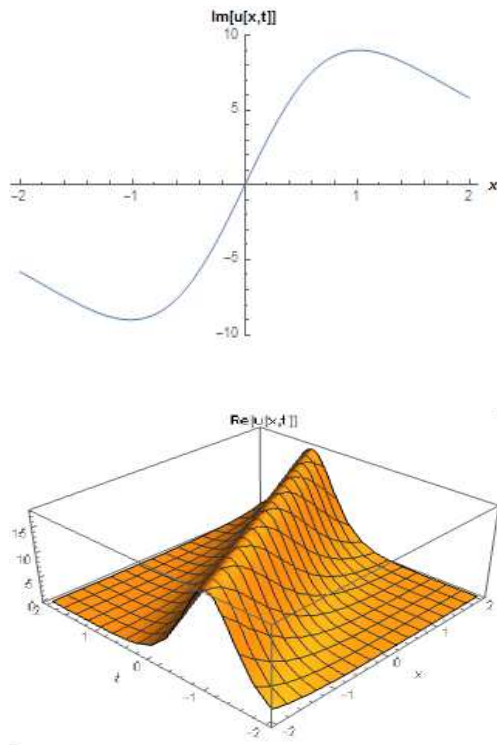
Figure 4. The 2D and 3D shape for the imaginary and real part of Eq. (17) with the values $k = 2$, $p = 1$, $c_0 = 1$, $-5 < x < 5$, $0 < t < 2$ and $t = 0$ for the graphics of 2D.

(4). The following solution is gotten by with case 4;

$$u_4(x,t) = \frac{k^2 - q^2}{q^2}(2 - 3\tanh[(-kt + x)\mu]^2) \quad (18)$$
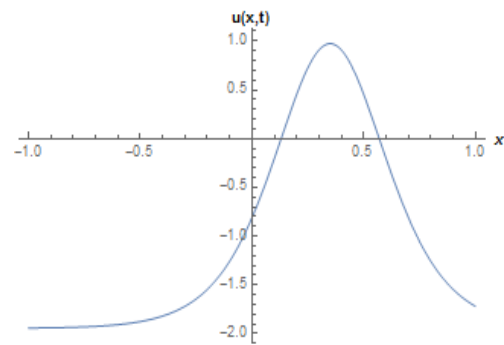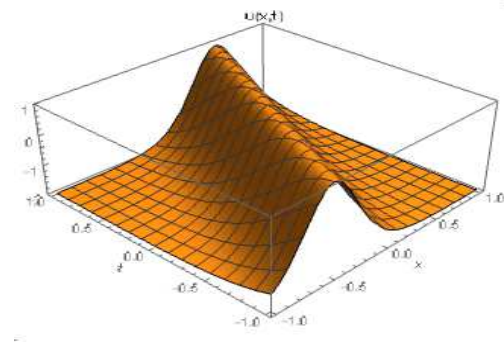
Figure 6. The 2D and 3D shape for the Eq. (19) with the values $k = 0.5$, $\mu = 3$, $p = 1$, $-0.5 < x < 1$, $0 < t < 2$ and $t = 0.7$ for the graphic of 2D.

(6). The following solution is gotten by with case 6;

$$u_6(x,t) = \frac{k^2 - q^2}{q^2}(-1 - 3\tan[\frac{\sqrt{k^2 - q^2}}{2k\sqrt{p}}(x - kt)]^2).$$
$$\quad (20)$$
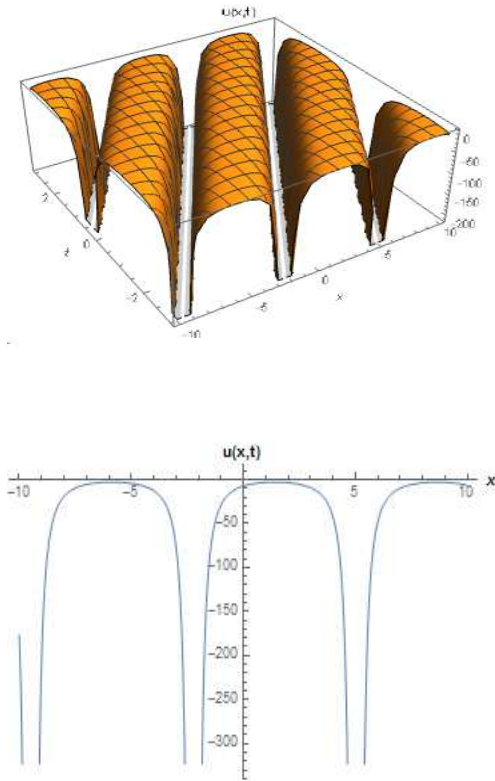
Figure 7. The 2D and 3D shape for the Eq. (20) with the values $k = 2$, $c_0 = 1$, $p = 1$, $-0.5 < x < 1$, $0 < t < 2$ and $t = 0.7$ for the graphic of 2D.

4. Results and Discussion

In [33] the improved $\exp(-\Phi(\xi))$-expansion function method is used in the solution of the magneto-electro-elastic circular rod longitudinal wave equation and the solution of different hyperbolic function forms is obtained. Secondly, the well-known improvement (G'/G)-expansion method [35] has been used for this equation and some precise hyperbolic and trigonometric functions are obtained. We observe that our results are new, but have the same solution structure. When compared with the existing, the results obtained by using these two methods. On the other hand, we observe that in the numerical simulations of the solutions we presented; Figure 1, Figure 2 and Figure 7 are singular soliton surfaces, Figure 3 is solit off surface, Figures 4-6 are soliton surfaces. We observe that some solutions in this study have important physical significance, such as the emergence of hyperbolic tangents in the calculation of magnetic moments and relative velocities, the emergence of hyperbolic secant in the profile of a laminar jet [40].

5. Conclusions

In this study, by utilizing the sine-Gordon extension method with the help of symbolic mathematical software, we investigated the solution of the magneto-electro-elastic circular rod longitudinal wave equation. We obtain some new solutions for complex hyperbolic and trigonometric functions. All solutions obtained in this study validate wave equations in magneto-electro-elastic circular rod and we examine this using the same procedure as symbolic mathematical software. We performed numerical simulations of all the solutions obtained in this paper. We observed that our results may be helpful in detecting transverse Poissons effect magneto-electro-elastic circular rod. The Sine-Gordon extension method is a powerful and efficient mathematical tool that can be used with the help of symbolic mathematical software to explore different non-linear methods arising in different fields of non-linear science.

Acknowledgments

References

[1] Ren, Y.J., Liu S.T., and Zhang, H.Q. (2007). A new generalized algebra method and its application in the (2+1)-dimensional Boiti-Leon-Pempinelli equation. *Chaos Solitons and Fractals*, 32, 1655-1665.

[2] Manafian, J. (2016). Optical soliton solutions for Schrödinger type nonlinear evolution equations by the $tan(F(\xi)/2)$-expansion method. *Optik-International Journal of Light and Electron Optics*, 127, 4222-4245.

[3] Feng, W.G., Li, K.M., Li, Y.Z. and Lin, C. (2009). Explicit exact solutions for (2+1)-dimensional Boiti-Leon-Pempinelli equation. *Communications in Non-linear Science and Numerical Simulation*, 14, 2013-2017.

[4] Fu, Z., Liu, S. and Zhao, Q. (2001) New Jacobi elliptic function expansion and new periodic solutions of nonlinear wave equations. *Physics Letters A*, 290, 72-76.

[5] Zayed, E.M.E. and Alurrfi, K.A.E. (2014). The homogeneous balance method and its applications for finding the exact solutions for nonlinear evolution equations. *Italian Journal of Pure and Applied Mathematics*, 33, 307-318.

[6] Islam, M.S, Khan, K. and Arnous, A.H. (2015). Generalized Kudryashov method for solving some (3+1)-dimensional nonlinear

evolution equations. *New Trends in Mathematical Sciences*, 3(3), 46-57.

[7] Lu, H.L., Liu, X.Q. and Niu, L. (2010). A generalized (G'/G)-expansion method and its applications to nonlinear evolution equations. *Applied Mathematics and Computation*, 215, 3811-3816.

[8] Wang, C. (2016). Dynamic behavior of traveling waves for the Sharma-Tasso-Olver equation. *Non-linear Dynamics*, 85(2), 1119-1126.

[9] Baskonus, H.M. and Bulut, H. (2016). Exponential prototype structures for (2+1)- dimensional Boiti-Leon-Pempinelli systems in mathematical physics. *Waves in Random and Complex Media*, 26(2), 201-208.

[10] Baskonus, H.M., Bulut, H. and Belgacem, F.B.M. (2017). Analytical Solutions for nonlinear long-short wave interaction systems with highly complex structure. *Journal of Computational and Applied Mathematics*, 312, 257-266.

[11] Kadkhoda, N. and Jafari, H. (2016). Kudryashov method for exact solutions of isothermal magnetostatic atmospheres. *Applied Mathematics and Computation*, 6(1), 43-52.

[12] El-wakil, S.A., El-labany, S.K., Zahran, M.A. and Sabry, R. (2002). Modified extended tanh-function method for solving nonlinear partial differential equations. *Physics Letters A*, 299, 179-188.

[13] Islam, M.S., Khan, K., Arnous, A.H. (2015). Generalized Kudryashov method for solving some (3+1)-dimensional nonlinear evolution equations. *New Trends in Mathematical Sciences*, 3(3), 46-57.

[14] Chen, H.T., Hong-Qing, Z. (2004). New double periodic and multiple soliton solutions of the generalized (2 + 1)-dimensional Boussinesq equation. *Chaos Solitons and Fractals*, 20, 765-769.

[15] Baskonus, H.M. and Bulut, H. (2015). On some new analytical solutions for The (2+1)-dimensional burgers equation and the special type of Dodd- Bullough-Mikhailov equation. *Journal of Applied Analysis and Computation*, 5(4), 613-625.

[16] Lu, Z. and Zhang, H. (2004). Soliton like and multi-soliton like solutions for the Boiti-Leon-Pempinelli equation. *Chaos, Solitons and Fractals*, 19, 527-531.

[17] Huang, D.J. and Zhang, H.Q. (2004). Exact Travelling Wave Solutions for the Boiti-Leon-Pempinelli Equation. *Chaos Solitons and Fractals*, 22, 243-247.

[18] Dai, C. and Wang, Y. (2009). Periodic structures based on variable separation solution of the (2+1)-dimensional Boiti-Leon-Pempinelli equation. *Chaos Solitons and Fractals*, 39, 350-355.

[19] Liang, Y. (2014). Exact solutions of the (3+1)-dimensional modified KdV- Zakharov-Kuznetsev equation and Fisher equations using the modified simple equation method. *Journal of Interdisciplinary Mathematics*, 17, 565-578.

[20] Zhang, H. (2007). Extended jacobi elliptic function expansion method and its applications. *Communications in Non-linear Science and Numerical Simulation*, 12(5), 627-635.

[21] Petrovic, N.Z. and Bohra, M. (2016). General jacobi elliptic function expansion method applied to the generalized (3 + 1)-dimensional nonlinear Schrodinger equation. *Optical and Quantum Electronics*, 48, (268).

[22] Yan, Z. (2003). Jacobi elliptic function solutions of nonlinear wave equations via the new sinh-gordon equation expansion method. *MM Research Preprints*, 22, 363-375.

[23] Manafian, J. and Lakestani, M. (2019). Lump-type solutions and interaction phenomenon to the bidirectional Sawada- Kotera equation. *Pramana*, 92:41.

[24] Manafian, J. (2018). Novel solitary wave solutions for the (3+1)-dimensional extended Jimbo-Miwa equations. *Computers & Mathematics with Applications*, 76(5), 1246-1260.

[25] Manafian, J., Mohammadi-Ivatloo, B. and Abapour, M. (2019). Lump-type solutions and interaction phenomenon to the (2+1)-dimensional breaking soliton equation. *Applied Mathematics and Computation Volume*, 356, 13-41.

[26] Ilhan, O.A., Manafian, J. and Shahriari M. (2019). Lump wave solutions and the interaction phenomenon for a variable coefficient Kadomtsev-Petviashvili equation. *Computers & Mathematics with Applications*, In press.

[27] Foroutan, M., Manafian, J. and Ranjbaran, A. (2018). Lump solution and its interaction to (3+1)-D potential-YTSF equation. *Nonlinear Dynamics*, 92, 2077-2092.

[28] Mohammad, S. and Manafian, J. (2019). Analytical behaviour of lump solution and interaction phenomenon to the Kadomtsev-Petviashvili-like equation. *Pramana*, 93:3.

[29] Seadawy, A.R. and Manafian, J. (2018). New soliton solution to the longitudinal wave equation in a magneto-electro-elastic circular rod. *Results in Physics*, 8, 1158-1167.

[30] Ramin, M.T., Manafian, J., Baskonus, H.M. and Duşunceli, F. (2019). Applications of He's semi-inverse variational method and ITEM to the nonlinear long-short wave interaction system, *IJAAS Journal*, 6(8), 53-64.

[31] Yan, C. (1996). A simple transformation for nonlinear waves. *Physics Letters A* , 22(4), 77-84.

[32] Bulut, H., Sulaiman, T.A. and Baskonus, H.M. (2016). New solitary and optical wave structures to the Korteweg-de Vries equation with dual-power law nonlinearity. *Optical and Quantum Electronics*, 48:564, 1-14.

[33] Baskonus, H.M., Bulut, H. and Atangana, A. (2016). On the complex and hyperbolic structures of longitudinal wave equation in a Magneto-Electro-Elastic circular rod. *Smart Materials and Structures*, 25(3), 035022.

[34] Xue, C.X., Pan, E. and Zhang, X.Y. (2011). Solitary waves in a Magneto-Electro-Elastic circular rod. *Smart Materials and Structures*, 20(10), 035022.

[35] Ma, X., Pan, Y. and Chang, L. (2013). Explicit travelling wave solutions in a Magneto-Electro-Elastic circular rod. *International Journal of Computer Science Issues*, 10(1), 62-68.

[36] Khan, K., Koppelaar, H. and Akbar, A. (2016). Exact and numerical soliton solutions to nonlinear wave equations. *Computational and Mathematical Engineering*, 2, 5-22.

[37] Younis, M. and Ali, S. (2015). Dark and singular solitons in Magneto-Electro-Elastic circular rod. *Waves in Random and Complex Media*, 25(4), 549-555.

[38] Yan, Z. and Zhang, H. (1999). New explicit and exact travelling wave solutions for a system of variant Boussinesq equations in mathematical physics. *Physics Letters A*, 252, 291-296

[39] Zhen-Ya, Y., Hong-oing, Z. and En-Gui, F. (1999). New explicit and travelling wave solutions for a class of nonlinear evolution equations. *Acta Physica Sinica*, 48(1), 1-5.

[40] Weisstein, E.W. (2002). *Concise encyclopedia of nathematics*. 2nd edition (New York: CRC Press).

Onur Alp İlhan *received his Masters Degree (2002) in Mathematics from Erciyes University and obtained a PhD Degree (2005) from National University of Uzbekistan. Mr. lhan is currently working as an Associate Professor at Faculty of Education in Erciyes University,. His research interests include Mathbio., ODE, PDE and Integral equations.*
iD *http://orcid.org/0000-0003-1618-6439*

Hasan Bulut *is currently professor of Mathematics in Firat University. His research interests include stochastic differential equations, fluid and heat mechanics, finite element method, analytical methods for nonlinear differential equations and numerical solutions of the partial differential equations.*
iD *http://orcid.org/0000-0002-6089-1517*

Tukur A. Sulaiman *is a research assistant at Firat University, Turkey and an assistant lecturer as Federal University Dutse, Nigeria. He is currently pursuing his PhD. (Applied Mathematics) in Firat University, Turkey. He has so far published 4 articles in various journals. His research interests include; stochastic optimization, analytical and numerical solutions of nonlinear ordinary/partial differential equations including the fractional differential equations.*
iD *http://orcid.org/0000-0001-7284-8332*

Haci Mehmet Baskonus *received the PhD degree in Mathematics from the Firat University, Turkey, in 2014. He is currently an Assoc. Prof. Dr at Faculty of Education in Harran University. His research interests include ordinary and partial differential equations, analytical methods for linear and nonlinear differential equations, mathematical physics, numerical solutions of the partial differential equations, fractional differential equations (of course ordinary and partial) and computer programming like Mathematica.*
iD *http://orcid.org/0000-0003-4085-3625*

RESEARCH ARTICLE

Simulation-based lateral transshipment policy optimization for s, S inventory control problem in a single-echelon supply chain network

Banu Y. Ekren[*], Bartu Arslan

Department of Industrial Engineering, Yasar University, Turkey
banu.ekren@yasar.edu.tr, bartuarslan@yahoo.com

ARTICLE INFO	ABSTRACT
	Since it affects the performance of whole supply chain significantly, definition of correct inventory control policy in a supply chain is critical. Recent technological development enabled real time visibility of a supply network by horizontal integration of each node in a supply network. By this opportunity, inventory sharing among stocking locations is also possible in the effort of cost minimization in supply chain management. Hence, lateral transshipment gained popularity and studies seeking the best lateral-transshipment policy is still under research. In this study, we aim to compare different lateral-transshipment policies for an s, S inventory control problem for a single-echelon supply chain network system. In this work, we consider a supply network with three stocking locations which may perform lateral transshipment among them when backorder takes place. We develop the simulation models of the systems in ARENA 14.5 commercial software and compare the performance of the policies by minimizing the total cost under a pre-defined fill rate constraint by using an optimization tool, OptQuest, integrated in that software. The results show that lateral transshipment works well compared to the scenario when there is no lateral transshipment policy in the network.

1. Introduction

Because it affects performance of the whole chain significantly, inventory control policy in a supply chain is important. Due to recent increase in e-commerce, order profile has changed towards more product variety with less volume and decreased response time. For instance, same day delivery concept is considered as strategy in many distribution companies to increase customer satisfaction. The recent order profile caused more variability in orders throughout the supply chain. Therefore, inventory control problem emerged as a significant issue in supply chain to overcome this variability and increase the performance of the supply chain. In a study, it is pointed out that inventory cost constitutes the 40% of the total logistics cost for fast moving consumer goods supply chain [1]. Therefore, recently, there are numerous studies focusing on inventory control problems, e.g., exploring optimal policies [2-4] by testing new and practical policies such as lateral transshipment [5-7] and inventory routing policies [8].

Companies tend to carry inventory in practice, to reduce their total cost and improve their customer service. However, information sharing also enabling inventory sharing among locations by transshipments has been less frequent [9-10]. For providing an effective mechanism, transshipments are being made between stocking locations at the same echelon based on their available inventories and their distances to increase the efficiency of the network. For instance, allowing transshipments between stocking locations may lead cost reduction as well as service improvement resulting with customer satisfaction. In this study, we study a single-echelon supply chain network by focusing on determining the best lateral-transshipment policy from four pre-defined ones. Our goal is to minimize the total cost by determining the safety-stock and up-to levels of stocking locations under a pre-defined fill rate performance metric. In Fig. 1, the layout of the supply network is shown. Based on that, there are three stocking locations (depots) in which they have their own demand profiles. These stocking

---

[*]Corresponding author

locations may share their inventories when a backorder takes place in one of the locations based on the pre-defined scenarios. The details of the sharing policy scenarios are explained in Section 3.4.
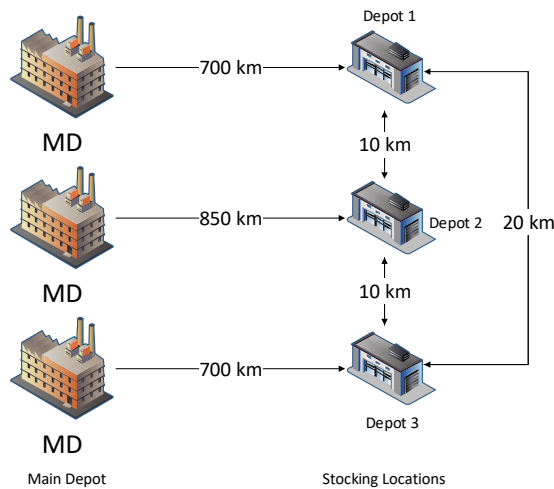
Figure 1. The studied single-echelon supply chain network.

In Fig. 1, we assume that the supply network's stocking locations are close to each other so that the backorder can be met instantaneously. Hence, we ignore the lead times for lateral transshipments, however, we consider a cost for these transshipments.

2. Literature review

Lateral transshipment can be defined as stock sharing policy among same echelon locations in an inventory network system. It is mostly motivated because that shorter lead times and decreased cost in redistribution of goods can be ensured compared to distribution from the main depot [11]. Recent comprehensive overviews on the problem are provided by [4] and [12]. In lateral transshipment mainly, there are two types of transshipment policies based on its timing: (1) reactive transshipments in response to an existing stock out [13-15], (2) proactive transshipments to prevent the future stock out [16-18]. In this work, we study reactive transshipment policies in which lateral transshipment may take place when backorder occurs in a stocking location.

Generally, the studies focus on the decision of how the lateral transshipment will take place between the locations [19-21]. For instance, Axsäter [19] studies a single-echelon inventory problem having a number of parallel local warehouses with Poisson demand. He evaluates the proposed decision rule via simulation model and concludes that the suggested rule performs quite good. Çapar et al. [20] study a decision rule by coordinating inventory and transportation policies in a two-stage supply chain network. They present that on average, their proposed rule outperforms the other alternative policies. A new heuristic approach for inventory sharing problem via lateral transshipment is introduced by Tiacci and Saetta [21]. Their work show that the proposed heuristic approach works well for

inventory balancing problem by lateral transshipment policies minimizing overall cost. A recent survey has also studied lateral transshipment problem for inventory models [4]. The interested reader can find further information in that review article.

The existing studies in the literature showed that transshipment is beneficial when the replenishment lead times are long from the upper echelon suppliers and when the stocking locations are close to each other at the same echelon level. This benefit increases drastically when backorder (shortage or penalty) cost is high. Recent transshipment studies explore different transshipment policies [20] by also investigating integration of proactive and reactive transshipment policies [22]. The existing studies show that when there is lateral transshipment policy in those networks, average cost is reduced by 11–17% compared to no lateral transshipment policy in that network [6].

The history of the (s, S) inventory control policies goes back to 1950s. Arrow et al. [23] developed a simple model determining the best order-up-to-level and the re-order level as a function of demand distribution, setup and stock out costs. The model considers immediate replenishment assumption. Freeman [24] studied (s, S) inventory policy with the inclusion of variable delivery time to derive the order size and the reordering point from an analytical perspective. Since then, lots of different variants of (s, S) policies have been analyzed and a considerable research has accumulated [25-28] because of its simple and efficient applications. An s, S inventory modelling application is shown by [29] which is a case study in a paint production company in Turkey.

The literature papers propose trial of new different transshipment policies as future study. Hence, different from the existing studies, we consider different four transshipment policies under reactive transshipment policy to test which one works better under the studied network. In the considered policies, based on the backorder amounts and the inventory levels of the other stocking locations, transshipments either take place or do not take place. Besides we compare the four transshipment policies among each other, we also compare these transshipment policies with the one when there is no lateral transshipment policy in the network.

3. Problem definition and simulation modelling

In this section, we provide the details of the studied inventory problem and the simulation modelling of the studied system with the assumptions.

3.1. Problem definition

In this paper, we study a single-item, one-echelon inventory problem in which items can be stored in three, $i = 1, 2, 3$, stocking locations. Those stocking locations are assumed to be supplied by an upper echelon supplier (i.e., main depots) with infinite

production capacity (see Fig. 1). In the model, it is assumed that demand for each stocking location i, arrives at the beginning of each period t (d_{it}). Demand, d_{it}, is fully satisfied, if there is sufficient amount of inventory level at the depot i at time t, I_{it}. After fully satisfying the demand, at the end of the time, inventory is checked at stocking locations for determining whether an order will be given from the main depot. Based on the considered inventory control policy of s, S, if the inventory level I_{it} is less than or equal to safety stock level, s_i, an order is placed for the stocking location i at time t. The order quantity, Q_{it}, is defined to be fulfilling the inventory level to up-to-levels of stocking locations, S_i. Hence, Q_{it} is equal to S_i - I_{it}. In the models, it is assumed that there is truck (transporter) capacity in carrying products both from the main depot and in the lateral transshipments. A lead time from the main depot to the stocking locations, L_{mi}, is also considered. If the demand exceeds the current inventory level, backorder occurs. When backorder takes place, lateral transshipment may realize between stocking locations based on the pre-defined scenario detailed in Section 3.4. The backorder amount that could not be satisfied by lateral transshipment is included as backorder cost in the total cost calculations.

3.2. Model notations

The utilized notations for the parameters are provided below:

b : backorder cost per demand,

k : number of stocking locations (i.e., $k = 3$)

h : holding cost per demand,

l_{ij} : a single truck's travel cost from stocking location i to j realizing lateral transshipment,

t_i : a single truck's transportation cost from the main depot to stocking location i,

d_{it} : demand amount for stocking location i at time t,

b_{it} : after lateral transshipments, backorder amount occurred at stocking location i at time t,

I_{it} : inventory level of stocking location i, at the end of time t,

d_{mi} : distance from the main depot to stocking location i (km.),

d_{ij} : distance from stocking location i to j,

L_{mi} : lead time from the main depot to stocking location i,

L_{ij} : lead time from stocking location i to stocking location j,

C_m : Truck capacity for main depot

C_t : Transporter capacity in lateral transshipment

In the above variables, h and b values are assumed to be, \$1/demand and \$5/demand, respectively. L_{mi} is set to: $L_{m1} = 1.5$ days; $L_{m2} = 1.75$ days; $L_{m3} = 1.5$ days and L_{ij} values are ignored. Demand for stocking locations is

considered to be same, stochastic and normally distributed: $N(100, 20)$.

Besides the above variables, we also consider the below decision variables for optimization.

s_i : safety stock level of stocking location i;

S_i : up-to-level of stocking location i;

n_{mit} : number of trucks sent from the main depot to the stocking location i, at time t,

n_{ijt} : number of transporters sent from stocking location i to j, at time t,

Q_{it} : order quantity of stocking location i, from the main depot at the end of time t,

q_{ijt} : amount of lateral transshipment from stocking location i to stocking location j, at time t,

It should be noticed that in an (s_i, S_i) inventory model, the a stocking location i (i.e. depot) places an order whenever its inventory level (I_{it}) decreases to a level less than the reorder level, s_i. At the end of each period t, the order quantity Q_{it} is calculated by Eq. (1) providing that the inventory is raised to an order-up-to level S_i:

$$Q_{it} = \begin{cases} S_i - I_{it} \ if \ I_{it} \leq s_i \\ 0, \ otherwise \end{cases} \quad (1)$$

Total cost (TC) is calculated by the Eq. (2) where T is the total simulation run time (i.e. 365 days) and k is the total number of stocking locations (i.e. $k = 3$):

$$TC = \sum_{t=1}^{T}\sum_{i=1}^{k}\sum_{j\neq i}^{k}[(h \times I_{it}) + (b \times b_{it}) + (n_{mjt} \times t_i) + (n_{ijt} * l_{ij})] \quad (2)$$

where n_{mit}, n_{ijt}, and t_i, l_{ij} values are computed by the below (3)-(6) equations:

$$n_{mit} = Q_{it} / C_m \quad (3)$$
$$n_{ijt} = q_{ijt} / C_t \quad (4)$$
$$t_i = d_{mi} \times \$0.4/km \quad (5)$$
$$l_{ij} = d_{ij} \times \$0.2/km \quad (6)$$

3.3. Simulation model assumptions

The inventory problem is simulated by the assumptions summarized below.

- In the simulation models, three stocking locations are considered where lateral transshipments may take place.

- We consider four different pre-defined lateral transshipment policies (P_1, P_2, P_3 and P_4) to compare their optimal results.

- Demand arrive each stocking location at the beginning of each day with normal distribution and the inventory levels are checked at the end of each day after demands are met from the inventory and lateral transshipments are completed.

- In total cost calculations, holding, transportation,

transshipment and backorder costs are considered.

- A new order is not sent to the main depot by the stocking locations until the previous ones arrive.

- Holding and backorder costs are considered to be $1 and $5 per demand, respectively.

- In transportation and transshipment cost calculations, we considered the distance (in km.) travelled in the network. For instance, travelling cost of a truck and lateral transshipment transporter is considered to be $0.4/km and $0.2/km, respectively.

- There is truck capacity constraint for both main depot transportation and lateral-transshipments which are considered to be 100 products/truck (C_m) and 25 products/transporter (C_t), respectively.

- It is assumed that there is infinite number of trucks and transporters in the system.

- Upper echelon supplier has infinite amount of items.

- The run length of simulation models is considered to be one year with 60 days of warm-up period for each scenario.

- To find out the optimal levels of s_i, S_i, we considered the minimization of total cost as objective function.

- Ten independent replications are completed in each scenario run.

- In the optimization, fill rate constraint is considered to be 0.95.

In the fill rate assumption, we considered that for instance, if a customer requests 100 units for demand, but the current inventory level is 80 units, then the fill rate is 80%. In the simulation models, because it is a popular and commonly utilized variance reduction technique, we used Common Random Numbers (CRN) variance reduction technique. In that technique, the same random number stream is used for different scenario configurations so that variance reduction is ensured.

3.4. Lateral transshipment policies

As mentioned previously, we determine four different lateral-transshipment policies to test how the lateral transshipment policies affect the inventory control problem and which works better based on the optimized total costs.

In each policy, first the arriving demands are satisfied by the current inventory in each stocking location. Then, a backorder existence is checked in the stocking locations of 1, 2 and 3, in sequence. If backorder occurs in a stocking location i at time t (b_{it}), then a lateral transshipment may take place for this location based on the below pre-defined policies:

Transshipment Policy 1, (P₁):
1- inventory availability is checked starting from the nearest stocking location j (where $i \neq j$).
2- backorder amount (b_{it}) is started to be met by transshipments from the nearest stocking locations in order. For instance, in the nearest stocking location, if the existence inventory amount does not meet the backorder amount then the remaining backorder is met from the following nearest stoking locations.
3- after transshipping all available inventories, if there is still a remaining backorder amount, then it is demanded from the upper supplier (main depot) at the end of the day.
4- the remaining amount of backorder that is supplied from the upper echelon is incurred as backorder cost in the total cost calculation.

Transshipment Policy 2, (P₂):
1- inventory availability is checked be starting from the nearest stocking location j (where $i \neq j$) whether single lateral transshipment can be done or not.
2- if available inventory at stocking location j at time t is larger than the backorder amount b_{it}, all the backorder is transshipped from this nearest stocking location j. Otherwise, the remaining nearest stocking locations' inventory levels are checked until it is found that there is available inventory level as backorder amount b_{it}.
3- if none of the stocking locations have enough inventory level as the backorder amount, b_i, then no lateral transshipment occurs and the backorder amount is demanded from the upper supplier at the end of the day.
4- the amount of backorder supplied from main depot is incurred as backorder cost in the total cost calculation.

Transshipment Policy 3, (P₃):
In this policy, first the availability of implementing P_2 is searched. Namely, the availability of meeting all the backorder amount at time t, b_{it}, by a single lateral transshipment is searched. However, if there is no any stocking location j having b_{it} amount of inventory level ($I_{jt} < b_{it}, \forall j$), then the total amount of inventory level in all stocking locations j at that time t where $i \neq j$ is calculated ($\sum_{j \neq i} I_{jt}$). If the total inventory levels in the stocking locations is larger than equal to b_{it} ($\sum_{j \neq i} I_{jt} \geq b_{it}$) then lateral transshipments take place starting from the nearest stocking locations until the b_{it} amount is met. Otherwise, no lateral transshipment occurs and all the b_{it} amount is met from the main depot.

Transshipment Policy 4, (P₄):
This policy is the combination of policies P_1 and P_2. Namely, first the availability of implementing the P_2 is searched. If P_2 cannot be implemented, in other words if the backorder amount at stocking location i at time t, cannot be met by a single lateral transshipment then P_1 is implemented.

Policy 5, (P₅):
Except the transshipment policies defined above, we also consider an inventory control system where no

lateral transshipments among the stocking locations take place.

3.5. Simulation model and OptQuest results

As an example, simulation flow chart of the developed ARENA model for Policy 1 and its OptQuest result's screenshots are shown by Figs. 2 and 3, respectively.
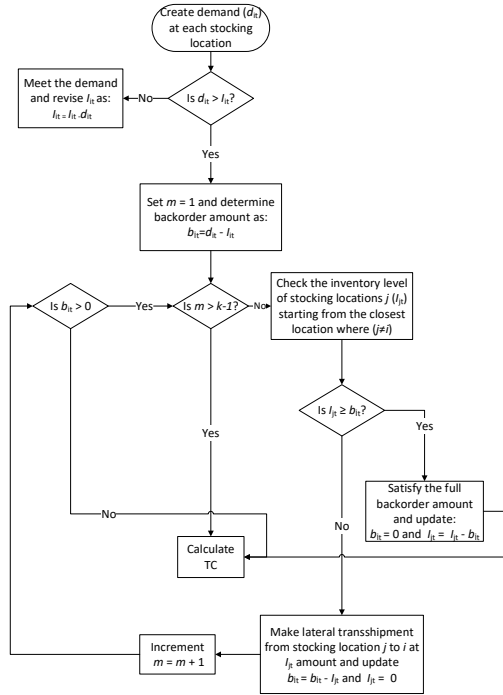
Figure 2. Flowchart of the simulation model for Policy 1

In an *s, S* inventory problem optimization, Kleijnen and Wan [30] showed the efficiency of the OptQuest optimization tool in the ARENA software. This optimization tool is a heuristic-based optimization tool combining the meta-heuristics of tabu search, neural networks, and scatter search into a single search heuristic [30]. It allows the user to explicitly define integer and linear constraints for the simulation inputs. Initially, it also requires the user to specify the lower, suggested, and the upper values for the decision variables to be optimized. The suggested values are for determining the starting points in the search procedure for s_i and S_i. In an effort to find a better result, first an initial optimization is run by heuristically determined suggested solution. Then, we utilize this initial optimization's result as suggested solution in the second optimization run.

3.6. Simulation results

Fig. 3 shows the OptQuest result namely the optimal s_i, S_i values for Policy 1. According to that, the optimal s_i, S_i values for P_1 are found to be (95, 200), (62, 100), (265, 360) for $i = 1, 2, 3$, respectively. The total cost is $483,694 at those levels.

The optimal s_i, S_i levels obtained by the OptQuest tool based on the pre-defined policies P_1-P_5 are summarized in Table 1 and Table 3. The output results for Table 1 are given by Table 2.
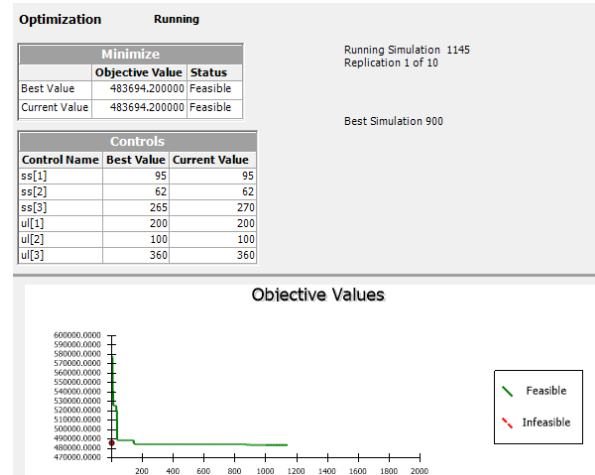
Figure 3. s_i, S_i values of P_1 in ARENA OptQuest

Table 1. *s, S* values of policies.

Policy	s_1	s_2	s_3	S_1	S_2	S_3
1	95	62	265	200	100	360
2	143	63	255	304	96	298
3	165	63	278	294	99	300
4	61	40	210	200	99	391
5	178	189	167	314	283	283

Once again, Table 2 shows total cost and fill rate results based on optimal Table 1 results.

Table 2. Total costs and fill rates of policies.

Policy	*TC*	*Fill rate*
1	$483,694	0.9535
2	$512,941	0.9501
3	$509,339	0.9546
4	$492,445	0.9523
5	$580,673	0.9508

Table 3 shows, the frequency of lateral transshipments took place among the stocking locations at the Table 1 values, optimal levels of s_i, S_i found by OptQuest.

Table 3. Lateral transshipment frequency among the stocking locations.

Policy	*Avg $\sum n_{1j}$*	*Avg $\sum n_{2j}$*	*Avg $\sum n_{3j}$*	**Total**
1	765.6	106.6	886.4	1,758.6
2	860.6	5.7	517.9	1,384.2
3	904.9	9.3	563.9	1,478.1
4	541.2	23.4	1,069.1	1,633.7
Total	**3,072.3**	**145**	**3,037.3**	

Table 4 illustrates the frequency of transportation taking place from the main depot to the stocking locations in the scenarios of P_1-P_4 at the Table 1 values, the optimal levels of s_i, S_i found by OptQuest.

Table 4. Transportation frequency from main depot to stocking locations.

Policy	$\sum n_{m1}$	$\sum n_{m2}$	$\sum n_{m3}$	$\sum n_{mi}$
1	364.6	182.9	582.7	1,130.2
2	536.1	182.9	488.2	1,207.2
3	532.6	182.9	491.0	1,206.5
4	335.8	181.7	625.4	1,142.9
5	438.4	417.0	413.6	1,269.0

Table 5 illustrates the total transportation and holding costs based on lateral transshipment policies.

Table 5. Transportation and holding costs of policies.

Policy	$\sum$Trans. Cost	$\sum$Holding Cost
1	$327,430	$122,680
2	$348,990	$129,600
3	$348,794	$128,650
4	$330,914	$127,590
5	$380,340	$170,020

In Table 1, we observe optimal s_i, S_i levels and total cost (TC) values in columns 1-7 in order. We also illustrate the obtained fill rate at the last column. The findings from Tables 1-5 are summarized below:

- It is observed that TC is always smaller when there is any lateral transshipment policy in the network. Note that TC in P_5 is the largest one in Table 2.
- It is observed that the minimum cost is obtained by P_1, P_4, P_3, P_2 in sequence.
- By Table 1, it is noted that when there is any lateral transshipment policy in the network, stocking location 2 carries lower inventory (s, S levels) than the other stocking locations. This is probably due to that the second stocking location is the furthest location to the main depot. This stocking location tends to decrease the number of transportations from the main depot by decreasing the frequency of lateral transshipments to the other locations. This result can also be validated in Table 3 that $\sum n_{2j} = 145$ which is almost 1/20 of the other transshipment frequencies: 3,072 and 3,037. It is also verified by Table 4 that $\sum n_{m2}$ values are relatively low compared to the others.
- From Table 3, it can be observed that maximum lateral transshipment frequencies occur in P_1, P_4, P_3, P_2 in sequence, showing that there is negative correlation between TC and lateral transshipment frequency. Namely, when total cost increases, lateral transshipment cost decreases in the policies.
- From Table 4, we observe that the least transportation frequency from the main depot took place in P_1. Since this policy has the least TC, this may mean that the highest lateral transshipment frequency might take place in this policy. This can also be verified by Table 3 that the highest lateral transshipment frequency happened in P_1 (1,759).
- In Table 4, since there is no lateral transshipment, the highest transportation frequency from the main

depot took place in P_5.

4. Conclusion

In this study, we compare different (i.e. four) lateral-transshipment policies in an *s, S* inventory control problem of a single-echelon supply chain network system in which there are three stocking locations. By the recent technological development, real time visibility of a supply network by horizontal integration of each node in a network is possible. By this opportunity, lateral transshipment gained popularity and studies seeking the best lateral-transshipment policy is still under research. In this work, we aim to contribute to literature by considering different lateral transshipment policies in a supply network and comparing their performances by optimizing their objective functions, total cost. We develop the simulation models of the systems in ARENA 14.5 commercial software and compare the performances of them by minimizing the total cost under a pre-defined fill rate constraint (i.e., 95%) by using OptQuest tool in this software. The results show that lateral transshipment works well for the studied supply system when it is compared with the scenario that there is no lateral transshipment policy in the network. It is observed that the minimum cost is obtained by the policies: P_1, P_4, P_3, P_2 in sequence.

Also, we noted that trial of different lateral transshipment policies under different supply network designs as well as parameter values (e.g. backlogging cost) and layout of chain is promising to be investigated as a future study.

References

[1] Cachon, G., & Terwiesch, C. (2006). *Matching supply with demand*. McGraw-Hill, Inc.

[2] Arrow, K. J., Harris, T., & Marschak, J. (1951). Optimal Inventory Policy. *Econometrica,*19(3), 250-272.

[3] Waller, M., Williams, B.D., & Tokar, T. (2008). A Review of Inventory Management Research in Major Logistics Journals: Themes and Future Directions. The International Journal of Logistics Management, 19(2), 212–232

[4] Bushuev, M.A., Guiffrida, A., Jaber, M.Y., & Khan, M. (2015). A Review of Inventory Lot Sizing Review Papers. Management Research Review, 38(3), 283–298.

[5] Paterson, C., Kiesmuller, G., Teunter, R., & Glazebrook, K. (2011). Inventory models with lateral transshipments: A review. European Journal of Operational Research, 210(2), 125–136.

[6] Pan, S., Nigrelli, M., Ballot, E., Sarraj, R., & Yang, Y. (2015). Perspectives of Inventory Control Models in the Physical Internet: A Simulation Study. Computers & Industrial Engineering, 84, 122–132.

[7] Yang, Y., Pan, S., & Ballot, E. (2015). Mitigating supply chain disruptions through interconnected

logistics services in the Physical Internet International. Journal of Production Research, 55(14), 3970–3983.

[8] Bertazzi, L., & Speranza, M.G. (2012). Inventory Routing Problems: An Introduction. EURO Journal on Transportation and Logistics, 1(4), 307–326

[9] Ozdemir D., Yucesan, E., & Herer, Y.T. (2006). Multi-location transshipment problem with capacitated transportation. European Journal of Operational Research, 175, 602-621.

[10] Ekren, B.Y. & Heragu, S.S. (2008). Simulation Based Optimization of Multi-Location Transshipment Problem with Capacitated Transportation. In Proceedings of the 2008 Winter Simulation Conference, edited by S. J. Mason et al., 2632–2638. Piscataway, New Jersey: IEEE.

[11] Allen, S.C. (1958). Redistribution of total stock over several user locations. Naval Research Logistics Quarterly, 5(4), 337-345.

[12] Seidscher, A., & Minner, S. (2013). A Semi-Markov decision problem for proactive and reactive transshipments between multiple warehouses. European Journal of Operational Research, 230(1), 42–52.

[13] Krishnan, K., & Rao, V. (1965). Inventory control in *N* warehouses. Journal of Industrial Engineering, 16, 212–215.

[14] Robinson, L.W. (1990). Optimal and approximate policies in multiperiod, multilocation inventory models with transshipments. Operations Research, 38(2), 278–295.

[15] Olsson, F. (2010). An inventory model with unidirectional lateral transshipments. European Journal of Operational Research, 200(3), 725-732.

[16] Diks, E.B., & de Kok, A.G. (1996). Controlling a divergent 2-echelon network with transshipments using the consistent appropriate share rationing policy. International Journal of Production Economics, 45(1–3), 369–379.

[17] Diks, E.B., & de Kok, A.G. (1998). Optimal control of a divergent multi-echelon inventory system. European Journal of Operational Research, 111(1), 75–97.

[18] Tagaras, G., & Vlachos, D. (2002). Effectiveness of stock transshipment under various demand distributions and nonnegligible transshipment times. Production and Operations Management, 11(2), 183–198.

[19] Axsäter, S. (2003). A New Decision Rule for Lateral Transshipments in Inventory Systems. Management Science, 49(9), 1168-1179.

[20] Capar, I., Eksioglu, B. & Geunes, J. (2011). A decision rule for coordination of inventory and transportation in a two-stage supply chain with alternative supply sources. Computers & Operations Research, 38(12), 1696–1704.

[21] Tiacci, L., & Saetta, S. (2011). A heuristic for balancing the inventory level of different locations through lateral shipments. International Journal of Production Economics, 131(1), 87-95.

[22] Paterson, C., Teunter, R. & Glazebrook, K. (2012). Enhanced lateral transshipments in a multi-location inventory system. *European* Journal of Operational Research, 221(2), 317–327.

[23] Arrow, K.J., Harris, T., & Marschak, J. (1951). Optimal Inventory Policy. Econometrica, 9(3), 250-272.

[24] Freeman R. (1957). *S s* Inventory Policy with Variable Delivery Time. Management Science, 3(4), 431-434.

[25] Axsäter, S. (2015). *Inventory Control*. International Series in Operations Research & Management Science. Springer International Publishing, 3rd Ed., Switzerland.

[26] Bashyam, S., & Fu, M.C. (1998). Optimization of (*s*, *S*) Inventory Systems with Random Lead Times and a Service Level Constraint. Management Science, 44(12), part-2, 243-256.

[27] Sethi, S., & Cheng, F. (1997). Optimality of (*s*, *S*) Policies in Inventory Models with Markovian Demand. Operations Research, 45(6), 931-939.

[28] Silver E.A., Pyke D.F., & Peterson R. (1998). Inventory management and production planning and scheduling. New York: Wiley.

[29] Ekren, B.Y. & Ornek, A. (2015). Determining Optimum (*s*, *S*) Levels of Floor Stock Items in a Paint Production Environment. Simulation Modelling Practice and Theory, 57, 133-141.

[30] Kleijnen, J.P.C., & Wan, J. (2007). Optimization of simulated systems: OptQuest and alternatives. Simulation Modelling Practice and Theory, 15(3), 354-362.

Banu Y. Ekren *is a full professor in the department of Industrial Engineering, at Yasar University, in Izmir, Turkey. She got her Ph.D., from University of Louisville in the Dept. of Industrial Engineering, in Kentucky, USA. Her research focus is on warehousing, stochastic and simulation modelling, supply chains, simulation-based optimization, and design of automated warehousing. Banu Y. Ekren holds associate professor position at her university and teaches simulation, stochastic modelling and facility logistics courses at the undergraduate level. She has several simulation-based journal and book chapter publications.*

https://orcid.org/0000-0001-6491-1389

Bartu Arslan *is an industrial engineer, graduated from the Department of Industrial Engineering, at Yasar University in Izmir, Turkey. He worked as a fellow under Dr. Banu Y. Ekren's supervision for a research project funded by TUBITAK focusing on agent-based simulation modelling of an automated warehousing system. His research interests include applications of simulation-*

optimization in supply chain, logistics, production, and warehousing. Arslan would like to do academic career in Industrial Engineering specifically focusing on dynamic modelling approaches in simulation.

ⓘ *https://orcid.org/0000-0003-2114-767X*

An International Journal of Optimization and Control: Theories & Applications (http://ijocta.balikesir.edu.tr)

RESEARCH ARTICLE

Route planning methods for a modular warehouse system

Elif G. Dayıoğlu [a] [*], Kenan Karagül [b], Yusuf Şahin [c], Michael G. Kay [d]

[a] *Department of Computer Technology and Information Systems, Mersin University, Turkey*
[b] *Department of Logistics, Pamukkale University, Turkey*
[c] *Department of Business Administration, Burdur Mehmet Akif Ersoy University, Turkey*
[d] *Edward P. Fitts Department of Industrial&Systems Engineering, NC State University, USA*
edayioglu@mersin.edu.tr, kkaragul@pau.edu.tr, ysahin@mehmetakif.edu.tr, kay@ncsu.edu

ARTICLE INFO	ABSTRACT
	In this study, procedures are presented that can be used to determine the routes of the packages transported within a modular storage system. The problem is a variant of robot motion planning problem. The structures of the procedures are developed in three steps for the simultaneous movement of multiple unit-sized packages in a modular warehouse. The proposed heuristic methods consist of route planning, tagging, and main control components. In order to demonstrate the solution performance of the methods, various experiments were conducted with different data sets and the solution times and qualities of the proposed methods were compared with previous studies. It was found that the proposed methods provide better solutions when taking the number of steps and solution time into consideration.

1. Introduction

Logistics is the process of strategically managing the supply, transport, and storage of raw materials, semi or finished products to ensure cost-effectiveness. The raw materials and semi-finished products used by a company and the finished products produced by the company must be moved from one location to another. Logistic activities, which have a significant impact on the success of the production and distribution operations of the company, are composed of many functional areas. The performance shown in these functional areas leads to an increase in service quality as well as a reduction in operating costs, and logistics has to provide high-quality service at a low or acceptable cost [1].

In logistic operations, it is an important challenge to meet the different products demanded by consumers [29]. One of the most critical functions in logistics processes is warehousing. During this process, the products are stored at certain points for a certain period of time. The primary purpose of the classical warehousing is to store the products in a correct and non-destructive way. On the other hand, many operations are carried out from the receipt of the products to the delivery of them to the customer in

today's modern warehousing concept. Such a system requires a high level of coordination between the seller and the buyer's decision-making [30]. In modern warehouse systems, activities such as classification of products, quality control, packaging, barcoding, labeling, keeping records of stock movements, providing the communication with the related units (sender, buyer, customer, producer, etc.) are carried out in addition to other activities [2,3].

It is possible to classify warehouses according to geographical distribution (central and decentralized), property structure (unique, general, and contract), product characteristics (parts, bulk) and operation (production, distribution). A public logistics networks (PLN) is a network that provides an alternative to private logistics networks for the transport of goods. A PLN consists of distribution centers (DCs), trucks, and package components. In these networks, which are inspired by the structure and operation of the Internet, a package is sent from a store to a public distribution center located in an area in a metropolitan area [4].

The use of automation systems for the activities carried out at the distribution centers provides a significant reduction in costs [5,6]. Fully automated warehouses (loading, unloading, sorting, stacking, automation of

---

[*]Corresponding author

packages storage and retrieval) have become an essential issue for effective cost minimization and warehouse management. These warehouses where the operations in the warehouse are fully automated are defined as modular warehouses. Kay [5] suggested a distribution center design that would meet these requirements. The proposed system consists of square modules with orthogonal pop-up powered wheels. Figure 1 shows the top view of one of the modules with orthogonal pop-up powered wheels. In each direction, the wheels of the module are raised and lowered relative to the wheels in the other direction. The guides (Fig. 2) used in this system can be raised and lowered when necessary to limit and direct the movement of the load [7].
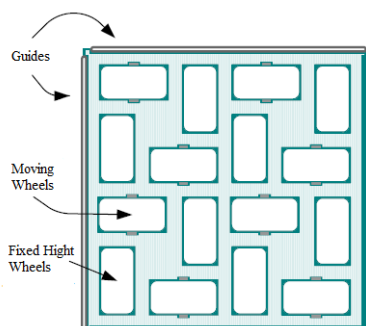
Figure 1. Top view of a single module [5]
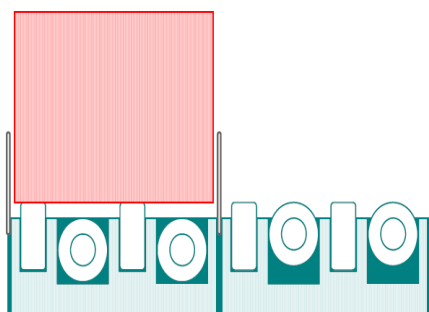
Figure 2. Guidelines in the ascended state [5]

In this study, heuristics algorithms based on the algorithm proposed by Datar [6] and Sittivijan [8] are used for the control of packages in a modular warehouse. The purpose of the problem addressed is that the packages can be delivered to the desired exit point in the shortest time and least number of steps. The conceptual framework of the study is presented in Section 2. In this context, 15-floating block, rush hour, and the warehouseman's problem along with studies related to these problems are presented. The details of the proposed methods are given in Section 3. Experimental studies and results were included in the fourth and fifth sections, respectively.

2. Route planning problem

In this study, heuristic algorithms based on the transport of unit-size packages in a modular warehouse with a limited number of free spaces are proposed. Therefore, the problem is closely related to the motion planning problem. In this section, 15-floating block,

rush hour and warehouseman's problems and the related literature are examined. These problems will provide a better understanding of this problem, which is considered within the scope of the study and related to the movement of objects within a limited space.

2.1. 15-floating block problem

The 15-floating block problem is similar to the structure of the problem discussed in this study. It is a purer form of the problem of transporting more than one object in a limited area [6]. In this problem, a square area of 4×4 has 15 full tiles and one empty tile numbered from 1 to 15, which will be rearranged according to a specific target configuration. An adjacent tile can be shifted to this position orthogonally by the described empty tile [9]. The goal is to reach the final target by moving the tiles only horizontally or vertically from the initial state as shown in Figure 3.

Figure 3. Floating block puzzle

There are many studies using various methods in the literature regarding this problem. Spitznagel [10] has proved that it is only possible to obtain the end configuration from the initial configuration by double-numbered permutation. Reinefeld [11] discussed the 8-floating block problem and evaluated the utility of node sequencing using the recursive deepening A* (YDA*) algorithm. It has been concluded that YDA* applications performed with a fixed operator (e.g., up, left, right, down) perform worse than those done with a simple random operator selection. Gue and Kim [12] developed a 15-block based warehouse system. Unlike the floating block problem, the calculation is made for more than one free space and as the number of free space increases, the retrieval time is reduced. Bauer [9] proposed the Manhattan Pair Distance Heuristics (MCU), which is a combination of YDA algorithm and Manhattan distance function. With the help of the proposed method, the number of nodes in the heuristic search has been reduced by 80% for the 15-floating block problem.

2.2. Rush hour problem

The rush hour problem, as seen in Figure 4, is a module-based game that consists of a target vehicle to be transported to the exit point and only a few vehicles moving in the horizontal or vertical direction [13]. Other cars in the module are moved to open the path to the designated exit of the target car.
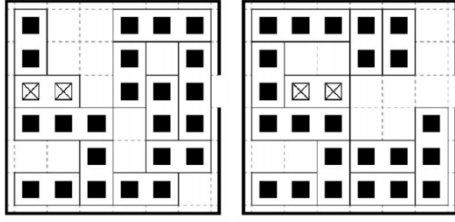
Figure 4. Rush hour problem

Flake and Baum [13] showed that the decision of whether the target vehicle would exit the module was PSPACE-Complete. Furthermore, unlike the original rush hour problem, they presented a generalized version of the traffic problem (GSH - Generalized Rush Hour Problem) with the option of arbitrary width and height and the possibility of the outlet to be anywhere in the vicinity of the grid. Hearn and Demaine [14] proposed a nondeterministic calculation model based on the inverse edge directions in the weighted directional charts with minimum flow constraints. The framework they developed was inspired by "Generalized Rush Hour Logic" developed by Flake and Baum [13]. Hauptman et al. [15] designed a novel IDA*-based heuristics solver for the Rush Hour domain.

2.3. Warehouseman's problem

The warehouseman's problem, which is an extension of the $n \times n$ floating block problem, involves coordinated movement planning of a large number of independent objects in a limited area [16]. The goal is to move objects in the repository from the initial configuration to the final configuration [6]. Coordinated motion planning of a large number of three-dimensional objects in the presence of obstacles is a computational problem in which it is important to regulate complexity [17]. Hopcroft et al. [17] proved that the problem of simultaneous motion planning for a limited number of discrete rectangular bodies of different sizes to move within a 2-dimensional rectangular area is PSPACE-hard. Yeung and Bekey [18] used a decentralized approach based on the problem being global and local road planning problem. Sanchez and Latombe [19] used probabilistic roadmaps (PRM) which plans free paths for multiple interacting robots without collision. They developed a new PRM planner that combines a single-query bi-directional sampling strategy with a lazy collision-checking connection strategy. Sharma and Aloimonos [20] proposed a solution method introducing constraints on the size of objects for non-unit sized objects and distributing the free space for warehouseman problem. Sarrafzadeh and Maddila [21] formed a two-dimensional warehouse system consisting of square objects (robots and obstacles) that were allowed to move horizontally and vertically along the grid lines.

LaValle and Hutchinson [22] used a dynamic programming-based solution algorithm to solve multiple robot motion planning problems. Azarm and Schmidt [23] developed a framework that is decentralized and allows for parallel decision for multiple robots to solve the collision problem. The framework allows parallel path computation and dynamic priority assignment. Svestka and Overmars [24] proposed a coordinated approach to the problem of multi-robot road planning, unlike conventional decentralized planning. In the proposed system, the data structure that stores multi-robot motion information is created in two steps. In the first step, a roadmap for only one robot is created using the PRM planner, and then some of these simple roadmaps have been made a roadmap for the composite robot in the second step.

Leroy et al. [25] developed a geometric-based method for motion planning of multiple robots. While the paths of all the robots are calculated independently of each other, the problem of coordinating the movements of the robots in their way so as not to collide with each other has been emphasized. Guo and Parker [26] proposed a distributed and best motion planning algorithm for multiple robots. This computational complexity problem is divided into two modules as path and speed planning, and D* search method is applied to both modules. Yamashita et al. [27] suggested a two-stage method for motion planning of multiple mobile robots in order to move a large object together in a three-dimensional environment. As a result, they have integrated their movement planner into two phases as a global road planner and a local motion planner. In global path planning, constraints of object motion are considered as a cost function and a heuristic function in the A* search. Liu et al. [28] presented a road planning scheme based on the ant colony algorithm with collision avoidance for multiple robot systems. In order to solve the collision between moving robots, they adopted a behavior strategy on "first come and first served" principle.

3. Proposed methods

In this study, five solution methods based on A* heuristic are proposed for planning the movement of packets to avoid collisions and deadlocks in a modular storage system. Although the proposed methods are diversified in some respects, they have the same components. With the help of these approaches, motion plans are prepared in the warehouse system where there are moving obstacles consisting of more than one object moving at the same time. Since there may be conflicts between moving objects, the route planning is not sufficient to bring the active objects to the desired targets at this stage. Therefore, the methods include components such as route planning, tagging, and main control. These components are described in the following sections.

3.1. Route planning phase

Each of the active objects has a planned path. An active object has the capability of planning a route from its initial position to its destination. On the other hand, if it has been tagged by a higher priority object and move

away from its current planned path, it can plan a new route from its initial location or its current location. Route planning is used to find this path. To find the path from the current position to the target position, the orthogonal neighboring modules around the current module are examined. In this study, A* based heuristic algorithms are used to select the next module to be moved. The lowest cost neighbor module is selected as the new module with this algorithm. The standard A* algorithm was modeled by making some arrangements because the warehouse system discussed in the study was not static. The location of the objects in the warehouse changes with time, so it is not a static but a dynamic environment. Therefore, a new function named $LB_{(x,y)}^T$ is used instead of the $F(n)$ function. The module with the smallest $LB_{(x,y)}^T$ value is selected as the module to be moved. According to Equation (1), the current position (a,b) of the object to be moved and the target position (x,y) of the object is assumed to be as follows:

$$LB_{(x,y)}^T = T_{(a,b)}^{(a_0,b_0)} + T_{(x,y)}^{(a,b)} + T_{(a_n,b_n)}^{(x,y)} \qquad (1)$$

where

$T_{(a,b)}^{(a_0,b_0)}$: the wandering time from the starting module (a_0, b_0) to some intermediary position (a,b)

$T_{(x,y)}^{(a,b)}$: the weighted estimated wander time to go to the neighboring module (x,y) during the next k time steps.

Since the configuration of the objects in the system may vary from one time step to another time step, at this stage, the weighted sums for each t time step are computed using Equation (2).

$$T_{(x,y)}^{(a,b)} = \sum_{t=1}^{k} w_t \times T_{(x,y),t}^{(a,b)} \qquad (2)$$

While the $T_{(x,y)}^{(a,b)}$ value is calculated, the occupancy gap state of the neighboring module is considered during the time off from the current time step ($t = 1$) to the kth time step ($t = k$). Because the state of the objects in the system can vary greatly from one time step to another time step. In this paper, the route planning is taken as $k = 3$ and the system state in each of the 3-time steps from the time step present in the route planning for each package is evaluated. The w_t value in Equation (2) is arbitrarily chosen, but it must satisfy the conditions of $\sum_{t=1}^{k} w_t = 1$ and $w_t > w_{t-1}$.

$T_{(x,y),t}^{(a,b)}$: estimated wander time to move the object from its current module (a, b) to the neighbor module (x,y) at the time step t.

$T_{(x,y),t}^{(a,b)} = 1$: At $t = k$ time step if the neighboring module is empty, the current module passes in a time step with the neighboring module.

$T_{(x,y),t}^{(a,b)} \geq 2$: At $t = k$ time step, if the neighboring module is filled with a low-priority object, switching to that module takes

place in one or more time steps.

$T_{(x,y),t}^{(a,b)} = \infty$: At $t = k$ time step, if there is a high priority object that has reached the target in the neighboring module, it can not be moved, and this variable takes the infinite value.

$T_{(a_n,b_n)}^{(x,y)}$: The distance from the neighboring module to the target point. The Manhattan distance method is used to calculate by Equation (3).

$$T_{(a_n,b_n)}^{(x,y)} = |x_{neighbour} - x_{\text{target}}| \\ + |y_{neighbour} - y_{\text{target}}| \qquad (3)$$

$LB_{(x,y)}^T$ is the lower bound value used on the path to be defined to go from point (a,b) to point (x,y). As in the case where the neighbor with the smallest $F(n)$ value is selected in the A* algorithm, here the neighbors with the smallest $LB_{(x,y)}^T$ value from the orthogonal neighbors of the current module is selected too.

3.2. Tagging phase

When moving on the planned path of the high priority object, if it encounters a lower priority or inactive object from the active object on the path, this process is used to move these objects away from the defined path. In Figure 5, the object numbered 8 tries to move from module (2,2) to module (2,3). Module (2,3) has an inactive object with 4 priority. For this reason, the object with 8 priority tags the object with 4 priority. In tagging, 8, which is the priority of the current object, is transferred to the object with 4 priority as the inheritance priority. Thus, the object with 4 priority can move 5, 6 or 7 priority objects. Because this object has a value of 8 as the inheritance priority during the tagging process. After the object with a priority of 4 has been tagged, it is checked whether they are empty neighbors that can move. Neighbors are (3,3) and (1,3) modules. The object with priority 4 selects the object with priority 2, which is the lowest priority neighbor. 4's inheritance priority passes to object with priority 2, but when the object with priority 2 tries to tag the object with priority 9, returns to the object with priority 4 because 9's priority is higher than 8. Here, backtracking is performed. The new object to be tagged is selected as 7, 7 → 5, 5 → 3, 3 → 6 tags and the module (3,1) is found as the last empty module. The tagging process ends in this way. In the latest case, it is moved to 6 → (3,1), 3 → (2,1).
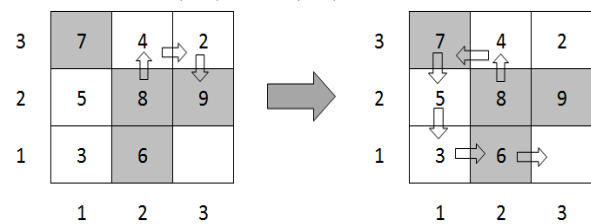
Figure 5. Example of the tagging process

3.3. Main control phase

The movement of all active and inactive objects is controlled by the main control component. At each time step, this part controls every active object that is not at the destination point and checks whether it is tagged by another high priority object. If it is not tagged, it performs route planning for the currently active object due to the changing system environment.

Possible situations in the main control process can be listed as follows:

✓ If the neighboring module is empty and not tagged by another object, then the currently active object passes to the neighboring module within that time step.

✓ If the neighboring module is empty and tagged by another object, then it is checked whether the neighboring module is tagged by the low priority. If the neighboring module is tagged by the low priority object, the tagging process of that object is released, and the currently active object is passed to the neighboring module at that time step. If the neighboring module is tagged by a high priority object, the higher priority object is expected to move from that module.

✓ If the neighboring module is not empty and is tagged by another object, it is also checked whether the neighboring module is tagged by the low priority. If the neighboring module is tagged by the low priority object, the tagging process for that object is released, and the tagging process is performed by the current object, and the current module is moved to the neighbor module. If the neighboring module is tagged by a high priority object, the higher priority object is expected to move from that module.

✓ If the neighboring module is not empty and is not tagged by another object, then the object's priority in the neighboring module is looked. If the priority of the neighboring module is lower than the priority of the active object, the labeling process is started by the active object for this module and if the tagging process is successful, the active object passes to the neighboring module. If the priority of the neighboring module is higher than the priority of the active object, the higher priority object is expected to move from that module.

4. Experimental study

In order to show the performance of the proposed algorithms, 23 test problems were produced for 3 group dataset. The dataset is divided into groups according to the density and dimensions of the warehouse. Table 1 shows the group numbers of the dataset and the size and density information of the warehouses. The first group contains 40% and 50% density warehouse test problems in 4×4 sizes. The second group has a 6×6 sizes of warehouse layout with the density ranging from 40-70%. Moreover, the last one consists of 20-99% density and 16×32 warehouse sizes.

Table 1. Details of the data set

The group of data sets	Number of data set	Density Interval (%)	Size of warehouse
Group 1	1, 2 and 3	40-50	4×4
Group 2	4, ..., 8	40-70	6×6
Group 3	9, ..., 23	20-99	16×32

All algorithms were implemented in the Eclipse environment using the Java programming language. Comparisons were made on a standard computer with 4 GB RAM and 2.67 GHz processor. In Table 2, the features and differences of all examined algorithms are shown in summary.

Table 2. Details of algorithms

Algorithm	Features and Differences
ALG-B1 (1)	✓ Sittivijan (2015) algorithm ✓ For each active object, an A * based intuitive route planning is performed at the beginning
ALG-B2 (2)	✓ Datar (2011) algorithm ✓ It is a greedy approach. ✓ It is an algorithm that is planned only for the movement at the next time step.
ALG-P1 (3)	✓ The algorithm in which ALG-B1 is restarted in every environment change ✓ For each active object, the route planning is performed again with an intuitive A * based always on the time step
ALG-P2 (4)	✓ An improved version of ALG-B1. ✓ Release process applied to tag object is removed from the main control and applied only during the tagging process
ALG-P3 (5)	✓ An improved version of ALG-P1 ✓ Release process applied to tag object is removed from the main control and applied only during the tagging process
ALG-P4 (6)	✓ An improved version of ALG-B1 ✓ The calculation of the $LB_{(x,y)}^{T}$ function has been changed
ALG-P5 (7)	✓ An improved version of ALG-P3 ✓ The calculation of the $LB_{(x,y)}^{T}$ function has been changed

In the experimental study, solution times and the number of steps in reaching the final solution were taken into consideration for the performance comparison of the methods. In Table 3, all algorithms were compared for data sets in terms of the number of steps required to reach the destination points of the packages.

Table 3. Number of solution steps

Data Set	METHODS							Size	Density
	1	2	3	4	5	6	7		
1	8	9	8	8	8	8	8	4*4	0,4
2	8	9	7	8	7	7	8	4*4	0,5
3	9	4	9	9	9	3	3	4*4	0,5
4	11	20	11	11	11	11	11	6*6	0,4
5	12	13	12	12	12	12	12	6*6	0,5
6	6	14	6	6	6	6	6	6*6	0,6
7	19	16	15	19	15	19	15	6*6	0,6
8	13	19	13	13	13	13	13	6*6	0,7
9	29	31	27	29	27	29	27	16*32	0,2
10	29	31	27	29	27	29	27	16*32	0,2
11	26	31	24	26	24	26	24	16*32	0,2
12	29	33	25	29	25	29	25	16*32	0,3
13	32	37	33	32	33	32	33	16*32	0,4
14	47	40	37	47	37	47	37	16*32	0,5
15	41	47	38	41	38	41	38	16*32	0,6
16	53	75	52	53	52	53	52	16*32	0,7
17	64	122	75	64	75	64	75	16*32	0,8
18	89	117	80	89	80	89	82	16*32	0,9
19	143	128	125	143	125	143	125	16*32	0,95
20	-	232	205	-	205	-	208	16*32	0,96
21	-	-	-	-	-	-	-	16*32	0,97
22	-	-	-	-	-	-	-	16*32	0,98
23	-	-	-	-	-	-	-	16*32	0,99
	582	692	543	582	543	582	545		

Table 4. Relative comparison of solution steps

Data Set	METHODS							Size	Density
	1	2	3	4	5	6	7		
1	1	1,13	1,00	1	1,00	1,00	1,00	4*4	0,40
2	1	1,13	0,88	1	0,88	0,88	1,00	4*4	0,50
3	1	0,44	1,00	1	1,00	0,33	0,33	4*4	0,50
4	1	1,82	1,00	1	1,00	1,00	1,00	6*6	0,40
5	1	1,08	1,00	1	1,00	1,00	1,00	6*6	0,50
6	1	2,33	1,00	1	1,00	1,00	1,00	6*6	0,60
7	1	0,84	0,79	1	0,79	1,00	0,79	6*6	0,60
8	1	1,46	1,00	1	1,00	1,00	1,00	6*6	0,70
9	1	1,07	0,93	1	0,93	1,00	0,93	16*32	0,20
10	1	1,07	0,93	1	0,93	1,00	0,93	16*32	0,20
11	1	1,19	0,92	1	0,92	1,00	0,92	16*32	0,20
12	1	1,14	0,86	1	0,86	1,00	0,86	16*32	0,30
13	1	1,16	1,03	1	1,03	1,00	1,03	16*32	0,40
14	1	0,85	0,79	1	0,79	1,00	0,79	16*32	0,50
15	1	1,15	0,93	1	0,93	1,00	0,93	16*32	0,60
16	1	1,42	0,98	1	0,98	1,00	0,98	16*32	0,70
17	1	1,91	1,17	1	1,17	1,00	1,17	16*32	0,80
18	1	1,31	0,90	1	0,90	1,00	0,92	16*32	0,90
19	1	0,90	0,87	1	0,87	1,00	0,87	16*32	0,95
20	-	-	-	-	-	-	-	16*32	0,96
21	-	-	-	-	-	-	-	16*32	0,97
22	-	-	-	-	-	-	-	16*32	0,98
23	-	-	-	-	-	-	-	16*32	0,99
Mean	1,00	1,23	0,95	1,00	0,95	0,96	0,92		

All algorithms were compared with ALG-B1 (1) according to the solution step numbers, and the results are shown in Table 4 and Figure 6. When the solutions are examined in terms of the number of steps, it has been observed that the proposed algorithms generally have better results than ALG-B1 (1) and ALG-B2 (2). For example, while the proposed methods reached a solution in 15 steps for the 7th dataset, ALG-B1 (1) and ALG-B2 (2) were able to reach solutions in steps of 19 and 16, respectively. For some datasets, the solution could not be obtained. The reason for this is that in the present configuration, no path can be defined for the arrival of the active packets to the destination points. A deadlock event occurs for these datasets. As a result, packages cannot move to any module.

Figure 6. Relative comparison of the solution steps

The second comparison of the obtained solutions is the solution times. The solution times of the methods for different datasets are shown in Table 5. On the other hand, the relative comparison is made according to the ALG-B1 method in Table 6. When the average solution times are taken into consideration, it is seen that the proposed methods provide better solutions in shorter times. The solution time comparison is shown in Figure 7.

Table 5. Solution times of the methods

Data Set	METHODS							Size	Density
	1	2	3	4	5	6	7		
1	4,88	5,40	4,89	4,86	4,88	4,90	4,87	4*4	0,4
2	4,91	5,45	4,41	4,88	4,39	4,38	4,92	4*4	0,5
3	5,40	2,84	5,41	5,43	5,38	2,38	2,32	4*4	0,5
Total	**15,1**	**13,7**	**14,7**	**15,1**	**14,6**	**11,6**	**12,1**		
4	1	1,82	1,00	1	1,00	1,00	1,00	6*6	0,4
5	1	1,08	1,00	1	1,00	1,00	1,00	6*6	0,5
6	1	2,33	1,00	1	1,00	1,00	1,00	6*6	0,6
7	1	0,84	0,79	1	0,79	1,00	0,79	6*6	0,6
8	1	1,46	1,00	1	1,00	1,00	1,00	6*6	0,7
Total	**36,2**	**46,9**	**34,2**	**35,9**	**33,9**	**36**	**34,01**		
9	1	1,07	0,93	1	0,93	1,00	0,93	16*32	0,2
10	1	1,07	0,93	1	0,93	1,00	0,93	16*32	0,2
11	1	1,19	0,92	1	0,92	1,00	0,92	16*32	0,2
12	1	1,14	0,86	1	0,86	1,00	0,86	16*32	0,3
13	1	1,16	1,03	1	1,03	1,00	1,03	16*32	0,4
14	1	0,85	0,79	1	0,79	1,00	0,79	16*32	0,5
15	1	1,15	0,93	1	0,93	1,00	0,93	16*32	0,6
16	1	1,42	0,98	1	0,98	1,00	0,98	16*32	0,7
17	1	1,91	1,17	1	1,17	1,00	1,17	16*32	0,8
18	1	1,31	0,90	1	0,90	1,00	0,92	16*32	0,9
19	1	0,90	0,87	1	0,87	1,00	0,87	16*32	0,95
20	-	-	-	-	-	-	-	16*32	0,96
21	-	-	-	-	-	-	-	16*32	0,97
22	-	-	-	-	-	-	-	16*32	0,98
23	-	-	-	-	-	-	-	16*32	0,99
Total	**344,8**	**673,8**	**456,1**	**342,3**	**439,1**	**336,5**	**437,1**		

Table 6. Relative comparison of solution times

Data Set	METHODS							Size	Density
	1	2	3	4	5	6	7		
1	1,00	1,11	1,00	0,99	1,00	1,00	1,00	4*4	0,4
2	1,00	1,11	0,90	0,99	0,89	0,89	1,00	4*4	0,5
3	1,00	0,53	1,00	1,01	1,00	0,44	0,43	4*4	0,5
4	1,00	1,70	1,01	1,01	1,00	1,01	1,01	6*6	0,4
5	1,00	1,06	1,00	0,99	0,99	0,99	0,99	6*6	0,5
6	1,00	2,05	1,03	0,99	0,99	1,00	1,00	6*6	0,6
7	1,00	0,87	0,81	1,00	0,81	1,01	0,82	6*6	0,6
8	1,00	1,38	0,99	0,98	0,99	0,98	0,98	6*6	0,7
9	1,00	0,97	0,94	1,00	0,93	1,00	0,92	16*32	0,2
10	1,00	0,98	0,94	0,98	0,91	0,98	0,91	16*32	0,2
11	1,00	1,13	0,96	0,99	0,96	1,00	0,96	16*32	0,2
12	1,00	1,08	0,88	0,98	0,86	1,00	0,87	16*32	0,3
13	1,00	1,05	0,98	0,97	0,94	0,99	0,95	16*32	0,4
14	1,00	0,80	0,78	0,99	0,78	0,98	0,78	16*32	0,5
15	1,00	1,08	0,94	0,98	0,92	0,98	0,93	16*32	0,6
16	1,00	1,35	0,98	0,99	0,97	1,00	0,97	16*32	0,7
17	1,00	1,96	1,16	0,99	1,16	0,99	1,16	16*32	0,8
18	1,00	1,28	0,89	0,99	0,88	1,00	0,88	16*32	0,9
19	1,00	0,90	1,02	1,01	0,88	0,93	0,86	16*32	0,95
20	-	-	-	-	-	-	-	16*32	0,96
21	-	-	-	-	-	-	-	16*32	0,97
22	-	-	-	-	-	-	-	16*32	0,98
23	-	-	-	-	-	-	-	16*32	0,99
Total	**344,8**	**673,8**	**456,1**	**342,3**	**439,1**	**336,5**	**437,1**		

Figure 7. Relative comparison of the solution times

5. Conclusion

In this paper, the modular warehouse management issue is discussed, and the new A* based heuristics algorithms for simultaneous movement of multiple unit-sized packages in the modular warehouse have been proposed. While some features are different, the proposed methods consist of three stages. The first stage is the route planning used to perform the movement of each package from the starting location to the destination location. The second stage is the tagging process based on packet priorities, used to prevent packet collisions. The final stage is the main control part where the movement of all packages in the warehouse is controlled.

The proposed methods are compared with the methods of Datar [6] and Sittivijan [8]. Datar [6] chose the next movement area in the route planning section only by looking at the distance to the target and the priorities of the packages. Sittivijan [8] has proposed an A* based heuristic for the movement of packages. The difference between the method proposed in this study and the method proposed by Sittivijan [8] is that the heuristic route planning is carried out at the beginning and subsequent route planning is not carried out as long as the packages do not leave their the planned routes. However, in the proposed method, the route planning process is applied again for the active packages in each environment change. Furthermore, in Sittivijan [8], a release is applied to the object subjected to the tagging process in both the main control and the tagging process. In the proposed method, this operation was removed from the main control section and applied only in the tagging process to ensure achieving high-quality results. Also, an improvement has been made to the conditions of high priority packages to reach their destination at close to their *LB* value calculations during the route planning phase, which provides improved results. In future studies, solution approaches using other heuristics will be developed for warehouse systems with different dimensions.

Acknowledgments

References

[1] Mason-Jones, R., & Towill, D.R. (1999). Total cycle time compression and the agile supply chain. *International Journal of Production Economics*, 62(1-2), 61-73.

[2] Çancı, M., & Erdal, M. (2009). *Lojistik Yönetimi: Freight Forwarder El Kitabı*. 3. Baskı, Uluslararası Taşımacılık ve Lojistik Hizmet Üretenleri Derneği, İstanbul.

[3] Sahin, Y., & Eroğlu, A. (2015). Hierarchical Solution of Order Picking and Capacitated Vehicle Routing Problems. *Suleyman Demirel University Journal of Engineering Sciences and Design*, 3(1), 15-28.

[4] Xiang, L., Kay, M.G., & Telford, J. (2007). *Public Logistics Network Protocol Design and Implementation* [online]. North Carolina, North Carolina State University, Available from: https://people.engr.ncsu.edu/kay/pln/IERC07.pdf [Accessed 15 August 2018]

[5] Kay, M. G. (2004). Protocol Design for a Public Logistic Network [online]. North Carolina, North Carolina State University, Available from: https://people.engr.ncsu.edu/kay/pln/IMHRC04.pdf [Accessed 16 August 2018]

[6] Datar, M. (2011). *Priority-based Control Algorithm for Movement of Packages in a Public Distribution Center*. PhD Thesis, North Carolina State University.

[7] Kay, M.G. (2013). *Home Delivery Logistics Networks using Driverless Delivery Vehicles [online]*. North Carolina, North Carolina State University, Available from: https://people.engr.ncsu.edu/kay/hdln/HDLNuDDV.pdf [Accessed 15 August 2018]

[8] Sittivijan, P. (2015). *Modular Warehouse Control: Simultaneous Rectilinear Movement of Multiple Objects within Limited Free Space Environment*. PhD Thesis, North Carolina State University.

[9] Bauer, B. (1994). *The Manhattan Pair Distance Heuristic for the 15-Puzzle [online]*. Paderborn, Universitat-GH Paderborn. Available from: http://citeseerx.ist.psu.edu/viewdoc/download?doi=10.1.1.58.7&rep=rep1&type=pdf [Accessed 25 July 2018]

[10] Spitznagel, E.L. (1967). A new look at the fifteen puzzle. *Mathematics Magazine*, 40(4), 171-174.

[11] Reinefeld, A. (1993). Complete Solution of the Eight-Puzzle and the Benefit of Node Ordering in IDA*. *Thirteenth International Joint Conference on Artificial Intelligence,* pp 248-253.

[12] Gue, K.R., & Kim, B.S. (2007). Puzzle-based storage systems. *Naval Research Logistics*, 54(5), 556-567.

[13] Flake, G.W., & Baum, E.B. (2002). Rush Hour is PSPACE-complete, or "Why you should generously tip parking lot attendants. *Theoretical Computer Science*, 270(1-2), 895-911.

[14] Hearn, R.A., & Demaine, E.D. (2005). PSPACE-completeness of sliding-block puzzles and other problems through the nondeterministic constraint logic model of computation. *Theoretical Computer Science*, 343(1-2), 72-96.

[15] Hauptman, A., Elyasaf, A., Sipper, M., & Karmon, A. (2009). GP-Rush: using genetic programming to evolve solvers for the Rush Hour puzzle. *11th Annual Conference on Genetic and Evolutionary Computation*, pp 955-962.

[16] Sharma, R., & Aloimonos, Y. (1992). Coordinated motion planning: the warehouseman's problem with constraints on free space. *IEEE Transactions on systems, man, and cybernetics*, 22(1), 130-141.

[17] Hopcroft, J.E., Schwartz, J.T., & Sharir, M. (1984). On the Complexity of Motion Planning for Multiple Independent Objects; PSPACE-Hardness of the Warehouseman's Problem. *The International Journal of Robotics Research*, 3(4), 76-88.

[18] Yeung, D.Y., & Bekey, G. (1987). A decentralized approach to the motion planning problem for multiple mobile robots. *IEEE International Conference on Robotics and Automation*, 4, 1779-1784.

[19] Sanchez, G., Latombe, & Jean-Claude (2002). Using a PRM planner to compare centralized and decoupled planning for multi-robot systems. *IEEE International Conference on Robotics and Automation*, 2, 2112-2119.

[20] Sharma, R., & Aloimonos, Y., (1992). Coordinated motion planning: the warehouseman's problem with constraints on free space. *IEEE Transactions on systems, man, and cybernetics*, 22(1), 130-141.

[21] Sarrafzadeh, M., & Maddila, S.R. (1995). Discrete warehouse problem. *Theoretical Computer Science*, 140(2), 231-247.

[22] LaValle, S.M., & Hutchinson, S.A. (1998). Optimal motion planning for multiple robots having independent goals. *IEEE Transactions on Robotics and Automation*, 14(6), 912-925.

[23] Azarm, K., & Schmidt, G. (1997). Conflict-free motion of multiple mobile robots based on decentralized motion planning and negotiation. *IEEE International Conference on Robotics and Automation*, 4, 3526-3533.

[24] Švestka, P., & Overmars, M.H. (1998). Coordinated path planning for multiple robots. *Robotics and Autonomous Systems*, 23(3), 125-152.

[25] Leroy, S., Laumond, J.P., & Siméon, T. (1999). Multiple path coordination for mobile robots: A geometric algorithm. *16th International Joint Conference on Artificial Intelligence*, pp 1118-1123.

[26] Guo, Y., & Parker, L.E. (2002). A distributed and optimal motion planning approach for multiple

mobile robots. *IEEE International Conference on Robotics and Automation*, 3, 2612-2619.

[27] Yamashita, A., Arai, T., Ota, J., & Asama, H. (2003). Motion planning of multiple mobile robots for cooperative manipulation and transportation. *IEEE Transactions on Robotics and Automation*, 19(2), 223-237.

[28] Liu, S. Mao, L., & Yu, J., (2006). Path planning based on ant colony algorithm and distributed local navigation for multi-robot systems. *IEEE International Conference on Mechatronics and Automation*, pp 1733-1738.

[29] Koç, Ç., Erbaş, M., & Ozceylan, E. (2018). A rich vehicle routing problem arising in the replenishment of automated teller machines. *An International Journal of Optimization and Control: Theories & Applications (IJOCTA)*, 8(2), 276-287.

[30] Uddin, M.F., & Kazushi, S.A.N.O., (2011). Coordination and Optimization: The integrated supply chain analysis with non-linear price-sensitive demand. *An International Journal of Optimization and Control: Theories & Applications (IJOCTA)*, 2(1), 83-94.

Elif G. Dayıoğlu *received her B.S. and M.S. degree from the Computer Engineering Department of Pamukkale University in 2014 and 2017 respectively. She is currently pursuing a PhD degree in Computer Engineering at Çukurova University. She has been working as a research assistant at the Computer Technology and Infirmation Systems Department of Mersin University since 2014.*

http://orcid.org/0000-0002-4716-952X

Kenan Karagül *studied industrial engineering for his Bachelor degree and business administration for M.S. and PhD degrees. His field of study includes operations research, logistics, vehicle routing problems, metaheuristics, and quantitative models. He worked at various firms between 1996 and 2001. He worked at Pamukkale University as an instructor until 2013. He has been working at the same university as an assistant professor since 2013. He was awarded the best Ph.D. thesis on Graduate Tourism Students Congress in Kuşadası (2014).*

http://orcid.org/0000-0001-5397-4464

Yusuf Şahin *received the M.S. degree in industrial engineering from Pamukkale University in 2009 and a PhD degree in business administration from Suleyman Demirel University in 2014. He has been an assistant professor of business administration at Burdur Mehmet Akif Ersoy University since 2014. His field of study includes operations research, logistics, warehouse management, vehicle routing, meta-heuristics, and quantitative models.*

http://orcid.org/0000-0002-3862-6485

Michael G. Kay *has been a professor of Industrial Engineering at North Carolina State University since 1992. He is Interim Director of the Operations Research Graduate Program and is Associate Director of Graduate Programs in the ISE Department. He is the current President of the College-Industry Council on Material Handling Education.*

http://orcid.org/0000-0002-1359-8270

Control of M/Cox-2/s make-to-stock systems

Özgün Yücel, Önder Bulut[*]

Department of Industrial Engineering, Yaşar University, Turkey
ozgun.yucel@yasar.edu.tr, onder.bulut@yasar.edu.tr

ARTICLE INFO	ABSTRACT
	This study considers a make-to-stock production system with multiple identical parallel servers, fixed production start-up costs and lost sales. Processing times are assumed to be two-phase Coxian random variables that allows us to model the systems having rework or remanufacturing operations. First, the dynamic programming formulation is developed and the structure of the optimal production policy is characterized. Due to the highly dynamic nature of the optimal policy, as a second contribution we propose an easy-to-apply production policy. The proposed policy makes use of the dynamic state information and controlled by only two parameters. We test the performance of the proposed policy at several instances and reveal that it is near optimal. We also assess the value of dynamic state information in general by comparing the proposed policy with the well-known static inventory position based policy.

1. Introduction

In a make-to-stock production system, there is always a tradeoff between excess inventory, shortages and production costs. Production control is the main tool handling this tradeoff and providing cost effective operation. In general, in a make-to-stock environment, optimal production control requires starting production at the right time and producing with the optimum number of channels (servers, lines, or machines) to provide sufficient amount of products.

Production policy strategies use the information of inventory status to trigger the production when the inventory status drops below certain threshold levels. Here, inventory status refers a function of the state variables that keep track of the required system information such as inventory level, number of outstanding production orders and their ages. The form of the optimal inventory status function would change from system to system but it is still unknown even for most of the basic make-to-stock production settings. Therefore, most of the studies in the literature, which consider only a single server, assumes that inventory status equals inventory level. There are limited number of studies on multi-server production-inventory systems but they only provide partial characterization of the optimal policy without any discussion on the performances of the static, which should take inventory status as inventory position, or alternative dynamic policies.

In real life production-inventory systems, due to the

nature of the environment and its technology, production times might have zero, moderate or high variance. Furthermore, such systems might have rework/inspection or remanufacturing operations. In order to deal with such real life systems, we assume phase-type, in specific two-phase Coxian production times. A busy server (worker or machine) might be either at the first phase (main operation) or at the second phase (inspection/rework) at any given time. A two-phase Coxian random variable has independent exponential phases and there is a certain visiting probability from phase-one to phase-two. Hence, we can create different systems at the boundaries of the visiting probability: when it is set to zero, processing time distribution becomes exponential (which is a typical assumption in the literature), when it is set to one, we can mimic the two-phase general Erlang processing times. Different values of this probability and production rates of phases correspond to systems with different rework characteristics and processing time moments. The representation of a production channel feeding the inventory after a two-phase Coxian processing time is shown in Figure 1. Coxian production times assumption would also help us to assess the value of dynamic state information, i.e. current status of production.
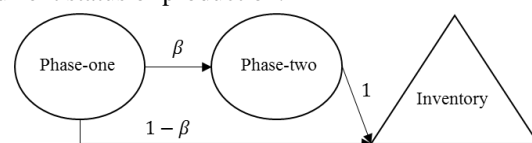
Figure 1. Representation of a Cox-2 production server

---

[*]Corresponding author

We charge fixed production (start-up) cost for activating servers, holding cost for each unit of inventory and lost sale cost for each unsatisfied demand. The studies that consider fixed costs in the literature are assuming only a single server. To the best of our knowledge, our study is the first considering multiple parallel production servers and fixed start-up cost at the same time in make-to-stock control environment. There is no study in the literature characterizing the optimal production policy for multi-server systems. For single server backordering systems, it is known that the optimal production policy is a two-critical-number policy. In this study, we aim to characterize the optimal production policy for lost sales multi-server systems with fixed production cost and propose easy to apply alternatives.

We provide the literature review in Section 2. Dynamic programming formulation of the problem is given in Section 3. In Section 4, we numerically characterize the optimal production policy. In Section 5, we propose an alternative production policy and evaluate its performance. Section 6 concludes the paper and provides future research directions.

2. Literature review

In this chapter, we review the production and inventory control literature in the make-to-stock environment. This problem is first attacked by considering the systems having single production channel and single customer/demand class. Analyses are mostly based on queueing theory techniques. Interestingly, the early studies consider the fixed startup or shut-down costs. More recent studies extend the literature by considering multiple production channels without fixed costs. Another common feature of the recent studies is the Markovian structure that enables them to develop Markov Decision Process (MDP) formulation for the control of make-to-stock systems.

Gavish and Graves [1] is the first to study the production-inventory problem assuming single channel, fixed and deterministic production times, independent Exponential inter-demand-arrival times, and backorders. They modeled the problem as an $M/D/1$ make-to-stock queue in the infinite horizon under the time-average cost criterion. This first study is actually the extension of Heyman [2] and Sobel [3] to the make-to-stock production environment. In [2] and [3], $M/G/1$ and $G/G/1$ queueing systems are studied, respectively, operating with server start-up and shutdown costs, and unit service and queue-time costs. For both of the settings, it is shown that the optimal policy is a *two critical number policy* denoted by (S, s) and (M, m) in [2] and [3], respectively. If the queue length is less than or equal to m (or s), service is not provided until queue length increases to M (or S). Service is triggered when the queue length is M and continued until it drops to m again. Although the analyses of [2] and [3] are specific for the queueing environment, we believe that their setting covers the

production control for make-to-order systems. The optimal policy structure, which is a *two critical number policy*, is preserved in the make-to-stock production environment setting of [1]. However, the control parameters of the policy are defined on the inventory level: start production when the inventory level hits to the lower control level and continue until it hits to the upper control level. For different settings where two critical number policy is still optimal, see [4] and [5]. Researchers apply different techniques for the analysis of the *two critical number policy*. For example, Lee and Srinivasan [6] considers $M/G/1$ make-to-stock queue with backordering and propose a renewal analysis in order to calculate expected cost. For compound Poisson demand extension of this study see [7].

Recent studies mostly apply MDP techniques for the settings having Markovian structure. This stream of literature usually assumes no fixed production/setup cost. In addition, production is triggered by a single server except Bulut and Fadıloğlu [8]. Ha [9] is the first that uses MDP techniques in problem modeling. [9] addresses $M/M/1$ make-to-stock queue with multiple demand classes and lost sales, and shows that base-stock is optimal production control policy. For backordering case, see [10]. [8] extends the setting by assuming multiple parallel exponential servers and optimal policy is defined as state-dependent base-stock. Ha [11] proves that work storage level is optimal production policy for $M/E_k/1$ make-to-stock queue. Gayon et al. [12] differs from [11] with the backordering assumption. However, in our study, preserving the Markovian structure, we consider multiple parallel production servers allowing reworks and fixed start-up costs at the same time. Interested readers are also directed to the study [13] that considers the control of hybrid make-to-stock/make-to-order systems.

3. Dynamic programming formulation

We consider a production system including s many identical parallel servers each having two production phases in order to produce a single type of product. Processing times are assumed to be two-phase Coxian random variables where each phase is exponentially distributed with rates μ_1 and μ_2, respectively. Production is started at phase-one, then items are either processed at second phase with a certain probability β, or leave the system without passing second stage with probability $1 - \beta$ (Figure 1). Visiting probability β facilitates us to work on more general systems than the ones having exponential processing times, which is a classical assumption in the literature. We model the system as $M/Cox_2/s$ make-to-stock queue with fixed start-up costs and lost sales. In the terminology of production-inventory control literature, the classical Kendall Lee queueing notation is used for the models of make-to-stock systems. However, the meaning of the queuing notation is slightly different in the make-to-stock environment. In our case, M denotes Markovian

inter-demand arrival times but the arrived demands do not enter a queue and trigger a production order. Instead, they are either directly satisfied from the inventory or lost, and immediately leave the system. The second entry in the notation, which is "Cox_2" in our case, is for the production time distribution. The inventory is replenished using s many available production channels according to a production policy in anticipation of the future demand arrivals. That is, Coxian-2 is not the "service" time of each demand arrival; it is the replenishment lead-time of any production order triggered according to the policy.

Customer demands arrive according to a stationary Poisson process with rate λ. Lost sale cost c is incurred for each unsatisfied demand. Fixed start-up cost of activating a server is K, inventory holding cost is h and discount rate is denoted by α.

System state is defined with three variables to keep track of the events. Let $x_i(t), i \in \{1,2\}$, be the number of active servers at i^{th} phase and $x_3(t)$ be the inventory level at time t. Then the system state space is

$$SS = \left\{ \begin{array}{c} (x_1(t), x_2(t), x_3(t)) \mid \sum_{i=1}^{2} x_i(t) \leq s, \\ x_i(t) \in Z^+ \cup \{0\}, i = 1,2,3 \end{array} \right\} \quad (1)$$

Through the Markovian property, decision can be made in either at a phase completion or a demand arrival. For this reason, system state definition $(x_1(t), x_2(t), x_3(t))$ is used regardless of time dimension. Since the original problem is a production-inventory control problem in continuous time, we obtain the discrete time equivalent of this problem via uniformization technique ([14]). The uniform transition rate is defined as $\nu = \lambda + s(\mu_1 + \mu_2)$. In our model, production is controlled by the decision variable $u \in \{x_1, \dots, s - x_2\}$, which is the number of busy servers at phase-1 (at the first stage of the production process). Model only controls the number of active servers at stage-one because whenever production is triggered on a server, it starts from stage-one. The production control variable is upper bounded by number of servers that are not at stage-two and lower bounded by number of active servers at phase-one since order cancellation is not allowed. Based on the above definitions, optimal cost-to-go function J is given by

$$J(x_1, x_2, x_3) = \frac{1}{\nu + \alpha} \min_{x_1 \leq u \leq s - x_2} \{hx_3 + K(u - x_1)$$
$$+ u\mu_1 \beta J(u - 1, x_2 + 1, x_3)$$
$$+ u\mu_1 (1 - \beta) min\{J(u - 1, x_2, x_3 + 1), J(u, x_2, x_3 + 1)\}$$
$$+ x_2 \mu_2 min\{J(u, x_2 - 1, x_3 + 1), J(u + 1, x_2 - 1, x_3 + 1)\}$$
$$+ (s(\mu_1 + \mu_2) - u\mu_1 - x_2\mu_2)J(u, x_2, x_3) + \lambda L(u, x_2, x_3)\} \quad (2)$$

where L is the lost sales operator expressed by

$$L(x_1, x_2, x_3) = \begin{cases} J(x_1, x_2, x_3 - 1), & x_3 > 0 \\ c + J(x_1, x_2, 0), & x_3 = 0 \end{cases} \quad (3)$$

We aim to identify how many production servers should be active/busy at any given state to minimize the expected discounted system cost. The minimization operation defined with rate $u\mu_1(1 - \beta)$ corresponds to the decision at the time of production completion at phase-one: it decides whether to continue production on the server that has just finished processing at the first phase and replenished inventory. The next optimizer, recalled with rate $x_2\mu_2$, is to decide whether to continue production on the server that has just finished processing at the second phase and replenished inventory. One should note that if fixed production cost is zero, these two continuation operators are redundant because the system can reactivate any server with zero cost whenever needed.

The term $(s(\mu_1 + \mu_2) - u\mu_1 - x_2\mu_2)J(u, x_2, x_3)$ is necessary for the fictitious self-transitions due to the uniformization. In equation (3), the operator L corresponds to the transitions triggered by demand arrivals: if there is inventory on-hand, it is decreased by one, otherwise lost sales cost is incurred and state remains the same.

4. Characterization of the optimal policy

In this section, we provide a numerical characterization of the optimal production policy under average system cost. Since the system dynamics can be very clearly expressed with discounted cost DPs, we developed our formulation accordingly. However, we conduct numerical studies under average system cost criteria in order to make the performance measure independent of the initial state and the discount factor. We apply the value iteration algorithm to the system defined by equations (2) and (3) with discount rate $\alpha = 0$. Average system cost is calculated as the convergent value of the ratio of the optimal cost-to-go function value and the number of iterations.

Gavish and Graves [4] shows that two-critical-number policy is optimal for backordering $M/G/1$ make-to-stock systems with fixed start-up cost. This policy dictates that production should be triggered when the inventory level drops to the lower control level (I^*) and it should be continued until the inventory level reaches to the upper control level (I^{**}). In Section 4.1, we numerically show that this optimal policy structure is also preserved for lost sales $M/Cox_2/1$ systems. On the

other hand, the numerical studies in Section 4.2 illustrates the dynamic nature of the optimal policy for multi-server systems.

4.1. Single server systems

Single server cases are relatively easy to handle because at any state production decision u is either 0 or 1. For the numerical study, we first define a base case as $[\mu_1, \mu_2, \beta, h, \lambda, c] = [3.25, 1.75, 0.15, 3, 6, 3]$. In this subsection, we set $s = 1$ and provide the results while we are changing K or λ. We first assume $K = 0$ and represent the optimal production decisions in Table 1. Rows are for the first two state variables, which are x_1 = the number of active servers at stage-one and x_2 = the number of active servers at stage-two. The columns are for the last state variable, x_3 = the inventory level. The numbers at the intersection of the row and the column axes represent the corresponding optimal decision.

Table 1. Optimal production policy: s=1, K=0

$u^*(x_1, x_2, x_3)$				x_3			
(x_1, x_2)	**0**	**1**	**2**	**3**	**4**	**5**	**6**
[0, 0]	1	1	1	1	0	0	0
[1, 0]	1	1	1	1	1	1	1
[0, 1]	0	0	0	0	0	0	0

The optimal decision at state (x_1, x_2, x_3), denoted by $u^*(x_1, x_2, x_3)$, is the optimal number of busy servers at phase-1 as explained in Section 3. For instance, $u^*(0,0,0) = 1$ implies that the server, which is currently idle, should be activated if the inventory level is zero. Since $K = 0$ and continuation decisions are redundant, Table 1 fully characterizes the optimal policy. For single server systems the decision is trivial at states $(1,0,x_3)$ and $(0,1,x_3)$. At such states the server is already busy (there is no idle server to activate) and the decision is automatically x_1. Hence, control is only for the states of $(0,0,x_3)$ type. Table 1, which is an example case for our extensive numerical study, shows that the optimal production policy is of base stock type: it is optimal to produce below the maximum inventory level (base stock level) and not to produce otherwise. For the setting considered in the table, optimal base stock level, BS^*, is 4.

However, when there is fixed start-up cost optimal production policy cannot be described with a single parameter as Gavish and Graves [4] shows for single-server backordering systems. As exemplified in Table 2, our numerical studies depict that *two-critical-number policy* is optimal also for lost sales systems. In the u^* part of the table, it is seen that production is started when inventory level drops to 2. When $K > 0$, in addition to the number of active servers decision, the continuation decisions are also required and provided in c_1^* and c_2^* parts of Table 2. Referring to the DP model of Section 3, we define c_1^* and c_2^* as follows: $c_1^* = 1$ if $min\{J(u - 1, x_2, x_3 + 1), J(u, x_2, x_3 + 1)\} =$

$J(u, x_2, x_3 + 1)$, i.e. it is optimal to immediately start new production at stage-1 when the whole production process has completed after stage-1 (without visiting stage-2). Similarly, $c_2^* = 1$ if $min\{J(u, x_2 - 1, x_3 + 1), J(u + 1, x_2 - 1, x_3 + 1)\} = J(u + 1, x_2 - 1, x_3 + 1)$ corresponding to the event where the production process has completed after stage-2 and it is optimal to immediately start new production on stage-1 of the process. Otherwise, the server that has just finished processing is turned-off and, c_1^* and c_2^* are set to zero. By definition, $c_1^*(x_1, x_2, x_3)$ and $c_2^*(x_1, x_2, x_3)$ are relevant only when $x_1 > 0$ and $x_2 > 0$, respectively. Otherwise continuation decisions are not applicable (NA). As seen from Table 2, $c_1^*(1,0, x_3) = c_2^*(0,1, x_3)$ for all inventory levels x_3. This holds because there is only one available server and the inventory level just after production completion would be the same independent of the last stage visited. If it is optimal to continue production on the server, then the new process is going to start at stage-1 in any case.

For the setting considered in Table 2 the parameters of the *two-critical-number policy* are $(I^*, I^{**}) = (2, 6)$ where I^* is the production trigger level and I^{**} is the maximum inventory level that the system reaches.

Table 2. Optimal production decisions, s=1, K=2

$u^*(x_1, x_2, x_3)$				x_3			
(x_1, x_2)	**0**	**1**	**2**	**3**	**4**	**5**	**6**
[0, 0]	1	1	1	0	0	0	0
[1, 0]	1	1	1	1	1	1	1
[0, 1]	0	0	0	0	0	0	0
$c_1^*(x_1, x_2, x_3)$				x_3			
(x_1, x_2)	**0**	**1**	**2**	**3**	**4**	**5**	**6**
[0, 0]	NA	NA	NA	NA	NA	NA	NA
[1, 0]	1	1	1	1	1	1	0
[0, 1]	NA	NA	NA	NA	NA	NA	NA
$c_2^*(x_1, x_2, x_3)$				x_3			
(x_1, x_2)	**0**	**1**	**2**	**3**	**4**	**5**	**6**
[0, 0]	NA	NA	NA	NA	NA	NA	NA
[1, 0]	NA	NA	NA	NA	NA	NA	NA
[0, 1]	1	1	1	1	1	1	0

After the characterization of the optimal policy we next show in Table 3 how the optimal policy parameters react to changes in traffic intensity and production start-up cost. We change the demand rate while keeping Coxian processing time parameters constant to obtain settings with different traffic intensity (ρ). The effect of Coxian parameters is discussed in Section 5.

Table 3 reveals that the optimal policy parameters are non-decreasing in ρ. At lower ρ values system prefers not to produce at all. That is, the optimal values of the policy parameters are all zero and corresponding average system cost equals λc. On the other hand, it is optimal to produce at some inventory levels beyond

certain traffic intensity and thus BS^* and the vector (I^*, I^{**}) are not zero for $K = 0$ and $K > 0$ cases, respectively. When $K > 0$, as the traffic getting heavier the increase in I^{**} is more pronounced than the increase in I^* because the system needs to hold more inventory to meet the increasing demand. For fixed ρ, a similar behavior is observed as the start-up cost K increases: in order to decrease the frequency of production start-up (so the total fixed cost) and to continue with the activated server as much as possible, the gap between the maximum inventory level and the production trigger point, $I^{**} - I^*$, is getting wider.

Table 3. Optimal policy parameters, $s = 1$

		$K = 0$		$K = 1$			$K = 2$			$K = 3$		
λ	ρ	AC	BS^*	AC	I^*	I^{**}	AC	I^*	I^{**}	AC	I^*	I^{**}
0.50	0.29	1.50	0	1.50	0	0	1.50	0	0	1.50	0	0
0.75	0.43	2.25	0	2.25	0	0	2.25	0	0	2.25	0	0
1.00	0.58	3.00	1	3.00	0	0	3.00	0	0	3.00	0	0
1.50	0.86	3.70	1	4.50	0	1	4.50	0	0	4.50	0	0
2.00	1.15	4.61	1	5.21	0	2	5.56	0	2	5.86	0	3
2.50	1.44	5.66	1	6.06	0	2	6.29	0	3	6.46	0	3
3.00	1.73	6.66	2	6.95	0	3	7.11	0	3	7.19	0	4
3.50	2.01	7.73	2	7.94	1	4	8.00	1	4	8.05	0	5
4.00	2.30	8.90	2	8.99	1	4	9.02	1	5	9.04	1	5

4.2. Multi-server systems

Although the structure of the optimal production policy is known for single server make-to-stock systems, it has not yet been fully characterized for multiple server systems. To the best of our knowledge, the only study addressing the production control of multi-server systems is Bulut and Fadıloğlu [8] and they only provide partial characterization of the policy for the $M/M/s$ case without fixed cost. In this section, we provide numerical analyses to describe the structure of the optimal policy for the $M/Cox_2/s$ make-to-stock systems with fixed cost for the first time in the literature. Single server assumption relatively eliminates the complexity because for such cases the decision is 0-1 for all inventory levels: whether to activate the only available server or not. However, when $s > 1$, the controller should decide how many servers should be active at any system state. Furthermore, this decision would be dependent on the status of the ongoing production, i.e. to the stage/phase information of the active servers.

Recalling the base case, we first set $s = 3$ and $K = 0$ and provide the optimal decisions in Table 4(a). Similar to the single-server case, u^* matrix is enough to describe the optimal policy when the production start-up cost is zero. We separate the decision matrix into four layers where each layer corresponds to a particular total number of active servers. In general, if there are s available servers, there would be $(s + 1)$ layers. For the setting presented in Table 4(a), we list our observations on the structure of the optimal policy below:

i. Since all the available servers are busy at the bottom layer, i.e. $x_1 + x_2 = s = 3$, and order cancellation cost is practically infinite, the optimal decision is trivial at all states of the bottom layer: $u^*(x_1, x_2, x_3) = x_1$.

ii. Production decisions are non-increasing in inventory level x_3, because shortage risk is reduced by increasing inventory.

iii. Unlike the classical static policies, which are based on either inventory level or inventory position (e.g. base stock), unit increase in inventory level does not always end up with unit decrease in the optimal number of active servers at phase-one, e.g. $u^*(0,0,1) = 2$ but $u^*(0,0,2) = 0$.

iv. In addition to (*iii*), there is a second level of dynamicity in the structure of the optimal policy; decisions are dependent on the status of ongoing production. One would expect that as the number of completed production stages increases, the total number of active servers decreases or remains the same. This is true for the processing time random variables having increasing failure rate (IFR) such as Erlangian production times. For such settings, as the number of completed stages increases remaining time to replenish the inventory stochastically decreases. However, for the case considered in Table 4(a), Coxian production time random variable has the parameters $(\mu_1, \mu_2, \beta) = (3.25, 1.75, 0.15)$ and more channels are needed if the item being processed visits stage-2. Since stage-1 is much faster than stage-2 and probability of visiting stage-2 is small, expected time to production completion is smaller when the current production is at stage-1 compared to the case where it is at stage-2. In order to make it clearer, let us consider the states $(1,0,1)$ and $(0,1,1)$ of Table 4(a): the inventory level is the same for both of the states but the (only) active server is at stage-1 in the first state and at stage-2 in the second. As seen from the table, $u^*(1,0,1) = u^*(0,1,1) = 2$ and the transitions are to $(2,0,1)$ and $(2,1,1)$ from $(1,0,1)$ and $(0,1,1)$, respectively. Before the decisions, both states have the same number of active servers, which is one, but after the transitions state $(2,1,1)$ has one more active server than $(2,0,1)$.

We increase μ_2 from 1.75 to 7.5 in Table 4(b) and observe completely different production decisions for the states $(1,0,1)$ and $(0,1,1)$: $u^*(1,0,1) = 2$, $u^*(0,1,1) = 0$ an the transitions are to $(2,0,1)$ and $(0,1,1)$ from $(1,0,1)$ and $(0,1,1)$, respectively That is, this time it is optimal to have more active servers when the current production is at phase-1.

v. With its three parameters Coxian production time random variable has the flexibility to obtain increasing and decreasing failure rate (IFR or DFR)

settings and, we show in Table 4 (a) and (b) that the structure of the optimal policy changes accordingly.

Table 4. Optimal production decisions s=3, K=0
(a) DFR, (b) IFR

(a) u^*			x_3				(b) u^*			x_3			
(x_1, x_2)	*0*	*1*	*2*	*3*	*4*	*5*	(x_1, x_2)	*0*	*1*	*2*	*3*	*4*	*5*
[0, 0]	3	2	0	0	0	0	**[0, 0]**	3	2	0	0	0	0
[1, 0]	3	2	1	1	1	1	**[1, 0]**	3	2	1	1	1	1
[0, 1]	2	2	0	0	0	0	**[0, 1]**	2	0	0	0	0	0
[2, 0]	3	2	2	2	2	2	**[2, 0]**	3	2	2	2	2	2
[1, 1]	2	2	1	1	1	1	**[1, 1]**	2	1	1	1	1	1
[0, 2]	1	1	0	0	0	0	**[0, 2]**	1	0	0	0	0	0
[3, 0]	3	3	3	3	3	3	**[3, 0]**	3	3	3	3	3	3
[2, 1]	2	2	2	2	2	2	**[2, 1]**	2	2	2	2	2	2
[1, 2]	1	1	1	1	1	1	**[1, 2]**	1	1	1	1	1	1
[0, 3]	0	0	0	0	0	0	**[0, 3]**	0	0	0	0	0	0

Fixed production cost adds more complexity to the structure of the optimal policy. To reveal this, one can compare Table 4(a) and Table 5 where the only difference is the value of the start-up cost K. When fixed cost is larger optimal policy tends to activate less

servers at all the states. Specifically, fixed cost prevents activating all the available servers even there is no inventory on hand. On the other hand, the optimal policy balance the holding and shortage trade-off mostly with the continuation decisions c_1^* and c_2^*; production continues with the previously activated servers for some time. However, continuation decisions are also state dependent and are not only determined by the total number of active servers. As opposed to the single-server case shown in Table 2, there exists a, b and x_3 values such that $c_1^*(a, b, x_3) \neq c_2^*(b, a, x_3)$. For instance, $c_1^*(2,0,3) = 0$ but $c_2^*(0,2,3) = 1$. This dynamic behavior of the optimal policy is due to the fact that any active server at stage-1 can replenish the inventory by two different realizations: with probability $(1 - \beta)$ inventory is replenished directly from stage-1, but with probability β stage-2 is visited and then the production is completed. On the other hand, any active server at stage-2 has only one possible realization path to replenish the inventory. Hence, when $s > 1$, continuation decisions are coupled with the number of active servers decision and depending on the values of the Coxian parameters (μ_1, μ_2, β), continuation decisions might be different even for the symmetric states at the same inventory level.

Table 5. Optimal production decisions, s=3 and K=2

u^*			x_3				c_1^*			x_3				c_2^*			x_3			
(x_1, x_2)	*0*	*1*	*2*	*3*	*4*	*5*	(x_1, x_2)	*0*	*1*	*2*	*3*	*4*	*5*	(x_1, x_2)	*0*	*1*	*2*	*3*	*4*	*5*
[0, 0]	2	1	0	0	0	0	**[0, 0]**	NA	NA	NA	NA	NA	NA	**[0, 0]**	NA	NA	NA	NA	NA	NA
[1, 0]	2	1	1	1	1	1	**[1, 0]**	1	1	1	1	1	0	**[1, 0]**	NA	NA	NA	NA	NA	NA
[0, 1]	1	0	0	0	0	0	**[0, 1]**	NA	NA	NA	NA	NA	NA	**[0, 1]**	1	1	1	1	1	0
[2, 0]	2	2	2	2	2	2	**[2, 0]**	1	1	1	0	0	0	**[2, 0]**	NA	NA	NA	NA	NA	NA
[1, 1]	1	1	1	1	1	1	**[1, 1]**	1	1	1	1	0	0	**[1, 1]**	1	1	1	0	0	0
[0, 2]	0	0	0	0	0	0	**[0, 2]**	NA	NA	NA	NA	NA	NA	**[0, 2]**	1	1	1	1	0	0
[3, 0]	3	3	3	3	3	3	**[3, 0]**	1	1	0	0	0	0	**[3, 0]**	NA	NA	NA	NA	NA	NA
[2, 1]	2	2	2	2	2	2	**[2, 1]**	1	1	0	0	0	0	**[2, 1]**	1	1	0	0	0	0
[1, 2]	1	1	1	1	1	1	**[1, 2]**	1	1	0	0	0	0	**[1, 2]**	1	1	0	0	0	0
[0, 3]	0	0	0	0	0	0	**[0, 3]**	NA	NA	NA	NA	NA	NA	**[0, 3]**	1	1	0	0	0	0

As the discussion on tables 4(a), 4(b) and 5 exhibits, for the multi-server systems, optimal policy is highly dynamic/state-dependent and cannot be fully described with two static parameters such as inventory level or inventory position. The highly dynamic structure of the optimal policy would reduce its value for practitioners. In practice, controllers are mostly after easy-to-apply approximate policies. We therefore propose an alternative production policy that is controlled by two parameters and can quickly adapt itself to IFR and DFR cases. Next two sections are devoted to the introduction and performance evaluation of our policy.

5. An alternative policy structure

As we have discussed in Section 4, the values of the Coxian parameters directly affect the structure of the optimal policy. In order to first guarantee that our policy structure responds to the changes in input parameters (μ_1, μ_2, β), we define $E(1,0,x_3)$ and $E(0,1,x_3)$ as the expected remaining production times if the current production is on stage-1 and stage-2, respectively. Since the stages are memoryless, $E(1,0,x_3) = \frac{1}{\mu_1} + \beta \frac{1}{\mu_2}$ and $E(0,1,x_3) = \frac{1}{\mu_2}$. We aim to identify IFR and DFR cases by comparing these expected times to production completion. $E(0,1,x_3) < E(1,0,x_3)$ or equivalently $r = \frac{E(0,1,x_3)}{E(1,0,x_3)} < 1$ implies that

expected remaining production time decreases when stage-2 is visited. For such settings, since the probability of demand arrivals before inventory replenishment decreases, our policy should demotivate activating new servers when stage-2 is visited. Otherwise, we are in a DFR case and the policy should motivate (or at least should not demotivate) activating new servers when stage-2 is visited.

Second, for the sake of applicability we aim to propose a policy structure that can be controlled by only two parameters. We stick to the notation used in Section 4: I^* and I^{**} are the production trigger and the maximum levels, respectively. This two-critical-number policy is optimal for single-server settings and the parameters of the policy are defined in terms of inventory level. So as to better capture the dynamic nature of the optimal policy of an $M/Cox_2/s$ make-to-stock system, we define I^* and I^{**} in terms of a function of the system state vector referred as *inventory status* (*IS*). At any state (x_1, x_2, x_3),

$$IS(x_1, x_2, x_3) = \sum_{i=1}^3 a_i x_i \tag{4}$$

where a_1, a_2 and a_3 are the weights of the number of active servers at stage-1, the number of active servers at stage-2 and inventory level, respectively. The above definition of inventory status allows us to trace a policy space including the classical inventory level ($IS = IL$ when $(a_1, a_2, a_3) = (0,0,1)$) and inventory position ($IS = IP$ when $(a_1, a_2, a_3) = (1,1,1)$) based policies.

Based on the above discussion, we propose the below policy structure that computes $u(x_1, x_2, x_3)$ = the number of active/busy servers at stage-1, $c_1(x_1, x_2, x_3)$ = continuation decision for the server that has just finished stage-1 and replenished inventory, and $c_2(x_1, x_2, x_3)$ = continuation decision for the server that has just finished stage-2 and replenished inventory.

$$u(x_1, x_2, x_3) = \begin{cases} \lfloor min\{(I^* + 1) - IS(x_1, x_2, x_3) + x_1, (s - x_2)\} \rfloor, & IS \le I^* \\ x_1 & , IS > I^* \end{cases} \tag{5}$$

$$c_1(x_1, x_2, x_3) = \begin{cases} 1, & IS(x_1 - 1, x_2, x_3 + 1) < I^{**} \\ 0, & IS(x_1 - 1, x_2, x_3 + 1) \ge I^{**} \end{cases} \tag{6}$$

$$c_2(x_1, x_2, x_3) = \begin{cases} 1, & IS(x_1, x_2 - 1, x_3 + 1) < I^{**} \\ 0, & IS(x_1, x_2 - 1, x_3 + 1) \ge I^{**} \end{cases} \tag{7}$$

For the states whose inventory status is at or below the production trigger level I^*, the proposed policy tries to raise *IS* to $(I^* + 1)$. This can only be achieved with $(I^* + 1) - IS(x_1, x_2, x_3)$ many new active servers at stage-1 additional to x_1. However, as discussed in the dynamic programming formulation of Chapter 3, $u(x_1, x_2, x_3)$ is bounded above by $(s - x_2)$. In Equation (5), $\lfloor \cdot \rfloor$ is to return the nearest integer for the calculated value as the number of active servers at stage-1. For the other states, $IS(x_1, x_2, x_3) > I^*$ and we do no nothing: $u(x_1, x_2, x_3)$ returns the current number of busy servers at stage-1.

Continuation decisions of the policy are defined by (6) and (7), which are only applicable when $x_1 > 0$ and $x_2 > 0$, respectively. The policy keeps the previously activated servers busy until target level I^{**} is reached. In (6) and (7), decisions are given just after production completion (in c_i, x_i is decreased by 1, $i = 1,2$) and inventory replenishment (x_3 is increased by 1).

The above defined policy structure has three weight and two control parameters: (a_1, a_2, a_3) and (I^*, I^{**}). First we develop the following approach to find the setting specific values of the weights (a_1, a_2, a_3): We structure our policy based on the relative values of a_i's. Thereby, the degrees of freedom of finding the values of a_i's is decreased to two. Without loss of generality, we set the value of an active server at stage-1 to 1, i.e.,

$$a_1 = 1 \tag{8}$$

Then, the weight of an on-hand inventory relative to the weight of an outstanding order at stage-1 is calculated as:

$$a_3 = \frac{\frac{1}{\mu_1} + \beta \frac{1}{\mu_2}}{\frac{1}{\mu_1}} = \frac{\beta \mu_1 + \mu_2}{\mu_2} \tag{9}$$

where $\frac{1}{\mu_1} + \beta \frac{1}{\mu_2}$ and $\frac{1}{\mu_1}$ are the expected time to complete the whole production and stage-1, respectively. That is, if an item at stage-1, which is going to spend (on the average) $\frac{1}{\mu_1}$ time units to complete the stage, has the weight a_1, then the relative weight of an item in the inventory, which has on the average spent $\frac{1}{\mu_1} + \beta \frac{1}{\mu_2}$ time units in the production facility, is $\frac{\frac{1}{\mu_1} + \beta \frac{1}{\mu_2}}{\frac{1}{\mu_1}}$ times a_1. The weight of an item at stage-2, a_2, on the other hand, is set to different values for IFR and DFR cases. As discussed in Section 4, depending on the values of Coxian parameters (μ_1, μ_2, β) more active servers might be needed if the item being processed visits stage-2. Our policy structure gains this flexibility with a_2. We let

$$a_2 = \begin{cases} \frac{a_1 + a_3}{2}, & if\ r = \frac{E(0,1,x_3)}{E(1,0,x_3)} < 1 \\ 0, & if\ r \ge 1 \end{cases} \tag{10}$$

The ratio $r = \frac{E(0,1,x_3)}{E(1,0,x_3)}$ is less than 1 if the expected time to production completion decreases when stage-2 is visited. For such IFR cases, we set the weight of an outstanding order at stage-2 to the average of the weights of the items that are at stage-1 and in the inventory. In this way, for the IFR cases we obtain a weight structure satisfying $a_1 < a_2 < a_3$.

On the other hand, if the case is DFR, the weight is set to 0 in order to motivate the system to activate more

servers whenever the slower stage (stage-2) is visited. In this case, $a_2 < a_1 < a_3$.

One can prefer to select the "best" values of (a_1, a_2, a_3) using an optimization routine applied over the DP formulation. However, the next section shows that the performance of the proposed structure under our *intelligent guesses* (8), (9) and (10) is very close to the optimal's. That is, without undertaking the computation cost of any optimization algorithm, we obtain a very good approximation to the "ideal" weights by exploring the structure of the optimal policy (in Section 4) and selecting the values accordingly.

On the other hand, it is hard to develop a similar intuition for the control levels (I^*, I^{**}). We therefore use our DP formulation (2) and (3) as the optimization routine: For given values of (I^*, I^{**}) such that $I^* < I^{**}$, DP is fed with the decision set of the proposed policy, (5), (6) and (7), and the value iteration algorithm is run to calculate the average system cost. We then search for the optimal values of the parameters in the integer domain. As the results presented in Section 6 show that optimizing (I^*, I^{**}) in the integer space results in a well-performing near-optimal policy.

It should be noted that a_1 can also be set to any arbitrary positive value. In such cases, the values of a_2, a_3 and thus *IS* would also be changed relative to a_1. Hence, the optimal values of (I^*, I^{**}) would be also altered/shifted in order to find the same cost minimizer $u(x_1, x_2, x_3)$ values. That is, larger values of a_i's would result in larger values of (I^*, I^{**}) so as to find the same u value.

Although we obtain the results in a reasonable amount of time one can further fasten the routine if the search first visits the space around $(I^{**} - I^*) = EOQ$. The approximate value of the batching decision of the classical inventory systems would here help us to capture the effect of the fixed cost on the length of the non-production period. One should note that our make-to-stock production-inventory environment is different than the classical inventory settings in terms of capacity (there are only *s* many servers) and one-at-a-time replenishment as the active servers complete production. A classical inventory system having stochastic lead-times that is controlled by lot-for-lot policy can be modeled using our approach only if *s* tends to infinity, which requires to guarantee an uncapacitated system.

6. Performance evaluation of the proposed policy

In this section we present the numerical study assessing the performance of the policy structure described by (5), (6) and (7). We test the performance of the structure with the (a_1, a_2, a_3) values given in (8), (9) and (10), which defines the specific policy that we propose. We also evaluate the performance of the inventory position based static policy (IP Policy) in order to quantify the value of dynamic state information. Our policy structure already has the flexibility to cover the IP Policy: in (4), we let $(a_1, a_2, a_3) = (1,1,1)$ to obtain the

inventory position as the sum of the on-hand inventory and the number of outstanding production orders (the number of items that are being processed).

The main goal of this section is to reveal the effects of Coxian parameters (μ_1, μ_2, β), the demand rate λ and the fixed cost K on the performances of the considered policies. The results of the numerical study are summarized in the tables provided at the end of the section. While changing the above mentioned parameters, without loss of generality we fix the values of the holding cost rate h and the unit lost sales cost c to 3. Each table includes five different instances with different traffic intensities (ρ) ranging from 0.50 to 1.50. For each instance, average costs of the optimal, proposed and IP policies, and their optimal control levels (I^*, I^{**}) are reported. For the proposed and IP policies, the optimality gap, defined as the percent cost deviation from the optimal, is also provided.

Table 6 shows the results when there are two parallel servers with no start-up costs and Coxian production times have decreasing failure rates (DFR). Our dynamic policy performs very well in the environment of Table 6. The optimality gap of the proposed policy is less than 0.5% at all the instances of the table. Furthermore, IP Policy is also a notable alternative of the optimal policy when the capacity is tight: when there are limited of number of servers or traffic intensity is high. As ρ increases or equivalently as the capacity is getting tighter, more and more servers would be activated independent of the status of the production. That is, all the plausible policies, including the optimal one, utilize all the servers at higher traffic intensity values. This observation is valid not just for Table 6 but for all the tables of the chapter: The proposed policy is near optimal at all the traffic intensities of all the considered cases, and IP Policy is a second alternative for the highly utilized systems.

Table 7 and Table 8 are the 'positive fixed cost' and 'more server' extensions of Table 6, respectively. When the production startup cost K is 0.5, the maximum optimality gap of the Proposed Policy is 2.55% and of the IP Policy is 3.22%. Both maximums are observed at the same instance where $\rho = 0.5$. For the systems with higher server activation cost, the distance between the upper and lower control limits, which is $(I^{**} - I^*)$, should be larger. That is, instead of activating servers at higher inventory levels it is more economical to increase I^{**} in order to both postpone the production cycle and to continue production on the previously activated servers once it is started until reaching I^{**} again.

On the other hand, in Table 8, the number of available servers is higher (and fixed cost is zero) and the maximum deviations from the optimality are 3.21% and 6.89% for the Proposed and IP policies, respectively. However, for the Proposed Policy, the average of the five optimality gaps reported in the table is only 1.54%, which is the highest average among all the tables. Since the Proposed Policy makes use of the

dynamic production status information of all the servers, it outperforms the static IP policy as the number of available servers increases.

Table 9 is for the IFR version of Table 8. The table shows that the optimality gap of the Proposed Policy is below 1% at all the instances. Although the performance of IP Policy is also improved from Table 8 to Table 9, that improvement is not as significant as the improvement attained by the Proposed Policy.

At all the tables from 6 to 9, the traffic intensity (ρ) is

increased by decreasing μ_1, the processing rate of stage-1. In order to eliminate any bias that can be due to this method, we reconsider the environment of Table 8 where $s = 5$ and $K = 0$, and increase the traffic intensity by decreasing μ_2 this time. The results are reported in the last table, Table 10. The probability of a WIP item being at phase-2 increases as μ_2 decreases that sharpens the DFR nature of the production times. Due to this fact, the performance of our policy is better than Table 8 in Table 10.

Table 6. Performances of the alternative policies: DFR cases with s=2 and K=0

			Optimal Policy	IP Policy			Proposed Policy		
#	ρ	(μ_1, μ_2, β)	Average Cost	Average Cost	(I^*, I^{**})	Optimality Gap %	Average Cost	(I^*, I^{**})	Optimality Gap %
1	0.5	(15, 0.5, 0.05)	7.05	7.20	(1, 2)	2.03	7.09	(1, 7)	0.48
2	0.75	(6.65, 0.5, 0.05)	8.21	8.49	(1, 2)	3.32	8.21	(2, 3)	0.00
3	1.00	(4.25, 0.5, 0.05)	9.19	9.26	(2, 3)	0.70	9.24	(2, 5)	0.45
4	1.25	(3.15, 0.5, 0.05)	9.98	10.01	(2, 3)	0.27	9.99	(3, 4)	0.09
5	1.50	(2.50, 0.5, 0.05)	10.70	10.72	(2, 3)	0.19	10.71	(3, 4)	0.10

Table 7. Performances of the alternative policies with DFR distribution, s=2 and K=0.5

			Optimal Policy	IP Policy			Proposed Policy		
#	ρ	(μ_1, μ_2, β)	Average Cost	Average Cost	(I^*, I^{**})	Optimality Gap %	Average Cost	(I^*, I^{**})	Optimality Gap %
1	0.5	(15, 0.5, 0.05)	8.82	9.12	(1, 5)	3.22	9.06	(1, 8)	2.55
2	0.75	(6.65, 0.5, 0.05)	9.71	9.79	(1, 6)	0.80	9.72	(2, 6)	0.03
3	1.00	(4.25, 0.5, 0.05)	10.24	10.28	(1, 6)	0.45	10.32	(1, 7)	0.80
4	1.25	(3.15, 0.5, 0.05)	10.77	10.78	(1, 7)	0.16	10.80	(2, 8)	0.30
5	1.50	(2.50, 0.5, 0.05)	11.20	11.21	(2, 7)	0.09	11.20	(2, 8)	0.02

Table 8. Performances of the Alternative Policies with DFR distribution, s=5 and K=0

			Optimal Policy	IP Policy			Proposed Policy		
#	ρ	(μ_1, μ_2, β)	Average Cost	Average Cost	(I^*, I^{**})	Optimality Gap %	Average Cost	(I^*, I^{**})	Optimality Gap %
1	0.5	(2.79, 0.5, 0.05)	7.85	8.43	(3, 4)	6.89	7.96	(3, 4)	1.47
2	0.75	(1.90, 0.5, 0.05)	8.47	8.92	(4, 7)	5.01	8.75	(3, 4)	3.21
3	1.00	(1.35, 0.5, 0.05)	9.29	9.43	(4, 5)	1.47	9.38	(4, 5)	0.99
4	1.25	(1.06, 0.5, 0.05)	9.97	10.09	(5, 8)	1.20	10.11	(5, 6)	1.37
5	1.50	(0.87, 0.5, 0.05)	10.67	10.70	(5, 8)	0.32	10.74	(5, 8)	0.68

Table 9. Performances of the Alternative Policies with IFR distribution, s=5 and K=0

			Optimal Policy	IP Policy			Proposed Policy		
#	ρ	(μ_1, μ_2, β)	Average Cost	Average Cost	(I^*, I^{**})	Optimality Gap %	Average Cost	(I^*, I^{**})	Optimality Gap %
1	0.5	(8.50, 2.65, 0.8)	7.97	8.33	(3, 6)	4.26	8.05	(7, 12)	0.99
2	0.75	(3.90, 2.65, 0.8)	8.46	8.81	(3, 6)	3.93	8.52	(5, 9)	0.75
3	1.00	(1.88, 2.65, 0.8)	9.27	9.40	(4, 7)	1.39	9.33	(6, 8)	0.62
4	1.25	(1.35, 2.65, 0.8)	9.88	9.94	(5, 7)	0.58	9.91	(6, 7)	0.24
5	1.50	(1.05, 2.65, 0.8)	10.52	10.57	(5, 7)	0.48	10.55	(7, 8)	0.26

Table 10. Performances of the Alternative Policies with DFR distribution, s=5, K=0 and μ_2 is changing

			Optimal Policy	IP Policy			Proposed Policy		
#	ρ	(μ_1, μ_2, β)	Average Cost	Average Cost	(I^*, I^{**})	Optimality Gap %	Average Cost	(I^*, I^{**})	Optimality Gap %
1	0.5	(14, 2.30, 0.8)	7.83	8.33	(3, 6)	6.00	7.93	(4, 11)	1.32
2	0.75	(14, 1.44, 0.8)	8.44	8.92	(4, 7)	5.34	8.51	(4, 16)	0.81
3	1.00	(14, 1.05, 0.8)	9.24	9.40	(4, 7)	1.73	9.28	(12, 23)	0.49
4	1.25	(14, 0.82, 0.8)	10.02	10.13	(4, 7)	1.10	10.02	(16, 28)	0.06
5	1.50	(14, 0.68, 0.8)	10.74	10.84	(5, 8)	0.92	10.74	(19, 33)	0.02

As the holding cost rate (h) and unit lost sales cost (c) are both set to 3 in all the above examples, we aim to depict the effect of changes in the (holding cost rate)/(unit lost sales cost) ratio in Figure 2. In order to visit different values of this ratio, without loss of generality we only change holding cost rate: h varies from 1 to 14 while c is kept constant at 3. The other parameters are assumed to be $[s, \mu_1, \mu_2, \beta, K, \lambda] = [2, 4.25, 0.5, 0.05, 0.5, 6]$. In the figure, for each increment of h, average system cost of both the optimal and proposed policies, and (I^*, I^{**}) values of the proposed policy are presented. As seen from the height of the bars representing the average system costs of the policies the performances are so close to each other: average and maximum optimality gaps are calculated as 0.25% and 0.90%, respectively. Furthermore, for both of the policies average system cost is concave in h that converges to a certain value (18) when h is above 12. As h increases and becomes larger relative to c, both policies demotivate production. In our example, when $h > 12$ it is optimal not to produce at all. In this case all the incoming demands are lost and the average system cost converges to $\lambda c = 18$ for both of the policies. In parallel to this observation, it is also seen from the figure that both of the optimal control parameters of the proposed policy, which are defined by (I^*, I^{**}), are non-increasing in h. Equivalently we can say that they would be non-decreasing in c. On the hand, when holding cost rate is getting smaller and smaller (compared to the unit lost sales cost) the average cost converges to zero. As h decreases both I^* and I^{**} increase in order to minimize shortage and start-up costs. At the extreme, when $h = 0$, I^* can be set to any value above a threshold that guarantees no shortage. Similarly, I^{**} can be any value (greater than I^*) such that fixed cost per each server is incurred only finitely many times. For all such control levels the long-run average system cost would be zero.
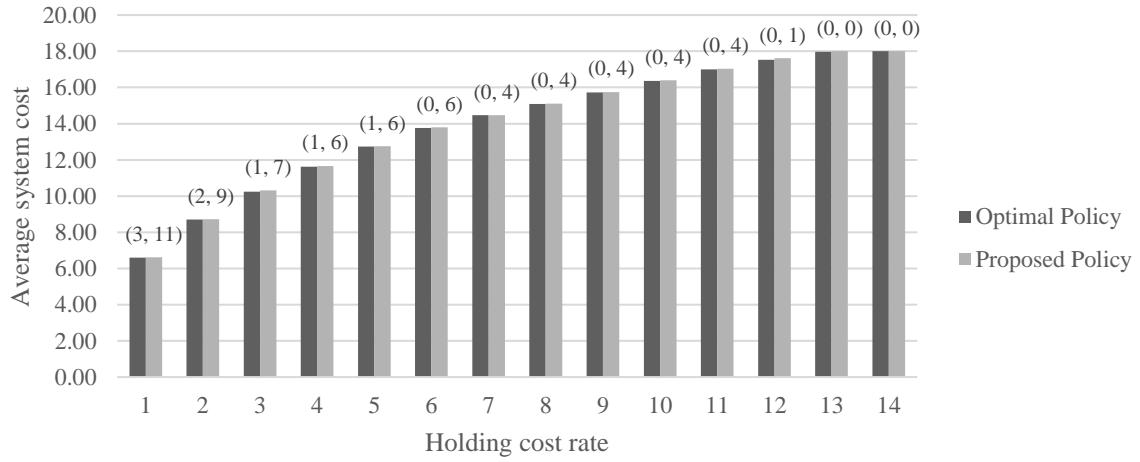
Figure 2. Effects of changes in h on the average cost and (I^*, I^{**})

7. Conclusion

This article considers a production-inventory system in a make-to-stock environment with multiple identical production channels (machines, servers or lines), fixed production start-up costs and lost sales. We assume that production times are 2-phase Coxian random variables that allows us to model rework/remanufacturing and repair operations within the production process. Demands are generated according to a stationary Poisson process and unsatisfied demands are immediately lost.

We extend the existing literature by considering phase-type production times and multiple servers with start-up costs in the same model. The system is modeled as an $M/Cox_2/s$ make-to-stock queue and dynamic programming formulation is developed. Thereafter, we first numerically characterize the optimal production policy and reveal that it has a highly dynamic nature. Secondly, we propose a policy structure that aims to capture the dynamic nature of the optimal policy with two control and three weight parameters. Control parameters are to define the maximum inventory and the production start-up levels. The other three parameters are the weights of the number of active servers at stage-1, the number of active servers at stage-2 and the number of items in inventory. This policy structure has the capability to trace a large space of several different policies. Using this structure we specifically propose a policy with fixed weight parameters and test its performance with respect to the optimal. Results reveal that our policy, which is controlled by only two parameters and thus easy-to-apply, is near optimal at all the instances.

References

[1] Gavish, B., Graves, S.C., (1980), A one-product production/inventory problem under continuous review policy, *Operations Research*, 28(5), 1228-1236.

[2] Heyman, D. P., (1968). Optimal operating policies for M/G/1 queuing systems, *Operations Research*, 16(2), 362-382.

[3] Sobel, M. J., (1969), Optimal average-cost policy for a queue with start-up and shut-down costs, *Operations Research*, 17(1), 145-162.

[4] Gavish, B., Graves, S.C., (1981), Production/Inventory Systems with a Stochastic Production Rate under a Continuous Review Policy, *Computers & Operations Research*, 8(3), 169-183.

[5] Graves, S. C., Keilson, J., (1981), The compensation method applied to a one-product production/inventory problem. *Mathematics of Operations Research*, 6(2), 246-262.

[6] Lee, H.S., Srinivasan, M.M., (1989), The Continuous Review (s, S) Policy for Production/Inventory Systems with Poisson Demands and Arbitrary Processing Times, *Technical Report*, 87-33.

[7] Lee, H.S., Srinivasan, MM., (1991), Random Review Production/Inventory Systems with Compound Poisson Demands and Arbitrary Processing Times, *Management Science*, 37(7), 813-833.

[8] Bulut, Ö., & Fadiloğlu, M. M., (2011), Production control and stock rationing for a make-to-stock system with parallel production channels, *IIE Transactions*, 43(6), 432-450.

[9] Ha, A. Y., (1997a), Inventory rationing in a make-to-stock production system with several demand classes and lost sales. *Management Science*, 43(8), 1093-1103.

[10] Ha, A.Y., (1997b), Stock rationing policy for a make-to-stock production system with two priority classes and backordering, *Naval Research Logistics*, 43, 458-472.

[11] Ha, A. Y., (2000), Stock rationing in an M/Ek/1 make-to-stock queue, *Management Science*, 46(1) 77-87.

[12] Gayon, J. P., De Vericourt, F., Karaesmen, F., (2009), Stock rationing in an M/Er /1 multi-class make-to-stock queue with backorders, *IIE Transactions*, 41(12), 1096-1109.

[13] Rafiei, H., Rabbani, M., Vafa-Arani, H., & Bodaghi, G. (2017). Production-inventory analysis of single-station parallel machine make-to-stock/make-to-order system with random demands and lead times. *International Journal of Management Science and Engineering Management*, *12*(1), 33-44.

[14] Lippman, S., (1975), Applying a new device in the optimization of exponential queuing systems, *Operations Research*, 23, 687-710.

Özgün Yücel *earned her master's degree from the Industrial Engineering Department of Yaşar University. She has been working as a research assistant and continuing her PhD studies at the same department.*
  ⓘ *http://orcid.org/0000-0003-0900-297X*

Önder Bulut *is Assistant Professor in the Department of Industrial Engineering at Yaşar University. He received his Ph.D. degree from the Industrial Engineering Department of Bilkent University. His research interests include stochastic modeling, production and inventory systems, simulation, dynamic programming and optimal control.*
  ⓘ *http://orcid.org/0000-0003-1476-6333*

RESEARCH ARTICLE

A multi-start iterated tabu search algorithm for the multi-resource agent bottleneck generalized assignment problem

Gulcin Bektur*

Department of Industrial Engineering, Iskenderun Technical University, Turkey
gulcin.bektur@iste.edu.tr

ARTICLE INFO	ABSTRACT
	In this study, a multi-resource agent bottleneck generalized assignment problem (MRBGAP) is addressed. In the bottleneck generalized assignment problem (BGAP), more than one job can be assigned to an agent, and the objective function is to minimize the maximum load over all agents. In this problem, multiple resources are considered and the capacity of the agents is dependent on these resources and it has minimum two indices. In addition, agent qualifications are taken into account. In other words, not every job can be assignable to every agent. The problem is defined by considering the problem of assigning jobs to employees in a firm. BGAP has been shown to be NP- hard. Consequently, a multi-start iterated tabu search (MITS) algorithm has been proposed for the solution of large-scale problems. The results of the proposed algorithm are compared by the results of the tabu search (TS) algorithm and mixed integer linear programming (MILP) model.

1. Introduction

Assignment problems (AP) are an important topic which is frequently studied in the literature. AP is generally considered in three classes. The simplest form of the AP, in which each agent can be assigned a job at most, is the classic AP. There are m number of agents and n number of jobs in this problem. Each job must be assigned to an agent so that the total cost is minimal. Each agent should also be assigned a job (one-to-one). Another class of the AP is generalized assignment problem (GAP). In GAP, more than one job can be assigned to an agent. Some subclasses of GAP are multi-resource generalized assignment problem (MRGAP), bottleneck generalized assignment problem (BGAP). Another class of AP is multidimensional AP. In multidimensional AP, jobs are assigned to at least two different resources. Detailed information can be reached from the literature review by Pentico [1].

In the GAP, each agent has a certain capacity. Jobs use this capacity and the capacity of the agent cannot be exceeded. In MRGAP, multiple resources are used for the completion of the jobs. Therefore, the capacity of agents depends on these resources. The capacity parameter of the agents has at least two indices due to parameter dependent on the agent and the resource. There are many applications of the MRGAP in real life.

For example, in vehicle routing problems, as vehicles are agents, and jobs are considered to be the places where vehicles should be visited, and the capacity of the vehicles depends on both the weight and the volume of the vehicle, the problem can be considered as the MRGAP [2].

In bottleneck assignment problems (BAP), the objective function is the minimization of the maximum assignment cost or maximum load over all agents. Completion time of the jobs also can be taken into account. In other words, completion time of the last job is minimized in the BAP [3].

GAP is an important problem frequently studied in literature. Studies in the literature can be categorized as studies that proposes exact algorithms and heuristic algorithms. In the studies that propose exact algorithms, the branch-bound algorithm ([4] and [5]), the cutting plane algorithm [6], the branch and price algorithm [7, 8], branch- and- cut algorithm for GAP with additional pair constraints [9] and with min- max regret criterion [10] were used. When the exact solution approaches are used, the solution time of the problem is quite prolonged. Since the GAP problem is an NP-hard problem, it is quite common to use heuristic algorithms that gives the near optimal solution in a short time [11]. In the studies using heuristic

---

*Corresponding author

algorithms, tabu search algorithm [12-14], genetic algorithm [15], bees algorithm [16], a heuristic based on Lagrangian relaxation [17, 18], LP- based heuristic [19], a hybrid heuristic based on scatter search [20], improved differential evolution algorithm [21], a parallel genetic algorithm [22] and simulated annealing algorithm [14] were used. Degroote et al. [23], poposed a methodology for selection the most suitable algorithm for GAP. Chakravarthy et al. [24], proposed a heuristic algorithm for bottleneck generalized assignment problem. For a strategic variant of GAP, approximation algorithm is proposed by Fadaei and Bichler [25]. Detailed information for GAP can be found in the literature review by Öncan [11].

Although there are many studies related to GAP, the number of studies dealing with the MRGAP is less. MRGAP is the generalization of the GAP. GAP has been shown to be NP- hard and MRGAP is also NP-hard [26]. Karsu and Azizoğlu [3], proposed a branch-bound algorithm for the multi-resource bottleneck GAP. Mazzola and Wilcox [2], proposed a three stage heuristic algorithm for the MRGAP. In the first stage, a suitable solution is obtained and at the other stages, this solution is improved. Yagiura et al. [27], proposed a new algorithm for multi-resource generalized quadratic assignment problem. In the algorithm, the path relinking approach was used in the neighborhood generation. Gavish and Pirkul [28], proposed a heuristic algorithm and branch-bound algorithm for the MRGAP. They also proposed some rules for the reduction of the problem dimensions. Yagiura et al. [29], proposed a TS-based heuristic algorithm for the MRGAP. Mitrovic-Minic and Punnen [30], proposed a heuristic algorithm based on a variable neighborhood search for the MRGAP.

The MRGAP problem is the generalized version of GAP and is a more difficult problem to solve. However, many studies in the literature propose an heuristic algorithm for GAP. Wu et al. [10], Souza et al. [20], Sethanan and Pitakaso [21], and Moussavi et al. [31] proposed an heuristic algorithm for the generalized assignment problem. Difference from the literature, in this study an heuristic algorithm is proposed for the multi-resource agent bottleneck generalized assignment problem with agent qualifications. The differences of the study from literature are agent qualifications are taken into account, a different heuristic algorithm is proposed for the larger size test problems than the problem sizes in the literature and the success of the proposed heuristic is shown through test problems by comparing with Tabu Search in the literature. The TS algorithm has been proposed by Karsu and Azizoğlu [3] in the literature for the problem of MRGAP. The proposed iterated local search algorithm is compared with the TS algorithm. Test problems are generated in two different ways as that takes into account agent qualification and not takes into account agent qualification. In addition, larger size test problems are solved and the success of the algorithm has been shown through test problems. In addition,

iterative local search algorithm was proposed for the first time for the MRGAP.

The remainder of this paper is organized as follows. The first section of the study is the introduction, in the second section the problem is defined and MILP model is given. In the third section, the proposed solution method is explained. In the fourth section, experimental results are given and conclusions are given in the final section.

2. Problem description

In this study, multi-resource bottleneck generalized assignment problem (MRBGAP) with agent qualifications was addressed. The problem addressed in this study is defined by the problem of assigning employees to jobs in a firm. Employees are considered as agents. Each jobs must be assigned to an employee. More than one job can be assigned to an employee. Employee capacities depend on employee and shift. The shifts are defined as morning, noon, afternoon, evening and night. Employees' capacities (working hours) vary according to shifts. For example, an employee can work more in the morning shift than in the evening shift. For this reason, the employees' capacity parameter has two indices due to the parameter depending on the employee and the shift. The objective function is the minimization of the completion time of the last job, and this objective function is the bottleneck objective function.

The sets, indices, parameters, decision variables, constraints and objective function of the MILP model are given below;

Sets and indices

N: Set of jobs, N= {1, 2, ..., n}

M: Set of agents, M= {1, 2,..., m}

R: Set of resource, R= {1, 2,...,r}

j: job indices where $j \in N$.

i: agent indices where $i \in M$.

k: resource indices where $k \in R$.

Parameters

p_{ijk}: processing time for job j on agent i and resource k

b_{ik}: capacity for agent i on resource k

$$h_{ij} = \begin{cases} 1; & if\ agent\ i\ capable\ of\ assigning\ job\ j \\ 0; & o.w. \end{cases}$$

M: very large positive number

Decision variables

$$x_{ij} = \begin{cases} 1; & if\ job\ j\ is\ assigned\ to\ agent\ i \\ 0; & o.w. \end{cases}$$

L_{max}: maximum completion time

Model

Min Z= L_{max} (1)

s.t.

$\sum_j p_{ijk} x_{ij} \le b_{ik}$ $\forall i, k$ (2)

$\sum_i x_{ij} = 1$ $\forall j$ (3)

$\sum_k \sum_j p_{ijk} x_{ij} \le L_{max}$ $\forall i$ (4)

$x_{ij} \le h_{ij}$ $\forall i, j$ (5)

$x_{ij} \in \{0,1\}$ ve $L_{max} \ge 0$ (6)

Constraint (1) shows the objective function, minimization of the maximum completion time. Constraint (2) ensures agent capacities are not exceeded. With constraint (3), each job is assigned to an agent. The constraint (4) calculates the completion of the last job. The constraint (5) ensures agent qualifications are satisfied. Constraints (6) are the sign constraints.

Table 1: Parameters of p_{ijk}

	pij1				pij2				pij3			
i	1	2	3	4	1	2	3	4	1	2	3	4
1	18	25	6	5	10	23	37	42	36	31	25	14
2	45	41	17	7	9	10	40	15	19	46	15	8
3	44	13	18	10	10	36	9	17	46	9	11	32
4	37	28	45	5	6	28	25	14	10	25	41	15
5	6	31	45	29	24	29	49	35	42	37	9	17
6	37	41	4	11	17	4	30	47	37	38	11	34
7	6	44	30	36	23	40	14	10	45	21	33	12
8	20	25	28	46	42	44	23	11	9	45	26	29
9	21	14	39	14	28	5	27	32	41	44	31	32
10	34	47	48	41	28	8	21	33	48	12	15	32
11	32	44	45	17	4	45	21	45	23	11	16	17
12	32	11	20	44	21	43	25	6	39	26	5	47
13	37	45	8	32	46	17	47	24	20	25	15	11
14	12	27	22	38	5	29	12	40	14	10	5	33
15	34	29	13	24	33	11	16	35	13	36	4	42

The problem is also explained by an example. In the example, there are 15 jobs, 4 employees (agents) and 3 shifts (resources). Table 1 shows p_{ijk} values. h_{ij} and b_{jk} parameters are given in Table 2 and Table 3, respectively.

Example: 15 jobs, 4 employee (agent) and 3 shifts (resource)

Table 2: Parameters of h_{ij}

i	hij														
	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15
1	0	1	0	1	0	1	1	0	0	1	0	1	1	0	1
2	1	0	1	1	1	0	1	1	0	1	1	1	1	0	1
3	0	1	1	1	0	1	1	0	1	1	0	1	1	1	0
4	1	0	1	1	1	1	1	1	1	1	1	0	1	1	1

Table 3: Parameters of b_{jk}

i	bik		
	1	2	3
1	138	102	147
2	120	100	110
3	130	132	95
4	90	105	100

In the best solution, jobs 2, 7 and 10 are assigned to agent 1; jobs 11, 13 and 15 are assigned to agent 2; jobs 3, 6, 9, 12 and 14 are assigned to agent 3; jobs 1, 4, 5, and 8 are assigned to agent 4. Loads of the agents are 254, 263, 269 and 262, respectively. The objective function value is 269.

3. Multi-start iterated tabu search algorithm

Since the problem is NP-Hard, a multi-start iterated tabu search algorithm has been proposed to solve large problem instances.

The iterated local search algorithm is a heuristic algorithm that has three basic stages. The first stage is the generation of the initial solution. At the second stage the solution is improved by a local search method. The third stage is the perturbation stage. The steps of the iterated local search algorithm are given in Figure 1. Once the initial solution is obtained, the algorithm repeats the local search and perturbation steps until the stopping criterion is achieved. If the solution obtained from the local search is better than the current solution, the solution is considered to be the current solution. The perturbation mechanism is intended to escape from local optimal. In the perturbation phase, the solution is changed slightly.

Iterated local search algorithm is applied to many combinatorial optimization problem successfully. Iterated local search algorithm is proposed for the scheduling problem [32], vehicle routing problem [33-36], quadratic assignment problem [37], quadratic knapsack problem [38], hub location problem [39] and shift scheduling [40].

Firstly, abbreviations used in algorithm are given:

$S_{0(c)(n)}$: Initial (current) (neighbor) solution;

S^p: Perturbated solution;

$E_{0(c)(n)}$: Obj. func. value of the initial (current) (neighbor) solution;

CL: Set of jobs;

TLL: Tabu list length;

v: maximum iteration number of TS;

MTLS: Maximum tabu list size;

E_{best}: Objective function value of the best solution; maxStart: Multi- start number of the algorithm

In this study, multi- start iterated tabu search algorithm is proposed for the multi resource bottleneck generalized assignment problem. Different features have been used to increase the success of the proposed algorithm.

Procedure ILS

Generate initial solution S_0;

Apply local search procedure to S_0 and obtain S^*;

While termination condition not meet

    Apply perturbation to S^* and obtain S^p;

    Apply local search procedure to S^p and obtain S'';

    If f(S'')<f(S*)

      S*←S'';

    End

End

Figure 1. Algorithm of the iterated local search

One of the important features of the proposed algorithm is to start the search process multiple times. This feature provides diversification. Initial solution of the algorithm is generated by randomly or by a greedy algorithm. Throughout our preliminary experiments, it was observed that the algorithm reached better solutions faster by using greedy algorithm as an initial solution finding mechanism. However, only the use of the greedy algorithm caused starting with very similar solutions. Thus, random solutions also taken as an initial solution for the investigation of the unexplored regions in the search space. For this, a random number is derived. If this random number is greater then q (a predetermined parameter) the algorithm uses the greedy algorithm. Otherwise random initial solution is generated. TS algorithm is used as a local search algorithm. The TS algorithm and perturbation mechanisms are applied respectively until the stopping criterion is achieved. Once the stopping criterion has been achieved, an initial solution is generated again and the steps are repeated until the number of multiple starting is reached.

In the next section, initial solution finding mechanisms, TS algorithm used in local search, perturbation mechanism and all steps of algorithm will be described.

3.1. Initial solution finding mechanism

3.1.1. A greedy construction heuristic

When generating the initial solution, a job is assigned to the agent with the smallest completion time as possible. For this, p_{ij} values are calculated using

Equation 7. p_{ij} denotes the total completion time of the job according to agent on the basis of resource.

$$p_{ij}=\sum_k p_{ijk} \qquad \forall i,j \qquad (7)$$

The p_{ij} values are sorted in ascending order and the sp_{ij} matrix is obtained. The aim is to assign the job to the first agent in the sp_{ij} matrix. However, since each agent has a capacity and agent qualifications are taken into account, the agent cannot be assigned to first agent in the sp_{ij} matrix. If the job is not assigned to the first agent, the job is assigned to second order of the agent in the sp_{ij} matrix. Algorithm is repeated until each job is assigned to an agent and a solution is obtained. The algorithm is given on Figure 2.

3.1.2. Random initial solution

In the random solution finding algorithm, the randomly selected job j* is assigned to the randomly selected agent i*. If job j* is not assignable to i*, another agent is randomly generated. The algorithm is working until a solution is obtained.

3.2. Local search algorithm

The TS algorithm was used as the local search algorithm in the proposed heuristic method.

Two methods are used to generate the neighboring solutions from current solution. The first method is to assign each job in the agent with the largest completion time to the other agents. The other method is the reciprocal displacement of jobs in the agent with the largest completion time with the jobs in other agents. All solutions are derived from the current solution by using neighboring solution generation methods.

The best of these solutions is taken, and if the movement in the generation of the neighbor solution is not in the tabu list, the solution is taken directly as the current solution. If a movement is made in the tabu list and the solution is not a better solution than the best solution, the neighbor solution with the second smallest objective function is chosen and the same test is also applied to this solution. This step is repeated until a solution is accepted.

The length of the tabu list is considered as fixed, and when the tabu list is full, the movement that has been in the list for the longest period is deleted. It is forbidden to carry out the movements in the tabu list. If a better solution is obtained than the best solution, the tabu is eliminated and the relevant solution is taken as the current solution.

If the solution is obtained as a result of the use of the first neighboring method, the job and the relevant agent are added to the tabu list. The movement of this job to the relevant agent during the tabu is prohibited. If the solution is obtained as a result of the use of the second neighboring method, replacement of these jobs is prohibited. The algorithm works until it reaches the predetermined number of iterations. The steps of the tabu search algorithm are given in Figure 3.

Procedure a greedy construction heuristic

Input: p_{ijk}, h_{ij}, b_{ik}, sp_{ij}

Output: Initial solution (S_0), Objective funct. value of S_0 (E_0)

exit1←0, n'←0; CL←{1,…,n}

While exit1==0

    exit2←0; x←1; flag←0;

    While exit2==0

      Select the job j* randomly from the CL and assign the j* to the x th order of the agent in the sp_{ij*} matrix;

      For k=1 **to** r

        If b_{i*k}<p_{i*j*k} **or** h_{i*j*}==0

        flag←1;

        End

      End

      If flag==1

      x←x+1;

      Else

      exit2←1; n'←n'+1;

      b_{i*k}←b_{i*k}-p_{i*j*k};CL←CL\{j*};

      End

      If x==m

        exit2←1; n'←0; Initialize CL and b_{ik};

      End

    End

    If n'==n

    exit1←1;

    End

End

Figure 2. Greedy Construction Heuristic

Procedure TS algorithm

Input: *p_{ijk}, b_{ik}, h_{ij}, m, n, r, v, MTLS, S_0, E_0*

Output: Near optimal solution (S^*)

S^*←S_0; E^*←E_0; S_c←S_0; E_c←E_0; TLL←1;

***While** iter<v*

    Generate neighbor solutions and sort ascending order according to *obj. func.value* (S_n ᵗ); t←1; check←0;

    While check==0

      ***If** the movement of S_n ᵗ not tabu **or** E_n^t<E_{best}*

        S_c←S_n^t; E_c←E_n^t;

        Insert the movement of S_c at the tabu list;

        TLL←TLL+1; check←1;

      Else

        t←t+1;

      End

End

If *TLL==MTLS+1*

  Delete the first element in the tabu list;

    TLL←1;

End

If E_c<E^*

    S^*←S_c; E^*←E_c;

End

    iter←iter+1;

End

Figure 3. Tabu search algorithm

3.3. Perturbation mechanism

The iterative local search algorithm uses the perturbation mechanism to escape the local optimal. If the perturbation is too strong, the algorithm can move away from promising regions. If perturbation is too small, the algorithm may loop in previously searched regions. Therefore, it is very important to determine the appropriate perturbation length. With perturbation, a new solution (S'') is derived from one of the methods of the neighboring solution from the current solution (S'). If objective function value of S'' is less than objective function value S', then the perturbated solution will be S''.

If S'' is accepted, the value Ɣ is increased by λ. Otherwise, a new S'' solution is derived. λ is taken as a value between 0 and 1. If the number of consecutive rejected solutions reaches a predetermined value (maxTry), the value of Ɣ is increased by λ. The algorithm works until the number of applied moves equal to pertLength. The perturbation mechanism is given in Figure 4.

Procedure Perturbation

Input: S', pertLength, λ, maxtry

Output: Sp

p←1; Ɣ←1+ λ; Sp←S'; try←0;

While p≤pertLength

    Generate a random solution S'', from S' by applying a randomly selected neighborhood structure;

    If f(S'')≤ Ɣ* f(S')

      Sp←S''; Ɣ← Ɣ + λ; p←p+1; try←0;

    Else

      try←try+1;

      If try>maxtry

        Ɣ← Ɣ +λ; try←0;

      End

    End

End

Figure 4. Perturbation mechanism

3.4. Steps of the algorithm

The steps of the proposed multistart iterated tabu search algorithm are given in Figure 5.

Procedure MS- ITS

Input: problem data, maxStart, λ, maxTry, pertLength, *v, MTLS, q, maxiter*

Output: Near optimal solution (S)*

For s=1:maxStart

    pertLength←1;

    Generate a random number rnd;

    If rnd≤q

      Construct a random initial solution S_0 with the objective function E_0;

      (S_0,E_0)←**RandomInitialSolution**(problem data*)*

    Else

      Construct an initial solution S_0 with the objective function E_0 using greedy algorithm;

      (S_0,E_0)←**GreedyInitialSolution**(problem data,sp_{ij}*)*

    End

    (S^*,E^*)←**TabuSearchAlgorithm**(problem data, v, *MTLS,S_0,E_0*)

    $S'=S^*; E'=E^*;$

    While iteration<maxiter

    (S^p,E^p)←**Perturbation**(pertLength,λ,maxtry, *S', E')*

    (S'', E'')←**TabuSearchAlgorithm**(problem data, v, *MTLS,S^p,E^p)*

      If $E''<E^*$

        $S^*←S''; E^*←E'';$

      End

      iteration← iteration+1;

    End

End

Figure 5. Steps of the algorithm

4. Computational results

The success of heuristic algorithms strongly depends on the selection of the right parameters. The parameters of the heuristic are maxStart, λ, maxTry, pertLength, v, MTLS, maxiter and q. Taguchi experimental design (TED) reduces the number of experiments and it is a successful method for determining the parameters of heuristic algorithms. If the number of parameters is high TED is preferred because it reduces the number of experiments [41]. The parameters of the algorithm were determined with TED due to proposed algorithm has many parameters. Factor levels are given in Table 4. There has been 8 factors and 3 levels. L27 orthogonal array is selected. Since the objective function of the problem is minimization, smaller-the-better type function is selected for the Taguchi design. S/N ratio is given below (Eq- 8). n is the number of observations in each experiment and Y_i is the objective function value.

$$S/N\ ratio = -10 * log\left(\frac{1}{n}\sum_{i=1}^{n} Y_i^2\right) \qquad (8)$$

Table 4: Factor levels

Factors	Levels
maxStart (A)	10; 15 and 20
λ (B)	0.02; 0.03 and 0.04
maxTry (C)	25, 50 and 75
pertLength (D)	10; 20 and 30
v (E)	500; 750 and 1000
MTLS (F)	40; 60 and 80
maxiter (G)	50; 75 and 100
q (H)	0.3; 0.4 and 0.5

In this study, instead of applying TED to each test problem, the following test problems were selected and TED was applied. In Table 5 selected test problems and determined parameters are given. In the selected problems, U [15,25], number of agent is 10 and number of jobs is 200. The largest problem size was preferred. Test problems are generated as described in the study by Karsu and Azizoğlu [3].

Table 5. Selected test problems and determined parameters

c	h_{ij}	A	B	C	D	E	F	G	H
1.4	1	10	0.04	25	10	500	80	50	0.4
	0.7	10	0.02	25	20	500	40	50	0.4
1.6	1	10	0.03	50	20	500	60	50	0.5
	0.7	10	0.02	50	10	500	40	50	0.3

Response table of S/N ratios for the test problem with c=1.4 and h=1 in Table 6.

Table 6. Response table of S/N ratios

Factors	S/N Ratio		
	1	2	3
A	**-63,30**	-63,32	-63,33
B	-63,31	-63,30	**-63,27**
C	**-63,14**	-63,17	-63,16
D	**-63,26**	-63,31	-63,34
E	**-63,08**	-63,21	-63,10
F	-63,32	-63,30	**-63,26**
G	**-63,13**	-63,30	-63,25
H	-63,31	**-63,26**	-63,32

According to the highest S/N values, the levels of the factors are determined and given in Table 5. Taguchi experimental design is analyzed using Minitab 16 for

Windows (Minitab Inc.).

The MILP model was coded in GAMS 24.1.3 program and the Cplex solver was used. Heuristic algorithms were coded in the MATLAB R2012b and implemented on an Intel (R) Core ™ i7- 5500 U CPU at 2.40 GHz with 12 GB of RAM memory and the Windows 10 operating system. The proposed algorithm is compared with the Tabu Search algorithm. The tabu search algorithm and the proposed algorithm were run in equal iterations. For this reason, the tabu search algorithm has been run with a (maxStart * v * maxiter) number of iterations. The neighbor generation of the TS algorithm is same with proposed heuristic approach. The results of the proposed algorithm, the tabu search algorithm, and the MILP (mixed integer linear programming) model are given in Table A1-A2. In Table A1 agent number is 5 and job numbers are 75 and 100. In Table A2 agent number is 10 and job numbers are 150 and 200. The problem specifications including the number of jobs, distribution of the processing times, the value of the c, probability of the parameter h_{ij} are given in the first, second, third and fourth columns, respectively. The MILP solution and CPU time, iterated local search solution and CPU time of the algorithm, TS solution and CPU time are given in Table A1-A2. The results of the ILS and TS algorithms are compared by the MILP solution. The heuristic error was calculated as follows:

$$\%Error = \frac{Heuristic\ solution - MILP\ solution}{MILP\ solution} \times 100$$

In the study, 48 test problems were solved by MILP model, ILP algorithm and TS algorithm. The success of the proposed algorithm is shown by comparing the results of the MILP model and TS algorithm. The proposed heuristic algorithm gave better results than TS algorithm in all test problems except 2 test problems. ILS algorithm gave the same result with TS for 5 test problems. The average error rates in heuristic algorithms according to the number of jobs are given in the Table 7.

Table 7. Average error according to job number

Job number	Average Error	
	ILS	TS
75	0	0,31
100	0,005	0,28
150	0,005	0,87
200	0,13	1,23
Average	0,035	0,6725

The proposed heuristic algorithm found the optimal results for test problems with 75 jobs. When the number of jobs is taken as 100, ILS found the optimal results in all test problems except 1 test problem. MILP model cannot find the best results for test problems with 150 jobs at 3600 sec. The proposed heuristic algorithm found a better solution in a shorter time than the MILP model for 5 test problems out of 12 test problems. When the number of jobs was 200, the ILS algorithm

gave better solutions for 6 test problems out of 12 test problems in a shorter time than the MILP model. If the tables are interpreted considering the number of agents, the MILP model found optimal solution for all test problems with 5 agents. The proposed heuristic algorithm found optimal solution for test problems with 5 agents except 1 test problem. MILP model could not find the best solution in 3600 seconds for test problems with 10 agents. The proposed heuristic algorithm provided better solution in 11 test problems out of 24 test problems in a shorter time than the MILP model. When the number of agents was 10, the TS algorithm provided a better solution for only 2 test problems than the ILS algorithm. The proposed heuristic algorithms were run in equal iteration number and ILS algorithm provided solutions in a shorter time than TS algorithm for all test problems. As a result, ILS algorithm displayed better performance than TS algorithm for MRBGAP.

5. Conclusion

In this article, multi-resource agent bottleneck generalized assignment problem (MRBGAP) is considered. The MRBGAP problem is the generalized version of GAP and is a more difficult problem to solve. However, many studies in the literature propose heuristic algorithms for GAP. Agent qualifications are examined firstly with this study for the MRBGAP. Due to the NP hardness of the problem, a multi- start iterated tabu search algorithm is proposed for the solution of the problem. In addition, proposed heuristic algorithm is compared with TS algorithm. According to experimental comparisons, ILS algorithm yielded better results than TS algorithm. ILS algorithm found better results than the MILP model in a shorter time for 10 agents. With this study, large problem instances are generated for MRBGAP. This study is the first to use ILS algorithm for the MRBGAP. Future studies will focus on matheuristic algorithm, other meta heuristic algorithms and stochastic version of the problem.

References

[1] Pentico, D.W. (2007). Assignment problems: A golden anniversary survey. *European Journal of Operational Research*, 176, 774- 793.

[2] Mazzola, J. B., & Wilcox, S.P. (2001). Heuristics for the multi- resource generalized assignment problem. *Naval Research Logistics*, 48, 468- 483.

[3] Karsu, Ö., & Azizoğlu, M. (2012). The multi- resource agent bottleneck generalised assignment problem. *International Journal of Production Research*, 5(2), 309- 324.

[4] Ross, G. T. (1975). A branch and bound algorithm for the generalized assignment problem. *Mathematical Programming*, 8, 91- 103.

[5] Posta, M., Ferland, J.A., & Michelon, P. (2012). An exact method with variable fixing for solving the generalized assignment problem. *Computational Optimization and Applications*, 52, 629- 644.

[6] Avella, P., Boccia, M., & Vasilyev, I. (2010). A computational study of exact knapsack separation

for the generalized assignment problem. *Computational Optimization and Applications*, 45, 543- 555.

[7]   Savelsbergh, M. (1997). A branch- and- price-algorithm for the generalized assignment problem. *Operations Research*, 831- 841.

[8]   Pigatti, A., Poggi, M., & Uchoa, E. (2005). Stabilized branch- and- cut- and- price for the generalized assignment problem. *Electronic Notes in Discrete Mathematics*, 19, 389- 395.

[9]   Öncan, T., Şuvak, Z., Akyüz, M. H., & Altınel, İ. K. (2019). Assignment problem with conflicts. *Computers and Operations Research* (In press).

[10] Wu, W., Iori, M., Martello, S., & Yagiura, M. (2018). Exact and heuristic algorithms for the interval min- max regret generalized assignment problem. *Computers and Industrial Engineering*, 125, 98- 110.

[11] Öncan, T. (2007). A survey of the generalized assignment problem and its applications. *Information Systems and Operational Research*, 45 (3), 123- 141.

[12] Higgins, A. J. (2001). A dynamic tabu search for large- scale generalised assignment problems. *Computers and Operations Research*, 28, 1039-1048.

[13] Diaz, J. A., & Fernandez, E. (2001). A tabu search heuristic for the generalized assignment problem. *European Journal of Operational Research*, 132, 22- 38.

[14] Osman, İ. (1995). Heuristics for the generalized assignment problem: simulated annealing and tabu search approaches. *Operations Research- Spektrum*, 17, 211- 225.

[15] Chu, P. L., & Beasley, J. E. (1997). A genetic algorithm for the generalized assignment problem. *Computers and Operations Research*, 24 (1), 17- 23.

[16] Ozbakır, L., Baykasoğlu, A., & Tapkan, P. (2010). Bees algorithm for generalized assignment problem. *Applied Mathematics and Computation*, 11, 3782-3795.

[17] Jeet, V., & Kutanoglu, E. (2007). Lagrangian relaxation guided problem space search heuristics for generalized assignment problem. *European Journal of Operational Research*, 182 (3), 1039-1056.

[18] Litvinchev, I., Mata, M., Rangel, S., & Saucedo, J. (2010). Lagrangian heuristic for a class of the generalized assignment problems. *Computers and Mathematics with Applications*, 60 (4), 1115- 1123.

[19] French, A. P., & Wilson, J. M. (2007). An LP- based heuristic procedure for the generalized assignment problem with special ordered sets. *Computers and Operations Research*, 34 (8), 2359- 2369.

[20] Souza, D. S., Santos H. G., & Coelho, I. M. (2017). A hybrid heuristic in GPU- CPU based on scatter search for the generalized assignment problem. *Procedia Computer Science*, 108, 1404- 1413.

[21] Sethanan, K., & Pitakaso, R. (2016). Improved differential evolution algorithms for solving generalized assignment problem. *Expert Systems with Applications*, 45, 450- 459.

[22] Liu, Y. Y., & Wang, S. (2015).A scalable parallel genetic algorithm for the generalized assignment problem. *Parallel Computing*, 46, 98- 119.

[23] Degroote, H., Velarde, J., & Causmaecker, P. (2018). Applying algorithm selection- a case study for the generalized assignment problem. *Electric Notes in Discrete Mathematics*, 69, 205- 212.

[24] Chakravarthy, V. K., Ramana, V., & Umashankar, C. (2018). Investigation of task bottleneck generalized assignment problems in supply chain optimization using heuristic techniques. *IOSR Journal of Business and Management*, 20 (5), 41- 47.

[25] Fadaei, S., & Bichler, M. (2017). Generalized assignment problem: truthful mechanism design without money. *Operations Research Letters*, 45 (1), 72-76

[26] Sahni, S., & Gonzalez, T. (1976). P-complete approximation problems. *Journal of the Association for Computing Machinery*, 23, 556- 565.

[27] Yagiura, M., Komiya, A., Kojima, K., Nonobe, K., Nagamochi, H., Ibaraki, T., & Glover, F. (2007). A path- relinking approach for the multi- resource generalized quadratic assignment problem. *Engineering stochastic local search algorithms, designing, implementing and analyzing effective heuristics*, 4638, 121- 135.

[28] Gavish, B., & Pirkul, H. (1991). Algorithms for the multi- resource generalized assignment problem. *Management Science*, 37 (6).

[29] Yagiura, M., Iwasaki, S., Ibaraki, T., & Glover, F. (2004). A very large- scale neighborhood search algorithm for the multi- resource generalized assignment problem. *Discrete Optimization*, 1, 87-98.

[30] Mitrovic- Minic, S., & Punnen, A. (2009). Local search intensified: Very large scale variable neighborhood search for the multi- resource generalized assignment problem. *Discrete Optimization,* 6, 370- 377.

[31] Moussavi, S. E., Mahdjoub, M., & Grunder, O. (2018). A hybrid heuristic algorithm for the sequencing generalized assignment problem in an assembly line. *IFAC- Papers OnLine*, 51(2), 695-700.

[32] Qin, T., Peng, B., Benlic, U., Cheng, T. C. E., Wang, Y., & Lü, Z. (2015). Iterated local search based on multi- type perturbation for single- machine earliness/ tardiness scheduling. *Computers and Operations Research*, 61, 81- 88.

[33] Ibaraki, T., Imahori, S., Nonobe, K., Sobue, K., Uno, T., & Yagiura, M. (2008). An iterated local search algorithm for the vehicle routing problem with convex time penalty functions. *Discrete Applied Mathematical*, 156 (11), 2050- 2069.

[34] Subramanian, A., Drummond, L. M. D. A., Bentes, C., Ochi, L. S., & Farias, R. (2010). A parallel heuristic for the vehicle routing problem with simultaneous pickup and delivery. *Computers and Operations Research*, 37 811), 1899- 1911.

[35] Penna, P. H. V., Subramanian, A., & Ochi, L. S. (2013). An iterated local search heuristic for the heterogeneous fleet vehicle routing problem. *Journal of Heuristics*, 19 (2), 201- 232.

[36] Michallet, J., Prins C., Amodeo, L., Yalaoui, F., & Vitry, G. (2014). Multi- start iterated local search for the periodic vehicle routing problem with time

Windows and time spread constraints on services. *Computers and Operations Research*, 41, 196- 207.

[37] Stützle, T. (2006). Iterated local search for the quadratic assignment problem. *European journal of Operations Research*, 174 (3), 1519- 1539.

[38] Avcı, M., & Topaloglu, S. (2017). A multi- start iterated local search algorithm for the generalized quadratic multiple knapsack problem. *Computers and Operations Research*, 83, 54- 65.

[39] Guan, J., Lin, G., & Feng, H. (2018). A multi- start iterated local search algorithm for the uncapacitated single allocation hub location problem. *Applied Soft Computing*, 73, 230- 241

[40] Meignan, D., & Knust, S. (2019). A neutrality- based local search for shift scheduling optimization and interactive reoptimization. *European Journal of Operational Research*, 279 (2), 320- 334.

[41] Mozdgir, A., Mahdavi, I., Badeleh, I. S., & Solimanpur, M. (2013). Using the Taguchi method to optimize the differential evolution algorithm parameters for minimizing the workload smoothness index in simple assembly line balancing. *Mathematical and Computer Modelling,* 57 (1-2), 137- 151.

Gülçin Bektur *received her MSc (2013) and PhD (2018) degrees from Department of Industrial engineering, Eskisehir Osmangazi University, Turkey. Her research areas include optimization and heuristic search. She is an Assistant Professor at the Department of Industrial engineering, Iskenderun Technical University.*
 http://orcid.org/0000-0003-4313-7093

Appendices
Table A1. Five agent and 75 or 100 jobs (* denotes optimal solutions)

				MILP			ILS			TS		
n	pij1	c	hij	Solution		CPU	Solution	CPU	%Error	Solution	CPU	%Error
75	U[5,25]	1,4	1	**697,35**	*	0,49	**697,35**	45,22	0	699,77	85,79	0,35
75	U[5,25]	1,4	0,7	**728,3**	*	1,08	**728,3**	27,41	0	729,64	40,42	0,19
75	U[5,25]	1,6	1	**574,27**	*	1,31	**574,27**	48,98	0	575,07	82,75	0,14
75	U[5,25]	1,6	0,7	**814,46**	*	1,2	**814,46**	20,13	0	**814,46**	42,42	0
75	U[15,25]	1,4	1	**1153,9**	*	0,81	**1153,9**	35,26	0	**1153,9**	82,19	0
75	U[15,25]	1,4	0,7	**1219,56**	*	1,7	**1219,56**	22,19	0	1220,53	45,42	0,08
75	U[15,25]	1,6	1	**1157,64**	*	1,42	**1157,64**	39,84	0	1169,14	88,46	1
75	U[15,25]	1,6	0,7	**978,86**	*	1,03	**978,86**	20,46	0	**978,86**	42,46	0
75	U[25,35]	1,4	1	**1806,47**	*	3,56	**1806,47**	41,21	0	1812,16	85,64	0,32
75	U[25,35]	1,4	0,7	**1915,14**	*	2,57	**1915,14**	25,12	0	**1915,14**	41,53	0
75	U[25,35]	1,6	1	**1770,45**	*	2,7	**1770,45**	41,54	0	1775,76	86,79	0,3
75	U[25,35]	1,6	0,7	**1921,74**	*	1,25	**1921,74**	21,89	0	1948,96	48,86	1,42
100	U[5,25]	1,4	1	**814,47**	*	2,8	814,96	81,69	0,06	818,29	102,85	0,47
100	U[5,25]	1,4	0,7	**917,57**	*	1,76	**917,57**	67,88	0	917,57	85,69	0
100	U[5,25]	1,6	1	**860,59**	*	5,53	**860,59**	82,87	0	861,32	105,18	0,09
100	U[5,25]	1,6	0,7	**1008,28**	*	0,89	**1008,28**	61,95	0	1008,28	80,25	0
100	U[15,25]	1,4	1	**1522,29**	*	5,44	**1522,29**	89,22	0	1522,49	109,18	0,02
100	U[15,25]	1,4	0,7	**1616,18**	*	1,43	**1616,18**	69,28	0	1628,33	81,43	0,76
100	U[15,25]	1,6	1	**1537,62**	*	2,18	**1537,62**	87,87	0	1545,82	106,17	0,54
100	U[15,25]	1,6	0,7	**1645,26**	*	2,06	**1645,26**	68,51	0	1649,96	88,88	0,29
100	U[25,35]	1,4	1	**2451,96**	*	5,5	**2451,96**	86,54	0	2463,93	101,6	0,49
100	U[25,35]	1,4	0,7	**2514,61**	*	42,54	**2514,61**	64	0	**2514,61**	87,67	0
100	U[25,35]	1,6	1	**2435,7**	*	4,41	**2435,7**	83,75	0	2451,57	107,62	0,66
100	U[25,35]	1,6	0,7	**2645,9**	*	3,59	**2645,9**	55,88	0	2647,86	80,86	0,08

Table A2. Ten agent and 150 or 200 jobs

n	pij1	c	hij	MILP		ILS			TS		
				Solution	CPU	Solution	CPU	%Error	Solution	CPU	%Error
150	U[5,25]	1,4	1	**500,19**	3600	501,18	1656,85	0,2	501,68	2258,5	0,3
150	U[5,25]	1,4	0,7	568,62	3600	**562,13**	1258,45	-1,14	570,8	1874,6	0,38
150	U[5,25]	1,6	1	**487,87**	3600	489,23	1845,74	0,28	499,49	2156,8	2,38
150	U[5,25]	1,6	0,7	**585,68**	3600	588,74	1152,41	0,52	587,16	1745,7	0,25
150	U[15,25]	1,4	1	**1068,17**	3600	1069,5	1985,12	0,12	1071,63	2275,3	0,32
150	U[15,25]	1,4	0,7	1115,84	3600	**1114,14**	1158,46	-0,15	1120,07	1856,4	0,38
150	U[15,25]	1,6	1	**1062,19**	3600	1064,89	1585,45	0,25	1074,3	2041,9	1,14
150	U[15,25]	1,6	0,7	**1111,28**	3600	1114,23	985,12	0,27	1119,24	1765,2	0,72
150	U[25,35]	1,4	1	1683,83	3600	**1683,55**	1289,13	-0,02	1684,24	2256,5	0,02
150	U[25,35]	1,4	0,7	1766,41	3600	**1762,5**	1258,45	-0,22	1784,49	1946,4	1,02
150	U[25,35]	1,6	1	**1685,86**	3600	1684,28	1365,75	-0,09	1720,34	2178,7	2,04
150	U[25,35]	1,6	0,7	**1770,04**	3600	1770,95	1058,43	0,05	1798,02	1845,5	1,58
200	U[5,25]	1,4	1	**655,5**	3600	660,76	2156,52	0,8	**658,02**	2985,4	0,38
200	U[5,25]	1,4	0,7	**748,35**	3600	752,63	1985,63	0,57	762,46	2441,1	1,89
200	U[5,25]	1,6	1	**666,2**	3600	670,19	2045,12	0,6	669,13	2874,3	0,44
200	U[5,25]	1,6	0,7	769,65	3600	**766,38**	1756,42	-0,42	789,1	2045,5	2,53
200	U[15,25]	1,4	1	1425,66	3600	**1425,28**	2378,41	-0,03	1425,37	2796,1	-0,02
200	U[15,25]	1,4	0,7	1494,56	3600	**1494,46**	1974,23	-0,01	1499,15	2248,6	0,31
200	U[15,25]	1,6	1	1432,5	3600	**1431,54**	2685,42	-0,07	1498,45	2213,9	4,6
200	U[15,25]	1,6	0,7	1465,53	3600	1468,01	1545,78	0,17	1496,42	2016,2	2,11
200	U[25,35]	1,4	1	2251,16	3600	**2250,43**	1985,45	-0,03	2289,15	2845,3	1,69
200	U[25,35]	1,4	0,7	2338,99	3600	2339,41	1845,12	0,02	2339,95	2156,2	0,04
200	U[25,35]	1,6	1	2234,67	3600	**2232,55**	2156,45	-0,1	2256,78	2845,3	0,99
200	U[25,35]	1,6	0,7	2330,87	3600	2333,75	1585,12	0,12	**2328,86**	2045,9	-0,09

RESEARCH ARTICLE

A new iterative linearization approach for solving nonlinear equations systems

Gizem Temelcan[*a], Mustafa Sivri[b], Inci Albayrak[b]

[a]Department of Computer Programming, Istanbul Aydin University, Turkey
[b]Department of Mathematical Engineering, Yildiz Technical University, Turkey
temelcan.gizem@gmail.com, msivri@yildiz.edu.tr; ibayrak@yildiz.edu.tr

ARTICLE INFO	ABSTRACT
	Nonlinear equations arise frequently while modeling chemistry, physics, economy and engineering problems. In this paper, a new iterative approach for finding a solution of a nonlinear equations system (NLES) is presented by applying a linearization technique. The proposed approach is based on computational method that converts NLES into a linear equations system by using Taylor series expansion at the chosen arbitrary nonnegative initial point. Using the obtained solution of the linear equations system, a linear programming (LP) problem is constructed by considering the equations as constraints and minimizing the objective function constructed as the summation of balancing variables. At the end of the presented algorithm, the exact solution of the NLES is obtained. The performance of the proposed approach has been demonstrated by considering different numerical examples from literature.

1. Introduction

Numerical analysis and computers are intimately related with each other regarding to solve mathematical problems. With the development of computers, numerical methods have been increased for solving scientific and engineering problems. The numerical methods are used to find approximate solution of such problems because it is not possible to obtain exact solution by using algebraic processes. One of the most important issues for solving NLES in science and engineering is to find a solution that is frequently arising in optimization and computational mathematics. Because NLESs cannot be solved as easily as linear systems, iterative methods are improved as a new class of numerical solution methods.

Iterative method is a procedure repeated over and over again to find either the root of an equation or the solution of an NLES. In numerical methods, the sequence of approximate solutions converges

to the root of the system. If the convergence rate of an iterative method is rapid, then a solution may be found in less iterations compared with other methods. As the iterations begin to have successive same values, this is an indication that the obtained solution is the exact solution of the NLES. However, when the obtained solution of the system does not converge, it is indicated that there is an error in the computations or there is no solution. Therefore, an NLES has finite or infinite number of solutions or no solution. There are numerous conventional methods to solve NLESs having algebraic and transcendental equations. One of the most popular and traditional numerical methods is Newton method which is widely used for finding roots of the NLES. This method is based on Taylor series expansion of a function, and converges rapidly to the exact solution of the NLES. It can be presented as an advantage that Newton method requires less iterations to reach the solution compared to other known methods.

*Corresponding Author

Another advantage of the Newton method is the framework is clear, and therefore it can be used to solve a variety of problems. On the contrary, due to the difficulty in computation of both Jacobian matrix and its inverse at each iteration. Using the Newton method would be time-consuming regarding to the size of the system. To avoid these impracticabilities, some developments and modifications are made to the Newton method, such as Quasi-Newton method, Dimension Reducing method, Modified Reducing Dimension method and Perturbed Dimension Reducing method.

Grapsa and Vrahatis [1] reviewed a class of methods for solving NLESs and optimization problems named Dimension Reducing methods. Frontini and Sormani [2] extended to p-dimensional case the modification of Newton method. This method is used to solve NLES and compared with Newton method and Halley-Chebyshev method. Babolian et al. [3] extended the Adomian decomposition method for solving the NLES. Nie [4] transformed the NLES into a constrained nonlinear optimization problem and used null space algorithm to solve the problem. Also, Nie [5] proposed a new approach by converting an NLES into a constrained nonlinear programming problem, and solved this problem by using a line search sequential quadratic programming approach. Jafari and Daftardar-Gejji [6] suggested a modification of Adomian decomposition method and demonstrated that series solution obtained converges faster than that of standard Adomian decomposition method. Darvishi and Barati [7] presented an iterative third-order Newton-type method based on Adomian decomposition method for solving NLESs. Golbabai and Javidi [8] considered homotopy perturbation method to construct an iterative method for solving the NLES, compared the results with that of the revised Adomian decomposition method in [6] obtained, and showed the accuracy and fast convergence of the proposed method. Biazar and Ghanbary [9] constructed a new iterative approach based on the concept of Jacobi method and presented the effectiveness of the proposed method as the number of equations and variables increases. Grosan and Abraham [10] proposed a novel approach transforming NLES to a multiobjective optimization problem and revealed that it deals with the large scale system of equations. Hosseini and Kafash [11] presented an algorithm based on Adomian decomposition convergence basis method for solving functional equations. Gu and Zhu [12] presented an effective filter algorithm for solving both

the nonlinear systems of equalities and inequalities. They transformed the system into a nonlinear programming problem, and used the nonmonotone technique and the global line search strategy in the algorithm. Vahidi et al. [13] implemented the restarted Adomian decomposition method for solving the NLESs and showed that the proposed method converges to the exact solutions more rapidly than the Adomian decomposition method. Sharma and Gupta [14] presented two iterative methods for solving NLES. One of the methods is a third-order method having two-steps which are the Newton iteration and the weighted-Newton iteration, respectively. The other method is a fifth and sixth-order method having three-steps of which the first two steps are same as that of third-order method and third step is the weighted-Newton iteration again. Wang and Pu [15] proposed a nonmonotone filter trust region method to solve the NLES. The system is converted to a nonlinear programming problem in which some equations are treated as constraints whereas the others are taken as objective function. Zhang [16] reviewed some methods, especially iterative methods, of solving system of nonlinear equations in the technical report. Dhamacharoen [17] proposed a new hybrid method having less computations than others. This hybrid method is composed of two methods that are the Newton method and the Broyden method. The proposed method is compared with the Newton method and the Darvishi−Barati method [7], and it is seen that the number of computations is fewer than the compared ones even if it requires more iterations to reach the solution. Izadian et al. [18] proposed a new approach combining Newton method and Homotopy Analysis method to solve the algebraic and transcendental equations system. The main purpose of this combined approach is to accelerate the rate of convergence and to obtain the local convergence. Narang et al. [19] presented a fourth order two parameter Chebyshev-Halley like two-point family for solving the nonlinear equations of large-scale systems. Saheya et al. [20] presented an improved Newton method based on iterative rational approximation model. Wang and Fan [21] presented two high computational efficient derivative-free iterative methods. The methods have low computational cost by reducing the number of lower-upper decomposition of matrix in each iteration. Xiao and Yin [22] presented a technique using the extended Newton iteration for increasing the order of convergence for iterative methods. They applied the proposed technique to several known methods and obtained new

methods having higher order of convergence. Balaji et al. [23] solved the NLES by using the integrated restarted Adomian decomposition method and Adomian decomposition method. Madhu et al. [24] proposed a new method which is an improvement of double-step Newton method. It is two-step fifth-order method in which two functions and two first order Frechet derivatives are used. Sharma and Arora [25] proposed Newton-like iterative methods of fifth and eighth-order of convergence to solve NLESs.

There are numerious traditional approaches such as Muller method and the Secant method for solving NLESs, however, these methods have many shortcomings. The methods are very sensitive to the choice of initial values and may show oscillatory behavior or even diverge in the case of closeness between the initial value chosen and the root of the system [26]. Moreover, most of these methods require continuously differentiable nonlinear equations. To avoid the negative aspects of the traditional methods, some approaches based on metaheuristic optimization methods such as Genetic Algorithm, Particle Swarm Optimization, Simulated Annealing have been presented. These methods are used with no assumptions about the function being optimized such as smoothness, convexity or differentiability. Dai et al. [27] mixed Genetic Algorithm and quasi-Newton method for solving NLES. Hirsch et al. [28] proposed a modified metaheuristic GRASP method in which all roots are found through the multiple minimizations of an objective function to find all real solutions of NLES. Pourjafari and Mojallali [26] proposed a novel optimization-based method finding all real and complex roots of a system.

In this paper, we introduce a new iterative approach to solve an NLES as an optimization problem. By means of the first order Taylor series expansion and by choosing an arbitrary nonnegative initial point, a system of linearized equations is solved at each iteration. New variables are obtained by adding balancing variables to the initial solution of the system of linearized equations, and then Maclaurin series expansion is used to linearize the NLES reconstructed by substituting these new variables in the system. At each iteration, a LP problem is constructed of which the linearized equations are considered as constraints whereas the objective function is the minimization of the summation of balancing variables. The iterative approach is processed until all ballancing variables are zero, and the optimal solution of the NLES is found.

The organization of the paper is as follows. In Section 2, some brief information is given. In section 3, the proposed approach is presented. In Section 4, some numerical examples and results are demonstrated and the paper ends with conclusion at Section 5.

2. Preliminaries

In this section, some definitions are given related with the proposed approach. In this paper, it is assumed that each equation in the NLES are continuously differentiable.

Definition 1. *[29] An NLES is a set of equations as follows:*

$$f_1(x_1, ..., x_n) = 0$$
$$f_2(x_1, ..., x_n) = 0$$
$$\vdots$$
$$f_m(x_1, ..., x_n) = 0$$

where $(x_1, ..., x_n) \in \mathbf{R}^n$ is a vector, $x_j \in \mathbf{R}$, $(j = 1, ..., n)$ and each $f_i(x)$, $(i = 1, ..., m)$ is a nonlinear real function.

Definition 2. *A solution of an NLES having m equations in n variables is a point $A = (a_1, ..., a_n) \in \mathbf{R}^n$ such that*

$$f_1(a_1, ..., a_n) = \cdots = f_m(a_1, ..., a_n) = 0.$$

Definition 3. *A function f is continuously differentiable if and only if the first (and possibly higher) order derivative of f is continuous.*

Definition 4. *[29] Taylor series expansion generated by $f(x)$ at $x = a$ is*

$$
\begin{aligned}
f(x) =\ & f(a) + f'(a)(x - a) \\
& + \frac{1}{2!} f''(a)(x - a)^2 \\
& + \cdots + \frac{1}{n!} f^{(n)}(a)(x - a)^n + \cdots.
\end{aligned}
$$

For linearization,

$$f(a) + f'(a)(x - a) = 0$$

is considered. Accordingly, the first two terms of Taylor series expansion generated by $f(x_1, ..., x_n)$ at $A = (a_1, ..., a_n)$, i.e.

$$f(A) + \frac{\partial}{\partial x_1} f(A)(x_1 - a_1) + \cdots + \frac{\partial}{\partial x_n} f(A)(x_n - a_n) = 0$$

linearizes the function f in n variables.

Definition 5. *[29] A set of vectors converges if the norm is zero, i.e.*

$$|x^k - x^{k-1}|| =$$
$$= \sqrt{(x_1^k - x_1^{k-1})^2 + \cdots + (x_n^k - x_n^{k-1})^2} = 0$$

where k is the number of iterations. The vector $x = (x_1, ..., x_n)$ is the root of the function if it satisfies that $|f_i(x)| < \epsilon$, $i = 1, ..., m$ where $\epsilon \geq 0$ is a given tolerance.

3. The proposed approach

A linearization method based on Taylor series expansion is adopted. Each nonlinear multi variable function of the NLES given in Definition 2.1 is considered as $f_i(x_1, ..., x_n)$, $(i = 1, ..., m)$ and $A = (a_1, ..., a_n)$ is a nonnegative chosen point. By using the linear terms of Taylor series generated at the point A as presented in Definition 2.4, each original nonlinear equation of the NLES is reduced to a linear equation. Because the higher order terms will be close to zero while x_j is sufficiently close to a_j, we omit them to obtain the approximation. Thus, by using the expansion, each nonlinear function f_i in n variables is linearized and a linear equations system is obtained. Using the linear equations system obtained, the algorithm generated to solve NLES is presented below.

Step 1. Load an NLES having m equations in n variables such that

$$\begin{aligned}
f_1(x_1, ..., x_n) &= 0 \\
f_2(x_1, ..., x_n) &= 0 \\
&\vdots \\
f_m(x_1, ..., x_n) &= 0.
\end{aligned} \tag{1}$$

Step 2. Choose any initial arbitrary nonnegative point such that $A = (a_1, ..., a_n)$.

Step 3. Linearize each equation in (1) by generating Taylor series expansion at the chosen point A, and construct a linear equations system having m equations in n variables as follows

$$\begin{aligned}
f_1(A) + \sum_{i=1}^{n} \frac{\partial f_1(A)}{\partial x_i}(x_i - a_i) &= 0 \\
f_2(A) + \sum_{i=1}^{n} \frac{\partial f_2(A)}{\partial x_i}(x_i - a_i) &= 0 \\
&\vdots \\
f_m(A) + \sum_{i=1}^{n} \frac{\partial f_m(A)}{\partial x_i}(x_i - a_i) &= 0.
\end{aligned} \tag{2}$$

Step 4. Solve the linearized equations system (2), and obtain a solution $(\bar{x}_1, ..., \bar{x}_n)$.

Step 5. Consider the solution $(\bar{x}_1, ..., \bar{x}_n)$ and introduce new variables $\underline{x}_j$, $(j = 1, ..., n)$ by adding balancing variables

$$\underline{x}_j = \bar{x}_j + u_j - v_j \tag{3}$$

where u_j and v_j, $(j = 1, ..., n)$ are nonnegative and defined as $0 \leq u_j \leq 1$ and $0 \leq v_j \leq 1$.

Step 6. Substitute the new variables (3) in the NLES (1).

Step 7. Linearize the NLES obtained in Step 6 by generating Maclaurin series expansion.

Step 8. Construct a LP problem such that

$$\begin{aligned}
&\text{Min } \sum_{j=1}^{n}(u_j + v_j) \\
&s.t. \\
&f_{1L}(u_j, v_j) = 0 \\
&f_{2L}(u_j, v_j) = 0 \\
&\vdots \\
&f_{mL}(u_j, v_j) = 0
\end{aligned} \tag{4}$$

where the subscript L defines the linearization, and solve (4).

Step 9. If all u_j and v_j, $(j = 1, ..., n)$ are zero, $\underline{\bar{x}}_j$, $(j = 1, ..., n)$ is a solution for the NLES (1), and STOP. Else, determine $\underline{x}_j$, assign $\underline{x}_j$ to $\bar{x}_j$, go to Step 5, and continue.

The flowchart of proposed approach is given in Figure 1.

4. Numerical experiments

Example 1 [7] Consider the following NLES:

$$\begin{aligned}
x_1 + 2x_2 - 3 &= 0 \\
2x_1^2 + x_2^2 - 5 &= 0.
\end{aligned} \tag{5}$$

Linearize each equation in (5) by generating Taylor series expansion at arbitrary nonnegative point $A(3,5)$. Thus, we have the following linearized equations system as

$$\begin{aligned}
x_1 + 2x_2 &= 3 \\
12x_1 + 10x_2 &= 48.
\end{aligned} \tag{6}$$

The solution of linearized system (6) is $(x_1, x_2) = (4.7143, -0.8571)$. Then, introduce new variables $x_1 = 4.7143 + u_1 - v_1$, $x_2 = -0.8571 + u_2 - v_2$, respectively, and substitute these variables in the NLES (5). After linearizing the NLES (5) by generating Maclaurin series expansion, the following LP problem is constructed:
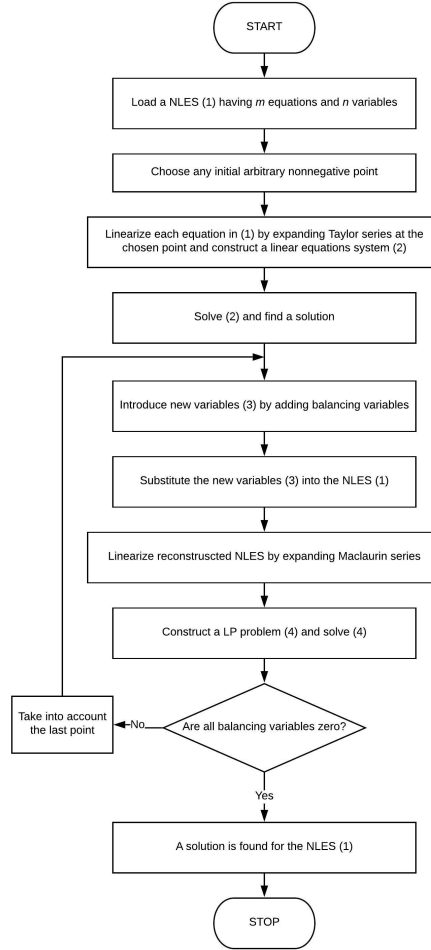
Figure 1. The flowchart of finding solution of NLES.

$$\text{Min} \; \sum_{j=1}^{2} (u_j + v_j)$$

s.t.

$$1(u_1 - v_1) + 2(u_2 - v_2)$$
$$+ f_{1L}(0, 0, 0, 0) = 0 \quad (7)$$
$$18.8572(u_1 - v_1) - 1.7142(u_2 - v_2)$$
$$+ f_{2L}(0, 0, 0, 0) = 0.$$

Table 1. Summarized Results of Example 1 (k is the number of iterations).

k	x_1^k	x_2^k	$||x^k - x^{k-1}||$
0	3.0000	5.0000	-
1	2.6760	0.1620	4.8488
2	1.7892	0.6054	0.9915
3	1.5192	0.7404	0.3019
4	1.4884	0.7558	0.0344
5	1.4880	0.7560	0.0004
6	1.4880	0.7560	0.0000

Example 2 [7] Consider the following NLES:

$$x_1^2 + x_2^2 + x_3^2 - 1 = 0$$
$$2x_1^2 + x_2^2 - 4x_3 = 0 \quad (8)$$
$$3x_1^2 - 4x_2^2 + x_3^2 = 0.$$

Optimal solution of the LP problem (7) is found as
$(u_1, v_1, u_2, v_2) = (0, 2.0383, 1.0191, 0)$, and it is used to determine new variables as $x_1 = 2.6760 + u_1 - v_1$, $x_2 = 0.1620 + u_2 - v_2$, respectively. This approach is applied recurrently until all balancing variables are found zero. The summarized results are given in Table 1.

Linearize each equation in (8) by generating Taylor series expansion at arbitrary nonnegative point $A(1, 1, 1)$. The solution of linearized equations system is found as $(x_1, x_2, x_3) = (0.8269, 0.7308, 0.4423)$. New variables are introduced as $x_1 = 0.8269 + u_1 - v_1$, $x_2 = 0.7308 + u_2 -$

v_2 and $x_3 = 0.4423 + u_3 - v_3$, respectively. Constructed LP problems are solved until all balancing variables are found zero, and the desired solution is obtained after four iterations. The summarized results are given in Table 2.

Table 2. Summarized Results of Example 2 (k is the number of iterations).

k	x_1^k	x_2^k	x_3^k	$\|\|x^k - x^{k-1}\|\|$
0	1.0000	1.0000	1.0000	-
1	0.7114	0.6371	0.3457	0.8019
2	0.6984	0.6286	0.3426	0.0158
3	0.6983	0.6285	0.3426	0.0001
4	0.6983	0.6285	0.3426	0.0000

Example 3 [2] Consider the following NLES:

$$\begin{aligned} \exp x_1 - x_2 - 2 &= 0 \\ \cos x_1 + x_2 - 1 &= 0. \end{aligned} \tag{9}$$
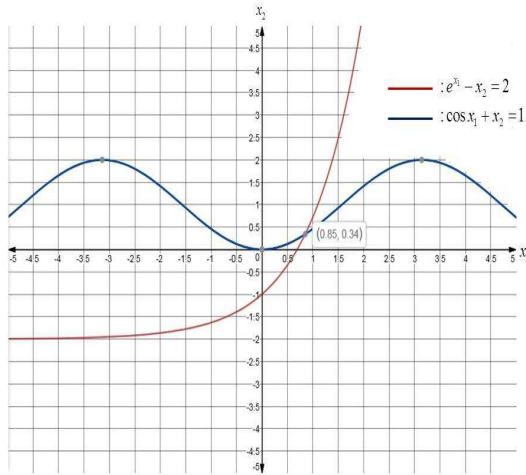

Figure 2. The graph of Example 3.

Linearize each equation in (9) by generating Taylor series expansion at point $A(0, \pi/2)$. The solution of linearized equations system is found as $(x_1, x_2) = (1, 0)$. New variables are introduced as $x_1 = 1 + u_1 - v_1$ and $x_2 = 0 + u_2 - v_2$, respectively. The approach is processed and the solution of (9) is found that is illustrated in Figure 2. The summarized results are given in Table 3.

Table 3. Summarized Results of Example 3 (k is the number of iterations).

k	x_1^k	x_2^k	$\|\|x^k - x^{k-1}\|\|$
0	0.0000	$\pi/2$	-
1	0.8622	0.3438	1.4996
2	0.8503	0.3402	0.0124
3	0.8502	0.3402	0.0001
4	0.8502	0.3402	0.0000

5. Conclusion

In this paper, a linearization approach is proposed to solve NLESs. Although our approach based on linearization using Taylor series involves more iterations than many other methods used in the literature, the fundamental of the approach is based on a very basic and important formation. Therefore, this proposed approach can be used to have less computational complexity and easier application and to obtain more accurate results. Numerical experiments are presented from the literature to demonstrate the ability and accuracy of the proposed approach for solving NLES.

References

[1] Grapsa, T.N. and Vrahatis, M.N. (2003). Dimension reducing methods for systems of nonlinear equations and unconstrained optimization: A review. *Recent Advances in Mechanics and Related Fields*, 215-225.

[2] Frontini, M. and Sormani, E. (2004). Third-order methods from quadrature formulae for solving systems of nonlinear equations. *Applied Mathematics and Computation*, 149(3), 771-782.

[3] Babolian, E., Biazar, J. and Vahidi, A.R. (2004). Solution of a system of nonlinear equations by Adomian decomposition method. *Applied Mathematics and Computation*, 150(3), 847-854.

[4] Nie, P. (2004). A null space method for solving system of equations. *Applied Mathematics and Computation*, 149(1), 215-226.

[5] Nie, P. (2006). An SQP approach with line search for a system of nonlinear equations. *Mathematical and Computer Modelling*, 43(3-4), 368-373.

[6] Jafari, H. and Daftardar-Gejji, V. (2006). Revised Adomian decomposition method for solving a system of nonlinear equations. *Applied Mathematics and Computation*, 175(1), 1-7.

[7] Darvishi, M.T. and Barati, A. (2007). A third-order Newton-type method to solve systems of nonlinear equations. *Applied Mathematics and Computation*, 187(2), 630-635.

[8] Golbabai, A. and Javidi, M. (2007). A new family of iterative methods for solving system of nonlinear algebric equations. *Applied Mathematics and Computation*, 190(2), 1717-1722.

[9] Biazar, J. and Ghanbary, B. (2008). A new approach for solving systems of nonlinear equations. *International Mathematical Forum*, 3(38), 1885-1889.

[10] Grosan, C. and Abraham, A. (2008). A new approach for solving nonlinear equations systems. *IEEE Transactions on Systems, Man, and Cybernetics-Part A: Systems and Humans*, 38(3), 698-714.

[11] Hosseini, M.M. and Kafash, B. (2010). An efficient algorithm for solving system of nonlinear equations. *Applied Mathematical Sciences*, 4(3), 119-131.

[12] Gu, C. and Zhu, D. (2012). A filter algorithm for nonlinear systems of equalities and inequalities. *Applied Mathematics and Computation*, 218(20), 10289-10298.

[13] Vahidi, A.R., Javadi, S. and Khorasani, S.M. (2012). Solving system of nonlinear equations by restarted Adomains method. *Applied Mathematical Sciences*, 6(11), 509-516.

[14] Sharma, J.R. and Gupta, P. (2013). On some efficient techniques for solving systems of nonlinear equations. *Advances in Numerical Analysis*, 2013.

[15] Wang, H. and Pu, D. (2013). A nonmonotone filter trust region method for the system of nonlinear equations. *Applied Mathematical Modelling*, 37(1-2), 498-506.

[16] Zhang, W. (2013). Methods for solving nonlinear systems of equations (Technical report). Department of Mathematics, University of Washington, Seattle, WA, USA.

[17] Dhamacharoen, A. (2014). An efficient hybrid method for solving systems of nonlinear equations. *Journal of Computational and Applied Mathematics*, 263, 59-68.

[18] Izadian, J., Abrishami, R. and Jalili, M. (2014). A new approach for solving nonlinear system of equations using Newton method and HAM. *Iranian Journal of Numerical Analysis and Optimization*, 4(2), 57-72.

[19] Narang, M., Bhatia, S. and Kanwar, V. (2016). New two-parameter Chebyshev-Halley-like family of fourth and sixth-order methods for systems of nonlinear equations.

[20] Saheya, B., Chen, G., Sui, Y. and Wu, C. (2016) A new Newton-like method for solving nonlinear equations. *SpringerPlus*, 5(1), 1269.

[21] Wang, X. and Fan, X. (2016). Two efficient derivative-free iterative methods for solving nonlinear systems. *Algorithms*, 9(1), 14.

[22] Xiao, X.Y. and Yin, H.W. (2016). Increasing the order of convergence for iterative methods to solve nonlinear systems. *Calcolo*, 53(3), 285-300.

[23] Balaji, S., Venkataraman, V., Sastry, D. and Raghul, M. (2017). Solution of system of nonlinear equations using integrated RADM and ADM. *International Journal of Pure and Applied Mathematics*, 117(3), 367-373.

[24] Madhu, K., Babajee, D.K.R. and Jayaraman, J. (2017). An improvement to double-step Newton method and its multi-step version for solving system of nonlinear equations and its applications. *Numerical Algorithms*, 74(2), 593-607.

[25] Sharma, J.R. and Arora, H. (2017). Improved Newton-like methods for solving systems of nonlinear equations. *SeMA Journal*, 74(2), 147-163.

[26] Pourjafari, E. and Mojallali, H. (2012). Solving nonlinear equations systems with a new approach based on invasive weed optimization algorithm and clustering. *Swarm and Evolutionary Computation*, 4, 33-43.

[27] Dai, J., Wu, G., Wu, Y. and Zhu, G. (2008). Helicopter trim research based on hybrid genetic algorithm. 7th World Congress on Intelligent Control and Automation, 2007-2011.

[28] Hirsch, M.J., Pardalos, P.M. and Resende, M.G.C. (2009). Solving systems of nonlinear equations with continuous GRASP. *Nonlinear Analysis: Real World Applications*, 10(4), 2000-2006.

[29] Remani, C. (2012). Numerical methods for solving systems of nonlinear equations (Technical report). Lakehead University, Thunder Bay, Ontario, Canada.

Gizem Temelcan *is a PhD candidate in the Department of Mathematical Engineering at Yildiz Technical University and a lecturer at Istanbul Aydin University. Her research interests include optimization, operational research and fuzzy mathematics.*
http://orcid.org/0000-0002-1885-0674

Mustafa Sivri *was born in Turkey in 1953. He is a Professor in Mathematical Engineering department at Yildiz Technical University, and a faculty member since 1999. He received his B.E. and M.S. degrees from*

Ege University and PhD. degree from Yildiz Technical University in 1975,1978,1983; respectively. His research interests lie on the area of optimization, ranging from theory to implementation.

http://orcid.org/0000-0002-0524-8502

Inci Albayrak *was born in Turkey in 1969. She is a Professor in Mathematical Engineering department at Yildiz Technical University, and a faculty member since 1992. She received her B.E., M.S. and*

PhD. degrees in Mathematical Engineering from the Yildiz Technical University in 1990,1993,1997; respectively. Her current research interests lie on the area of fuzzy mathematical programming, ranging from theory to implementation. She actively cooperates with researchers.

http://orcid.org/0000-0001-6906-9880

RESEARCH ARTICLE

Optimal control of fractional integro-differential systems based on a spectral method and grey wolf optimizer

Raheleh Khanduzi^{*a}, Asyieh Ebrahimzadeh^{b}, and Samaneh Panjeh Ali Beik^{c}

^{a} Department of Mathematics and Statistics, Gonbad Kavous University, Golestan, Iran
^{b} School of Basic Sciences, Farhangian University, Tehran, Iran
^{c} Young Researchers and Elite Club, Karaj Branch, Islamic Azad University, Karaj, Iran
khanduzi@gonbad.ac.ir, ebrahimzadeh263@gmail.com, panjehali@alumni.iust.ac.ir

ARTICLE INFO	ABSTRACT
	This paper elaborated an effective and robust metaheuristic algorithm with acceptable performance based on solution accuracy. The algorithm applied in solution of the optimal control of fractional Volterra integro-differential (FVID) equation which be substituted by nonlinear programming (NLP). Subsequently the FIVD convert the problem to a NLP by using spectral collocation techniques and thereafter we execute the grey wolf optimizer (GWO) to improve the speed and accuracy and find the solutions of the optimal control and state as well as the optimal value of the cost function. It is mentioned that the utilization of the GWO is simple, due to the fact that the GWO is global search algorithm, the method can be applied to find optimal solution of the NLP. The efficiency of the proposed scheme is shown by the results obtained in comparison with the local methods. Further, some illustrative examples introduced with their approximate solutions and the results of the present approach compared with those achieved using other methods.

1. Introduction

The main purpose of this essay is to introduce an efficient approach for solving following optimal control problem (OCP):

Problem A: Find optimal control u^* and corresponding optimal state x^* that minimize the quadratic performance index

$$J = \int_0^T \left(x^2(t) + u^2(t) + f(t)x(t) + g(t)u(t) \right) dt, \tag{1}$$

subject to the fractional Volterra integro-differential (FVID) equation

$$D^\alpha x(t) = a(t)x(t) + b(t)u(t) + \int_0^t (K(t,s))\varphi(x(s))ds, \tag{2}$$

where $a(t)$, $b(t)$, $g(t)$, $f(t)$ are known and real valued functions which are belonged to $L^2[0,T]$ and $\varphi(x(s))$ is a nonlinear function in terms of the unknown function $x(s)$.

In various problems of physics, mechanics and engineering, fractional differential equations have been proved to be a valuable tool in the modeling of many phenomena. There are many applications in viscoelasticity, electrochemistry, control and electromagnetic, [1, 2]. In consequence, the subject of fractional equations is gaining much importance and attention. Meanwhile, the study of OCP governed by fractional integro differential equations is also important as such systems occur in various problems of applied nature. Some approaches for numerical solutions of fractional optimal control problems can be found in [3–6].

---

*Corresponding Author

We need to be mindful only special cases of OCPs can be solved analytically, so choosing the best numerical schemes in terms of rapidity of convergence and accuracy is significant. The method implemented for discritizing mentioned OCP is spectral method, which is one of the most accurate method which used by several author and in different kind of functional problems for example see the [7,8] and the references in. The idea is to write the solution of OCP as a sum of Bernoulli polynomials, substituting these approximations in the OCP yields a NLP in the coefficients which can be solved using any metaheuristic algorithm presented in the literature for solving an optimization problem.

Among these algorithms, nature-inspired metaheuristic algorithms are appropriate for global searches according to ability in exploring globally and exploiting locally. Mirjalili et al. [9] proposed grey wolf optimizer (GWO) algorithm inspired by the behavior of grey wolves in nature. Indeed, the GWO algorithm simulated the leadership hierarchy and hunting behavior of grey wolves. GWO has shown a good performance when applied to solve nonlinear continuous optimization problems. The GWO algorithm is also compared with particle swarm optimization, gravitational search algorithm, differential evolution, evolutionary programming, and evolution strategy to confirm its results.

So, GWO is theoretically able to solve our NLP. Some points on the advantages of the GWO have been expressed:

- The social hierarchy helped GWO to visit the best solutions generated over the course of iteration.
- The encircling procedure determined a circle-shaped neighborhood around the solutions which can be developed to higher dimensions as a hyper-sphere.
- The random parameters helped candidate solutions to have hyper-spheres with different random radii.
- The hunting approach accepted candidate solutions to detect the probable location of the prey.
- Exploration and exploitation are warranted by the adaptive values of two parameters.
- The adaptive values of parameters helped GWO to efficiently trade off between exploration and exploitation.
- The GWO had only two main parameters to be controlled.

The paper is organized as follows: In section 2, the basic concepts about the Bernoulli polynomials and how to approximate the functions in terms of these polynomials is interpreted. Also, the operational matrices of fractional integration are mentioned. As we have provided some definition of fractional calculus. In section 3, the outline of our spectral scheme for discritizing aforementioned optimal control problem and obtaining the resulted NLP is presented. Section 4 is devoted to explain the grey wolf optimizer algorithm for solving the problem under consideration.

In section 5, numerical results are reported to verify the applicability of the presented method in comparison with the other methods in the literature. Through these examples, the superiority of these three bases functions are also discussed. Finally, section 6 ends this paper with a brief conclusion and some remarks.

2. Preliminaries

In this section, we give some basic concepts we require.

2.1. Fractional Calculus

This section, reviews some basic definitions and notations of fractional integral and derivative which are applied further in this work [10].

Definition 1. *The Riemann-Liouville fractional integral operator of order α, is defined by*

$$I^{\alpha}\xi(t) = \frac{1}{\Gamma(\alpha)} \int_0^t (t-s)^{\alpha-1}\xi(s)ds, \quad \alpha > 0, t > 0. \tag{3}$$

in which $\Gamma(.)$ denotes the Gamma function and for $\alpha = 0$, we set $I^0\xi(t) = \xi(t)$.

Definition 2. *Let $n = \lceil \alpha \rceil$ ($\lceil . \rceil$ denotes ceiling function, $\lceil t \rceil = min\{z \in Z : z \geq t\}$), the operator D^{α}, defined by*

$$D^{\alpha}\xi(t) = D^n I^{n-\alpha}\xi(t),$$

is called the Riemann-Liouville fractional differential operator of order α. For $\alpha = 0$, we set $D^0 = I$, the identity operator.

The one type of fractional derivative is Caputo fractional derivative, which is frequently used in applications.

Definition 3. *The Caputo fractional derivative of f, is defined as*

$$D_*^\alpha \xi(t) = \begin{cases} I^{n-\alpha} D^n \xi(t), & n-1 < \alpha < n, n \in N, \\ \frac{d^n}{dt^n} \xi(t), & \alpha = n. \end{cases}$$

$$(4)$$

Lemma 1. *Let* $\alpha, \beta \geq 0$, $c_1, c_2 \in \Re$ *and* $f(t), g(t) \in L_1[0, T]$. *Then*

1) $I^\alpha I^\beta f(t) = I^\beta I^\alpha f(t)$,
2) $I^\alpha I^\beta f(t) = I^{\beta+\alpha} f(t)$,
3) $D^\alpha(c_1 f(t) + c_2 g(t)) = c_1 D^\alpha(f(t)) + c_2 D^\beta(g(t))$

Note that for $n-1 < \alpha < n$, $n \in N$

$$I^\alpha D_*^\alpha \xi(t) = \xi(t) - \sum_{k=0}^{n-1} \xi^{(k)}(0^+) \frac{t^k}{k!}. \quad (5)$$

hold almost everywhere on $[0, T]$.

2.2. An overview on Bernoulli polynomials

Bernoulli polynomials of order m can be defined with the following formula [11],

$$\beta_m(t) = \sum_{i=0}^m \binom{m}{i} \alpha_i t^{m-i}, \quad (6)$$

where α_i, $i = 0, 1, \cdots, m$ are Bernoulli numbers. These numbers are a sequence of signed rational numbers which arise in the series expansion of trigonometric functions [12] and can be defined by the identity

$$\frac{t}{e^t - 1} = \sum_{i=0}^m \alpha_i \frac{t^i}{i!}. \quad (7)$$

The first few Bernoulli numbers are

$$\alpha_0 = 1, \quad \alpha_1 = -\frac{-1}{2}, \quad \alpha_2 = -\frac{1}{6}, \quad \alpha_4 = \frac{-1}{30}. \quad (8)$$

with $\alpha_{2i+1} = 0$, $i = 1, 2, 3, \cdots$. Bernoulli polynomials form a complete basis over the interval [0, 1] [13]. These polynomials satisfy the following formula [12]

$$\int_0^1 \beta_n(t) \beta_m(t) dt = (-1)^{n-1} \frac{m! n!}{(m+n)!} \alpha_{n+m}, \quad (9)$$
$$m, n \geq 1$$

The first few Bernoulli polynomials are

$$\beta_0(t) = 1,$$

$$\beta_1(t) = t - \frac{1}{2},$$

$$\beta_2(t) = t^2 - t + \frac{1}{6},$$

$$\beta_3(t) = t^3 - \frac{3}{2}t^2 + \frac{1}{2}t.$$

Presume that $H := L^2[0, 1]$ and

$$Y = span\{\beta_0, \beta_1, \ldots, \beta_m\},$$

where $m \in \mathbb{N} \cup \{0\}$ and β_i's are the Bernoulli polynomials. Since $Y \subset H$ is a finite dimensional vector space, for every $f \in H$, there exists a unique $y_0 \in Y$ such that

$$\|f - y_0\|_2 \leq \|f - y\|_2 \qquad \forall y \in Y,$$

in which $\|f\|_2 = \sqrt{\langle f, f \rangle}$. Here, the function y_0 is called the best approximation to f out of Y. As $y_0 \in Y$, we may conclude that

$$f(t) \approx y_0(t) = \sum_{j=0}^m c_j \beta_j(t) = C^T \Psi(t),$$

where

$$\Psi^T(t) = (\beta_0(t), \beta_1(t), \ldots, \beta_m(t)), \quad (10)$$

and
$C^T = (c_0, c_1, \ldots, c_m)$ such that C uniquely calculated by

$$C = Q^{-1} \int_1^1 f(t) \Psi(t) dt, \quad (11)$$

where $Q \in \mathbb{R}^{(m+1) \times (m+1)}$ is said the dual matrix of $\Psi(t)$ and given by

$$Q = \int_0^1 \Psi(t) \Psi^T(t) dt.$$

For more details about best approximation see [13].

2.3. Bernoulli operational matrix of the fractional integration

In in recent years, the operational matrices have attracted researchers attention and applied to solving problems consisted of continuous operators (such as integral, derivative, delay, etc.). Moreover, the numerical methods via these operational matrices are easily implemented and have the following characteristics:

⋄ play a significant role as a preconditioner in inverse problems,

⋄ have higher accuracy due to their sparsity.

The RiemannLiouville fractional integration of the vector $\Psi(t)$ given in Equation (10) can be expressed by [4]

$$I^\gamma \Psi(t) = F^\gamma \Psi(t), \qquad (12)$$

in which F^γ is the $(m+1) \times (m+1)$ RiemannLiouville fractional operational matrix of integration. Although F^γ given in [4] we use the different way and notations to show this matrix. For this purpose, Assume that S

$$I^\gamma \beta_i(t) = I^\gamma (\sum_{k=0}^{i} \binom{i}{k} \alpha_k t^{i-k})$$

$$= \sum_{k=0}^{i} \binom{i}{k} \alpha_k I^\gamma t^{i-k}$$

$$= \sum_{k=0}^{i} \frac{\binom{i}{k} \Gamma(i-k+1)\alpha_k}{\Gamma(i-k+1+\gamma)} t^{i-k+\gamma}$$

$$(13)$$

Now if $t^{i-k+\gamma}$ approximated in terms of Bernoulli polynomials we can define each elements of $S = [s_{ij}]_{(m+1)\times(m+1)}$ as

$$s_{ij} = \sum_{k=0}^{i} \sum_{l=0}^{j} \frac{\Gamma(i-k+1)\binom{i}{k}\binom{j}{l}\alpha_k \alpha_l}{\Gamma(i-k+1+\gamma)(-l-k+\alpha+1+i+j)}, \quad i,j = 0, 1, \ldots, m. \qquad (14)$$

As a results F^γ can be expressed as

$$F^\gamma = SQ^{-1} \qquad (15)$$

3. Bernoulli polynomial collocation method

For discretization of the integro-differential dynamic system (2), we express the fractional state rate $D^\gamma x(t)$ and control variable $u(t)$ in terms of Bernoulli polynomial as

$$D^\gamma x(t) \simeq X^T \Psi(t),$$

$$u(t) = U^T \Psi(t), \qquad (16)$$

where X^T and U^T are unknown vectors and $\Psi(t)$ given in (10). Using Lemma 2.1. Equation (12), $x(t)$ can be represented by

$$x(t) = I^\gamma D^\gamma x(t) + x(0) \simeq (X^T F^\gamma + E^T)\Psi(t). \qquad (17)$$

F^γ is the fractional operational matrix of integration and $E^T = [x_0, 0, \ldots, 0]_{(1 \times (m+1))}$. Now we replace (16) and (17) in dynamic system (2)

$$X^T \Psi(t) - a(t)(X^T F^\gamma + E^T)\Psi(t) - b(t)U^T \Psi(t)$$

$$- \int_0^t k(t,s)\varphi((X^T F^\gamma + E^T)\Psi(s)) \, ds = 0, \qquad (18)$$

In order to specify the unknown coefficients in (18), we collocate this equation at $m+1$ collocation points. So (18) can be rewrite as

$$X^T \Psi(t_i) - a(t_i)(X^T F^\gamma + E^T)\Psi(t_i) - b(t_i)U^T \Psi(t_i)$$

$$- \int_0^{t_i} k(t_i,s)\varphi((X^T F^\gamma + E^T)\Psi(s)) \, ds = 0. \qquad (19)$$

In above equation, t_i, $i = 0, \ldots, m$ are the Chebyshev-Gauss-Lobatto nodes in $[0,1]$ which we chose them as suitable collocation points. In order to utilize the Gauss-Legendre (GL) quadrature formula, by means of transformation $s = \frac{t_i}{2}(\tau + 1)$, (19) convert to

$$X^T \Psi(t_i) - a(t_i)(X^T F^\gamma + E^T)\Psi(t_i)$$

$$- b(t_i)U^T \Psi(t_i) - \frac{t_i}{2} \sum_{j=0}^{N} \omega_j k(t_i, \frac{t_i}{2}(\tau_j + 1))$$

$$\times \varphi((X^T F^\gamma + E^T)\Psi(\frac{t_i}{2}(\tau_j + 1))) = 0, \qquad (20)$$

where τ_js are GL nodes, zeros of Legendre polynomials in the interval $[-1,1]$ and ω_js are the corresponding weights. Although explicit formulas for quadrature nodes are not known, the weights can be expressed in closed form by the following relation consequently, the controlled FVID (2) is reduced to $m+1$ nonlinear algebraic.

For discritization of the performance index stated in (1) , we approximate $f(t)$ and $g(t)$ by Bernoulli polynomials respectively as

$$f(t) = F^T \Psi(t), \quad g(t) = G^T \Psi(t). \qquad (21)$$

Substituting (21) in (1) conclude that

$$J = \int_0^1 (X^T \Psi(t)\Psi^T(t)X + U^T \Psi(t)\Psi^T(t)U$$

$$+ F^T \Psi(t)\Psi^T(t)X + G^T \Psi(t)\Psi^T(t)U) \, dt \qquad (22)$$

Integrating (21) on $[0, 1]$ results

$$J = X^T Q X + U^T Q U + F^T Q X + G^T Q U, \quad (23)$$

in which Q given in (11). So, The OCP given in (1) and (23) is converted to a NLP with objective functional (22) and constraints (20).

The resulted NLP problem is also large scale. So, it is of great importance to use an efficacious and compatible metaheuristic algorithm which generates the solutions with high computational decisions. This research utilizes a new metaheuristic algorithm, called grey wolf optimizer (GWO) to solve the problem under consideration. The next section describe the GWO and its elements and mechanisms to solve NLP governed by OCP.

4. Grey wolf optimizer

The proposed mathematical programming problem is the nonlinear and large scale. So, we need to solve the problem by studying both local and metaheuristic approaches. Among the local algorithms, the trust region method plays a vital role in solving large-scale nonlinear optimization problems because of its efciency [14, 15]. However, it finds local solutions in a long amount of time as the $T >> 1$ or the problem dimension increases. To prevent this type of imperfection, a grey wolf optimizer (GWO) algorithm is proposed in this research. The GWO algorithm is a nature-inspired metaheuristic algorithm which mimitates the leadership hierarchy and hunting structure of grey wolves.Therefore, high-performing metaheuristic approach with high computational high-precision numerical solutions and short execution time is implemented to solve the problem. The GWO obtains near-optimal solutions or the global minimum of objective functional in more efcient way. About the proposed algorithm, it is necessary to note that the GWO is really suitable and appropriate for the nonlinear optimization problems with the number of more variables and constraints, specially when solving large-sized instances of the problems [16–19]. For the problem, the values of objective functionals show that GWO's performance is better than local method in terms of the approximate solution of functions $x(t)$ and $u(t)$. So, on the base of above-mentioned points, one can come to the conclusion that the GWO is a favorable candidate for solving the problem if $T >> 1$.

The GWO algorithm has derivation-free procedures. In contrast to gradient-based optimization algorithms, this metaheuristic algorithm minimizes the problems stochastically. The optimization mechanism begins with random solution, and there is no need to compute the derivative of search regions or gradient information of the objective functionals to obtain the global minimum of the problem. This makes the GWO algorithm highly applicable for the NLP problems with unknown derivative information. On the other hand, the simplicity of the GWO is particularly advantageous in the presence of non-smooth objective functionals, for which exact algorithms may fail to reach their global solutions. Viability of the GWO is analyzed using some non-smooth mathematical functions and engineering design problems [20–22].So, the GWO algorithm is a favorable choice and a competitive algorithm when considering non-smooth, and non-linear functions.

In this section, the essential nature of the GWO algorithm is explained. GWO algorithm is an new nature-inspired metaheuristic algorithm which was first introduced by Mirjalili et al. [9].

GWO is a technique inspired from the nature and grey wolves. The GWO algorithm simulated the leadership hierarchy and hunting behavior of grey wolves.

In leadership hierarchy, alpha, beta, delta, and omega were applied as four grey wolves. Also, hunting, searching for prey, encircling prey, and attacking prey were as the three main components of GWO. These new steps are discussed in the following section.

4.1. Social hierarchy

To formulate the social hierarchy of wolves, the fittest solution is considered as the alpha (α). Accordingly, the second and third best solutions are called beta (β) and delta (δ), respectively. The remaining candidate solutions are then represented as omega (ω). Also, the hunting mechanism is constructed by α, β, and δ. The ω wolves followed these three wolves.

4.2. Encircling prey

As seen in nature, grey wolves surround prey during the hunt. This surrounding behavior is given by:

$$\vec{D} = |\vec{C}.\vec{X}_p(t) - \vec{X}(t)| \quad (24)$$

$$\vec{X}(t+1) = \vec{X}(t) - \vec{A}.\vec{D} \quad (25)$$

where t shows the current iteration, $\vec{A}$ and $\vec{C}$ are coefficient vectors, $\vec{X}_p$ is the position vector of the

prey, and $\vec{X}$ shows the position vector of a grey wolf. The vectors $\vec{A}$ and $\vec{C}$ are computed followed by:

$$\vec{A} = 2\vec{a}.\vec{r}_1 - \vec{a} \qquad (26)$$

$$\vec{C} = 2.\vec{r}_2 \qquad (27)$$

where r_1 and r_2 are random vectors in $[0, 1]$. Over the course of iterations, components of $\vec{a}$ are linearly reduced from 2 to 0.

4.3. Hunting

Grey wolves have the capability to identify the position of prey and envelop them. The hunt is normally conducted by the alpha. The beta and delta also partake in hunting sometimes. So, the first three best solutions is saved and the other search agents (including the omegas) is performed to update their positions based on the position of the best search agents. This process is stated with the following equations:

$$\vec{D}_\alpha = |\vec{C}_1.\vec{X}_\alpha - \vec{X}|, \vec{D}_\beta = |\vec{C}_2.\vec{X}_\beta - \vec{X}|,$$
$$\vec{D}_\delta = |\vec{C}_3.\vec{X}_\delta - \vec{X}|, \qquad (28)$$
$$\vec{X}_1 = \vec{X}_\alpha - \vec{A}_1.(\vec{D}_\alpha), \vec{X}_2 = \vec{X}_\beta - \vec{A}_2.(\vec{D}_\beta),$$
$$\vec{X}_3 = \vec{X}_\delta - \vec{A}_3.(\vec{D}_\delta), \qquad (29)$$
$$\vec{X}_{t+1} = \frac{\vec{X}_1 + \vec{X}_2 + \vec{X}_3}{3} \qquad (30)$$

4.4. Attacking prey

At the end of the hunt, grey wolves rush at the prey when it stops moving. To model nearing the prey, the value of $\vec{a}$ is reduced from 2 to 0. Then, the variation range of $\vec{A}$ is also reduced by $\vec{a}$. Especially, $\vec{A}$ is a random value in the interval $[-2a, 2a]$.

4.5. Search for prey

Grey wolves chiefly explore based on the position of the alpha, beta, and delta. They get away from each other to search for prey and converge to rush prey. To formulate divergence, $\vec{A}$ with random values greater than 1 or less than -1 is applied to enforce the search agent for diverging from the prey. So, the GWO algorithm employing global search strategy and this confirms exploration.

As it is seen in Eq. (27), the $\vec{C}$ vector is also another factor of exploration. This factor obtains random weights for prey to stochastically accentuate ($C > 1$) or unaccentuate ($C < 1$) the efficacy

of prey in determining the distance in Eq. (24). The C vector can be also considered as the efficacy of barriers to nearing prey in nature. Usually, the barriers in nature exist in the hunting paths of wolves and impede them from swiftly and comfortably nearing prey.

Briefly, the search procedure starts with generating a random population of grey wolves. Over the course of iterations, alpha, beta, and delta wolves suggest the possible location of the prey. The distance from the prey is updated by each candidate solution The parameter a is reduced from 2 to 0 to accentuate exploration and exploitation, respectively.

Candidate solutions favor divergence of the prey when $|\vec{A}| > 1$ and move towards (converge) the prey when $|\vec{A}| < 1$. Finally, the GWO algorithm stops when an end criterion is satisfied.

The pseudo code of the GWO algorithm is presented in Algorithm 1.

Algorithm 1. *GWO algorithm*
1: *Initialize the grey wolf population $X_i = (x_i(t), u_i(t))(i = 1, \cdots, n)$*
2: *Initialize a, A, and C*
3: *Calculate the fitness of each search agent*
4: *X_α=the best search agent*
5: *X_β=the second best search agent*
6: *X_δ=the third best search agent*
7: **while** *(k < Max number of iterations)* **do**
8:     **for** *each search agent* **do**
9:        *Update the position of the current search agent by equation (30)*
10:     **end for**
11:     *Update a, A, and C*
12:     *Calculate the fitness of all search agents*
13:     *Update X_α, X_β, and X_δ*
14:     *k = k + 1*
15: **end while**
16: *return X_α*

5. Numerical experiments

In these examples, firstly the OCP is converted to a NLP with proposed method in section 3. The resulted NLP is solved by using *fmincon* function in MATLAB and GWO algorithm to find local and global minimum of constrained nonlinear function, respectively.

For ease of references in tables, we use the proposed method and local method to demonstrate the results obtained from solving NLP with *fmincon* function and GWO algorithm, respectively.

In order to demonstrate and justify the performance and the accuracy of our scheme on OCPs governed by fractional integro-differential equation, we consider the following examples.

Example 1. *Consider the following OCP*

$$J = \int_0^1 \left((x(t) - e^t)^2 + (u(t) - e^{3t})^2 \right) dt \quad (31)$$

subject to the nonlinear fractional integro-differential equation

$$D^\alpha x(t) - \frac{3}{2}x(t) + \frac{1}{2}u(t) - \int_0^t \left(e^{t-s}x^3(s) \right) ds = 0,$$

$$x(0) = 1. \quad (32)$$

The problem is to find the optimal control $u^*(t)$, which minimizes the quadratic performance index (31). For this problem, the exact solution in the case of $\alpha = 1$ is given by [23]

$$x(t) = e^t, \quad u(t) = e^{3t}.$$

In Table 1, one can compare the optimal value of objective functional by utilizing GOW algorithm as well as local method in $M = 7$ and different values of α. The numerical results for $x(t)$ and $u(t)$ in $M = 7$ and $\alpha = 0.5, 0.7, 0.9$ and 1 are plotted in Figures 1-2. In these figures, we see that our approximate solutions converge to exact solution. The results obtained with GOW method demonstrate validity and effectiveness of proposed method.
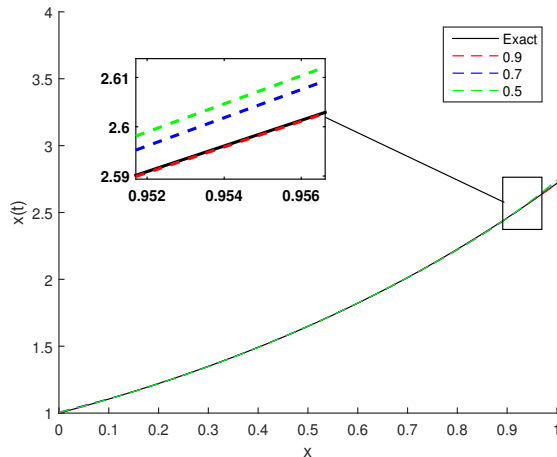

Figure 1. State $x(t)$ as a function of t for the Example 1 for $m = 7$ and different values of α (green: $\alpha = 0.5$, blue: $\alpha = 0.7$, red: $\alpha = 0.9$.
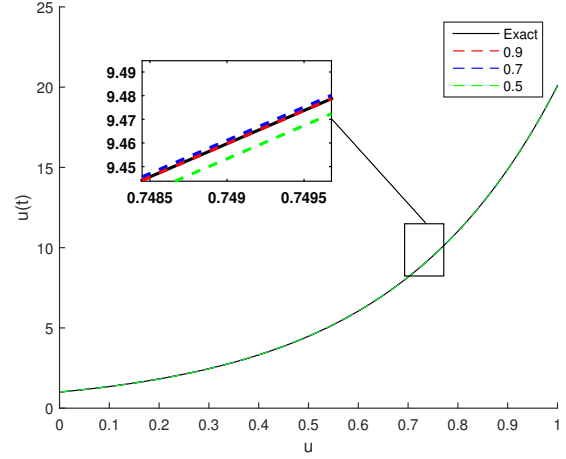

Figure 2. Control $u(t)$ as a function of t for the Example 1 for $m = 7$ and different values of α (green: $\alpha = 0.5$, blue: $\alpha = 0.7$, red: $\alpha = 0.9$.

Example 2. *Consider the following nonlinear problem [23]*

$$\min \ J = \int_0^1 \left((x(t) - e^{t^2})^2 + (u(t) - (1 + 2t))^2 dt, \right.$$

$$D^\alpha x(t) + x(t) - u(t)$$
$$- \int_0^t \left(t(1 + 2t)e^{s(t-s)}x(s) \right) ds = 0$$

The optimal control u^* and corresponding optimal state x^* for $\alpha = 1$ are respectively $1 + 2t$ and e^{t^2}.

Figures 3-4 show the approximate solution of functions $x(t)$ and $u(t)$ using GWO algorithm and local method for $M = 7$ and $\alpha = 0.5, 0.7, 0.9$. The exact solution for $\alpha = 1$ is also represented. The value of objective function with GWO and local methods for $M = 7$ and different values of α are given in Table 2. It is obvious that we can achieve a better approximation with GWO algorithm against local method.

Table 1. The value of J^* for Example 1 ($m = 7$ and different α).

	Local Method			Proposed Method		
	$\alpha = 0.7$	$\alpha = 0.9$	$\alpha = 1$	$\alpha = 0.7$	$\alpha = 0.9$	$\alpha = 1$
J^*	0.389877	0.0946404	4.17536×10^{-11}	2.45×10^{-5}	8.64×10^{-8}	7.74×10^{-12}

Table 2. The value of J^* for Example 2 ($m = 7$ and different α).

	Local Method			Proposed Method		
	$\alpha = 0.7$	$\alpha = 0.9$	$\alpha = 1$	$\alpha = 0.7$	$\alpha = 0.9$	$\alpha = 1$
J^*	0.0515124	0.00494284	1.29×10^{-11}	2.50×10^{-6}	1.22×10^{-8}	4.70×10^{-12}
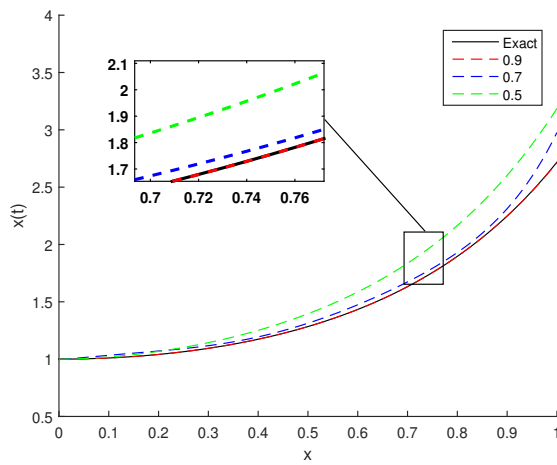
Figure 3. State $x(t)$ as a function of t for the Example 2 for $m = 7$ and different values of α (green: $\alpha = 0.5$, blue: $\alpha = 0.7$, red: $\alpha = 0.9$).
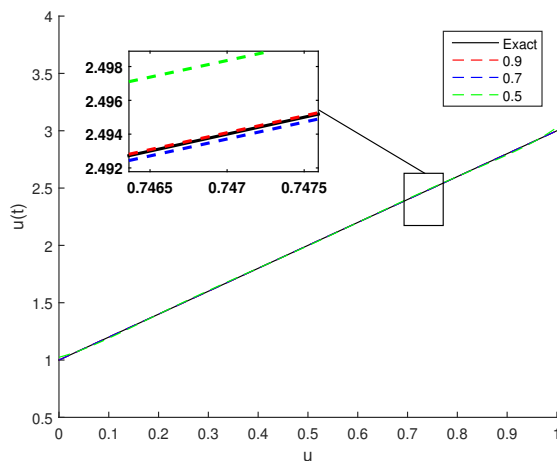
Figure 4. Control $u(t)$ as a function of t for the Example 2 for $m = 7$ and different values of α (green: $\alpha = 0.5$, blue: $\alpha = 0.7$, red: $\alpha = 0.9$).

Example 3. *Consider the minimization of fractional [23]*

$$J = \int_0^1 \left((x(t) - t)^2 + (u(t) - te^{t^2})^2 \right) dt,$$

subject to dynamic state

$$D^\alpha x(t) - x(t) - u(t) + 2 \int_0^t \left(tse^{-x^2(s)} \right) ds = 0.$$

The optimal control $u^*(t)$ and corresponding optimal state $x(t)$ for $\alpha = 1$ are as follows:

$$x^*(t) = t,$$
$$u^*(t) = 1 - te^{-t^2}$$

We solve this OCP using GWO and local methods for for $M = 7$ and various α. Figures 5-6 show that as $\alpha \to 1$, the approximate solutions with GWO algorithm tend to the exact solution in the case of $\alpha = 1$. The value of objective function with GWO and local methods for $M = 7$ and different values of α, is shown in Table 3. From Table 3, we can see that the value of objective function based on GWO is all less than the least value of objective function obtained by local method.

Table 3. The value of J^* for Example 3 ($m = 7$ and different α).

	Local Method			*Proposed Method*		
	$\alpha = 0.7$	$\alpha = 0.9$	$\alpha = 1$	$\alpha = 0.7$	$\alpha = 0.9$	$\alpha = 1$
J^*	0.0366084	0.00374084	8.84×10^{-14}	4.71×10^{-7}	9.37×10^{-8}	2.12×10^{-14}
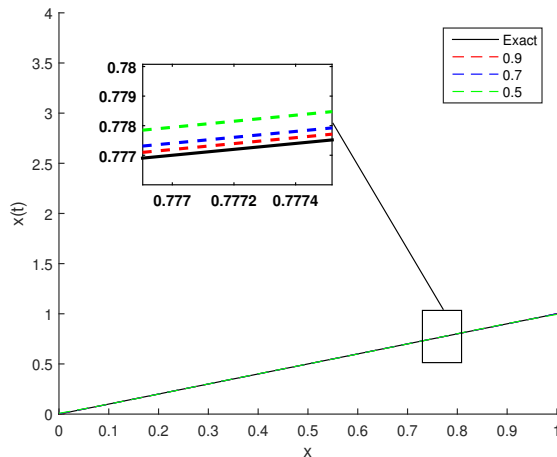
Figure 5. State $x(t)$ as a function of t for the Example 3 for $m = 7$ and different values of α (green: $\alpha = 0.5$, blue: $\alpha = 0.7$, red: $\alpha = 0.9$).
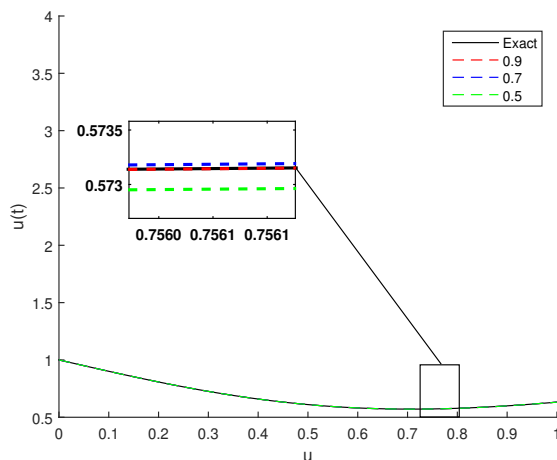
Figure 6. Control $u(t)$ as a function of t for the Example 3 for $m = 7$ and different values of α (green: $\alpha = 0.5$, blue: $\alpha = 0.7$, red: $\alpha = 0.9$).

6. Conclusions

By utilizing spectral method, OCP governed by fractional Volttera-integro differential equation is converted to a NLP.

In this research, a powerful and efficacious meta-heuristic algorithm called Grey Wolf Optimizer (GWO) is utilized to obtain the solutions of the optimal control and state as well as the optimal value of the objective function.

The GWO algorithm imitated the leadership hierarchy with four types of grey wolves and hunting procedure with searching for prey, encircling prey, and attacking prey. These strategies confirmed the preferable exploitation, exploration capability and efficient escape from local optimum of the GWO.

Numerical experiments verify the validity and the applicability of the proposed method. Comparisons with the exact solution and other methods show that this technique is a powerful and efficient tool for solving the fractional OCP.

Acknowledgments

References

[1] Diethelm, K., Ford, N.J. and Freed, A.D. (2004). Detailed error analysis for a fractional Adams method. *Numerical Algorithms*,36, 31–52.

[2] Huang, L., Li, X.F., Zhao, Y.L. and Duan, X.Y. (2011). Approximate solution of fractional integro-differential equations by Taylor expansion method. *Computers & Mathematics with Applications*, 62(3), 1127–1134.

[3] Tohidi, E. and Nik, H.S. (2015). A bessel collocation method for solving fractional optimal control problems. *Applied Mathematical Modelling*, 39(2), 455–465.

[4] Keshavarz, E., Ordokhani, Y. and Razzaghi, M. (2015). A numerical solution for fractional optimal control problems via Bernoulli polynomials. *Journal of Vibration and Control*, 22(18), 3889–3903.

[5] Zaky, M.A. and Machado, J.A.T. (2017). On the formulation and numerical simulation of distributed-order fractional optimal control problems. *Communications in Nonlinear Science and Numerical Simulation*, 52, 177–189.

[6] Salati, A.B., Shamsi, M., Torres, D.F.M. (2019). Direct transcription methods based on fractional integral approximation formulas for solving nonlinear fractional optimal control problems. *Communications in Nonlinear Science and Numerical Simulation*, 67, 334–350.

[7] Youssri, Y.H., and Abd-Elhameed, W.M. (2018). Spectral tau algorithm for solving a class of fractional optimal control problems via Jacobi polynomials. *An International Journal of Optimization and Control: Theories & Applications (IJOCTA)*, 8(2), 152–160.

[8] Youssri, Y.H. and Hafez, R.M. (2019). Chebyshev collocation treatment of VolterraFredholm integral equation with error analysis. *Arabian Journal of Mathematics*, 1–10.

[9] Mirjalili, S.A., Mirjalili, S.M. and Lewis, A. (2014). Grey wolf optimizer. *Advances in Engineering Software*, 69, 46–61.

[10] Baleanu, D., Diethelm, K., Scalas, E. and Trujillo, J.J. (2012). **Fractional calculus models and numerical methods**. Nonlinearity and Chaos, Series on Complexity, World Scientific.

[11] Costabile, F., Dellaccio, F. and Gualtieri, M.I. (2006). A new approach to Bernoulli polynomials. *Rendiconti di Matematica, Serie VII*, 26, 1–12.

[12] Arfken G. (1985). *Mathematical methods for physicists*. 3rd edn. San Diego, CA: Academic Press.

[13] Kreyszig, E. (1978). *Introductory functional analysis with applications*. New York: John Wiley and Sons.

[14] Yuan, Y.X. (2000). A review of trust region algorithms for optimization. *In Iciam*, 99(1), 271–282.

[15] Sadjadi, S.J. and Ponnambalam, K. (1999). Advances in trust region algorithms for constrained optimization. *Applied Numerical Mathematics*, 29(3), 423–443.

[16] Faris, H., Aljarah, I., Al-Betar, M. A. and Mirjalili, S. (2018). Grey wolf optimizer: a review of recent variants and applications. *Neural computing and applications*, 30(2), 413–435.

[17] Wang, J.S. and Li, S.X. (2019). An improved grey wolf optimizer based on differential evolution and elimination mechanism. *Scientific reports*, 9(1), 71–81.

[18] Liu, H., Hua, G., Yin, H. and Xu, Y. (2018). An intelligent grey wolf optimizer algorithm for distributed compressed sensing. *Computational intelligence and neuroscience*, 2018.

[19] Gupta, S. and Deep, K. (2018). An opposition-based chaotic grey wolf optimizer for global optimisation tasks. *Journal of Experimental & Theoretical Artificial Intelligence*, 1–29.

[20] Abdo, M., Kamel, S., Ebeed, M., Yu, J. and Jurado, F. (2018). Solving non-smooth optimal power flow problems using a developed grey wolf optimizer. *Energies*, 11(7), 1692.

[21] Pradhan, M., Roy, P. K. and Pal, T. (2016). Grey wolf optimization applied to economic load dispatch problems. *International Journal of Electrical Power & Energy Systems*, 83, 325–334.

[22] Tung, N.S. and Chakravorty, S. (2015). Grey Wolf optimization for active power dispatch planning problem considering generator constraints and valve point effect. International Journal of Hybrid Information Technology, 8(12), 117–134.

[23] Maleknejad, K., Ebrahimzadeh, A. (2014). Optimal control of volterra integrodifferential systems based on legendre wavelets and collocation method. *Journal of Mathematical, Computational, Physical an Quantum Engineering*,8(7), 1040–1044.

Raheleh Khanduzi *is an Assistant Professor in the Department of Mathematics at Gonbad Kavous University, Gonbad Kavous, Golestan, Iran. She received her Ph.D. in Operations Research from the Faculty of Mathematics, Shiraz University of Technology, Shiraz, Iran, M.Sc. in Operations Research from the Faculty of Mathematics, K.N. Toosi University of Technology, Tehran, Iran and B.Sc. from the Faculty of Mathematics, Shahrood University of Technology, Shahrood, Iran. Her research interests include Linear and Nonlinear Optimization, Mathematical Modeling, Convex Optimization, Combinatorial Optimization, Facility Location Problem, Service Industries or Systems, System Protection, Stackelberg Game, Optimal Control, Metaheuristic Algorithms, Exact Algorithms, Data Mining, and Neural Networks.*
http://orcid.org/0000-0002-0979-4041

Asyieh Ebrahimzadeh *is an Assistant Professor in the School of Basic Sciences, Farhangian University, Tehran, Iran. She received her Ph.D. in Numerical Analysis from the Department of Applied Mathematics at Iran University of Science and Technology, Tehran, Iran, M.Sc. in Applied Mathematics from the Faculty of Mathematics, K.N. Toosi University of Technology, Tehran, Iran and B.Sc. in Mathematics at the ValiAsr University of Rafsanjan, Kerman, Iran. Her scientific interests include Optimal control, Wavelets, Integral Equations, and Orthogonal functions.*
http://orcid.org/0000-0002-4684-7640

Samaneh Panjeh Ali Beik *is a Ph.D. in Numerical Analysis from the Department of Applied Mathematics at Iran University of Science and Technology, Tehran, Iran. She received her M.Sc. in Applied Mathematics from the Faculty of Mathematics, Alzahra University, Tehran, Iran. Her scientific interests include Optimal control, Wavelets, Fractional Equations, and Orthogonal functions.*
http://orcid.org/0000-0002-6559-3279

RESEARCH ARTICLE

New complex-valued activation functions

Nihal Özgür*[a], Nihal Taş[a] and James F. Peters[b]

[a] *Department of Mathematics, Balıkesir University, Turkey*
[b] *Computational Intelligence Laboratory, University of Manitoba, Canada*
nihal@balikesir.edu.tr, nihaltas@balikesir.edu.tr, James.Peters3@umanitoba.ca

ARTICLE INFO	ABSTRACT
	We present a new type of activation functions for a complex-valued neural network (CVNN). A proposed activation function is constructed such that it fixes a given ellipse. We obtain an application to a complex-valued Hopfield neural network (CVHNN) using a special form of the introduced complex functions as an activation function. Considering the interesting geometric properties of the plane curve ellipse such as focusing property, we emphasize that these properties may have possible applications in various neural networks.

1. Introduction

Recently, complex-valued neural networks (CVNN) have been used in various fields such as optoelectronics, imaging, signal processing, quantum neural devices and artificial neural information processing by many researchers (see [1–8] for more details). For example, Gandal et al. tried to evaluate and compare the relative performance of CVNN using different error functions [3]. Hirose studied what merits of CVNNs arise from [4]. Jalab and Ibrahim introduced a new type of complex-valued sigmoid function for a fully multi-layered CVNN [5]. Zimmerman et al. gave the differences between complex-valued and real-valued neural networks and studied the problems of CVNNs gradients computations by combining the global and local optimization algorithm [8]. Oladipo and Gbolagade investigated modified logistic sigmoid as relates to analytic univalent functions by means of subordination properties in terms of starlikeness, convexity and close-to-convexity [6]. In [7], it was proposed a

wind prediction system for the wind power generation using ensemble of multiple complex extreme learning machines and used the elegant theory of conformal mapping to find better transformations in the complex domain for enhancing its prediction capability.

In a CVNN, one of the main problems is selecting of nodes activation function (see [9]). In this paper, we propose a new type of complex-valued functions as an activation function for a complex-valued Hopfield neural network (CVHNN). These functions fix a given ellipse on the complex plane. To construct an appropriate activation function, a suitable ellipse can be chosen according to the particular problem. The main advantage of this choice is the increase in the number of fixed points of a neural network with a geometrical meaning.

On the other hand, an ellipse has some interesting properties such as the focusing property (see [10]). We recall the focusing property of an ellipse. Let E be the ellipse of the normal form with semi-major axis a and semi-minor axis b:

$$E : \frac{x^2}{a^2} + \frac{y^2}{b^2} = 1. \qquad (1)$$

*Corresponding Author

The foci of the ellipse E of the form (1) are as follows:
$$c_{1,2} = \mp\sqrt{a^2 - b^2}.$$

The radius of the ellipse E is $r = 2a$ and so we rewrite the equation of the ellipse of the form (1) as
$$E : |z - c_1| + |z - c_2| = r.$$

It is a well-known fact from geometry that a light ray which leaves a focus c_1 of an ellipse will be reflected to other focus c_2 (see [10] and [11] for more details). Using this interesting property and the following proposition, Frantz proposed an application to the open problem about trapped reflections described in [12]. It was seen that the light ray gradually approaches a horizontal trajectory and never leaves the container (see [10] for more details).

Proposition 1. *[10] Let a light ray leave a focus of an ellipse with departure angle $\theta_0 \in (0, \pi)$ and let the successive departure angles of the ray be $\theta_1, \theta_2, \dots$. Then $\theta_n \uparrow \pi$.*

Therefore, it is possible to get some applications of these kind properties of an ellipse in neural networks. It is known that the plane curve ellipse has appeared in many applications in real life problems (for example, see [13–21]). We expect that our study will help to generate some new researches and applications on complex-valued neural networks.

2. Complex Functions That Fix an Ellipse

In this section, we investigate a new type of complex-valued function which fixes an ellipse. We begin with the following definition.

Definition 1. *Let E be any ellipse on the complex plane. If a complex function T satisfies the condition $T(z) = z$ for each complex number $z \in E$, then the ellipse E is called the fixed ellipse of T.*

Now we consider an ellipse E of the form (1). If we take $x = \frac{z+\overline{z}}{2}$ and $y = \frac{z-\overline{z}}{2i}$, then we can rewrite the equation of this ellipse as
$$\alpha\left(z^2 + \overline{z}^2\right) + \beta z\overline{z} - 1 = 0, \qquad (2)$$
where $\alpha = \frac{1}{4a^2} - \frac{1}{4b^2}$ and $\beta = \frac{1}{2a^2} + \frac{1}{2b^2}$.
Conversely, let us consider the following general equation
$$\alpha\left(z^2 + \overline{z}^2\right) + \beta z\overline{z} + \gamma = 0. \qquad (3)$$

The equation (3) defines an ellipse if the following conditions hold:

(1) $\alpha, \beta, \gamma \in \mathbb{R}$ and $\beta > 0$,

(2) $\gamma < 0$, $2\alpha + \beta > 0$ and $2\alpha - \beta < 0$.

Indeed, if we write $z = x + iy$ then we have
$$\alpha\left(z^2 + \overline{z}^2\right) + \beta z\overline{z} + \gamma = 0$$
$$\Rightarrow \alpha\left[(x+iy)^2 + (x-iy)^2\right] + \beta\left(x^2 + y^2\right) + \gamma = 0$$
$$\Rightarrow 2\alpha x^2 + \beta x^2 - 2\alpha y^2 + \beta y^2 + \gamma = 0$$
$$\Rightarrow (2\alpha + \beta)x^2 + (\beta - 2\alpha)y^2 + \gamma = 0$$
$$\Rightarrow \frac{2\alpha+\beta}{-\gamma}x^2 + \frac{\beta-2\alpha}{-\gamma}y^2 = 1$$
$$\Rightarrow \frac{x^2}{\frac{-\gamma}{2\alpha+\beta}} + \frac{y^2}{\frac{\gamma}{2\alpha-\beta}} = 1.$$

If we choose a and b such as
$$a = \sqrt{\frac{-\gamma}{2\alpha + \beta}} \text{ and } b = \sqrt{\frac{\gamma}{2\alpha - \beta}}, \qquad (4)$$
then the equation (3) defines the ellipse $\frac{x^2}{a^2} + \frac{y^2}{b^2} = 1$.

Now we present a complex function which fixes an ellipse of the form (3). For any complex number z on the ellipse E, we get
$$\alpha\left(z^2 + \overline{z}^2\right) + \beta z\overline{z} + \gamma = 0$$
$$\Rightarrow -\alpha z^2 - \beta z\overline{z} = \alpha\overline{z}^2 + \gamma$$
$$\Rightarrow z\left(-\alpha z - \beta\overline{z}\right) = \alpha\overline{z}^2 + \gamma$$
$$\Rightarrow \frac{-\gamma - \alpha\overline{z}^2}{\alpha z + \beta\overline{z}} = z.$$

Hence we obtain the following theorem.

Theorem 1. *Let E be any ellipse with the equation (3). If we define the transformation T_1 as*
$$T_1(z) = \frac{-\gamma - \alpha\overline{z}^2}{\alpha z + \beta\overline{z}}, \qquad (5)$$
then T_1 fixes the ellipse E.

If we consider the following transformation T_2 defined as
$$T_2(z) = \frac{-\gamma - \alpha\overline{z}^2 - \alpha z^2}{\beta\overline{z}},$$
then it can be easily seen that T_2 also fixes the ellipse E. Clearly, the transformations $T_1 \circ T_2$ and $T_2 \circ T_1$ fix also the ellipse E. The transformations T_1 and T_2 are not always commutative, that is, it can be $T_1 \circ T_2 \neq T_2 \circ T_1$. For example, if we consider $\gamma = -1$, $\alpha = 1$, $\beta = 3$ and $z = 1$ then we get
$$T_1 \circ T_2(1) = -\frac{2}{3}$$
and
$$T_2 \circ T_1(1) = \infty.$$
Consequently, we can give the following corollary.

Corollary 1. *For each ellipse E, there are at least three transformations T such that*
$$E = \{z \in \mathbb{C} : T(z) = z\}.$$
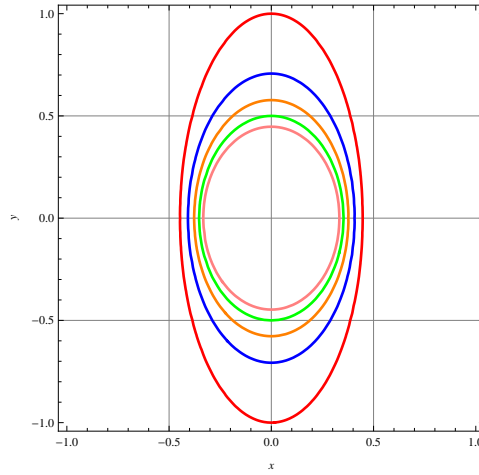*Then E is exactly the set of fixed points of each T.*

Figure 1. The ellipses E_β for $\beta \in \{3, 4, 5, 6, 7\}$.

Taking $\alpha = 0$, $\gamma = -1$, that is, $a = b$ in the equation (4), then we get the transformation

$$T_3(z) = \frac{1}{\beta \overline{z}}$$

and E becomes the circle with the center $z_0 = 0$ and the radius $r = \frac{1}{\sqrt{\beta}}$. Hence, the transformation T_3 fixes the circle E and it is known that the transformation T_3 is an anti-conformal map on the complex plane.

Now we consider the following two families of ellipses:

1) Let $\alpha = c$ be fixed in the equation (5). Then we get $\beta \in (2c, \infty)$ when $c \geq 0$ and $\beta \in (-2c, \infty)$ when $c < 0$. For example, if we choose $\alpha = 1$ in the equation (5), we get $\beta \in (2, \infty)$. In this case, the transformation T_1 fixes the following ellipses E_β defined according to β:

$$E_\beta : \frac{x^2}{a_\beta^2} + \frac{y^2}{b_\beta^2} = 1,$$

where $a_\beta = \sqrt{\frac{1}{\beta+2}}$ and $b_\beta = \sqrt{\frac{1}{\beta-2}}$ for each $\beta \in (2, \infty)$. In the following figure, which has been drawn by Mathematica [22], it is seen how the ellipses E_β change (see Figure 1). The ellipses E_β are indicated with different colors: E_3 is the red ellipse, E_4 is the blue ellipse, E_5 is the orange ellipse, E_6 is the green ellipse and E_7 is the pink ellipse.

2) Let $\beta = c$ be fixed in the equation (5). Then we get $\alpha \in \left(-\frac{c}{2}, \frac{c}{2}\right)$. For example, if we consider $\beta = 1$ in the equation (5), we get $\alpha \in \left(-\frac{1}{2}, \frac{1}{2}\right)$. In this case, the transformation T_1 fixes the following ellipses E_α defined according to α:

$$E_\alpha : \frac{x^2}{c_\alpha^2} + \frac{y^2}{d_\alpha^2} = 1,$$

where $c_\alpha = \sqrt{\frac{1}{1+2\alpha}}$ and $d_\alpha = \sqrt{\frac{1}{1-2\alpha}}$ for each $\alpha \in \left(-\frac{1}{2}, \frac{1}{2}\right)$. It is seen from Figure 2 that how the ellipses E_α change. The ellipses E_α are indicated with different colors: $E_{\frac{1}{4}}$ is the red ellipse, $E_{\frac{1}{8}}$ is the blue ellipse, $E_{\frac{1}{10}}$ is the orange ellipse, $E_{\frac{1}{12}}$ is the green ellipse and $E_{\frac{1}{14}}$ is the pink ellipse.

3. An Application to Complex-Valued Hopfield Neural Networks

Möbius transformations and some related (anti-conformal) maps have been used as activation functions in complex-valued neural networks using some different point of views such as fixed points or fixed circles. It is known that Möbius transformations are the conformal mappings of the complex plane $\mathbb{C}$. A Möbius transformation is a rational function of the form

$$T(z) = \frac{az + b}{cz + d}, \tag{6}$$

where a, b, c, d are complex numbers satisfying $ad - bc \neq 0$. A point z on the complex plane is said to be a fixed point of the Möbius transformation $T(z)$ if $T(z) = z$. A Möbius transformation $T(z)$ has at most two fixed points if it is not identity transformation (see [23], [24] and [25] for the basic properties of Möbius transformations). In [26], it was identified the activation function of a neuron and a single-pole all-pass digital filter section as Möbius transformations and then, the existence of fixed points of a neural network were guaranteed by the underlying Möbius transformation. In [27], Özdemir et al. proposed new types of activation functions which fix a circle for a CVNN. The usage of these types of activation functions leads us to guarantee the existence of the fixed points of a CVHNN.
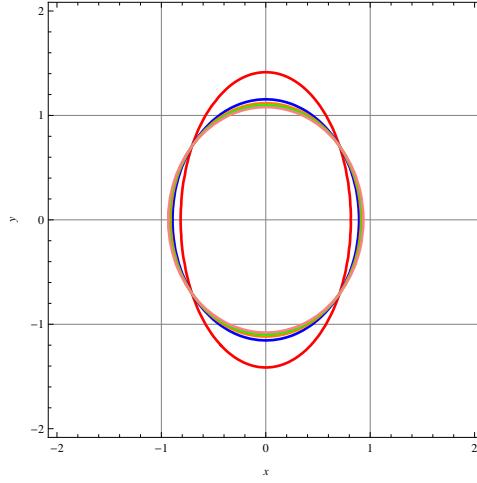
Figure 2. The ellipses E_α for $\alpha \in \left\{ \frac{1}{4}, \frac{1}{8}, \frac{1}{10}, \frac{1}{12}, \frac{1}{14} \right\}$.

In this section, we consider the special form of the transformation (5) defined as follows:

$$S_\alpha(z) = \frac{1 - \alpha \overline{z}^2}{\alpha z + (2\alpha + 1)\overline{z}}, \qquad (7)$$

where $\alpha \geq 0$ (notice that $\gamma = -1$ and β is chosen as $2\alpha + 1$). We propose this function as an activation function for a CVHNN.

If we take $\alpha = 0$ in the equation (7), then we get the following activation function used in a CVHNN in [27]:

$$S_0(z) = \frac{1}{\overline{z}}.$$

Therefore, the transformation S_α defined in (7) which fixes an ellipse of the form (3) with $\gamma = -1$, $\alpha \geq 0$, $\beta = 2\alpha + 1 > 0$ becomes a transformation S_0 which fixes the unit circle. The transformation $S_0(z) = \frac{1}{\overline{z}}$ was used to guarantee the existence of the fixed points of a complex-valued Hopfield neural network (CVHNN). The transformation $S_\alpha(z)$ is not injective while $S_0(z)$ is injective. Also this transformation $S_\alpha(z)$ maps an ellipse of the form (3) onto itself, outside of the ellipse to its inside and inside of the ellipse to its outside. For example, in the following Figure 3, we see the image of the outside of the ellipse $E : 5x^2 + y^2 = 1$ under the transformation defined as

$$S_1(z) = \frac{1 - \overline{z}^2}{z + 3\overline{z}}.$$

At first, we give a brief summary about CVHNNs. In [28], Hopfield presented a recurrent neural network model referred to as the Hopfield neural network (HNN). HNN has been generalized to CVHNN and this generalized neural network has been studied by many authors using different aspects. For example, Kobayashi defined the concept of a hyperbolic neuron and constructed hyperbolic Hopfield neural network [18]. Also he described the symmetric complex-valued Hopfield neural networks using the complex-valued multistate neurons [29].

Following the studies given in [27], here we consider the class of system in $\mathbb{C}$ in order to interest CVHNN given by

$$\dot{z}(t) = -H(z(t))(-Tz(t) + F(z(t)) - U), \quad (8)$$

where $T \in \mathbb{C}^{n \times n}$, $U \in \mathbb{C}^n$ are matrices, $z(t) \in \mathbb{C}^n$ is state vector, $H(z) : \mathbb{C}^n \to \mathbb{C}^{n \times n}$ is a nonlinear function and $F(z) = (S_1(z_1), S_2(z_2), \ldots, S_n(z_n))^T : \mathbb{C}^n \to \mathbb{C}^n$ is an activation function with

$$S_k(z_k) = S_\alpha(z_k) = \frac{1 - \alpha \overline{z_k}^2}{\alpha z_k + (2\alpha + 1)\overline{z_k}}, \quad (9)$$

for some fixed $\alpha \geq 0$ and all $k \in \{1, 2, \ldots, n\}$. We note that the parameter α can be chosen appropriately according to the studied problem. We choose $T \in \mathbb{R}^{n \times n}$ and $U = 0$ in the equation (8) to obtain a relationship between the fixed points of the activation function $F(z)$ and the fixed points of the network. Fixed points of the equation $\dot{z}(t) = -H(z(t))(-Tz(t) + F(z(t)))$ can be obtained by the equation $-H(z)(-Tz + F(z)) = 0$. Suppose that $H(z)$ is a nonsingular matrix then the fixed points are $F(z) = Tz$, which correspond to the fixed points of the activation function.

In our approach, we increase the number of fixed points using the activation function defined in (9) by a point of geometric view. We use the Lyapunov stability to determine whether the fixed points are stable or not (see [27], [30] and [31] and for more details). The fixed points of the CVHNN are isolated since they are on an ellipse. Following the steps used in the proof of Theorem 2 on page 4701 in [27] and using the property $\overline{S_k(\overline{z_k})} = S_k(z_k)$, it can be easily obtained that

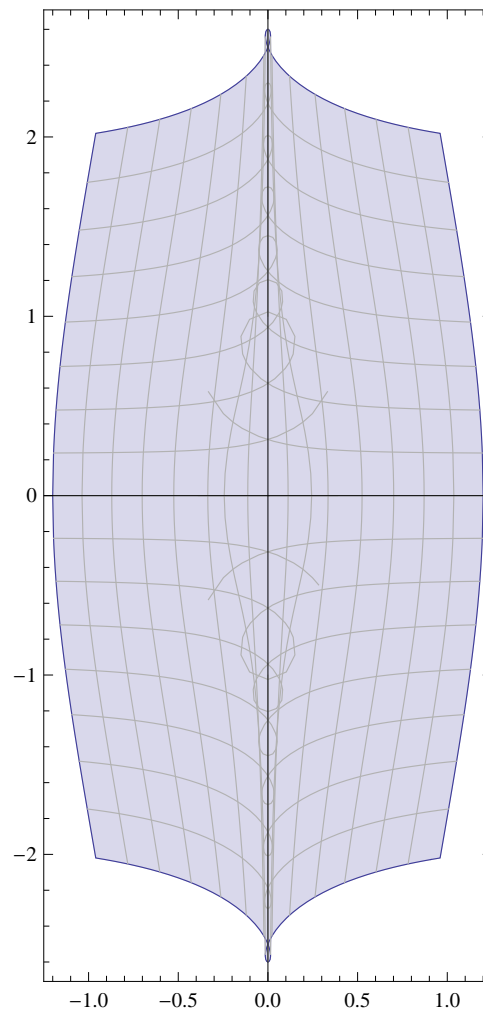$$\dot{E}(z) = -Re\left[(Tz - F(z))(Tz - F(z))^* H(z)^*\right],$$

Figure 3. The geometric interpretation of the transformation $S_1(z)$ for the outside of the ellipse.

which is negative for positive definite matrix $Re\,[H(z)]$ and also equal to zero if and only if $\dot{z}(t) = 0$. So the following theorem gives the stability of the fixed points.

Theorem 2. *Let the inner product be defined on $\mathbb{C}^n$ as $\langle z_1, z_2 \rangle = z_2^* z_1$ where $z_1, z_2 \in \mathbb{C}^n$ and $(.)^*$ denotes the conjugate transpose. Assume that the matrix $T \in \mathbb{R}^{n \times n}$ is symmetric and the matrix $Re\,[H(z)]$ is positive definite. Then the function*

$$E(z) = -\frac{1}{2} z^* T z + Re \left[\sum_{k=1}^{n} \int_0^{\overline{z_k}} \overline{S_k}(s) ds \right]$$

is a Lyapunov function of the CVHNN given by the equation $\dot{z}(t) = -H(z(t))(-Tz(t) + F(z(t)))$.

4. Remarks and Conclusion

We note that a general activation function for a CVHNN can be obtained using the transformation T_1 given in (5). The fixed points of this activation function are on an ellipse with the form

(3). This allows us to choose the appropriate activation function according to the considered problem on a neural network. This activation function can be helpful to construct several neural nets and lead to interesting applications. Proposed activation functions can be considered as the generalizations of ones used in [27].

Finally, we emphasize that the properties of the ellipse, which is fixed by the chosen activation function, are applicable to the neural networks. Therefore, our results have possible applications in real life problems.

Acknowledgments

References

[1] Ceylan, M., Özbay, Y., Uçan, O.N., Yıldırım, E. (2010). A novel method for lung segmentation on chest CT images: complex-valued artificial neural network with complex wavelet transform. *Turk. J. Elec. Eng. and Comp. Sci.*, 18(4), 613-623.

[2] Ceylan, M., Yaşar, H. (2016). A novel approach for automatic blood vessel extraction in retinal images: complex ripplet-I transform and complex valued artificial neural network. *Turk. J. Elec. Eng. and Comp. Sci.*, 24(4), 3212-3227.

[3] Gandal, A.S., Kalra, P.K., Chauhan, D.S. (2007). Performance evaluation of complex valued neural networks using various error functions. *International Journal of Electrical, Computer, Energetic, Electronic and Communication Engineering*, 1(5), 732-737.

[4] Hirose, A. (2009). Complex-valued neural networks: The merits and their origins. *Proceedings of the Internatinal Joint Conference on Neural Networks (IJCNN)*, Atlanta, June 14-19, 1237-1244.

[5] Jalab, H.A., Ibrahim, R. W. (2011). New activation functions for complex-valued neural network. *International Journal of the Physical Sciences*, 6(7), 1766-1772.

[6] Oladipo, A.T., Gbolagade, M. (2014). Some subordination results for logistic sigmoid activation function in the space of univalent functions. *Advances in Computer Science and Engineering*, 12(2), 61-79.

[7] Singh, R.G., Singh, A.P. (2015). Multiple complex extreme learning machine using holomorphic mapping for prediction of wind power generation system. *International Journal of Computer Applications*, 123(18), 24-33.

[8] Zimmermann, H.G., Minin, A., Kusherbaeva, V. (2011). Comparison of the complex valued and real valued neural networks trained with gradient descent and random search algorithms. *ESANN 2011 proceedings, European Symposium on Artificial Neural Networks, Computational Intelligence and Machine Learning. Bruges (Belgium)* 27-29 April 2011.

[9] Kim, T., Adalı, T. (2002) Fully complex multi-layer perceptron network for nonlinear signal processing. *J. VLSI Sig. Process.*, 32, 29-43.

[10] Frantz, M. (1994). A focusing property of the ellipse. *Amer. Math. Monthly*, 101(3), 250-258.

[11] Wilker, J.B. (1995). Further thoughts on a focusing property of the ellipse. *Bull. Belg. Math. Soc.*, 2, 153-159.

[12] Connett, J.E. (1992). Trapped reflections?. *Amer. Math. Monthly*, 99, 178-179.

[13] Di Concilio, A., Guadagni, C., Peters, J.F., Ramanna, S. (2018). Descriptive proximities. Properties and interplay between classical proximities and overlap. *Math. Comput. Sci.*, 12(1), 91-106. *arXiv:1609.06246. MR3767897.*

[14] Ferrer, S., Hanßmann, H., Palacián, J., Yanguas, P. (2002). On perturbed oscillators in 1-1-1 resonance: the case of axially symmetric cubic potentials. *J. Geom. Phys.*, 40(3-4), 320-369.

[15] Grandon, J., Derpich, I. (2011). A Heuristic for the Multi-knapsack Problem. *WSEAS Transactions on Mathematics*, 10(3), 95-104.

[16] Kanan, H.R., Faez, K., Ezoji, M. (2006). An efficient face recognition system using a new optimized localization method, In Pattern Recognition. *ICPR 2006, 18th International Conference on Vol. 3*, 564-567.

[17] Kellner, M.A., Hanning, T., Farr, H. (2002). Real-time analysis of the grain on wooden planks, Machine Vision Applications in Industrial Inspection X. *Vol. 4664. International Society for Optics and Photonics.*

[18] Kobayashi, M., (2013). Hyperbolic Hopfield neural networks. *IEEE Trans. Neural Netw. Learn Syst.* 24(2), 335-341.

[19] Li, J., Zhang, J. (2004). Bifurcations of travelling wave solutions for the generalization form of the modified KdV equation. *Chaos Solitons Fractals* 21(4), 889-913.

[20] Peters, J.F. (2018). Proximal Vortex Cycles and Vortex Nerves. Non-Concentric, Nesting, Possibly Overlapping Homology Cell Complexes. *Journal of Mathematical Sciences and Modelling*, 1 (2), 80-85. *arXiv:1805.03998.*

[21] Zhang, G., Jayas, D.S., White, N.D. (2005). Separation of touching grain kernels in an image by ellipse fitting algorithm. *Biosystems engineering*, 92(2), 135-142.

[22] Wolfram Research. (2019). Inc., Mathematica, Version 12.0, Champaign, IL.

[23] Beardon, A.F. (1983). *The geometry of discrete groups.* Graduate texts in mathematics, vol 91. Springer, New York.

[24] Jones, G.A., Singerman, D. (1987). *Complex functions. An algebraic and geometric viewpoint*, Cambridge University Press, Cambridge.

[25] Kwok, Y.K. (2010). *Applied complex variables for scientists and engineers.* Cambridge University Press, New York.

[26] Mandic, D.P. (2000). The use of Möbius transformations in neural networks and signal processing. *Neural Networks for Signal Processing X*, 1, 185-194.

[27] Özdemir, N., İskender, B.B., Özgür, N.Y. (2011). Complex valued neural network with Möbius activation function. *Commun. Nonlinear Sci. Numer. Simul.*, 16(12), 4698-4703.

[28] Hopfield, J.J. (1982). Neural networks and physical systems with emergent collective computational abilities. *Proc. Nat. Acad. Sci. United States Amer.*, 79(8), 2554-2558.

[29] Kobayashi, M. (2017). Symmetric complex-valued Hopfield neural networks. *IEEE Trans. Neural Netw. Learn Syst.*, 28(4), 1011-1015.

[30] Khalil, H.K. (1996). *Nonlinear systems.* 2nd ed. United States of America: Prentice Hall.

[31] Kuroe, Y., Yoshida, M., Mori, T. (2003). On activation functions for complex-valued neural networks - existence of energy functions. *In: Kaynak O et al., editors. ICANN/ICONIP 2003. LNCS 2714. Berlin Heidelberg: Springer-Verlag*, 985-992.

Nihal Özgür *is currently a professor at Balıkesir University in Turkey. Her research interests include complex functions with one variable, soft set theory, fixed point theory and fixed circle problem.*
http://orcid.org/0000-0002-8152-1830

Nihal Taş *is currently a research assistant at Balıkesir University in Turkey. Her research interests include topological spaces, soft set theory, fixed point theory and fixed circle problem.*
http://orcid.org/0000-0002-4535-4019

James Francis Peters *is currently a professor at University of Manitoba in Canada. His research interests include pattern recognition, engineering, applied, computational mathematics and signal, image, video processing.*
https://orcid.org/0000-0002-1026-4638

An International Journal of Optimization and Control: Theories & Applications (http://ijocta.balikesir.edu.tr)

RESEARCH ARTICLE

Analytical and approximate solution of two-dimensional convection-diffusion problems

Hatıra Günerhan[*]

Department of Mathematics, Faculty of Education, Kafkas University, Kars, Turkey
gunerhanhatira@gmail.com

ARTICLE INFO	ABSTRACT
	In this work, we have used reduced differential transform method (RDTM) to compute an approximate solution of the Two-Dimensional Convection-Diffusion equations (TDCDE). This method provides the solution quickly in the form of a convergent series. Also, by using RDTM the approximate solution of two-dimensional convection-diffusion equation is obtained. Further, we have computed exact solution of non-homogeneous CDE by using the same method. To the best of my knowledge, the research work carried out in the present paper has not been done, and is new. Examples are provided to support our work.

1. Introduction

We consider two-dimensional convection-diffusion equation as follows:

$$\frac{\partial u(a,b,t)}{\partial t} + \beta_a \frac{\partial u(a,b,t)}{\partial a} + \beta_b \frac{\partial u(a,b,t)}{\partial b}$$
$$= \alpha_a \frac{\partial^2 u(a,b,t)}{\partial a^2} + \alpha_b \frac{\partial^2 u(a,b,t)}{\partial b^2} + f(a,b,t),$$
$$\text{in} \quad \Omega \times (0,T], u(a,b,t) = g(a,b,t), \quad (1)$$
$$(a,b) \in \partial\Omega, \quad t \in (0,T], u(a,b,0) = h(a,b),$$
$$(a,b) \in \Omega,$$

where β_a and β_b are progressive velocity components in the direction of a and b respectively, and $\alpha_a > 0$ and $\alpha_b > 0$ are the coefficients of diffusivity in the a and b directions, respectively. And $\alpha_a > 0$ and $\alpha_b > 0$ are $g(a,b,c)$ and $h(a,b)$ are smooth functions and Ω is a subset of R^2 and $(0,T]$ is the time interval, and $\partial\Omega$ is the boundary of Ω.

This equation is frequently used in applied sciences and engineering especially in modeling and simulations of various complex phenomena in science and engineering. This paper first describes RDTM and then uses it to solve the Convection-diffusion equation. In recent years, studies conducted on findings new analytical solutions of differential equations have attracted attention of scientists from all over the world (see [1]- [9]).

And some numerical solutions have been developed to solve these types of convection-diffusion problems. likes: Higher-Order ADI method [10] or rational high-order compact ADI method [11], the alternating direction implicit method [12], the finite element method [13], fourth-order compact finite difference method [14], decomposition Method [15], the finite difference method [16], restrictive taylors approximation [17], The fundamental solution [18], finite difference method [19], combined compact difference scheme and alternating direction implicit method [20], higher order compact schemes method [21], the finite volume method [22], the finite difference and legendre spectral method [23] and even the Monte

---

*Corresponding Author

Carlo method [24]. Keskin in [25] proposed the RDTM to solve various PDE and fractional non-linear partial differential equations.

This method is a repetitive procedure for the solution of a Taylor series differential equations. This technique reduces the size of the computational work and can be easily applied to numerous physical problems. We organize the paper as follows. In section RDTM is used to four types of TDCDP, and section 4 concludes the paper.

2. Analysis of the RDTM

We have a function with three variables $u(a, b, t)$, and presume that it can be shown as an invention of multiple of two functions $u(a, b, t) = v(a, b)w(t)$. $u(a, b, t)$ can be denoted as

$$
u(a, b, t) =
$$
$$
(\sum_{n_1=0}^{\infty} \sum_{n_2=0}^{\infty} V(n_1, n_2)a^{n_1}b^{n_2}) \cdot (\sum_{n_3=0}^{\infty} W(n_3)t^{n_3})
$$
$$
= \sum_{n_1=0}^{\infty} \sum_{n_2=0}^{\infty} \sum_{n_3=0}^{\infty} V(n_1, n_2)W(n_3)a^{n_1}b^{n_2}t^{n_3}
$$
$$
= \sum_{k=0}^{\infty} U_k(a, b)t^k, \quad (2)
$$

where $U_k(a, b)$ is called t-dimensional spectrum function of $u(a, b, t)$. The three-dimensional RDTM are introduced are defined in [26] as follows:

Definition 1. *Assume $u(a, b, t)$ is an analytic function in the domain of interest. The RDTM of $u(a, b, t)$ is defined as*

$$
U_k(a, b) = \frac{1}{k!}[\frac{\partial^k}{\partial t^k}u(a, b, t)]_{t=0}. \quad (3)
$$

Definition 2. *The differential inverse transform of $U_k(a, b)$ is defined as:*

$$
u(a, b, t) = \sum_{k=0}^{\infty} U_k(a, b)t^k. \quad (4)
$$

By inserting equation (3) in (4), we obtain

$$
u(a, b, t) = \sum_{k=0}^{\infty} \frac{1}{k!}[\frac{\partial^k}{\partial t^k}u(a, b, t)]_{t=0}t^k, \quad (5)
$$

Some basic properties of RDTM are presented in Table1 below.

Table 1. The operations for the reduced differential transform method.

Original function	Transformed function
$g(a, b, t) \pm h(a, b, t)$	$G_k(a, b) \pm H_k(a, b)$
$e^{\gamma t}$	$\frac{\gamma^k}{k!}$
$\frac{\partial^c}{\partial t^c}g(a, b, t)$	$\frac{(k+c)!}{K!}G_{k+c}(a, b)$
$g(a, b, t)h(a, b, t)$	$\sum_{l=0}^{k} G_l(a, b)H_{k-l}(a, b)$
$\frac{\partial^w}{\partial a^w}g(a, b, t)$	$\frac{\partial^w}{\partial a^w}G_k(a, b)$
$a^w b^v t^c$	$a^w b^v \delta(k - c) = \begin{cases} a^w b^v, k = c \\ 0, k \neq c \end{cases}$
$\frac{\partial^{w+v+c}}{\partial a^w \partial b^v \partial t^c}g(a, b, t)$	$\frac{\partial^{w+v}}{\partial a^w \partial b^v}\frac{(k+c)!}{k!}G_{k+c}(a, b)$

3. Applications

We used the basic definitions (in Section 2) of the three-dimensional RDTM for solving four examples of Convection-diffusion equations (CDE).

Example 1. Consider the TDCDP (see [15])

$$
\frac{\partial u}{\partial t} - \frac{\partial^2 u}{\partial a^2} - \frac{\partial^2 u}{\partial b^2} = 0, (a, b, t) \in \Omega \times J, \quad (6)
$$

with the initial condition

$$
u(a, b, 0) = \sin(\pi a)\sin(\pi b). \quad (7)
$$

By using the RDTM in equations (6) and (7), we obtain

$$
(k+1)U_{k+1}(a, b) - \frac{\partial^2}{\partial a^2}U_k(a, b) - \frac{\partial^2}{\partial b^2}U_k(a, b) = 0, \quad (8)
$$

from initial condition(7), we have

$$
U_0(a, b) = \sin(\pi a)\sin(\pi b). \quad (9)
$$

By using Eq. (9) in Eq. (8), we obtain $U_k(a, b)$ values for $k = \{0, 1, 2, 3, \cdots\}$ as follows:

$$
\begin{aligned}
U_1(a, b) &= -2\pi^2\sin(\pi a)\sin(\pi b), \\
U_2(a, b) &= 2\pi^4\sin(\pi a)\sin(\pi b), \\
U_3(a, b) &= -\frac{4}{3}\pi^6\sin(\pi a)\sin(\pi b), \\
U_4(a, b) &= \frac{2}{3}\pi^8\sin(\pi a)\sin(\pi b), \\
U_5(a, b) &= -\frac{4}{15}\pi^{10}\sin(\pi a)\sin(\pi b), \\
U_6(a, b) &= \frac{4}{45}\pi^{12}\sin(\pi a)\sin(\pi b), \\
U_7(p, q) &= -\frac{8}{315}\pi^{14}\sin(\pi a)\sin(\pi b), \ldots,
\end{aligned}
$$
$$(10)$$

by using the differential inverse reduced transform of $U_k(a, b)$, we get

$$u(a, b, t) = \sum_{k=0}^{\infty} U_k(a, b)t^k$$

$$= U_0(a, b) + U_1(a, b)t + U_2(a, b)t^2 + \dots$$
$$= \sin(\pi a)\sin(\pi b)(1 - 2\pi^2 t + 2\pi^4 t^2 - \frac{4}{3}\pi^6 t^3$$
$$+ \frac{2}{3}\pi^8 t^4 - \frac{4}{15}\pi^{10}t^5 + \frac{4}{45}\pi^{12}t^6 + \dots),$$

$$(11)$$

by using the closed form in the solution of (11), we obtain following approximate solution

$$u(a, b, t) = \sin(\pi a)\sin(\pi b)e^{-2\pi^2 t}. \qquad (12)$$

Example 2. We consider the non-homogeneous convection-diffusion problem see ([15])

$$\frac{\partial u}{\partial t} + \frac{\partial u}{\partial a} + \frac{\partial u}{\partial b} - \frac{\partial^2 u}{\partial a^2} - \frac{\partial^2 u}{\partial b^2} = 3a^2 - 6a + 2t + 1,$$

$$(a, b, t) \in \Omega \times J, \qquad (13)$$

subject to the initial condition

$$u(a, b, 0) = a^3 + b. \qquad (14)$$

By using the basic properties of RDTM in equations (13) and (14), we obtain the following relations

$$(k + 1)U_{k+1}(a, b) + \frac{\partial}{\partial a}U_k(a, b) + \frac{\partial}{\partial b}U_k(a, b)$$
$$- \frac{\partial^2}{\partial a^2}U_k(a, b) - \frac{\partial^2}{\partial b^2}U_k(a, b)$$
$$= 3a^2\delta(k) - 6a\delta(k) + 2\delta(k-1) + \delta(k),$$

$$(15)$$

Taking the differential transform of Eq.(14), we write

$$U_0(a, b) = a^3 + b. \qquad (16)$$

By using Eq. (16) in Eq. (15), we obtain $U_k(a, b)$ values for $k = \{0, 1, 2, 3, \cdots\}$ as follows

$$U_1(a, b) = 0, U_2(a, b) = 1, U_i(a, b) = 0,$$
$$for(i = 3, 4, 5, \dots). \qquad (17)$$

The exact solution of the equation (13) will assume the following form:

$$u(a, b, t) = \sum_{k=0}^{\infty} U_k(a, b)t^k = a^3 + b + t. \qquad (18)$$

Example 3. We consider the non-homogeneous CDE (see [14])

$$\frac{\partial u}{\partial t} - \frac{\partial^2 u}{\partial a^2} - \frac{\partial^2 u}{\partial b^2} + \frac{\partial u}{\partial a} = (2\pi^2 - 1)e^{-t}\sin(\pi a)\cos(\pi b)$$

$$+ \pi e^{-t}\cos(\pi a)\cos(\pi b), \ (a, b, t) \in \Omega \times J, \quad (19)$$

with the initial condition

$$u(a, b, 0) = \sin(\pi a)\cos(\pi b). \qquad (20)$$

By using the basic properties of RDTM in equations (19) and (20), we obtain the following relations

$$(k + 1)U_{k+1}(a, b) - \frac{\partial^2}{\partial a^2}U_k(a, b)$$
$$- \frac{\partial^2}{\partial b^2}U_k(a, b) + \frac{\partial}{\partial a}U_k(a, b)$$
$$= (2\pi^2 - 1)\frac{(-1)^k}{k!}\sin(\pi a)\cos(\pi b) \qquad (21)$$
$$+ \pi\frac{(-1)^k}{k!}\cos(\pi a)\cos(\pi b),$$

from initial condition(20), we have

$$U_0(a, b) = \sin(\pi a)\cos(\pi b). \qquad (22)$$

By using Eq. (22) in Eq. (21), we obtain $U_k(a, b)$ values for $k = \{0, 1, 2, 3, \cdots\}$ as follows:

$$
\begin{array}{lcl}
U_1(a, b) & = & -sin(\pi a)cos(\pi b), \\
U_2(a, b) & = & \frac{1}{2}sin(\pi a)cos(\pi b), \\
U_3(a, b) & = & -\frac{1}{6}sin(\pi a)cos(\pi b), \\
U_4(a, b) & = & \frac{1}{24}sin(\pi a)cos(\pi b), \\
U_5(a, b) & = & -\frac{1}{120}sin(\pi a)cos(\pi b), \\
U_6(a, b) & = & \frac{1}{720}sin(\pi a)cos(\pi b), \\
U_7(a, b) & = & -\frac{1}{5040}sin(\pi a)cos(\pi b), \dots,
\end{array}
\qquad (23)
$$

by using the differential inverse reduced transform of $U_k(a, b)$,we get

$$u(a, b, t) = \sum_{k=0}^{\infty} U_k(a, b)t^k = U_0(a, b) + U_1(a, b)t + \dots$$

$$= sin(\pi a)cos(\pi b)(1 - t + \frac{t^2}{2!} - \frac{t^3}{3!} + \frac{t^4}{4!} - \frac{t^5}{5!} + \frac{t^6}{6!} - \dots),$$

$$(24)$$

by using the closed form in the solution of (24), we obtain the following exact solution

$$u(a, b, t) = e^{-t}sin(\pi a)cos(\pi b). \qquad (25)$$

Example 4. Consider the TDCDP (see [14])

$$\frac{\partial u}{\partial t} = \frac{\partial^2 u}{\partial a^2} + \frac{\partial^2 u}{\partial b^2} - \frac{\partial u}{\partial a}$$
$$+ e^{-t}(2\pi^2 - 1)sin(\pi a)sin(\pi b) \qquad (26)$$
$$+ \pi e^{-t}cos(\pi a)sin(\pi b), (a, b, t) \in \Omega \times J,$$

with the initial condition

$$u(a, b, 0) = sin(\pi a)sin(\pi b). \qquad (27)$$

By using the basic properties of RDTM in equations (26) and (27), we obtain the following relations

$$(k+1)U_{k+1}(a,b) = \frac{\partial^2}{\partial a^2}U_k(a,b) + \frac{\partial^2}{\partial b^2}U_k(a,b)$$

$$-\frac{\partial}{\partial a}U_k(a,b) + (2\pi^2 - 1)\frac{(-1)^k}{k!}sin(\pi a)sin(\pi b)$$

$$+ \pi\frac{(-1)^k}{k!}cos(\pi a)sin(\pi b), \quad (28)$$

from initial condition(28), we have

$$U_0(a,b) = sin(\pi a)sin(\pi b). \quad (29)$$

By using Eq. (29) in Eq. (28), we obtain $U_k(a,b)$ values for $k = \{0,1,2,3,\cdots\}$

$$\begin{array}{rcl}
U_1(a,b) & = & -sin(\pi a)sin(\pi b), \\
U_2(a,b) & = & \frac{1}{2}sin(\pi a)sin(\pi b), \\
U_3(a,b) & = & -\frac{1}{6}sin(\pi a)sin(\pi b), \\
U_4(a,b) & = & \frac{1}{24}sin(\pi a)sin(\pi b), \quad (30) \\
U_5(a,b) & = & -\frac{1}{120}sin(\pi a)sin(\pi b), \\
U_6(a,b) & = & \frac{1}{720}sin(\pi a)sin(\pi b), \\
U_7(a,b) & = & -\frac{1}{5040}sin(\pi a)sin(\pi b), \ldots,
\end{array}$$

by using the differential inverse reduced transform of $U_k(a,b)$,we get

$$\begin{array}{rcl}
u(a,b,t) & = & sin(\pi a)sin(\pi b)(1 - t + \frac{t^2}{2} \\
& - & \frac{t^3}{6} + \frac{t^4}{24} - \frac{t^5}{120} + \frac{t^6}{720} - \cdots), \quad (31)
\end{array}$$

by using the closed form in the solution of (31) we obtain the following exact solution

$$u(a,b,t) = e^{-t}sin(\pi a)sin(\pi b). \quad (32)$$

4. Conclusion

In this study, we used RDTM to solve convection-diffusion problems and showed that RDTM is an effective and appropriate technique for finding exact solutions of the TDCDP which we have investigated here. On the other hand the results are quite reliable for solving this problem. The exact closed form solution was obtained for all the examples presented in this paper. RDTM offers an excellent opportunity for future research.

References

[1] Baskonus, H.M. (2019). Complex Soliton Solutions to the Gilson-Pickering Model. *Axioms*, 8(1), 18.

[2] Ilhan, O.A., Esen, A., Bulut, H., & Baskonus, H.M. (2019). Singular Solitons in the Pseudo-parabolic Model Arising in Nonlinear Surface Waves. *Results in Physics*, 12, 17121715.

[3] Cattani, C., Sulaiman, T.A., Baskonus, H.M., & Bulut, H. (2018). Solitons in an inhomogeneous Murnaghan's rod. *European Physical Journal Plus*, 133(228), 1-12.

[4] Baskonus, H.M., Sulaiman, T.A. & Bulut, H. (2018). Dark, bright and other optical solitons to the decoupled nonlinear Schrdinger equation arising in dual-core optical fibers. *Optical and Quantum Electronics*, 50(4), 1-12.

[5] Ciancio, A., Baskonus, H.M., Sulaiman, T.A., & Bulut, H. (2018). New Structural Dynamics of Isolated Waves Via the Coupled Nonlinear Maccari's System with Complex Structure. *Indian Journal of Physics*, 92(10), 12811290.

[6] Ilhan, O.A., Sulaiman, T.A., Bulut, H., & Baskonus, H.M. (2018). On the New Wave Solutions to a Nonlinear Model Arising in Plasma Physics. *European Physical Journal Plus*, 133(27), 1-6.

[7] Yel, G., Baskonus, H.M., & Bulut, H. (2017). Novel archetypes of new coupled KonnoOono equation by using sineGordon expansion method. *Optical and Quantum Electronics*, 49(285), 1-10.

[8] Baskonus, H.M. (2017). New Complex and Hyperbolic Function Solutions to the Generalized Double Combined Sinh-Cosh-Gordon Equation. *AIP Conference Proceedings*. 1798, 1-9 (020018).

[9] Baskonus, H.M. (2016). New acoustic wave behaviors to the DaveyStewartson equation with power-law nonlinearity arising in fluid Dynamics. *Nonlinear Dynamics*, 86(1), 177183.

[10] Karaa, S., & Zhang, J. (2004). Higher order ADI method for solving unsteady convection-diffusion problems. *Journal of Computational Physics*, 198, 1-9.

[11] Tian, Z. (2011). A rational high-order compact ADI method for unsteady convection-diffusion equations. *Computer Physics Communications*, 182, 649-662.

[12] Rui, H. (2003). An alternative direction iterative method with second-order upwind scheme for convection-diffusion equations. *International Journal of Computer Mathematics*, 80(4), 527-533.

[13] Mekuria, G.T., & Rao, J.A. (2016). Adaptive finite element method for steady convection-diffusion equation. *American Journal of Computational Mathematics*, 6(3), 275-285.

[14] Li, L., Jiang, Z., & Yin, Z. (2018). Fourth-order compact finite difference method for solving two-dimensional convection-diffusion equation. *Advances in Difference Equations*, 2018, 1-24.

[15] Momani, S. (2008). A Decomposition Method for Solving Unsteady Convection-Diffusion Problems. *Turkish Journal of Mathematics*, 32, 51-60.

[16] Saqib, M., Hasnain, S., & Mashat, D.S. (2017). Computational solutions of two dimensional convection-diffusion equation using Crank Nicolson and time efficient ADI. *American Journal of Computational Mathematics*, 7(3), 208-227.

[17] Ismail, H.N.A., Elbarbary, E. M. E., & Salem, G. S. E. (2004). Restrictive Taylor's approximation for solving convection-diffusion equation. *Applied Mathematics and Computation*, 147, 355-363.

[18] Castillo, M., & Power, H. (2008). The Neumann series as a fundamental solution of the two-dimensional convection-diffusion equation with variable velocity. *Journal of Engineering Mathematics*, 62, 189-202.

[19] Noye, B.J., & Tan, H.H. (1989). Finite difference methods for solving the two-dimensional advection-diffusion equation. *International Journal for Numerical Methods in Fluids*, 9(1), 75-98.

[20] Sun, H., & Li, L. (2014). A CCD-ADI method for unsteady convection-diffusion equations. *Computer Physics Communications*, 185, 790-797.

[21] Kalita, J.C., Dalal, D.C., & Dass, A.K. (2002). A class of higher order compact schemes for the unsteady two-dimensional convection-diffusion equation with variable convection coefficients. *International Journal for Numerical Methods in Fluids*, 38, 1111-1131.

[22] Shu, C.W. (2017). Bound-preserving high order finite volume schemes for conservation laws and convection-diffusion equations. Finite Volumes for Complex Applications VIII-Methods and Theoretical Aspects, Springer Proceedings, 3-14.

[23] Ammi, M.R.S., & Jamiai, I. (2017). Finite difference and Legendre spectral method for a time-fractional diffusion-convection equation for image restoration. *Discrete & Continuous Dynamical Systems*, 11(1), 103-117.

[24] Koley, U., Risebro, N.H., Schwab, C., et al. (2017). A multilevel monte carlo finite difference method for random scalar degenerate convection-diffusion equations. *Journal of Hyperbolic Differential Equations*, 14(3), 415-454.

[25] Keskin, Y. (2010). *Solving partial differential equations by the reduced differential transform method.* PhD Thesis. Selcuk University.

[26] Ziqan, A.M., Armiti, S., & Suwan, L. (2016). Solving three-dimensional volterra integral equation by the reduced differential transform method. *International Journal of Applied Mathematical Research*, 5(2), 103-106.

Hatıra Günerhan *is currently an assistant professor at Kafkas University Faculty of Education, Department of Mathematics, Kars, Turkey. Her research interests are in the areas of applied mathematics including the numerical methods for special models of differential equations and differential-algebraic equations and integral equations.*

http://orcid.org/0000-0002-7802-477X

IJOCTA
An International Journal of
Optimization and Control:
Theories & Applications
2010

RESEARCH ARTICLE

Some Hermite-Hadamard type inequalities for (P,m)-function and quasi m-convex functions

Mahir Kadakal

Department of Mathematics, Giresun University, Gure Campus, Giresun,Turkey
mahirkadakal@gmail.com

ARTICLE INFO	ABSTRACT
	In this paper, we introduce a new class of functions called as (P,m)-function and quasi-m-convex function. Some inequalities of Hadamard's type for these functions are given. Some special cases are discussed. Results represent significant refinement and improvement of the previous results. We should especially mention that the definition of (P,m)-function and quasi-m-convexity are given for the first time in the literature and moreover, the results obtained in special cases coincide with the well-known results in the literature.

1. Preliminaries

Inequalities present an attractive and active field of research. In recent years, various inequalities for convex functions and their variant forms are being developed using innovative techniques. For some inequalities, generalizations and applications concerning convexity see [1,2]. Recently, in the literature there are so many papers about P-function, quasi-convex and m-convex functions. Many papers have been written by a number of mathematicians concerning inequalities for P-function, quasi-convex functions and m-convex functions see for instance the recent papers [3–8] and the references within these papers.

Definition 1. *A function $\omega : I \subseteq \mathbb{R} \to \mathbb{R}$ is said to be convex if the inequality*

$$\omega (t\lambda + (1 - t)\mu) \leq t\omega (x) + (1 - t)\omega(y)$$

is valid for all $\lambda, \mu \in I$ and $t \in [0,1]$. If this inequality reverses, then ω is said to be concave

on interval $I \neq \emptyset$. *This definition is well known in the literature. Denote by $C(I)$ the set of the convex functions on the interval I.*

Definition 2. *Let $\omega : I \subseteq \mathbb{R} \to \mathbb{R}$ be a convex function defined on the interval I of real numbers and $\lambda, \mu \in I$ with $\lambda < \mu$. The following inequality*

$$\omega \left(\frac{\lambda + \mu}{2} \right) \leq \frac{1}{\mu - \lambda} \int_\lambda^\mu \omega(x)\, dx \leq \frac{\omega (\lambda) + \omega (\mu)}{2} \tag{1}$$

holds.

The inequality (1) is known as Hermite-Hadamard (H-H) integral inequality for convex functions in the literature.

Some refinements of the H-H inequality on convex functions have been extensively studied by researchers (*e.g.*, [1,9]) and the researchers obtained a new refinement of the H-H inequality for convex functions.

Definition 3. *A function $\omega : I \subseteq \mathbb{R} \to \mathbb{R}$ is said to be quasi-convex if the inequality*

$$\omega\left(t\lambda + (1-t)\mu\right) \le \max\left\{\omega\left(\lambda\right), \omega(\mu)\right\}$$

holds for all $\lambda, \mu \in I$ and $t \in [0,1]$. Denote by $QC(I)$ the set of the quasi-convex functions on the interval I.

Definition 4. *A nonnegative function $\omega : I \subseteq \mathbb{R} \to \mathbb{R}$ is called P-function if the inequality*

$$\omega\left(t\lambda + (1-t)\,\mu\right) \le \omega\left(\lambda\right) + \omega\left(\mu\right)$$

holds for all $\lambda, \mu \in I$ and $t \in (0,1)$.

We will denote by $P(I)$ the set of P-function on the interval I. Note that $P(I)$ contain all nonnegative quasi-convex and convex functions.

In [10], Dragomir et al. proved the following inequality of Hadamard type for class of P-functions.

Theorem 1. *Let $\omega \in P(I)$, $\lambda, \mu \in I$ with $\lambda < \mu$ and $\omega \in L[\lambda, \mu]$. Then*

$$\omega\left(\frac{\lambda + \mu}{2}\right) \le \frac{2}{\mu - \lambda}\int_{\lambda}^{\mu}\omega(x)\,dx \le 2\left[\omega\left(\lambda\right) + \omega\left(\mu\right)\right].$$
$$(2)$$

Definition 5. *[11] The function $\omega : [0, \tau] \to \mathbb{R}$, $\tau > 0$, is said to be an m-convex function, where $m \in [0,1]$; if we have*

$$\omega\left(t\lambda + m\left(1-t\right)\mu\right) \le t\omega\left(\lambda\right) + m\left(1-t\right)\omega\left(\mu\right)$$

for all $\lambda, \mu \in [0,\tau]$ and $t \in [0,1]$. We say that f is an m-concave function if $(-\omega)$ is m-convex. Denote by $K_m(\tau)$ the set of the m-convex functions on $[0,\tau]$ for which $\omega(0) \le 0$.

Obviously, this definition recaptures the concept of standard convex functions on $[0, \tau]$ for $m = 1$; and the concept star-shaped functions for $m = 0$.

2. Some new definitions and their properties

In this section, we will define the (P, m) and quasi-m-convex function supply several properties of this kind of functions.

Definition 6. *A function $\omega : [0, \tau] \to \mathbb{R}$ is called quasi-m-convex if the inequality*

$$\omega\left(t\lambda + m\left(1-t\right)\mu\right) \le \max\left\{\omega\left(\lambda\right), m\omega(\mu)\right\}$$

holds for all $\lambda, \mu \in [0, \tau]$, $m \in [0,1]$ and $t \in [0,1]$. We will denote by $Q_m C(\tau)$ the set of quasi-m-convex function on the interval $[0, \tau]$.

It is clear that quasi-convexity obtained in quasi-m-convexity for $m = 1$.

Definition 7. *A nonnegative function $\omega : [0, \tau] \to \mathbb{R}$ is called (P, m)-function if the inequality*

$$\omega\left(t\lambda + m\left(1-t\right)\mu\right) \le \omega\left(\lambda\right) + m\omega\left(\mu\right)$$

holds for all $\lambda, \mu \in [0, \tau]$, $m \in [0,1]$ and $t \in (0,1)$. We will denote by $P_m(\tau)$ the set of (P, m)-function on the interval $[0, \tau]$.

It is clear that P-function obtained in (P, m)-function for $m = 1$. Note also that $P_m(\tau)$ contain all nonnegative m-convex and quasi-m-convex functions. Since

$$\omega\left(t\lambda + m\left(1-t\right)\mu\right) \le t\omega\left(\lambda\right) + m\left(1-t\right)\omega\left(\mu\right)$$
$$\le \omega\left(\lambda\right) + m\omega\left(\mu\right),$$
$$\omega\left(t\lambda + m\left(1-t\right)\mu\right) \le \max\left\{\omega\left(\lambda\right), m\omega\left(\mu\right)\right\}$$
$$\le \omega\left(\lambda\right) + m\omega\left(\mu\right).$$

Theorem 2. *Let $m \in [0,1]$ and $\omega : [0,\tau] \to \mathbb{R}$. If ω is a quasi-m-convex function, then, for $c \in \mathbb{R}$ $(c \ge 0)$, $c\omega$ is a quasi-m-convex function.*

Proof. For $c \in \mathbb{R}$ $(c \ge 0)$,

$$(c\omega)\left(t\lambda + m\left(1-t\right)\mu\right)$$
$$\le c.\max\left\{\omega\left(\lambda\right), m\omega\left(\mu\right)\right\}$$
$$= \max\left\{(c\omega)\left(\lambda\right), m\left(c\omega\right)\left(\mu\right)\right\}.$$

$\square$

Remark 1. *If ω and φ are quasi-m-convex functions, then it is not necessary that the function $\omega + \varphi$ is a quasi-m-convex function.*

Example 1. *Let $\omega, \varphi : [0, \tau] \to \mathbb{R}$, $\omega\left(u\right) = u$, $\varphi\left(u\right) = 1$. Then ω and φ are quasi-m-convex functions. Now, if we choose $\lambda, \mu \in [0, \tau]$, $m \in [0,1]$ as numbers which satisfy the conditions $m\mu \ge \lambda$ and $m\left(\mu + 1\right) \le \lambda + 1$. Then, $(\omega + \varphi)\left(u\right) = u + 1$. Moreover, $\omega + \varphi$ is not quasi-m-convex function. Indeed, we can write following equality: for all $t \in [0,1]$,*

$$\begin{aligned}
&(\omega + \varphi)(t\lambda + m(1-t)\mu) \\
&= \quad t\lambda + m(1-t)\mu + 1 \\
&= \quad t(\lambda + 1) + (1-t)(m\mu + 1).
\end{aligned}$$

Since $m\mu \geq \lambda$,

$$\begin{aligned}
&(\omega + \varphi)(t\lambda + m(1-t)\mu) \\
&= \quad t(\lambda + 1) + (1-t)(m\mu + 1) \\
&\geq \quad t(\lambda + 1) + (1-t)(\lambda + 1) \\
&= \quad \lambda + 1,
\end{aligned}$$

and since $m(\mu + 1) \leq \lambda + 1$,

$$\begin{aligned}
&(\omega + \varphi)(t\lambda + m(1-t)\mu) \\
&= \quad t(\lambda + 1) + (1-t)(m\mu + 1) \\
&\geq \quad tm(\mu + 1) + (1-t)(m\mu + 1).
\end{aligned}$$

Since $m \leq 1$,

$$\begin{aligned}
&(\omega + \varphi)(t\lambda + m(1-t)\mu) \\
&\geq \quad tm(\mu + 1) + (1-t)(m\mu + m) \\
&= \quad tm(\mu + 1) + m(1-t)(\mu + 1) \\
&= \quad m(\mu + 1).
\end{aligned}$$

So,

$$\begin{aligned}
&(\omega + \varphi)(t\lambda + m(1-t)\mu) \\
&\geq \quad \max\{\lambda + 1, m(\mu + 1)\} \\
&= \quad \max\{(\omega + \varphi)(\lambda), m(\omega + \varphi)(\mu)\}.
\end{aligned}$$

Theorem 3. *Let $m \in [0,1]$ and $\omega_\alpha : [0,\tau] \to \mathbb{R}$ be an arbitrary family of quasi-m-convex functions and let $\omega(x) = \sup_\alpha \omega_\alpha(x)$ for all $x \in [0,\tau]$. If*

$$J = \{u \in [0,\tau] : \omega(u) < \infty\}$$

is nonempty, then J is an interval and ω is a quasi-m-convex functions on J.

Proof. Let $t \in [0,1]$ and $\lambda, \mu \in J$ be arbitrary. Then

$$\begin{aligned}
&\omega(t\lambda + m(1-t)\mu) \\
&= \quad \sup_\alpha \omega_\alpha(t\lambda + m(1-t)\mu) \\
&\leq \quad \sup_\alpha [\max\{\omega_\alpha(\lambda), m\omega_\alpha(\mu)\}] \\
&\leq \quad \max\left\{\sup_\alpha \omega_\alpha(\lambda), m\sup_\alpha \omega_\alpha(\mu)\right\} \\
&\leq \quad \max\{\omega(\lambda), m\omega(\mu)\} < \infty
\end{aligned}$$

This shows that J is an interval since it contains every point between any two of its points and ω is a quasi-m-convex functions on J. $\qquad\square$

Theorem 4. *Let $m \in [0,1]$ and $\omega : [0,\tau] \to \mathbb{R}$ be a m-convex function. If φ is a quasi-m-convex functions and increasing on $[0,\tau]$, then the function $\varphi \circ \omega$ is a quasi-m-convex function.*

Proof. For $\lambda, \mu \in [0,\tau]$ and $t \in [0,1]$,

$$\begin{aligned}
&(\varphi \circ \omega)(t\lambda + m(1-t)\mu) \\
&= \quad \varphi(\omega(t\lambda + m(1-t)\mu)) \\
&\leq \quad \varphi(\omega(\lambda) + m(1-t)\omega(\mu)) \\
&\leq \quad \max\{(\varphi \circ \omega)(\lambda), m(\varphi \circ \omega)(\mu)\}.
\end{aligned}$$

$\qquad\square$

Theorem 5. *Let $m \in [0,1]$ and $\omega, \varphi : [0,\tau] \to \mathbb{R}$. If ω is a quasi-m-convex and non-negative function, φ is a (P,m)-function. Then, $\omega + \varphi$ is a (P,m)-function.*

Proof. For $\lambda, \mu \in [0,\tau]$ and $t \in [0,1]$,

$$\begin{aligned}
&(\omega + \varphi)(t\lambda + m(1-t)\mu) \\
&= \quad \omega(t\lambda + m(1-t)\mu) + \varphi(t\lambda + m(1-t)\mu) \\
&\leq \quad \max\{\omega(\lambda), m\omega(\mu)\} + \varphi(\lambda) + m\varphi(\mu) \\
&\leq \quad \omega(\lambda) + m\omega(\mu) + \varphi(\lambda) + m\varphi(\mu) \\
&= \quad (\omega + \varphi)(\lambda) + m(\omega + \varphi)(\mu).
\end{aligned}$$

$\qquad\square$

Theorem 6. *Let $m \in [0,1]$ and $\omega, \varphi : [0,\tau] \to \mathbb{R}$. If ω and φ are (P,m)-functions, then*

(1) $\omega + \varphi$ is a (P,m)-function ,
(2) For $c \in \mathbb{R}$ $(c \geq 0)$, $c\omega$ is a (P,m)-function .

Proof. i) For $\lambda, \mu \in [0,\tau]$ and $t \in [0,1]$,

$$\begin{aligned}
&(\omega + \varphi)(t\lambda + m(1-t)\mu) \\
&= \quad \omega(t\lambda + m(1-t)\mu) + \varphi(t\lambda + m(1-t)\mu) \\
&\leq \quad \omega(\lambda) + m\omega(\mu) + \varphi(\lambda) + m\varphi(\mu) \\
&\leq \quad (\omega + \varphi)(\lambda) + m(\omega + \varphi)(\mu).
\end{aligned}$$

ii) For $c \in \mathbb{R}$ $(c \geq 0)$,

$$
\begin{aligned}
(c\omega)(t\lambda + m(1-t)\mu) &\leq c[\omega(\lambda) + m\omega(\mu)] \\
&= (c\omega)(\lambda) + m(c\omega)(\mu).
\end{aligned}
$$

$\square$

Theorem 7. *Let $m \in [0,1]$ and $\omega_\alpha : [0,\tau] \to \mathbb{R}$ be an arbitrary family of (P,m)-functions and let $\omega(x) = \sup_\alpha \omega_\alpha(x)$ for all $x \in [0,\tau]$. If*

$$
J = \{u \in [0,\tau] : \omega(u) < \infty\}
$$

is nonempty, then J is an interval and ω is a (P,m)-functions on J.

Proof. Let $t \in [0,1]$ and $\lambda, \mu \in J$ be arbitrary. Then

$$
\begin{aligned}
&\omega(t\lambda + m(1-t)\mu) \\
&= \sup_\alpha \omega_\alpha(t\lambda + m(1-t)\mu) \\
&\leq \sup_\alpha [\omega_\alpha(\lambda) + m\omega_\alpha(\mu)] \\
&\leq \sup_\alpha \omega_\alpha(\lambda) + m\sup_\alpha \omega_\alpha(\mu) \\
&= \omega(\lambda) + m\omega(\mu) < \infty.
\end{aligned}
$$

This shows simultaneously that J is an interval since it contains every point between any two of its points and ω *is a (P,m)-function on the interval J.* $\square$

Theorem 8. *Let $m \in [0,1]$ and $\omega : [0,\tau] \to \mathbb{R}$ be an m-convex function. If the function φ is a (P,m)-function and increasing, then the function $\varphi \circ \omega$ is a (P,m)-function.*

Proof. For $\lambda, \mu \in I$ and $t \in [0,1]$,

$$
\begin{aligned}
&(\varphi \circ \omega)(t\lambda + m(1-t)\mu) \\
&= \varphi(\omega(t\lambda + m(1-t)\mu)) \\
&\leq \varphi(t\omega(\lambda) + m(1-t)\omega(\mu)) \\
&\leq \varphi(\omega(\lambda)) + m\varphi(\omega(\mu)) \\
&= (\varphi \circ \omega)(\lambda) + m(\varphi \circ \omega)(\mu).
\end{aligned}
$$

$\square$

3. Hermite-Hadamard integral inequality for (P,m)-function and quasi-m-convex functions

The main purpose of this paper is to develop concepts of the (P,m)-function and quasi-m-convex functions and to obtain some inequalities of H-H type for these classes of functions.

Theorem 9. *Let $m \in [0,1]$ and $\omega : [0,\tau] \to \mathbb{R}$ be a (P,m)-function. If $0 \leq \lambda < \mu < \tau$ and $\omega \in L[\lambda, \mu]$, then the following inequalities holds:*

$$
\begin{aligned}
&\frac{1}{m\mu - \lambda} \int_\lambda^{m\mu} \omega(x)dx \\
&\leq \min\{\omega(\lambda) + m\omega(\mu), \omega(\mu) + m\omega(\lambda)\}.
\end{aligned}
$$

Proof. By using (P,m)-function property of ω and changing variable as $u = t\lambda + m(1-t)\mu$

$$
\begin{aligned}
&\int_0^1 \omega(t\lambda + m(1-t)\mu)dt \\
&= \frac{1}{m\mu - \lambda} \int_\lambda^{m\mu} \omega(u)du \\
&\leq \int_0^1 [\omega(\lambda) + m\omega(\mu)]dt \\
&= \omega(\lambda) + m\omega(\mu)
\end{aligned}
$$

and similarly for $z = t\mu + m(1-t)\lambda$, then

$$
\begin{aligned}
&\int_0^1 \omega(t\mu + m(1-t)\lambda)dt \\
&= \frac{1}{m\mu - \lambda} \int_\lambda^{m\mu} \omega(z)dz \\
&\leq \int_0^1 [\omega(\mu) + m\omega(\lambda)]dt \\
&= \omega(\mu) + m\omega(\lambda).
\end{aligned}
$$

So, we have

$$
\begin{aligned}
&\frac{1}{m\mu - \lambda} \int_\lambda^{m\mu} \omega(x)dx \\
&\leq \min\{\omega(\lambda) + m\omega(\mu), \omega(\mu) + m\omega(\lambda)\}.
\end{aligned}
$$

$\square$

Remark 2. *Under the conditions of Theorem 9, if $m = 1$ then, the following inequality holds:*

$$
\frac{1}{\mu - \lambda} \int_\lambda^\mu \omega(x)dx \leq \omega(\lambda) + \omega(\mu)
$$

The above inequality is the right hand side of the inequality 2.

Theorem 10. *Let $m \in (0,1]$ and $\omega : [0,\tau] \to \mathbb{R}$ be an (P,m)-function. If $0 \leq \lambda < \mu < \tau$ and $\omega \in L[\lambda, \tau]$, then the following inequalities holds:*

$$
\omega\left(\frac{\lambda + m\mu}{2}\right) \leq \frac{2}{m\mu - \lambda} \int_\lambda^{m\mu} \omega(x)dx.
$$

Proof. By the (P, m)-function property of ω, we have

$$
\begin{aligned}
&\omega\left(\frac{\lambda + m\mu}{2}\right) \\
&= \omega\left(\frac{[t\lambda + m(1-t)\mu] + [(1-t)\lambda + mt\mu]}{2}\right) \\
&= \omega\left(\frac{1}{2}[t\lambda + m(1-t)\mu] + \frac{1}{2}[(1-t)\lambda + mt\mu]\right) \\
&\leq \omega(t\lambda + m(1-t)\mu) + \omega((1-t)\lambda + mt\mu).
\end{aligned}
$$

Now, if we take integral in the last inequality on $t \in [0, 1]$ and choose $x = t\lambda + m(1-t)\mu$ and $y = (1-t)\lambda + mt\mu$, we deduce

$$
\begin{aligned}
\omega\left(\frac{\lambda + m\mu}{2}\right) &\leq \frac{1}{m\mu - \lambda}\int_\lambda^{m\mu} \omega(x)\, dx \\
&+ \frac{1}{m\mu - \lambda}\int_\lambda^{m\mu} \omega(y)\, dy.
\end{aligned}
$$
$\square$

Remark 3. *Under the conditions of Theorem 10, if $m = 1$, then, the following inequality holds:*

$$
\omega\left(\frac{\lambda + \mu}{2}\right) \leq \frac{2}{\mu - \lambda}\int_\lambda^\mu \omega(x)\, dx
$$

This inequality is the left hand side of the inequality 2.

Theorem 11. *Let $m \in (0, 1]$ and $\omega : [0, \tau] \to \mathbb{R}$ be a (P, m)-function. If $0 \leq \lambda < \mu < \tau$ and $\omega \in L[\lambda, \mu]$, then the following inequalities holds:*

$$
\begin{aligned}
\omega\left(\frac{\lambda + \mu}{2}\right) &\leq \frac{1}{\mu - \lambda}\int_\lambda^\mu \left[\omega(x) + m\omega\left(\frac{x}{m}\right)\right] dx \\
&\leq \min\{I_1, I_2\}. \qquad (3)
\end{aligned}
$$

where

$$
I_1 = \omega(\lambda) + m\omega\left(\frac{\mu}{m}\right) + m\omega\left(\frac{\lambda}{m}\right) + m^2\omega\left(\frac{\mu}{m^2}\right)
$$

and

$$
I_2 = \omega(\mu) + m\omega\left(\frac{\lambda}{m}\right) + m\omega\left(\frac{\mu}{m}\right) + m^2\omega\left(\frac{\lambda}{m^2}\right).
$$

Proof. Using the (P, m)-function property of ω, we have

$$
\omega\left(\frac{x + y}{2}\right) \leq \omega(x) + m\omega\left(\frac{y}{m}\right)
$$

for all $x, y \in [0, \tau]$. If we take $x = t\lambda + (1-t)\mu$, $y = (1-t)\lambda + t\mu$, we get

$$
\begin{aligned}
&\omega\left(\frac{\lambda + \mu}{2}\right) \\
&\leq \omega(t\lambda + (1-t)\mu) + m\omega\left((1-t)\frac{\lambda}{m} + t\frac{\mu}{m}\right)
\end{aligned}
$$

for all $t \in [0, 1]$. Here, if we take integral over $t \in [0, 1]$, we get

$$
\begin{aligned}
\omega\left(\frac{\lambda + \mu}{2}\right) &\leq \int_0^1 \omega(t\lambda + (1-t)\mu)\, dt \qquad (4) \\
&+ m\int_0^1 \omega\left((1-t)\frac{\lambda}{m} + t\frac{\mu}{m}\right) dt.
\end{aligned}
$$

Taking into account that

$$
\int_0^1 \omega(t\lambda + (1-t)\mu)\, dt = \frac{1}{\mu - \lambda}\int_\lambda^\mu \omega(x)dx,
$$

and

$$
\begin{aligned}
&\int_0^1 \omega\left((1-t)\frac{\lambda}{m} + t\frac{\mu}{m}\right) dt \\
&= \frac{m}{\mu - \lambda}\int_{\frac{\lambda}{m}}^{\frac{\mu}{m}} \omega(x)dx \\
&= \frac{1}{\mu - \lambda}\int_\lambda^\mu \omega\left(\frac{x}{m}\right) dx,
\end{aligned}
$$

we deduce from (4) the first part of (3). That is

$$
\omega\left(\frac{\lambda + \mu}{2}\right) \leq \frac{1}{\mu - \lambda}\int_\lambda^\mu \left[\omega(x) + m\omega\left(\frac{x}{m}\right)\right] dx.
$$

By the (P, m)-function property of ω we also have

$$
\begin{aligned}
&\omega(t\lambda + (1-t)\mu) + m\omega\left((1-t)\frac{\lambda}{m} + t\frac{\mu}{m}\right) \quad (5) \\
&\leq \omega(\lambda) + m\omega\left(\frac{\mu}{m}\right) + m\omega\left(\frac{\lambda}{m}\right) + m^2\omega\left(\frac{\mu}{m^2}\right).
\end{aligned}
$$

for all $t \in [0, 1]$. Integrating the last equality (5) over t on $[0, 1]$, we deduce

$$
\begin{aligned}
&\frac{1}{\mu - \lambda}\int_\lambda^\mu \left[\omega(x) + m\omega\left(\frac{x}{m}\right)\right] dx \qquad (6) \\
&\leq \omega(\lambda) + m\omega\left(\frac{\mu}{m}\right) + m\omega\left(\frac{\lambda}{m}\right) + m^2\omega\left(\frac{\mu}{m^2}\right).
\end{aligned}
$$

By a similar argument, if we take

$$\omega\left(t\mu + (1-t)\lambda\right) + m\omega\left(t\frac{\lambda}{m} + (1-t)\frac{\mu}{m}\right) \quad (7)$$
$$\leq \quad \omega\left(\mu\right) + m\omega\left(\frac{\mu}{m}\right) + m\omega\left(\frac{\lambda}{m}\right) + m^2\omega\left(\frac{\lambda}{m^2}\right),$$

we have

$$\frac{1}{\mu-\lambda}\int_\lambda^\mu \left[\omega\left(x\right) + m\omega\left(\frac{x}{m}\right)\right] dx \quad (8)$$
$$\leq \quad \omega\left(\mu\right) + \omega\left(\frac{\lambda}{m}\right) + \omega\left(\frac{\mu}{m}\right) + m^2\omega\left(\frac{\lambda}{m^2}\right).$$

From (6) and (8), we obtain

$$\frac{1}{\mu-\lambda}\int_\lambda^\mu \left[\omega\left(x\right) + m\omega\left(\frac{x}{m}\right)\right] dx \leq \min\{I_1, I_2\}.$$
$$\square$$

Remark 4. *For $m = 1$, (3) exactly becomes the inequality 2 (the Hermite-Hadamard integral inequality for P-functions given in [10]).*

Theorem 12. *Let $m \in (0,1]$ and $\omega : [0, \tau] \to \mathbb{R}$ be a (P,m)-function. If $0 \leq \lambda < \mu < \tau$ and $\omega \in L[\lambda, \mu]$, then the following inequalities holds:*

$$\frac{1}{m\mu-\lambda}\int_\lambda^{m\mu} \omega(x)dx + \frac{1}{\mu-m\lambda}\int_{\lambda m}^\mu \omega(x)dx$$
$$\leq \quad (m+1)\left[\omega\left(\lambda\right) + \omega\left(\mu\right)\right] \quad (9)$$

Proof. By the (P,m)-function property of ω we have that

$$\omega\left(t\lambda + m\left(1-t\right)\mu\right) \leq \omega\left(\lambda\right) + m\omega\left(\mu\right),$$
$$\omega\left(t\mu + m(1-t)\lambda\right) \leq \omega\left(\mu\right) + m\omega\left(\lambda\right)$$

for all $t \in [0,1]$ and $\lambda, \mu \in [0, \tau]$. By adding the above inequalities we get

$$\omega\left(t\lambda + m\left(1-t\right)\mu\right) + \omega\left(t\mu + m(1-t)\lambda\right)$$
$$\leq \quad (m+1)\left[\omega\left(\lambda\right) + \omega\left(\mu\right)\right].$$

Integrating over $t \in [0,1]$, we obtain

$$\int_0^1 \omega(t\lambda + m\left(1-t\right)\mu)\, dt \quad (10)$$
$$+ \int_0^1 \omega(t\mu + m(1-t)\lambda)\, dt$$
$$\leq \quad (m+1)\left[\omega\left(\lambda\right) + \omega\left(\mu\right)\right].$$

As it is easy to see that

$$\int_0^1 \omega(t\lambda + m\left(1-t\right)\mu)\, dt$$
$$= \quad \frac{1}{m\mu-\lambda}\int_\lambda^{m\mu} \omega(x)dx$$

and

$$\int_0^1 \omega(t\mu + m(1-t)\lambda)\, dt$$
$$= \quad \frac{1}{\mu-m\lambda}\int_{m\lambda}^\mu \omega(x)dx,$$

from (10) we deduce the desired result, namely, the inequality (9). $\square$

Remark 5. *For $m = 1$, (9) exactly becomes the right hand side of the inequality 2.*

References

[1] Dragomir, S.S. and Pearce, C.E.M. (2000). Selected topics on Hermite-Hadamard inequalities and applications. *RGMIA Monographs*, Victoria University.

[2] Maden, S. Kadakal, H. Kadakal, M. and İşcan, İ. (2017). Some new integral inequalities for n-times differentiable convex functions. *Journal of Nonlinear Sciences and Applications*, 10 12, 6141-6148.

[3] Barani, A. Barani, S. and Dragomir, S.S. Hermite-Hadamard Inequalities for Functions Whose Second Derivatives Absolute Values are P-Convex, *http://www.ajmaa.org/RGMIA/papers /v14/v14a73.pdf*.

[4] Maden, S. Kadakal, H. Kadakal, M. and İşcan, İ. (2017). Some new integral inequalities for n-times differentiable P-functions. *AIP Conference Proceedings* 1833, 020015, doi: 10.1063/1.4981663.

[5] Kadakal, H. İşcan, İ. and Kadakal M. (2017). New type integral inequalities for p-quasi convex functions. *Ordu Univ. J. Sci. Tech.*,7(1), 124-130.

[6] Pearce, C.E.M. and Rubinov, A.M. (1999). P-functions, quasi-convex functions and Hadamard-type inequalities. *Journal of Mathematical Analysis and Applications*, 240, 92-104.

[7] Hussain, S. and Qaisar, S. (2013). New integral inequalities of the type of Hermite-Hadamard through quasi convexity. *Punjab University, Journal of Mathematics*, 45, 33-38.

[8] Lara, T., Rosales, E. and Sánchez, J.L. (2015). New properties of m-convex functions. *International Journal of Mathematical Analysis,* 9(15), 735-742.

[9] Hadamard, J. (1893). Etude sur les proprietes des fonctions entieres en particulier d'une fonction consideree par Riemann. *J. Math. Pures Appl.,* 58, 171-215.

[10] Dragomir, S.S., Pečarić, J. and Persson, L.E. (1995). Some inequalities of Hadamard type. *Soochow Journal of Mathematics,* 21(3), 335-341.

[11] Dragomir, S.S. (2002). On some new inequalities of Hermite-Hadamard type for m-convex functions. *Tamkang Journal of Mathematics,* 33(1), 45-55.

Mahir Kadakal *is a Professor at the Department of Mathematics, Art and Science Faculty, Giresun University. He received his B. Sc. (1993) degree from Department of Mathematics Teaching, Faculty of Education, Ondokuz Mayıs University and M. Sc. (1996) degree and Ph.D (2000) from Ondokuz Mayıs University, Turkey. He has many research papers about the Theory of inequalities, Integral equations and Transforms, Boundary Value Problems.*

http://orcid.org/0000-0002-0240-918X

RESEARCH ARTICLE

Modeling the impact of temperature on fractional order dengue model with vertical transmission

Ozlem Defterli[*]

Department of Mathematics, Faculty of Arts and Sciences, Cankaya University, 06790 Ankara, Turkey
defterli@cankaya.edu.tr

ARTICLE INFO	ABSTRACT
	A dengue epidemic model with fractional order derivative is formulated to analyze the effect of temperature on the spread of the vector-host transmitted dengue disease. The model is composed of a system of fractional order differential equations formulated within Caputo fractional operator. The stability of the equilibrium points of the considered dengue model is studied. The corresponding basic reproduction number R_0^α is derived and it is proved that if $R_0^\alpha < 1$, the disease-free equilibrium (DFE) is locally asymptotically stable. $L1$ method is applied to solve the dengue model numerically. Finally, numerical simulations are also presented to illustrate the analytical results showing the influence of the temperature on the dynamics of the vector-host interaction in dengue epidemics.

1. Introduction

A deeper understanding of mathematical models is essential to represent and to reliably control the transmission of the (epidemic/pandemic)diseases. The vector-borne diseases becomes an extensive threat with a significant affect on human and animal health. Distribution of vector-borne diseases is determined by complex demographic, environmental and social factors. Vector-borne diseases are responsible from more than 700 000 deaths annually as taking a part of 17% of all infectious diseases. Dengue fever, as a severe, flu-like illness influences infants, young children and adults, is mainly faced in urban and semi-urban areas of the countries in tropical and sub-tropical climates [1,2]. Dengue fever disease has no concrete treatment but early detection and efficient medical treatment reduces the death rates below 1%.

Dengue virus is carried from vector-host-vector mainly by the bites of infected female mosquitoes of the type Aedes aegypti. After virus incubation for 410 days, an infected mosquito is able to transmit the virus for the rest of its life [2].

All mosquito species go through four distinct phases during their life cycle: Egg, Larva (plural: larvae), Pupa (plural: pupae), Adult. The first three phases take place in water, but the adult is an active flying insect (see Figure 1). Only the female mosquitoes bite and they feed on the blood of humans or other animals.
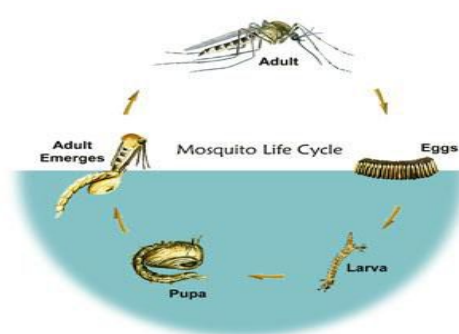

Figure 1. Mosquito Life Cycle [3].

In order to keep the dengue infection under control some adequate and powerful mathematical (compartmental) models and analysis have been

---

*Corresponding Author

suggested (see [4–15]). Abdelrazec et al. [15] developed a model for the dynamic study of transmission of dengue fever by means of a nonlinear rate of recovery to analyze the spread and control of the disease. A mathematical model for dengue is constructed in [8] to determine the chain of two epidemic diseases with different human populations. The reproduction number R_0^α as a threshold quantity of the epidemics is explained by means of the stability analysis. Their model shown that the ecological administration alone as a the vector control is not enough; it may postpone the spread of epidemics. The usage of a vaccine may control simultaneously against some serotypes. In Andraud et al. [16], deterministic models of dengue transmission are surveyed in terms of the assumptions for parameters, threshold values and control measures. The effect of seasonal variations in temperature and some other climate factors on the transmission dynamics of dengue diseases are epidemiologically discusses in several experimental research (e.g. see [17] the references therein) and mathematically analyzed in the recent studies [18–21]. The mathematical analysis in these references are based on the compartmental integer-order epidemic models that contain a system of ODE's. However, in general, integer-order systems are memoryless ([4–8, 10, 11, 15–20]).

Fractional calculus is the study of an extended form of integrals and derivatives in fractional orders. The most vital aspect of fractional derivatives is that, the models based on these operators hold memory which provides an important advantage for a well understanding of the behaviour of the entomological factors and so the dynamics of the epidemic diseases. Sardar et al. [12] investigated a compartmental dengue transmission model having the memory, in which the memory incorporated in the model exhibiting an arbitrary order differential operator. A threshold quantity R_0^α is derived, having the similarity with elementary reproduction number and determined that although the value of R_0^α is less than one, the disease-free equilibrium would not be constantly stable, and the system reveals a Hopf-type bifurcation. The SIR model of the fractional order differential equation of the dengue fever is investigated in [22], and a fractional order SEIR model with vertical transmission in a nonconstant population is studied by [23]. In [24] and [25], three definitions of fractional operators are carried out for MSEIR models of some other type of epidemics as varicella disease validated by an outbreak data. Moreover, in the studies of [26–29], dengue fever epidemics is modeled within real

data by the help of fractional operators of some types that varies by the definition of their kernel. Our goal in this study is to investigate the fractional order dynamical model of the dengue fever with temperature effect centered at the distribution of the human population into three categories (susceptible, infected, and resistant humans), whereas the population of the Aedes aegypti pre-adult female mosquitos (eggs and larvae) is distributed in two parts (susceptible and infected) and adult mosquitos population is divided in three parts (susceptible, infected but not infectious, and infectious) to understand the dynamics of dengue disease in a more accurate and realistic way.

The manuscript is organized as follows: after the related literature info in Introduction part in Section 1, the generalized non-integer order mathematical model of the temperature effect in the vector-host (vertical) transmission dynamics of dengue fever epidemics is introduced in Section 2 with a preliminary info about its parameter list and initial conditions. The disease-free equilibrium points of the system are newly obtained and the local asymptotic stability conditions are derived correspondingly which results the basic reproduction number of the system. Section 3 is dedicated to the numerical solution of the discretized version of generalized Caputo-based dengue model. Followingly, the numerical simulations are performed for the model (for different temperature values and fractional orders) in order to analyze the temperature effect on the dengue transmission dynamics by using the real data of the FongShan district, Kaohsiung, Taiwan. The discussions of the simulations are provided in Section 3 and the concluding remarks are given in Section 4.

2. Preliminaries and the proposed generalized dengue model

In this part, we present fractional operators where the Caputo fractional derivative is mainly considered. For a function $x(t)$ defined on a time interval $[0, T]$, the Caputo derivative and integral of $x(t)$ are denoted by ${}^C D^\alpha x(t)$ and ${}^C I^\alpha x(t)$, respectively, and defined as [30]

$$ {}^C D^\alpha x(t) := \frac{1}{\Gamma(1-\alpha)} \int_0^t (t-\xi)^{-\alpha} \dot{x}(\xi) d\xi, \quad (1) $$

$$ {}^C I^\alpha x(t) := \frac{1}{\Gamma(\alpha)} \int_0^t (t-\xi)^{\alpha-1} x(\xi) d\xi, \quad (2) $$

where $0 < \alpha < 1$ represents the order of the fractional operator.

We will examine a mathematical model of the dengue fever having arbitrary order, the parameter values used in this mathematical model are estimated based on the demographic data of Fong-Shan district, Kaohsiung in southern Taiwan ([20], [31]). Kaohshiung is the second largest cosmopolitan city of Taiwan. The data showed that Kaohsiung was at high risk for dengue fever from 1998 to 2010 and in total 2415 number of cases of dengue fever were reported. According to the WHO Dengue situation update by July 2019, approx. 29 300 cases reported in 2019 only in the Western Pasific Region ([1], [2]). In this study, a set of weekly meteorological data has been used which belongs to Kaohsiung district that was recorded by 11 supervising observatory locations of the Taiwan Environmental Protection Agency [3] in 2011, further weekly maximum, minimum and mean temperatures between the years of 2001 and 2010 are incorporated [20].

2.1. Fractional order vector-host dengue model

This study is based upon the model of vector-host transmission dynamics proposed in [32] to represent the transmission patterns of the dengue fever. The population has been separated into three main parts as host (human), vector (pre-adult female mosquito), and vector (adult female mosquito) population. There are two compartments of Aedes aegypti pre-adult female mosquito population (effective eggs and larvae) namely susceptible E_s and infected E_i. The three compartments of vector (adult) population are stated accordingly as: $M_s, M_e,$ and M_i, which are the values evaluated at time t of susceptible, infected (but not being infectious) and infectious female mosquitoes. Also, host (human population) is divided into three compartments as: $H_s, H_i,$ and H_r, these are the numbers at the time t of the susceptible, infected/infectious and recovered/immune human populations, respectively. This way dengue model defined by the following system [32] and correspondingly the model parameters are listed in Table 1:

$$\begin{cases} {}^{C}D^{\alpha}E_s = e_v^{\alpha}(1 - p(\frac{M_i}{M_s+M_e+M_i})) - \eta^{\alpha}E_s, \\ {}^{C}D^{\alpha}E_i = e_v^{\alpha}p(\frac{M_i}{M_s+M_e+M_i}) - \eta^{\alpha}E_i, \\ {}^{C}D^{\alpha}M_s = \eta^{\alpha}E_s - b^{\alpha}\frac{H_i}{N_h}M_s - \delta^{\alpha}M_s, \\ {}^{C}D^{\alpha}M_e = b^{\alpha}\frac{H_i}{N_h}M_s - \gamma^{\alpha}M_e - \delta^{\alpha}M_e, \\ {}^{C}D^{\alpha}M_i = \gamma^{\alpha}M_e + \eta^{\alpha}E_i - \delta^{\alpha}M_i, \\ {}^{C}D^{\alpha}H_s = R_{hb}^{\alpha}N_h - b^{\alpha}\frac{H_s}{N_h}M_i - R_{hd}^{\alpha}H_s, \\ {}^{C}D^{\alpha}H_i = b^{\alpha}\frac{H_s}{N_h}M_i - \xi^{\alpha}H_i - R_{hd}^{\alpha}H_i, \\ {}^{C}D^{\alpha}H_r = \xi^{\alpha}H_i - R_{hd}^{\alpha}H_r. \end{cases} \quad (3)$$

The initial values are $H_s(0) = 341094$ (total human population), $H_i(0) = 26$ (number of confirmed cases), and $H_r(0) = 0$ according to the collected info in December, 2010 in FongShan district (Kaohsiung, Taiwan). The transmissible biting rate was taken as 0.33 on a daily basis (meaning that one bite occurs per three days for one female mosquito). At an initial time $t = 0$, the values for pre-adult and adult vector populations are set as $E_s(0) = 0$, $E_i(0) = 0$, $M_s(0) = 341120$, $M_e(0) = 0$, and $M_i(0) = 0$, respectively. The total adult mosquito population N_m was 341120 female mosquitoes that is same value with the human population size. The detailed assumptions of the considered (vertical) transmission dengue dynamics model can be seen in [20] and the references therein.

2.2. Equilibrium Points

The system has two types of disease free equilibrium points namely trivial disease-free equilibrium (DFE) E_0^{α} and biologically realistic disease-free equilibrium (BRDFE) $\widehat{E}_1^{\alpha}$. To find equilibrium points, we will solve the following system:

$${}^{C}D^{\alpha}E_s = 0, {}^{C}D^{\alpha}E_i = 0, {}^{C}D^{\alpha}M_s = 0, {}^{C}D^{\alpha}M_e = 0,$$
$${}^{C}D^{\alpha}M_i = 0, {}^{C}D^{\alpha}H_s = 0, {}^{C}D^{\alpha}H_i = 0, {}^{C}D^{\alpha}H_r = 0.$$

By solving above system, we obtain the following equilibrium points:

$$E_0^{\alpha} = (0, 0, 0, 0, 0, \overline{H}_s, 0, 0)$$
$$\widehat{E}_1^{\alpha} = (\widehat{E}_s, 0, \widehat{M}_s, 0, 0, \widehat{H}_s, 0, 0)$$

where

$$\overline{H}_s = \frac{R_{hb}^{\alpha}N_h}{R_{hd}^{\alpha}}, \widehat{E}_s = \frac{e_v^{\alpha}}{q^{\alpha}}, \widehat{M}_s = \frac{e_v^{\alpha}}{\delta^{\alpha}}, \widehat{H}_s = \frac{R_{hb}^{\alpha}N_h}{R_{hd}^{\alpha}}.$$

2.3. Stability analysis

Theorem 1. *The biologically realistic disease-free equilibrium (BRDFE) $\widehat{E}_1^{\alpha}$ of the system in Eq. (3) is locally asymptotically stable if*

$$R_1^{\alpha} = \frac{N_m + \eta^{\alpha}e_v^{\alpha} - \eta^{\alpha}\delta^{\alpha}N_m}{N_m} < 1.$$

Proof. The Jacobian matrix of the system in Eq. (3) evaluated biologically realistic disease-free equilibrium (BRDFE) $\widehat{E}_1^{\alpha}$ is given by

Table 1. Dengue model parameters (at temperature 25°C)

Symbol	Meaning and unit	Range of values	References
p	"proportion of eggs"	0.028	[33]
η	"pre-adult mosquito maturation rate (per day)"	0.099	Estimated
b	"the biting rate (per day)"	0.33	[32]
e_v	"Oviposition rate (per day)"	6.218	Estimated
γ	"Virus incubation rate in mosquito (per day)"	0.0607	Estimated
δ	"Adult mosquito death rate (per day)"	0.0331	Estimated
ξ	"Human recovery rate (per day)"	1/7	[32]
R_{hd}	"Human death rate (per day)"	0.000016	[31]
R_{hb}	"Human birth rate(per day)"	0.00002	[31]
N_m	"Total number of mosquitoes"	341120	Assumed
N_h	"Total size of human population"	341120	[31]

$$J(\widehat{E}_1^\alpha) =$$

$$
\begin{bmatrix}
\eta^\alpha & 0 & 0 & 0 & \frac{e_v^\alpha p}{N_m} & 0 & 0 & 0 \\
0 & -\eta^\alpha & 0 & 0 & \frac{e_v^\alpha p}{N_m} & 0 & 0 & 0 \\
\eta^\alpha & 0 & -\delta^\alpha & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & -\gamma^\alpha - \delta^\alpha & 0 & 0 & 0 & 0 \\
0 & \eta^\alpha & 0 & \gamma^\alpha & -\delta^\alpha & 0 & 0 & 0 \\
0 & 0 & 0 & \frac{-b^\alpha R_{hb}^\alpha}{R_{hd}^\alpha} & 0 & -R_{hd}^\alpha & 0 & 0 \\
0 & 0 & 0 & 0 & \frac{b^\alpha R_{hb}^\alpha}{R_{hd}^\alpha} & 0 & -\xi^\alpha - R_{hd}^\alpha & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & \xi^\alpha & -R_{hd}^\alpha
\end{bmatrix}
$$

$$(4)$$

The calculated eigenvalues are given by

$$\lambda_1 = -\eta^\alpha, \lambda_2 = -\delta^\alpha, \lambda_3 = -R_{hd}^\alpha,$$

$\lambda_4 = -R_{hd}^\alpha, \lambda_5 = -(\xi^\alpha + R_{hd}^\alpha), \lambda_6 = -(\gamma^\alpha + \delta^\alpha)$, remaining eigenvalues are the roots of the quadratic polynomial

$$\lambda^2 + (\eta^\alpha + \delta^\alpha)\lambda + (1 - R_1^\alpha) = 0,$$

by the Routh-Hurwitz stability criterion, $\widehat{E}_1^\alpha$ is locally asymptotically stable if and only if $R_1^\alpha < 1$. $\qquad\square$

2.4. The basic reproduction number (R_0^α)

The basic reproduction number R_0^α of the epidemic disease is known as the number of secondary infections caused by a unique infected individual. Hypothetically, when $R_0^\alpha < 1$ the transmissions chains are not beneficial and disease will eradicate from the population. Although if $R_0^\alpha > 1$, each generation will be raised by the number of infected humans and infection will be taken advantage of with the usage of next generation matrix approach. In this situation R_0^α is equal to the spectral radius of $K = FV^{-1}$, where F is a non-negative matrix of the infection items,

and V is the M-Matrix of the corresponding transition terms given as in below [13]

$$
F = \begin{pmatrix}
0 & 0 & 0 & b^\alpha \\
0 & 0 & 0 & p\delta \\
b^\alpha \frac{N_m}{N_h} & 0 & 0 & 0 \\
0 & 0 & 0 & 0
\end{pmatrix},
$$

$$
V^{-1} = \begin{pmatrix}
\frac{1}{\xi^\alpha + R_{hd}^\alpha} & 0 & 0 & 0 \\
0 & \frac{1}{\eta^\alpha} & 0 & 0 \\
0 & 0 & \frac{1}{\gamma^\alpha + \delta^\alpha} & 0 \\
0 & \frac{1}{\eta^\alpha} & \frac{1}{\gamma^\alpha} & \frac{1}{\delta^\alpha}
\end{pmatrix},
$$

and so

$$
FV^{-1} = \begin{pmatrix}
0 & \frac{b^\alpha}{\delta^\alpha} & \frac{b^\alpha}{\gamma^\alpha} & \frac{b^\alpha}{\delta^\alpha} \\
0 & p & \frac{p\delta^\alpha}{\gamma^\alpha} & p \\
\frac{b^\alpha N_m}{N_h}(\xi^\alpha + R_{hd}^\alpha) & 0 & 0 & 0 \\
0 & 0 & 0 & 0
\end{pmatrix}.
$$

The eigenvalues of FV^{-1} is:

$$\frac{p}{2} \pm \frac{1}{2}\sqrt{p^2 + \frac{4b^\alpha N_m}{N_h \gamma^\alpha (\xi^\alpha + R_{hd}^\alpha)}}$$

Therefore, the dominant eigenvalue of FV^{-1} is

$$R_0^\alpha = \frac{p}{2} + \frac{1}{2}\sqrt{p^2 + \frac{4b^{2\alpha} N_m}{N_h \gamma^\alpha (\xi^\alpha + R_{hd}^\alpha)}}. \quad (5)$$

3. Numerical method and simulations

In this section, we will construct the discretization of the model given by Eq. (3) involving the Caputo fractional operator.

First, we rewrite system in Eq. (3) in a compact form so that the relations can be simplified

$$
\begin{cases}
{}^C D^\alpha Y(t) = \mathcal{F}(Y(t)), & 0 < t < b < \infty, \\
Y(0) = Y^0,
\end{cases}
$$

$$(6)$$

where $Y = (E_s, E_i, M_s, M_e, M_i, H_s, H_i, H_r) \in \mathbb{R}_+^8$, where $\mathcal{F}$, is a real-valued continuous vector

function having the Lipschitz condition

$$\|\mathcal{F}(Y_1(t)) - \mathcal{F}(Y_2(t))\| \leq L\|Y_1(t) - Y_2(t)\|, \quad L > 0, \tag{7}$$

and $Y^0 = (E_s^0, E_i^0, M_s^0, M_e^0, M_i^0, H_s^0, H_i^0, H_r^0)$ is the initial vector. By applying the fractional integral operator in Eq. (2) to Eq. (6) the following result is obtained

$$Y(t) = Y_0 + {}^C I^\alpha \mathcal{F}(Y(t)), \quad 0 < t < b < \infty, \tag{8}$$

where ${}^C I^\alpha$ represents the Riemann-Liouville integral operator. In order to propose a numerical scheme, the discretization of the time interval $[0, b]$ is done by equally spaced nodes so that time step size is $\tau = \frac{b}{N}$. Let Y_n denote the approximation of the true solution $Y(t_n)$ at point $t_n = n\tau$ and $0 = t_0 < t_1 < ... < t_N = b$, with $t_{n+1} - t_n = \tau$, $n = 0, 1, 2, ..., (N-1)$. Then, we derive following numerical scheme for the Caputo operator using Euler method [34]

$$Y^{n+1} = Y^0 + \frac{\tau^q}{\Gamma(\alpha+1)} \sum_{k=0}^{n} w_{k+1,j} \mathcal{F}(Y_k), \tag{9}$$

where (n=0,1,2,...,(N-1)) and

$$w_{k+1,j} = (k+1-j)^\alpha - (k-j)^\alpha$$

are the weights of the fractional Euler method. Thus, we obtain the following discretization of the model in Eq. (3)

$$\begin{cases} E_s^{n+1} = E_s^0 \\ \quad + \frac{\tau^q}{\Gamma(\alpha+1)} \sum_{k=0}^{n} w_{k+1,j} \left(e_v^\alpha (1 - p(\frac{M_i^k}{N_m})) - \eta^\alpha E_s^k \right), \\ E_i^{n+1} = E_i^0 \\ \quad + \frac{\tau^q}{\Gamma(\alpha+1)} \sum_{k=0}^{n} w_{k+1,j} \left(e_v^\alpha p(\frac{M_i^k}{N_m}) - \eta^\alpha E_i^k \right), \\ M_s^{n+1} = M_s^0 \\ \quad + \frac{\tau^q}{\Gamma(\alpha+1)} \sum_{k=0}^{n} w_{k+1,j} \left(\eta^\alpha E_s^k - b^\alpha \frac{H_i^k}{N_h} M_s^k - \delta^\alpha M_s^k \right), \\ M_e^{n+1} = M_e^0 \\ \quad + \frac{\tau^q}{\Gamma(\alpha+1)} \sum_{k=0}^{n} w_{k+1,j} \left(b^\alpha \frac{H_i^k}{N_h} M_s^k - \gamma^\alpha M_e^k - \delta^\alpha M_e^k \right), \\ M_i^{n+1} = M_i^0 \\ \quad + \frac{\tau^q}{\Gamma(\alpha+1)} \sum_{k=0}^{n} w_{k+1,j} \left(\gamma^\alpha M_e^k + \eta^\alpha E_i^k - \delta^\alpha M_i^k \right), \\ H_s^{n+1} = H_s^0 \\ \quad + \frac{\tau^q}{\Gamma(\alpha+1)} \sum_{k=0}^{n} w_{k+1,j} \left(R_{hb}^\alpha N_h - b^\alpha \frac{H_s^k}{N_h} M_i^k - R_{hd}^\alpha H_s^k \right), \\ H_i^{n+1} = H_i^0 \\ \quad + \frac{\tau^q}{\Gamma(\alpha+1)} \sum_{k=0}^{n} w_{k+1,j} \left(b^\alpha \frac{H_s^k}{N_h} M_i^k - \xi^\alpha H_i^k - R_{hd}^\alpha H_i^k \right), \\ H_r^{n+1} = H_r^0 \\ \quad + \frac{\tau^q}{\Gamma(\alpha+1)} \sum_{k=0}^{n} w_{k+1,j} \left(\xi^\alpha H_i^k - R_{hd}^\alpha H_r^k \right), \end{cases} \tag{10}$$

where $n = 0, 1, 2, ..., (N-1)$, and $N_m = M_s + M_e + M_i$.

3.1. Simulation results and discussion

We solve the newly proposed fractional dengue model given by Eq. (3) numerically. The effect of temperature on the dynamics of dengue epidemics

is presented by Fig 2 where the temperature values are considered as 18° C, 25° C and 28° C since the parameter values in Table 1 are known at these temperatures. It is stated that (see ref. [20] and the related references therein) e_v-oviposition rate, γ-virus incubation rate in mosquito, δ-adult mosquito death rate and η-pre-adult mosquito maturation rate are found as temperature dependent entomological parameters in the selected district of Taiwan. It is seen that the temperature is a significant environmental factor that affect the transmission in dengue epidemics. The obtained results in Fig 2 shows that the infected human (host) population and infected (pre-adult and adult) mosquito populations reached to their peak values with a higher amount at the temperature of 28° C approximately in the period of 50-100 days. Correspondingly, the susceptible human population showed a sharper decrease whereas the recovered human population size has a sharper increase (after 100 days approx.) at 28° C.

The impact of the fractional order is depicted in Fig 3. In this sense, the temperature effect in vector-host (vertical) transmission dynamics in dengue epidemics formulated by fractional order dengue model is investigated for different values of the fractional order such as $\alpha = 1, 0.8$ and $\alpha = 0.6$ at a fixed temperature of 25° C. The results for all considered values of α reach to steady state for each population and integer order case is recovered when $\alpha = 1$. The susceptible human and pre-adult mosquito populations, recovered human populations reach to different values in the asymptotic case as α differs for the fixed temperature value of 25° C. The fractional order α here enables us to include the memory into the evolution of the epidemic dynamics and also it gives us the opportunity to capture different behaviors of the system components inside the same model.

Moreover, Table 2 verify the Theorem 1 as the condition $R_1^\alpha < 1$ is satisfied for the considered 3-different temperature cases which means that the biologically realistic disease free equilibriums of the system are all asymptotically stable.

Table 2. Stability of $\widehat{E}_1^\alpha$

Equilibrium	R_1^1	$R_1^{0.8}$	$R_1^{0.6}$
$\widehat{E}_1^\alpha$ at 18° C	0.9761	0.9931	0.9761
$\widehat{E}_1^\alpha$ at 25° C	0.9677	0.9897	0.9677
$\widehat{E}_1^\alpha$ at 28° C	0.9633	0.9878	0.9633
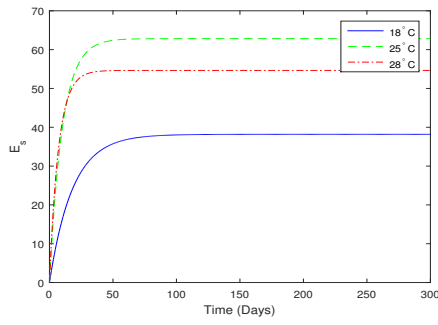
(a) Pre-adult female mosquito-Susceptible
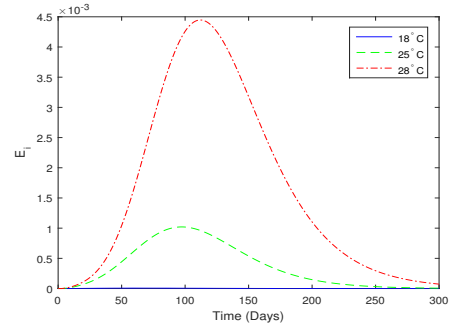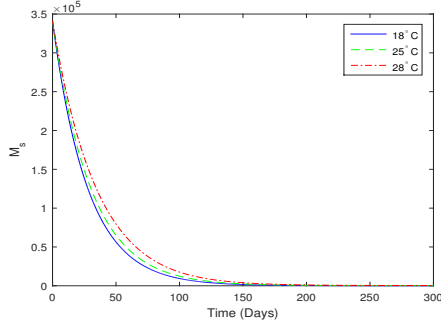
(b) Pre-adult female mosquito-Infected

(c) Adult female mosquito-Susceptible
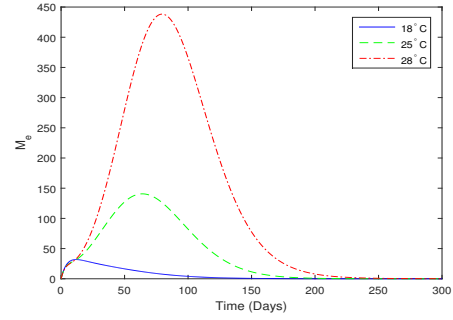
(d) Adult female mosquito-Infected
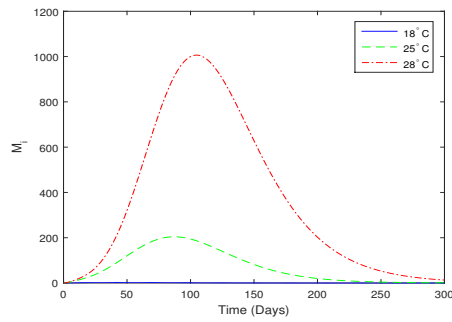
(e) Adult female mosquito-Infectious
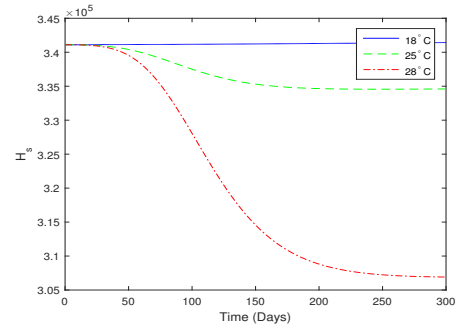
(f) Human population-Susceptible
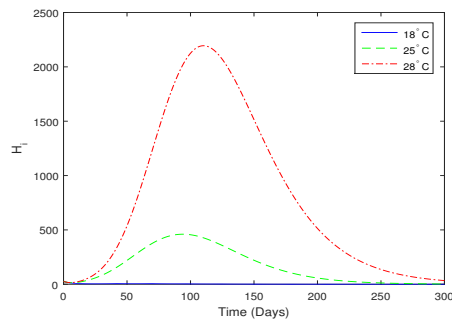
(g) Human population-Infected/Infectious
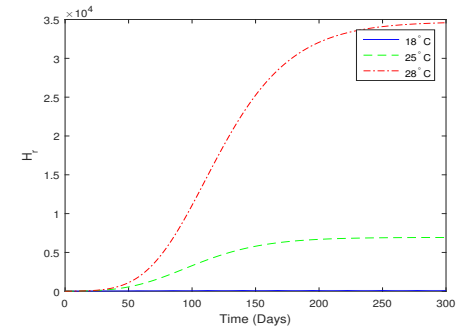
(h) Human population-Recovered/Immune

Figure 2. Simulation results of the fractional order dengue model at temperature 18° C, 25° C and 28° C with $\alpha = 1$.

4. Conclusion

A fractional order dengue model is newly investigated that incorporate temperature dependent features in entomological parameters. In the modeling process four entomological parameters are used for this purpose, that includes the oviposition rate (e_v), pre-adult mosquito maturing rate

η, adult mosquito death rate δ, and the virus incubation rate in mosquitoes γ. Weekly mean temperature data in Kaohsiung (2001-2010) [20] has been utilized to perform the simulations. We applied $L1$ method to solve the model numerically for various fractional order and temperatures. Further, the basic reproduction number is derived for the newly considered fractional order

(a) Pre-adult female mosquito-Susceptible

(b) Pre-adult female mosquito-Infected

(c) Adult female mosquito-Susceptible

(d) Adult female mosquito-Infected

(e) Adult female mosquito-Infectious

(f) Human population-Susceptible

(g) Human population-Infected/Infectious

(h) Human population-Recovered/Immune
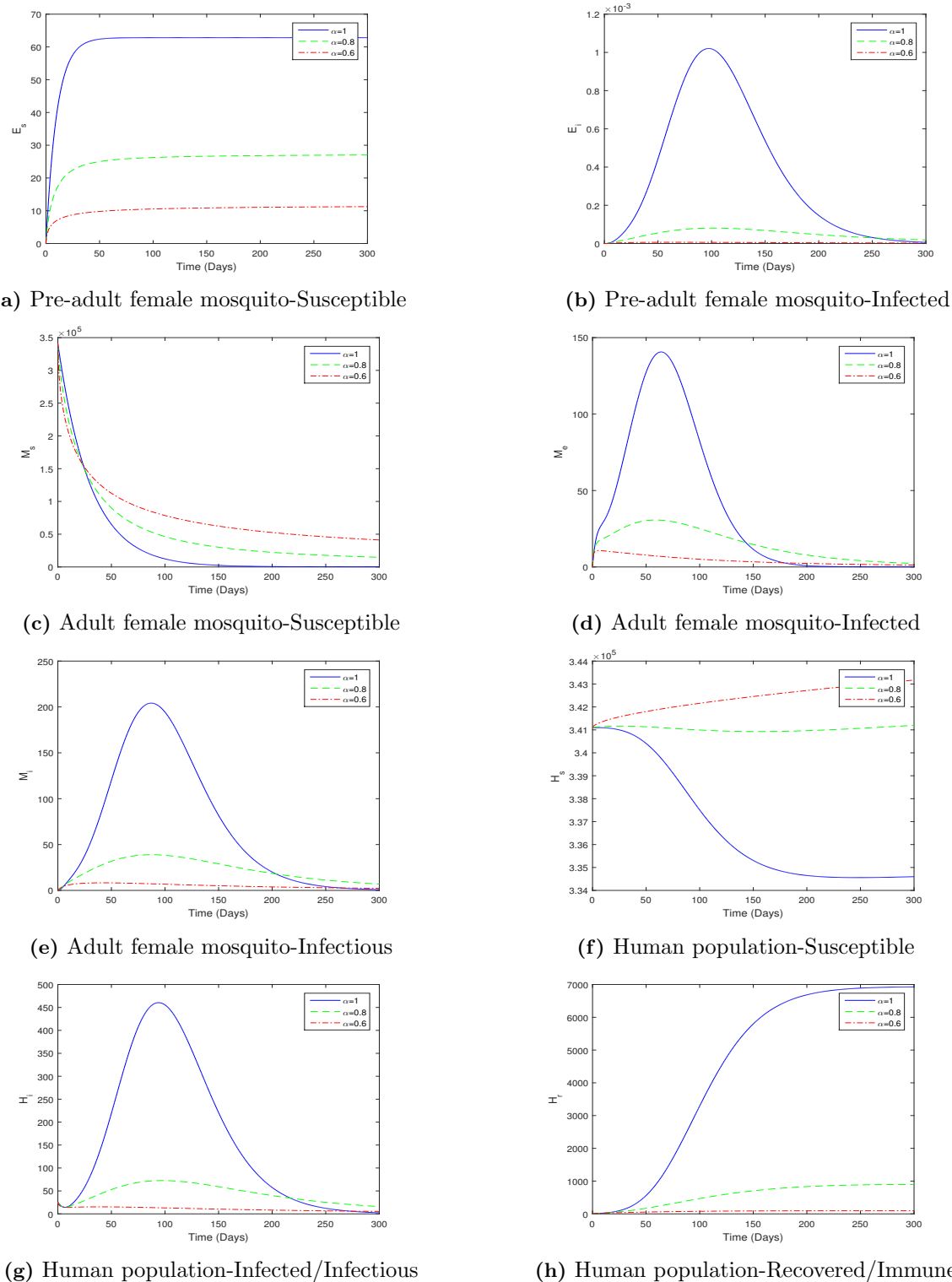
Figure 3. Simulation results of the fractional order dengue model at $\alpha = 1$, $\alpha = 0.8$, $\alpha = 0.6$ at temperature $25°$ C.

dengue model (that is formulated by using Caputo fractional operators). Correspondingly, stability analysis is performed and the local asymptotic stability of the disease-free equilibria is obtained. Simulation results refers that the highest danger of dengue transmission exists at temperature $28°$ C. The control of dengue transmission is based on the control of the aquatic and adult stages of the mosquito and so this study can contribute to adapt effective control strategies like the use of pesticide and other techniques.

Acknowledgments

References

[1] World Health Organization(WHO). (2019). Dengue Situation Update [online]. Available from: `https:www.who.int/westernpacific/emergencies/surveillance/dengue` [Accessed August 2019]

[2] World Health Organization(WHO). (2019). Dengue and Severe Dengue [online]. Available from: `https:www.who.int/news-room/fact-sheets/detail/dengue-and-severe-dengue` [Accessed June 2019]

[3] United States Environmental Protection Agency(EPA). (2019). Mosquito Life Cycle [online]. Available from: `https:www.epa.gov/mosquitocontrol/mosquito-life-cycle` [Accessed July 2019]

[4] Chowell, G., Diaz-Duenas, P., Miller, J.C., Alcazervelazco, A., Hyman, J.M., Fenimore, P.W., & Chavez, C. (2007). Estimation of the reproduction number of dengue fever from spatial epidemic data. *Mathematical Biosciences*, 208, 571–589.

[5] Derouich, M., & Boutayeb, A. (2006). Dengue fever: Mathematical modelling and computer simulation. *Applied Mathematics and Computation*, 177, 528–544.

[6] Dietz, K. (1974). Transmission and control of arbo virus diseases. *SIAM*, 975, 104–21.

[7] Esteva, L., & Vargas, C. (1998). Analysis of dengue transmission model. *Mathematical Biosciences*, 150(2), 131-51.

[8] Estweva, L., & Yang, H. (2005). Mathematical model to assess the control of Aedes aegypti mosquitoes by the sterile insect technique. *Mathematical Biosciences*, 198(2), 132–47.

[9] Garba, S.M., Gumel, A.B., & Abu Bakar, M.R. (2008). Backward bifurcations in dengue transmission dynamics. *Mathematical Biosciences*, 215(1), 11–25.

[10] Thome, R.C., Yang, H.M., & Esteva, L. (2010). Optimal control of Aedes aegypti mosquitoes by the sterile insect technique and insecticide. *Mathematical Biosciences*, 223(1), 12–23.

[11] Pinho, S.T.R., et al. (2010). Modelling the dynamics of dengue real epidemics. *Philosophical Transactions of The Royal Society*, 368, 5679–5693.

[12] Sardar, T., Rana, S., & Chattopadhyay, J. (2015). A mathematical model of dengue transmission with memory. *Communnications in Nonlinear Science and Numerical Simulation*, 22(1), 511–525.

[13] Stanislavsky, A. (2000). Memory effects and macroscopic manifestation of randomness. *Physical Review E*, 61(5), 4752–4759.

[14] Syafruddin, S., & Noorani, M.S.M. (2012). SEIR model for transmission of dengue fever in Selangor Malaysia. *International Journal of Modern Physics: Conference Series*, 9, 380-389.

[15] Abdelrazec, A., Belair, J., Shan, C., & Zhu, Y. (2016). Modeling the spread and control of dengue with limited public health resources. *Mathematical Biosciences*, 271, 136–145.

[16] Andraud, M., Hens, N., Marais, C., & Beutels, P. (2012). Dynamic epidemiological models for dengue transmission: a systematic review of structural approaches. *PloS One*, 7(11), e49085.

[17] Robert, Michael A., Christofferson, R.C., Weber, P.D., & Wearing, H.J. (2019). Temperature impacts on dengue emergence in the United States: Investigating the role of seasonality and climate change. *Epidemics*, 28, UNSP 10034.

[18] Alto, B., & Bettinardi, D. (2013). Temperature and dengue virus infection in mosquitoes: Independent effects on the immature and adult stages. *The American Journal of Tropical Medicine and Hygiene*, 88, 497–505.

[19] Taghikhani, R., & Gumel, A.B. (2018). Mathematics of dengue transmission dynamics: Roles of vector vertical transmission and temperature fluctuations. *Infectious Disease Modelling*, 3, 266–292.

[20] Chen, S.C., & Hsieh, M.H. (2012). Modeling the transmission dynamics of dengue fever: Implications of temperature effects. Science of the Total Environment, 431, 385-391.

[21] Yang, H.M., Macoris, M.L.G., Galvani, K.C., Andrighetti, M.T.M., & Wanderley, D.M.V. (2009). Assessing the effects of temperature on dengue transmission. *Epidemiology & Infection*, 137(8), 1179–1187.

[22] Hamdan, N.I., & Kilicman, A. (2018). A fractional order SIR epidemic model for dengue transmission. *Chaos*, Solutions and Fractals, 114, 55–62.

[23] Ozalp, N., & Demirci E. (2011). A fractional order SEIR model with vertical transmission. *Mathematical and Computer Modelling*, 54(1-2), 1–6.

[24] Qureshi, S., Yusuf, A., Shaikh, A.A., & Inc, M. (2019). Transmission dynamics of varicella zoster virus modeled by classical and novel fractional operators using real statistical data. *Physica A: Statistical Mechanics and its Applications*, 534, 122149.

[25] Qureshi, S., & Yusuf, A. (2019). Fractional derivatives applied to MSEIR problems: Comparative study with real world data. *The European Physical Journal Plus*, 134 (4), 171 (13 pages).

[26] Pooseh, S., Rodrigues, H.S., & Torres, D.F.M. (2011). *Fractional derivatives in dengue epidemics*. In: T.E. Simos, G. Psihoyios, C. Tsitouras and Z. Anastassi, eds. Numerical Analysis and Applied Mathematics, American Institute of Physics, Melville, 739–742.

[27] Diethelm, K. (2013). A fractional calculus based model for the simulation of an outbreak of dengue fever. *Nonlinear Dynamics*, 71, 613-619.

[28] Qureshi, S., & Atangana, A. (2019). Mathematical analysis of dengue fever outbreak by novel fractional operators with field data. *Physica A*, 526, 121127.

[29] Jajarmi, A., Arshad, S., & Baleanu, D. (2019). A new fractional modelling and control strategy for the outbreak of dengue fever. *Physica A: Statistical Mechanics and its Applications*, Volume 535, 122524.

[30] Samko, S.G., Kilbas, A.A., & Marichev, O.I. (1993). *Fractional Integrals and Derivatives: Theory and Applications*. Gordon and Breach, New York.

[31] Civil Affairs Bureau, Kaohsiung city government. Available from [online]: `http: cabu.kcg.gov.tw/cabu2/statis61B2.aspx` [Accessed December 2011]

[32] Adams, B., & Boots, M. (2010). How important is vertical transmission in mosquitoes for the persistence of dengue? Insights from a mathematical model. *Epidemics*, 2, 1–10.

[33] Joshi, V., Mourya, D.T., & Sharma, R.C. (2002). Persistence of dengue-3 virus through transovarial transmission passage in successive generations of Aedes aegypti mosquitoes. American Journal of Tropical Medicine and Hygiene, 67(2), 158-61.

[34] Li, C., & Zeng, F. (2013). The finite difference methods for fractional ordinary differential equations. *Numerical Functional Analysis and Optimization*, 34(2), 149–79.

Ozlem Defterli *received her B.Sc. degree major in Mathematics and Computer Science and minor in Computer Engineering from Cankaya University (Ankara, Turkey) with full scholarship. After completing her M.Sc. education in Mathematics and Computer Science Program (with full scholarship) in the same university, she got her Ph.D. degree from the Department of Mathematics in Middle East Technical University (METU) in August 2011. She has been a faculty member in Cankaya University - Department of Mathematics since 2002. Her research interests lie in the areas of numerical methods for classical and fractional order differential equations, applied optimal control, mathematical modelling and mathematical biology.*

http://orcid.org/0000-0003-0918-9834

RESEARCH ARTICLE

An algebraic stability test for fractional order time delay systems

Münevver Mine Özyetkin[a*] and Dumitru Baleanu[b,c]

[a] Department of Electrical and Electronics Engineering, Aydın Adnan Menderes University, Turkey
[b] Department of Mathematics, Faculty of Art and Sciences, Cankaya University, Turkey
[c] Institute of Space Sciences, P.O. Box 077125, Magurele-Bucharest, Romania
m.ozyetkin@adu.edu.tr, dumitru@cankaya.edu.tr

ABSTRACT

In this study, an algebraic stability test procedure is presented for fractional order time delay systems. This method is based on the principle of eliminating time delay. The stability test of fractional order systems cannot be examined directly using classical methods such as Routh-Hurwitz, because such systems do not have analytical solutions. When a system contains the square roots of s, it is seen that there is a double value function of s. In this study, a stability test procedure is applied to systems including $\sqrt{s}$ and/or different fractional degrees such as s^{α} where $0 < \alpha < 1$, and $\alpha \epsilon \mathbb{R}$. For this purpose, the integer order equivalents of fractional order terms are first used and then the stability test is applied to the system by eliminating time delay. Thanks to the proposed method , it is not necessary to use approximations instead of time delay term such as Padé. Thus, the stability test procedure does not require the solution of higher order equations.

1. Introduction

The systems shown by differential equations with real orders instead of integer orders are called fractional order systems [1]. Fractional order systems (FOS) are one of the most popular research topics of today. Although the mathematical analysis of such systems has been known since 1695, mostly, it has been discussed and investigated by mathematicians because of its complexity [1]. The most important feature of this subject is that it expresses real systems better than integer order ones [2]. As it is known time delays are intrinsic of a variety of electrical, electronic, and communication systems, control applications, power systems with long transmission lines, and many real world applications [3–5]. In control applications, there are many examples of neutral-type time-delay systems as well as discrete-continuous hybrid systems regarded as delay differential algebraic equations (DDAEs) [4]. Besides, power systems with long transmission lines can be modeled as DDAEs

for certain assumptions [4]. Systems with neutral delay differential equations (NDDEs) contains delays in both the state variables and their time derivatives [3]. If fractional order systems include time delay, the analysis of such systems becomes more and more complicated. And, the studies to obtain analytical solutions of fractional order systems with delays are very restricted.

Many studies have been carried out in relation to FOS, in the literature [6–14]. Analytical stability test procedures of FOS are still important research topics. Analytical stability test procedures such as the Routh-Hurwitz method cannot be applied to FOS,directly. There are some studies on the stability of FOS in [1, 15–22]. In [15], a method for stability analysis of distributed parameter systems having delay is presented. This method is also applicable to FOS. In [18], internal and external stabilities of fractional differential systems in the state-space form are investigated. In [16], stability for a certain class of linear and nonlinear fractional order systems is presented. A

---

*Corresponding Author

test procedure based on the Nyquist stability criterion is presented in [1]. However, the studies related to the stability of FOS continue, and an analytical stability test technique does not exist, to the best of the author's knowledge. In addition, examining of time response analysis of such systems is very complicated since calculating inverse Laplace transforms of them is difficult, in spite of this, time response analysis of FOS can be made by using integer order approximations. Some toolboxes designed to examine such systems can be found in [23–26].The frequency domain based methods can be considered advantageous for FOS since the stability of FOS can be tested thanks to frequency domain methods such as the Nyquist curve.

In the literature, the most preferred method to investigate FOS is to use integer-order approximations [27, 28]. That is, in order to apply methods in classical control to such systems, integer-order equivalent transfer functions can be used. There are many approximation methods to obtain integer order equivalencies of fractional order differential equations. For example, the continued fraction expansion method (CFE), Oustaloup, Carlson and Matsuda's method and Maclaurin series etc. [29]. In this study, the CFE method is preferred to obtain integer order approximations of FOS. Then, the proposed algebraic stability test is applied to the system. According to results that are obtained in [28], it is observed that when the degree of used approach increases the obtained results are closer to the original system. However, this makes the process mathematically more complicated. As aforementioned, the analytical stability test of FOS cannot be performed by classical methods, directly. Therefore, this study is aimed to fill this gap. For this purpose,in the first step, the integer order equivalents of fractional order terms are used. In the second step, the stability test is applied to the system by eliminating time delay. As it is known, in general, analytical stability test procedures of time delay systems require to use some approximation methods such as Padé. Besides, we need to use higher order Padé approximations instead of time delay term to obtain more reliable results. This process makes the analysis of time delay systems more complicated. However, using the proposed method, it is not necessary to use approximations instead of time delay term since it is eliminated. Thus, the stability test procedure does not require the solution of higher order equations. This makes the proposed method practical and preferable. For the future studies, this method can be extended for systems with multiple time delays. It can also be applied to systems controlled by fractional order controllers. Besides, power systems modeled by delayed differential equations can be investigated by using the proposed method.

This paper is organized as follows: In the first section, literature information has been presented. In the second section, fractional order time delay systems are introduced. In the third section, an algebraic stability test procedure is presented for fractional order time delay systems. In the last section, concluding remarks have been presented.

2. Fractional order time delay systems

Systems where derivatives are expressed in fractional orders instead of integer ones are called fractional order systems. A unity feedback control system is given in Fig.(1).

Definition 1. Fractional order time delay systems are represented as follows.

$$
\begin{aligned}
G(s) &= \frac{N(s)}{D(s)} e^{-hs} \\
&= \frac{b_m s^{\beta_m} + b_{m-1} s^{\beta_{m-1}} + \ldots + b_0 s^{\beta_0}}{a_n s^{\alpha_n} + a_{n-1} s^{\alpha_{n-1}} + \ldots + a_0 s^{\alpha_0}} e^{-hs}
\end{aligned}
\tag{1}
$$

Where, h represents time delay, a_k $(k = 0, ..., n)$, and b_k $(k = 0, ..., m)$ are constants, α_k $(k = 0, ..., n)$, and β_k $(k = 0, ..., m)$ are arbitrarily real numbers. Where, one can assume inequalities $\alpha_n > \alpha_{n-1} > \ldots > \alpha_0$ and $\beta_m > \beta_{m-1} > \ldots > \beta_0$ without loss of generality [30].

Time delay, which may cause poor performance or even instability in system response, is a common case in many industrial processes. It can be originated from the internal dynamics of the system [31]. Since stability test of time delay systems cannot be performed directly, some approximations such as Padé are used instead of time delay term. However, in some cases, the first order Padé approximation may not give correct results in terms of stability [31]. Therefore, to obtain more correct results, it is necessary to increase the degree of approach, which makes the processes mathematically more complicated. Thus, a stability test eliminating time delay will be important for simplicity.

3. A Stability test for fractional order time delay systems

Definition 2. The characteristic equation for a linear system having a single time delay is expressed as follows, where h is time delay [32].

$$
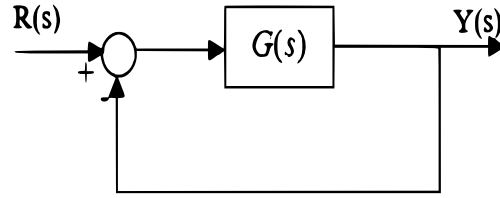\Delta(s, h) = \Delta_1(s) + \Delta_2(s) e^{-hs} = 0
\tag{2}
$$

Figure 1. A feedback system.

Definition 3. The general representation of this expression for systems having multiple commensurate time delay is given as follows [32].

$$\Delta_n(s,h) = \sum_{k=0}^{n} \Delta_k(s)e^{-ksh} \qquad (3)$$

For zero delay systems, the necessary and sufficient condition of asymptotic stability is known as the presence of all the roots of the characteristic equation on the left half of the complex s plane. For systems with time delay, this result may be stable or unstable for some values of h [32]. It has been concluded that the system is asymptotically stable regardless of delay for a particular case where all positive values of time delay h are not negative [32]. Here, the main problem is to determine h values when $\Delta(s,h) = \Delta_1(s) + \Delta_2(s)e^{-hs} = 0$ has root/or roots on the complex axis. $\Delta(s,h) = 0$ is an implicit function of s and h, which may exceed or not exceed the imaginary axis. Suppose that all the roots of $\Delta(s,0) = 0$ are in the left-half plane. So, the system is stable for zero time delay. If $\Delta(s,h) = 0$ has a root on the imaginary axis when $s = j\omega$ for some values of h, it can be said that $\Delta(-s,h) = 0$ has also a root on the imaginary axis for the same values of h and ω. Thus, with the same common root $\Delta(s,h) = 0$ and $\Delta(-s,h) = 0$ for the determination of the roots on the imaginary axis, we do not need to find h values. That is, a structure independent of time delay is obtained.

Theorem. *A system is asymptotically stable regardless of delay for a particular case where all positive values of time delay h are not negative. If $\Delta(s,h) = \Delta_1(s) + \Delta_2(s)e^{-hs} = 0$ has root/roots on the complex axis, the value range of h is calculated for the stability.*

Proof. *$\Delta(s,h) = 0$ is an implicit function of s and h, which may exceed or not exceed the imaginary axis. If all the roots of $\Delta(s,0) = 0$ are in the left-half of s plane, the system is stable for zero time delay. If $\Delta(s,h) = 0$ has a root on the imaginary axis when $s = j\omega$ for some values of h, $\Delta(-s,h) = 0$ has also a root on the imaginary axis for the same values of h and ω. Thus, with the same common root $\Delta(s,h) = 0$ and $\Delta(-s,h) = 0$ for the determination of the*

roots on the imaginary axis, it is not necessary to find h values.

Corollary 1. The Eq.(6) is obtained by eliminating the time delay h from Eq.(4) and Eq.(5). It is clear that this structure is independent of time delay.

$$\Delta(j\omega, h) = \Delta_1(j\omega) + \Delta_2(j\omega)e^{-j\omega h} = 0 \qquad (4)$$

$$\Delta(-j\omega, h) = \Delta_1(-j\omega) + \Delta_2(-j\omega)e^{+j\omega h} = 0 \qquad (5)$$

$$M(\omega^2) = \Delta_1(j\omega)\Delta_1(-j\omega) - \Delta_2(j\omega)\Delta_2(-j\omega) = 0 \qquad (6)$$

Corollary 2. *Where, it is clear that $M(\omega^2)$ is a polynomial in the form $\omega^2 = -s^2$. If $M(\omega^2) = 0$ does not have positive roots, the system is stable for all $h \geq 0$.*

The proposed stability test procedure is summarized as follows:

(1) In the first step, Eq.(2) is turned into the form of Eq.(7) for *h=0*.

$$\Delta(s,0) = \Delta_1(s) + \Delta_2(s) = 0 \qquad (7)$$

It is tested whether the zeros of the characteristic equation are in the left half of s plane. The system is stable for h if all the zeros are located in the left half of s plane. If so, the second step is applied.

(2) The presence of positive roots of $M(\omega^2) = 0$ is investigated. The system is stable for all $h \geq 0$ if $M(\omega^2) = 0$ does not have any positive roots. If M has at least one positive root, the range of h must be investigated for the stability.

(3) If the second condition is met, the following equations are used to determine the range of h [32].

$$\begin{aligned} \cos(\omega h) &= Re\left\{\frac{-\Delta_1(j\omega)}{\Delta_2(j\omega)}\right\}, \\ \sin(\omega h) &= Im\left\{\frac{\Delta_1(j\omega)}{\Delta_2(j\omega)}\right\} \end{aligned} \qquad (8)$$

For the given value of ω, $h_0(\omega)$ is the smallest positive value of h providing Eq.(8), and the general solution is given as follows.

$$h = h_0(\omega) + 2\pi n/\omega, n = 0, 1, 2, 3, ... \quad (9)$$

More information about the method can be found in [32, 33]. Let handle some examples to better understand the subject.

3.1. Example 1

Consider the characteristic equation given as follows.

$$\Delta(s, h) = s^2 + 4s + 4 - e^{-hs} = 0 \quad (10)$$

When the procedure described above is applied, one obtains

$$\Delta(s, 0) = s^2 + 4s + 3 = 0 \quad (11)$$

for $\Delta(s, 0) = 0$, we obtain $s = -3$ and $s = -1$. It is clear that the system is stable for $h = 0$. Thus, the second step is applied to the system as follows.

$$M(\omega^2) = (-\omega^2 + 4j\omega + 4)(-\omega^2 - 4j\omega + 4) - 1 = \omega^4 + 8\omega^2 + 15 \quad (12)$$

Since $\omega^2 = -3$ and $\omega^2 = -5$, there is no positive root of $M(\omega^2) = 0$. It means that there is no any point touching or crossing imaginary axis. Thus, this system is stable independent of time delay h.

3.2. Example 2

For a unity feedback system given in Fig.(1), $G(s)$ is given by Eq.(13).

$$G(s) = \frac{1}{s^{1.1} + 2} e^{-hs} \quad (13)$$

The characteristic equation of the system is obtained as follows.

$$\Delta(s, h) = s^{1.1} + 2 + e^{-hs} = 0 \quad (14)$$

In this equation, if we use the first order integer approximation instead of fractional order term, the characteristic equation is obtained as follows.

$$\Delta(s, h) = s^2 + 2.46s + 2 + (0.82s + 1)e^{-hs} = 0 \quad (15)$$

The characteristic equation is stable for $h = 0$ and it is obtained as follows.

$$\Delta(s, 0) = s^2 + 3.28s + 3 = 0 \quad (16)$$

In this case, the second stage is applied. Therefore, M is obtained by Eq.(17)

$$M(\omega^2) = (-\omega^2 + 2.46j\omega + 2)(-\omega^2 - 2.46j\omega + 2) - (1 + 0.82j\omega)(1 - 0.82j\omega) \quad (17)$$

Since Eq.(17) does not have a positive solution, the system is stable regardless of time delay. Let's examine the Nyquist curve of the system to confirm this result. The Nyquist diagrams of the original system and the first order approximation for $\omega = 0 : 0.01 : 5$, and $h = 1$ are shown in Fig.(2). As can be seen from the Fig.(2), the system is stable because the curve does not contain the critical point $(-1, j0)$. Besides, the results of original system and of first order approximation are very close to each other. In Fig.(3), the Nyquist diagrams of the original system are given for $\omega \in [0, 50]$, and $h = 0.1 : 0.1 : 2$. The Nyquist diagrams of original system and of first order approximation for $\omega \in [0, 50]$, and $h = 0.1 : 0.1 : 2$ are shown in Fig.(4). As can be seen from Fig.(3) and Fig.(4), the curves do not include the critical point for increasing values of h. Therefore, if the system is stable for condition 1 and 2, as stated in the stability test procedure in section 3, it can be said that it is stable for all values of $h \geq 0$. The Nyquist diagrams are shown in Fig.(3) and Fig.(4) also support this result.

In this example, if we use second order integer approximation instead of fractional order term, the characteristic equation is obtained as follows.

$$\Delta(s, h) = 1.351s^3 + 6.67s^2 + 10.34s + 2.702 + (s^2 + 4.67s + 1.351)e^{-hs} = 0 \quad (18)$$

If the procedure is applied to the system, one obtains

$$\Delta(s, 0) = 1.351s^3 + 7.67s^2 + 15.01s + 4.053 = 0 \quad (19)$$

where, $s_{1,2} = -2.6791 \pm j1.491$, and $s_3 = -0.3191$. It is clear that the system is stable for $h = 0$.

In the second step, M is obtained by Eq.(20)

$$M(\omega^2) = (-1.351j\omega^3 - 6.67\omega^2 + 10.34j\omega + 2.702) \times (1.351j\omega^3 - 6.67\omega^2 - 10.34j\omega + 2.702) - (-\omega^2 + 4.67j\omega + 1.351)(-\omega^2 - 4.67j\omega + 1.351) \quad (20)$$

Since Eq.(20) does not have a positive solution, the system is stable regardless of time delay. Fig.(5) shows unit step responses of the system (using first order approximation) with the second order Padé approximation for $h = 0.1 : 0.1 : 1.2$.

If we use a PI controller of the form $C(s) = (k_p s + k_i)/s$, as shown in Fig.(6), unit step responses of the system are depicted in Fig.(7) for $k_p = k_i = 1$. Here, It should be noted that M can
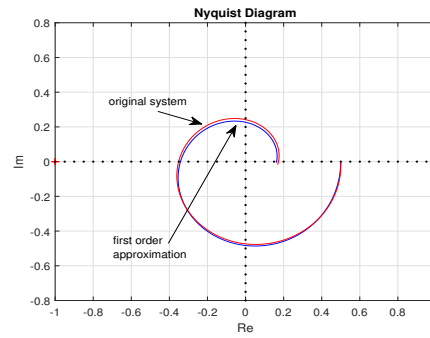
Figure 2. Nyquist diagrams of the original system and the first order approximation for $\omega \in [0, 5]$ and $h = 1$.
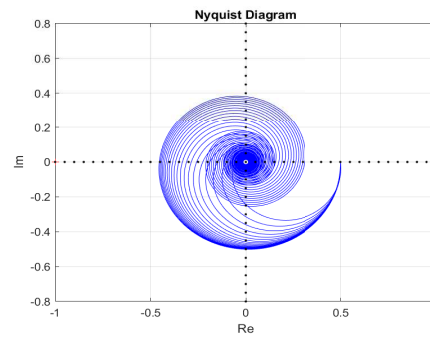
Figure 3. Nyquist diagrams of the original system for $\omega \in [0, 50]$ and $h = 0.1 : 0.1 : 2$.
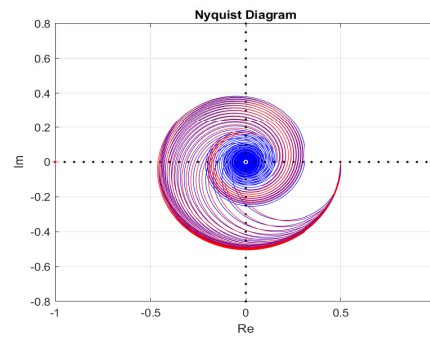
Figure 4. Nyquist diagram of the original system (blue) and the first order approximation (red) for $\omega \in [0, 50]$ and $h = 0.1 : 0.1 : 2$.
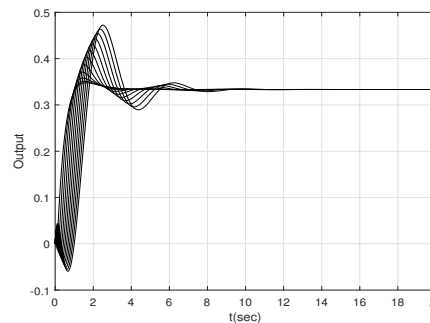
Figure 5. Unit step responses of the system (the first order approximation) with the second order Pade for $h = 0.1 : 0.1 : 1.2$.

have positive solutions. Thus, delay free system can be unstable for some values of k_p and k_i. In this case, it should be determined if the root touches the imaginary axis. If not, it means the system is unstable for $h = 0$, but it is stable for infinite small h, that is, it is stable for
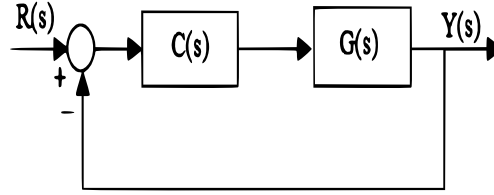
Figure 6. Feedback control system with PI controller.
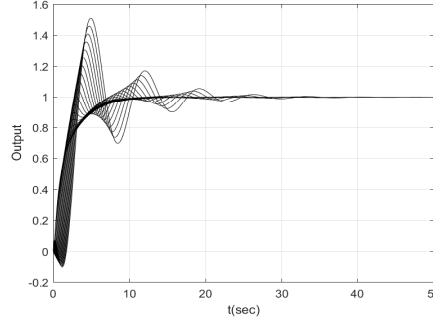
Figure 7. Unit step responses of the first order approximation and the second order Padé for $h = 0.1 : 0.1 : 1.2$, and k$_p$=k$_i$=1.

$0 < h < h_0(\omega)$. If the root touches the imaginary axis, then the system is unstable for $h = 0$ and corresponding values of k_p and k_i for that h.

3.3. Example 3

In this example, $G(s)$ is given by

$$G(s) = \frac{1}{\sqrt{s}(s+1)}e^{-hs} \qquad (21)$$

By using the first order approximation, the characteristic equation of the system is obtained as

$$\Delta(s,h) = 3s^2 + 4s + 1 + (s+3)e^{-hs} = 0 \quad (22)$$

The characteristic equation for $h = 0$ is obtained as follows

$$\Delta(s,0) = 3s^2 + 5s + 4 = 0 \qquad (23)$$

Where the roots of the characteristic equation are $s_{1,2} = -0.833 \pm j0.8$. Thus, the system is stable for $h = 0$.

Thus, the second step of the procedure is applied to the system. And one obtains

$$\begin{aligned} M(\omega^2) &= (-3\omega^2 + 4j\omega + 1)(-\omega^2 - 4j\omega + 1) \\ &\quad -(3+j\omega)(3-j\omega) \end{aligned}$$
$$(24)$$

From Equation Eq.(24), we obtain $\omega^2 = -1.5672$, and $\omega^2 = 0.5672$. Since $M(\omega^2) = 0$ has a positive solution, there is a root touching the imaginary axis. In this case, it is necessary to determine the stability range of h. For this purpose, using the

Eq.(8) and Eq.(9) the range of h making the system stable is calculated as $0 \le h < 2.1086$. The Nyquist curve for $h = 1$ and the critical point $h = 2.1086$ are shown in Fig.(8) according to the first order approximation. As can be seen from the Fig.(8), $h = 2.1086$ is the critical point for stability of the system. However, this value was obtained according to the first order approximation. That is, when the second, third and fourth order approximations are used, the range value of h would change. Depending on results in [28], it can be said that third and fourth order approximations provide the best results in capturing the original system. Thus, the value of h obtained using these approximations will probably provide the best results for the system. But this change may not involve big numerical differences. That is, using first order approximation may be sufficient to examine stability of the system in terms of simplicity. The unit step responses of the system according to the first order approximation are given in Fig.(9) for $h = 1$, and according to the critical point $h = 2.1086$. As can be seen from the Fig.(9), the critical point gives an oscillatory response as expected. Fig.(10) shows the unit step responses of the system for $h = 2.2$ and $h = 2.5$ values exceeding the critical point. The system becomes unstable after the critical point.

In this example, if we use the second order integer approximation instead of the fractional order parameter, the characteristic equation is obtained as follows.
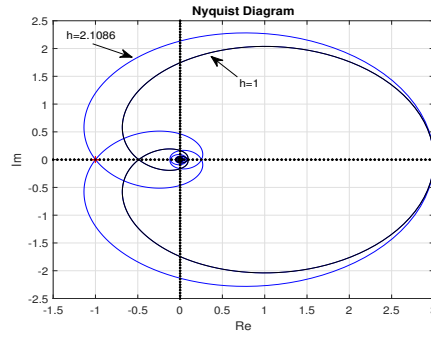
Figure 8. Nyquist diagram of the first order approximation for $h = 1$ and the critical point $h = 2.1086$.
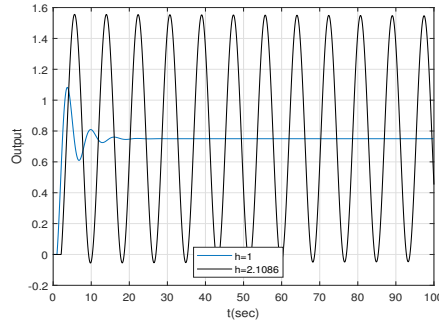
Figure 9. Unit step responses of Example 3.
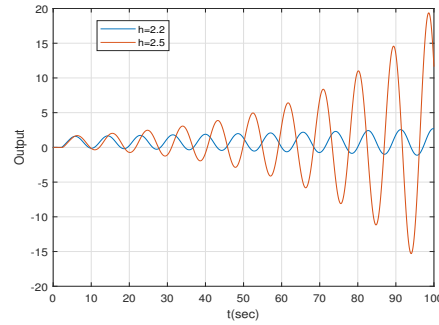
Figure 10. Unit step responses of Example 3 for $h=2.2$ and $h=2.5$.

$$\Delta(s, h) = 5s^3 + 15s^2 + 11s + 1 \\ +(s^2 + 10s + 5)e^{-hs} = 0 \tag{25}$$

If the procedure is applied to the system, one obtains

a.

$$\Delta(s, 0) = 5s^3 + 16s^2 + 21s + 6 = 0 \tag{26}$$

where, $s_{1,2} = -1.4074 \pm j1.0654$, and $s_3 = -0.3851$. It is seen that the system is stable for $h = 0$.

b. In the second step, M is obtained by Eq.(27)

$$M(\omega^2) = (-5j\omega^3 - 15\omega^2 + 11j\omega + 1) \\ \times (5j\omega^3 - 15\omega^2 - 11j\omega + 1) - (-\omega^2 + 10j\omega + 5) \\ \times (-\omega^2 - 10j\omega + 5) \tag{27}$$

From Equation Eq.(27), we obtain $\omega^2 = -4.5038$, $\omega^2 = -0.4906$, and $\omega^2 = 0.4344$. That is, $M(\omega^2) = 0$ has a positive solution. Using the Eq.(8) and Eq.(9) the range of h making the system stable is calculated as $0 \le h < 2.6962$. The unit step response of the system according to the second order approximation is given in Fig.(11) for the critical point $h = 2.6962$. As can be seen from the Fig.(11), the critical point gives an oscillatory response as expected.

4. Conclusion

In this study, an algebraic stability test procedure based on the principle of eliminating time delay is presented for fractional order systems with a single time delay. Thus, mathematical operations
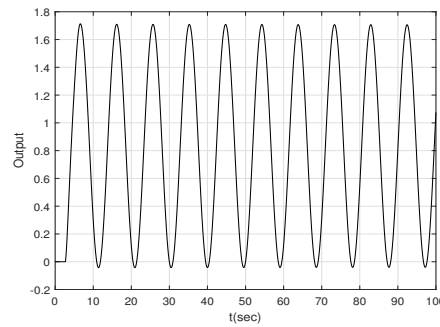
Figure 11. Unit step response of Example 3 for the second order approximation and the critical point $h = 2.6962$.

that are already complicated for FOS have become easier. The examples show that the proposed method gives very reasonable results. For this purpose, integer-order approximations have been used. Thus, a fractional order equation has been turned into an integer-order one, and then the stability test has been applied to the system. When using integer-order approximations, there can be a difference depending on the degree of approximation. Studies have shown that good approximation results for FOS are obtained when using third or fourth-order approximations. Therefore, when determining the stability range of h, the order of approximation can cause some differences in the calculations. However, it can be said that the first order approximation is sufficient for determining whether a system is stable or unstable because higher order approximations make mathematical operations quite complicated. Besides, too large values of the time delay can produce unwanted results in system performance. Thus, it is necessary to investigate of stability range of h to obtain reasonable results. For future works, stability analysis can be investigated for FOS having parameter uncertainty or different time delays. In addition, stability for different types of controllers can also be investigated. As there are no analytical methods in this area, the studies on this subject will contribute significantly to the field.

References

[1] Petráš, I. (2011). Stability test procedure for a certain class of the fractional-order systems. *Proceedings of the 2011 12th International Carpathian Control Conference*, ICCC'2011, (5), 303-307.

[2] Podlubny, I. (1999). Fractional-order systems and $PI^\lambda D^\mu$-controllers. *IEEE Transactions on Automatic Control*, 44(1), 208–214.

[3] Liu, M., Dassios I., Milano F. (2019). On the Stability Analysis of Systems of Neutral Delay Differential Equations. *Circuits, Systems, and Signal Processing*, 38(4), 1639-1653.

[4] Milano, F., Dassios, I. (2016). Small-Signal Stability Analysis for Non-Index 1 Hessenberg Form Systems of Delay Differential-Algebraic Equations. *IEEE Transactions on Circuits and Systems I: Regular Papers*, 63(9), 1521-1530.

[5] Liu, M., Dassios I., Tzounas, G., Milano F. (2019). Stability Analysis of Power Systems with Inclusion of Realistic-Modeling of WAMS Delays. *IEEE Transactions on Power Systems*, 34(1), 627-636.

[6] Petras, I. (2010). Fractional-Order Memristor-Based Chua's Circuit. *IEEE Transactions on Circuits and Systems*, 57(12), 975–979.

[7] Dorčák, L., Valsa, J., Gonzalez, E., Terpák, J., Petráš, I., & Pivka, L. (2013). Analogue realization of fractional-order dynamical systems. *Entropy*, 15(10), 4199-4214.

[8] Shah, P., & Agashe, S. (2016). Review of fractional PID controller. *Mechatronics*, 38, 29–41.

[9] Podlubny, I., Petráš, I., O'Leary, P., Dorčák, L., & Vinagre, B. M. (2002). Analogue realizations of fractional order controllers. *Nonlinear dynamics*, 29, 281-296.

[10] Luo, Y., & Chen, Y. (2012). Stabilizing and robust fractional order PI controller synthesis for first order plus time delay systems. *Automatica*, 48(9), 2159–2167.

[11] Tang, X., Shi, Y., & Wang, L. L. (2017). A new framework for solving fractional optimal control problems using fractional pseudospectral methods. *Automatica*, 78, 333-340.

[12] Razminia, A., & Baleanu, D. (2013). Complete synchronization of commensurate fractional order chaotic systems using sliding mode control. *Mechatronics*, 23(7), 873-879.

[13] Ozturk, N. (1995). Stability Independent of distributed Lag for A Special Class of Distributed Parameter Systems. In Proceedings of the 34th IEEE Conference on Decision and Control, 13-15 December, New Orleans, LA, USA, 3245-3246).

[14] Tan, N.,Özgüven, Ö. F., & Özyetkin,M. M. (2009). Robust stability analysis of fractional order interval polynomials. *ISA Transactions*, 48(2), 166-172.

[15] Ozturk, N. and Uraz, A. (1985). An Analysis Stability Test for a Certain Class of Distributed Parameter Systems with Delays.*IEEE Transactions on Circuits and Systems*, CAS-32, 393-396.

[16] Petras, I. (2008). Stability of Fractional-Order Systems with Rational Orders.*Fractional calculus and Applied Aalysis*, 12(3), 25.

[17] Sabatier, J., Moze, M., & Farges, C. (2010). LMI stability conditions for fractional order systems. *Computers and Mathematics with Applications*, 59(5), 1594-1609.

[18] Matignon, D. (1996). Stability results for fractional differential equations with applications to control processing. *Computational engineering in systems applications*, 963-968.

[19] Matignon, D. (1998). Stability properties for generalized fractional differential systems. ESAIM: *Proceedings Fractional Differential Systems: Models, Methods and Aplicaitons*,(5), 145-158.

[20] Lazarević, M. P., & Spasić, A. M. (2009). Finite-time stability analysis of fractional order time-delay systems: Gronwall's approach. *Mathematical and Computer Modelling*, 49(3-4), 475–481.

[21] Lazarević, M. P., & Debeljković, D. L. (2008). Finite time stability analysis of linear autonomous fractional order systems with delayed state. *Asian Journal of Control*, 7(4), 440-447.

[22] Hwang, C., & Cheng, Y. C. (2006). A numerical algorithm for stability testing of fractional delay systems. *Automatica*, 42(5), 825-831.

[23] Oustaloup, A., Melchior, P., Lanusse, P., Dancla, F., & Crone, E. (2000). The CRONE toolbox for Matlab.*In Proceedings of the 2000 IEEE International Symposium on Computer-Aided Control System Design.* Anchorage, Alaska, USA September 25-27, 2000,190-195.

[24] Melchior, P., Orsoni, B., Oustaloup, A., Cnrs, U. M. R., & Bordeaux, U. (2001). The CRONE toolbox for Matlab: Fractional Path Planning Design in Robotics.*In Proceedings*

[25] Costa, D. V. and J. S. da. (2004). NINTEGER: A Non-Integer Control Toolbox for MATLAB. In Proceedings of 1st IFAC Workshop on Fractional Differentation and Its Applications, Bordeaux, France, 1-6.

[26] Tepljakov, A., Petlenkov, E., & Belikov, J. (2011). FOMCON: Fractional-order modeling and control toolbox for MATLAB.*Proceedings of the 18th International Conference Mixed Design of Integrated Circuits and Systems-MIXDES 2011*, (4), 684-689.

[27] Ozyetkin, M. M. (2018). A simple tuning method of fractional order PI^{λ}-PD^{μ}controllers for time delay systems. *ISA Transactions*, 74, 77–87.

[28] Özyetkin, M. M., Yeroğlu, C., Tan, N., & Tağluk, M. E. (2010). Design of PI and PID controllers for fractional order time delay systems. *IFAC Proceedings Volumes (IFAC-PapersOnline)*, 43, 355-360.

[29] Krishna, B. T. (2011). Studies on fractional order differentiators and integrators: A survey. *Signal Processing*, 91(3), 386-426.

[30] Chen, Y. Q., Petráš, I., & Xue, D. (2009). Fractional order control - A tutorial. *Proceedings of the American Control Conference*, 1397-1411.

[31] Silva, G. J., Datta, A., & Bhattacharyya, S. P. (2005). *PID Controllers for Time-Delay Systems.* Birkhauser Boston.

[32] Walton, K., & Marshall, J. E. (1987). Direct method for TDS stability analysis.*IEE Proceedings D Control Theory and Applications*, 134(2), 101-107.

[33] Ozturk, N. (1988). Stability Intervals for Delay Systems.*In Proceedings of the 27th Conference on Decision and Control*, Austin, Texas, 2215-2216.

Münevver Mine Özyetkin *received the Ph.D. degree in Electrical and Electronics Engineering Department from Inonu University. She was awarded a grant by TUBITAK (The Scientific and Technological Research Council of Turkey) to study insulin control for diabetic patients (artificial pancreas) between 2010 and 2011, at Clemson University, USA. She is interested in fractional order control systems, design of fractional order controllers, robust control,and stability analysis. She is currently an Assistant Prof. at Aydın Adnan Menderes University.*

(iD) *http://orcid.org/0000-0002-3819-5240*

Dumitru Baleanu*'s research interests include fractional dynamics and its applications, fractional differential equations, discrete mathematics, dynamic systems on time scales, the wavelet method and its applications, quantization of the systems with constraints, Hamilton-Jacobi formalism, geometries admitting generic and non-generic symmetries. He has published more than 600 papers indexed in SCI. He is one of the editors of 5 books published by Springer. Dumitru is an editorial board member of the following journals indexed in SCI: Abstract and Applied Analysis, Central European Journal of Physics, Advances in Difference Equations, Scientific Research and Essays (SRE) and Fractional Calculus and Applied Analysis. He is also an editorial board member of 12 different journals which are not indexed in SCI. He is currently a faculty member of Department of Mathematics and Computer Sciences at Cankaya University, Ankara, Turkey. Besides, he is a Professor at Institute of Space Sciences, Magurele-Bucharest, Romania*

http://orcid.org/0000-0002-0286-7244

An International Journal of Optimization and Control: Theories & Applications (http://ijocta.balikesir.edu.tr)

RESEARCH ARTICLE

Maximum cut problem: new models

Hakan Kutucu[a]* and Firdovsi Sharifov[b]

[a] *Department of Computer Engineering, Karabuk University, Turkey*
[b] *V.M. Glushkov Institute of Cybernetics, Ukraine*
hakankutucu@karabuk.edu.tr, f-sharifov@yandex.ru

ARTICLE INFO	ABSTRACT
	The maximum cut problem is known to be NP-hard, and consists in determining a partition of the vertices of a given graph such that the sum of the weights of the edges having one end node in each set is maximum. In this paper, we formulate the maximum cut problem as a maximization of a simple non-smooth convex function over the convex hull of bases of the polymatroid associated with a submodular function defined on the subsets of vertices of a given graph. In this way, we show that a greedy-like algorithm with $O(mn^2)$ time complexity finds a base of a polymatroid that is a solution to the maximum cut problem with different approximation ratio. Moreover, with respect to a base of a polymatroid, we formulate the maximum cut problem as a maximum flow problem between a source and a sink. We then investigate the necessary and sufficient conditions on the optimality of the base in terms of network flow.

1. Introduction

The well-known maximum cut problem consists in determining a partition of the vertices of a given graph such that the sum of the weights of the edges having one end node in each set is maximum. The maximum cut problem is very easy to state but hard to solve. This problem is one of the first problems whose NP-hardness was established in [1] by Karp. Note that the problem remains NP-hard even for unit edge weights [2,3].

The solution of the maximum cut problem has been approached by mathematical programming. In terms of design variables for every vertex, an integer quadratic programming formulation is given in [4]. Further integer linear programming formulations of the maximum cut problem using the boolean design variables are given in [5]. The algorithm in [6] for finding solutions of the maximum cut problem is an efficient method from a practical point of view. Goemans and Williamson use a semidefinite relaxation technique. Their experiments show that exact solutions are obtained

in a reasonable time for any maximum cut instance of size up to 100 vertices. Using a semidefinite relaxation, the authors achieve an approximation ratio of 0.87856 for this difficult combinatorial optimization problem. Semidefinite programming is a convex optimization approach with a linear objective function of the design variables for a symmetric matrix, subject to linear constraints, and also convex constraints requiring the matrices to be positive semidefinite. Despite the fact that the algorithm in [6] has one of the best worst-case performance, Bertoni, Campadelli and Grossi [7] show that the algorithm improved by Goemans and Williamson has a complex design and its computation time may be prohibitive on large problem instances having more than 500 vertices. By solving experimental test problems on large random graphs, Bertoni et al. also show that their algorithm is better than the semidefinite programming algorithm of [6] and they define cuts with the same values in less time on standard benchmarks. Ben-Ameur et al. discuss the complexity of the maximum cut problem and some

---

*Corresponding Author

cases where the problem can be solved in polynomial time [8]. They also introduce some approximation methods for the maximum cut problem, both with and without guarantees.

In all references mentioned, the topological properties of a given graph did not play an essential role in proofs or in solving the maximum cut problem. Differently from the investigations mentioned above, to solve the maximum cut problem, some polynomial time algorithms have been developed based on topological properties of given graphs such as planar graphs [9,10], weakly bipartite graphs with non-negative edge weights [11], graphs without K_5 minors [12]. The problem is solved using a linear time algorithm for series-parallel graph [13].

For definitions used in the paper, we refer readers to [14,15]. Following the success of the theory of polymatroids in solving difficult combinatorial problems, we apply a polymatroid approach to the maximum cut problem.

Section 2 contains necessary notations and definitions in the theory of polymatroids used throughout the paper. In Section 3, we present the maximum cut problem as a maximization of a simple non-smooth function over a special polytope $P(f)$ called a polymatroid [14,16] associated with the submodular function $f(S)$ defined on subset S of V of a given graph $G = (V, E)$. This model includes variables for each node in V. The convexity of the objective function implies that an optimal solution to the maximum cut problem is among extreme points (bases) of the polytope (polymatroid) $P(f)$ [17].

It is well known that the greedy algorithm defines bases of $P(f)$ according to different linear ordering of vertices, in polynomial time (see [14,16]). One might say that for each maximum cut problem, *an optimal* linear ordering of vertices has to be chosen such that an optimal base of $P(f)$ (an optimal solution) can be defined by the greedy algorithm in polynomial time. Hardness of the maximum cut problem implies that an optimal linear ordering cannot be defined in polynomial time. In [18], Sharifov proposes a $O(mn^2)$ time algorithm which defines different linear ordering and related bases of $P(f)$ based on the topological properties of a given graph. We show that a solution to the maximum cut problem with the approximation ratio $0,75\lambda$ can be defined in $O(mn^2)$ time by this algorithm, where $\lambda \leq 1.3$ is some positive number and $m = |E|, n = |V|$. In Section 4, we present a new model of the maximum cut problem in terms of flows with respect to a base of $P(f)$. This model is used in the proof of new

necessary and sufficient conditions for optimality of a base of $P(f)$.

2. Basic notions and preliminary results

Consider an undirected graph $G = (V, E)$ with non-negative weights $c_e \geq 0$ on the edges $e \in E$. We assume that G is a graph without loops and parallel edges. An edge with endpoints v and u is denoted by (v, u) and uv denotes the arc whose tail is v, and head is u. We use $\overline{S} = V \setminus S$ for $S \subseteq V$ and $S + v$ for $S \cup \{v\}$ when $v \notin S$, and $S - v$ for $S \setminus \{v\}$ when $v \in S$.

Let $\gamma(S)$ and $\kappa(S)$ denote the subsets of edges having at least one of endpoints in $S \subseteq V$ and both endpoints in $S \subseteq V$, respectively. Consider functions

$$f(S) = \sum(c_e : e \in \gamma(S))$$
$$g(S) = \sum(c_e : e \in \kappa(S)).$$

Obviously $f(S)$ and $g(S)$ are monotone functions by definitions, moreover, it is well known that f is submodular, i.e.,

$$f(S) + f(T) \geq f(S \cup T) + f(S \cap T)$$

and g is supermodular, i.e.,

$$g(S) + g(T) \leq g(S \cup T) + g(S \cap T)$$

for any $S, T \subseteq V$ [16].

The cut given by a subset $S \subset V$ is denoted by $\delta(S)$. We will use $c(E)$ for $\sum_{e \in E} c_e$, and $c(\delta(S))$ for $\sum(c_{ij}; (i, j) \in E, i \in S, j \in \overline{S})$. The vector $d = (d_v = c(\delta(v)); v \in V)$ is called the *weighted degree vector* of the graph G. From the definition of the sets $\gamma(S)$ and $\kappa(S)$ it follows that

$$f(S) + g(S) = d(S) = \sum_{v \in S} d_v,$$

and

$$f(S) - g(S) = c(\delta(S))$$

for the cut $\delta(S)$ given by any $S \subset V$. Clearly, $f(S) - g(S)$ is a submodular function.

Let $\mathbb{R}^V$ denote the set $\{(u(v) \in \mathbb{R} : v \in V)\}$. For $u = (u(v) : v \in V) \in \mathbb{R}^V$ and a subset $S \subseteq V$, we denote $u(S) = \sum_{v \in S} u_v$. The following two sets of vectors in $\mathbb{R}^V$ associated with the functions f

and g are called polymatroid and superpolymatroid [14], respectively:

$$P(f) = \{x \in \mathbb{R}^V;\ x(S) \le f(S),\ S \subseteq V\},$$
$$Q(g) = \{y \in \mathbb{R}^V;\ y(S) \ge g(S),\ S \subseteq V\}.$$

The following polytope associated with the function $f - g$ is called extended polymatroid [14]:

$$EP(f-g) = \{w \in \mathbb{R}^V;\ w(S) \le f(S)-g(S),\ S \subseteq V\}.$$

Vectors $x \in P(f)$ and $y \in Q(g)$ are called bases of the polymatroid and the superpolymatroid if $x(V) = f(V)$ and $y(V) = g(V)$, respectively. Note that, for any bases $x \in P(f)$ and $y \in Q(g)$,

$$x(V) - y(V) = f(V) - g(V) = 0,$$

since

$$\gamma(V) = E = \kappa(V)$$

by definition of the sets $\gamma(S)$ and $\kappa(S)$. So, a vector $w \in EP(f - g)$ is a base of $EP(f - g)$ if $w(V) = 0$.

Let $x^L \in P(f)$ and $y^L \in Q(g)$ be bases computed by the greedy algorithm in [18] with respect to any linear ordering L of the vertices. The first observation is that the difference $w^L = x^L - y^L$ of the bases x^L and y^L is a base of $EP(f - g)$ which can also be found by the greedy algorithm with respect to the linear ordering L of the vertices. In what follows, we will write x, y and w instead of x^L, y^L and w^L, respectively.

We write $v \prec_L u$ if v precedes u in the linear ordering L of the vertices. According to the linear ordering L of vertices, one can orient the edges of the graph $G = (V, E)$ in such a way that the resulting digraph $G = (V, A)$ is an acyclic oriented graph. This requires each edge (v, u) to be replaced by an arc vu if $v \prec_L u$ or an arc uv if $u \prec_L v$. The opposite is also true; each acyclic orientation of the edges of the graph $G = (V, E)$ defines a linear ordering L of its vertices. In an acyclic oriented graph $G = (V, A)$ with weights c_{vw} on arcs, let $\delta_+(v)$ be the set of arcs entering to node v, and let $\delta_-(v)$ be the set of arcs leaving from node v.

Our key observation is that the bases $x \in P(f)$ and $y \in Q(g)$ satisfy the equalities

$$\sum_{u \in \delta_+(v)} c_{vu} = c(\delta_+(v_i)) = x_v,\ v \in V, \qquad (1)$$

$$\sum_{u \in \delta_-(v)} c_{uv} = c(\delta_-(v_i)) = y_v,\ v \in V. \qquad (2)$$

In other words, x_v is the sum of weights on the leaving arcs from the node v, and y_v is the sum of weights on the entering arc to the node v.

All the above equalities are satisfied by any bases of $x \in P(f)$ and $y \in Q(g)$ which are computed with respect to any linear ordering of the vertices in any graph. Their proof immediately follows from the greedy algorithm formula for computing bases of $P(f)$ and $Q(g)$ with respect to a given linear ordering of the vertices. So, we can state the following claims.

Claim 1. *Let $x \in P(f)$ and $y \in Q(g)$ be any bases computed by the greedy algorithm developed in [18] with respect to any linear ordering L of the vertices, then*

$$x + y = d$$

and the difference $x - y = w$ is a base of $EP(f-g)$, for which the following the zero sum equality

$$\sum(w_v;\ w_v > 0) = -\sum(w_v;\ w_v <= 0)$$

holds.

Proof. Since $d_v = c(\delta_+(v_i)) + c(\delta_-(v_i))$ in the graph, obtained by orientation of the edges $G = (V, E)$ according to the linear ordering L, then $x_v + y_v = d_v$ for any node $v \in V$. From $x(V) = y(V)$ it follows that $w(V) = x(V) - y(V) = 0$. Since $x \in P(f)$ and $y \in Q(f)$, $x(S) - y(S) \in EP(f - g)$. Besides, $w(V) = 0$ can be written as the zero sum equality by performing algebraic operations. Therefore, $x - y = w$ is a base of $EP(f - g)$. $\qquad \square$

Claim 2. *For a given linear ordering $L = \{v_1, ... v_n\}$ of the vertices in V, the bases $x(L) \in P(f)$, $y(L) \in Q(g)$ and $w(L) \in EP(f - g)$ can be found in $O(m)$ time.*

Proof. For $L_i = \{v_1, ... v_i\}$ and $i = 1, ..., n$, by the greedy formula

$$x_{v_i} = f(L_i) - f(L_{i-1}) = c(\delta_+(v_i)),$$

it follows that x_{v_i} is the sum of weights on the edges $(v_i, v_j) \in E$ for which $v_i \prec_L v_j$ in L. So, each edge of E appears only once in computing the base x. From $y = d - x \in Q(g)$ and $w = x - y \in EP(f - g)$, we obtain the bases y

and w in $O(m)$ time with respect to the linear order L. $\square$

We make the following useful observation whose proof immediately follows from equalities (1) and (2), for two graphs obtained after the orientation of the edges in G with respect to linear orderings $L = \{v_1, v_2, ..., v_n\}$ and $I = \{v_n, ..., v_2, v_1\}$.

Claim 3. *If the greedy algorithm defines the bases $x^1 \in P(f)$ and $x^2 \in P(f)$ with respect to the L and I, respectively, then $x^2 = y^1 = d - x^1 \in Q(g)$ and $x^1 = y^2 = d - x^2 \in Q(g)$.*

Proof. Consider two acyclic oriented graphs $G(L)$ and $G(I)$ obtained after replacing edges in E by arc according to L and I, as above. If $v \prec_L u$, then an edge (v, u) corresponds to arc vu in $G(L)$ and arc uv in $G(I)$. Let x_v^1 and $y_v^1 = d_v - x_v^1$ be bases defined by the greedy algorithm [18] with respect to L, that is, equalities (1) and (2) hold for x_v^1 and y_v^1 with respect to $G(L)$. Let x_v^2 and $y_v^2 = d_v - x_v^1$ be bases defined by the greedy algorithm with respect to I. Since an edge (v, u) corresponds to arc vu in $G(L)$ and arc uv in $G(I)$, then x^1 defined by (1) for $G(L)$ is y^2 for $G(I)$ and y^1 defined by (2) for $G(L)$ is x^2 for $G(I)$. By Claim 1, $x^1, x^2 \in Q(g)$. $\square$

The greedy algorithm [18] defines the following bases with respect to the linear order $L = (W, U)$ for a bipartite graph $H = (W, U, A)$:

$$x_v = d_v^H, \text{ for } v \in W, \ x_u = 0, \text{ for } u \in U,$$
$$y_v = 0, \text{ for } v \in W, \ y_u = d_u^H, \text{ for } u \in U.$$

Therefore,

$$\sum_{v \in W \cup U} |x_v - y_v| \tag{3}$$

is equal to the double weight of the maximum cut separating the sets W and U, in the bipartite graph $H = (W, U, A)$. To the best of our knowledge, there is no an algorithm to define a linear ordering L of vertices for non-bipartite graph in order to determine a maximum cut. Our goal in the next sections is to develop some new ideas for finding maximum cut in a non-bipartite graph $G = (V, E)$.

3. Models with convex objective function

The maximum cut problem of a graph $G = (V, E)$ is to find the set of vertices S that maximizes the weight of the edges in the cut $(S, \overline{S})$, i.e., the weight of the edges with one end node in S and the other in $\overline{S}$. In this section, we propose an alternative formulation for the maximum cut problem. First, the relationship between cuts and bases of the polymatroids $P(f)$ is established. In a linear ordering L of vertices, if $v \prec_L u$ for any vertices v and u such that $v \in S \subset V$ and $u \in \overline{S}$, we write it as $L = (S, \overline{S})$ for short.

Theorem 1. *The double weight of a cut $\delta(S)$ separating sets S and $\overline{S}$ is equal to*

$$\sum_{v \in V} |x_v^L - y_v^L|, \tag{4}$$

where bases $x^L \in P(f)$ and $y^L \in Q(g)$ are computed by the greedy algorithm with respect to the linear ordering $L = (S, \overline{S})$ of vertices in V.

Proof. Consider the linear ordering $L = (S, \overline{S})$ of the vertices in V, that is, $v \prec_L u$ for any node $v \in S$ and $u \in \overline{S}$, and let $x^L \in P(f)$ and $y^L = d - x^L$. According to the linear ordering L, we can direct each edge (v, u) of the graph as arc vu, if $v \prec_L u$ or as uv if $u \prec_L v$. Then all edges in E will be directed as arcs $v_1 v_2$, if $v_1 \prec_L v_2$ for vertices $v_1, v_2 \in S$, as arcs $u_1 u_2$ if $u_1 \prec_L u_2$ for vertices $u_1, u_2 \in \overline{S}$ and as arcs vu, where $v \in S$ and $u \in \overline{S}$. Clearly, after deleting the arcs $v_1 v_2$ with end nodes $v_1, v_2 \in S$ and the arcs $u_1 u_2$ with end nodes $u_1, u_2 \in \overline{S}$, in G, the resulting subgraph is a bipartite subgraph $H = (S, \overline{S}, A)$ ($S \subset V$ and $A \subseteq E$). With respect to H, one can define the functions f_0 and g_0, also the matroids $P(f_0)$, $Q(g_0)$ and $E(f_0 - g_0)$. Consider the bases $h \in P(f_0)$ and $t \in Q(g_0)$ defined by the greedy algorithm with respect to L, i.e.,

$$h_v = d_v^H, \text{ for } v \in S, \ h_u = 0, \text{ for } u \in \overline{S},$$
$$t_v = 0, \text{ for } v \in S, \ t_u = d_u^H, \text{ for } u \in \overline{S}.$$

Hence, $t_v = d_v^H - h_v$ for any node $v \in V$, and

$$2 \sum_{(u,v) \in A} c_{uv} = \sum_{v \in S} d_v^H + \sum_{v \in \overline{S}} d_v^H =$$
$$= h(S) - t(S) + |h(\overline{S}) - t(\overline{S})|.$$

Since h is defined with respect to L and $t_v = d_v^H - h_v$, then

$$h(T) - t(T) = x^L(T) - y^L(T), \text{ for } T = S, \overline{S}.$$

Therefore,

$$2 \sum_{(u,v) \in A} c_{uv} = x^L(S) - y^L(S) + |x^L(\overline{S}) - y^L(\overline{S})|$$

$$= \sum_{v \in V} |x_v^L - y_v^L|.$$

Thus, H is a maximum bipartite subgraph (the sum of weights on its edges is maximum) of G, if the last sum is maximum for the bases x^L and y^L. $\qquad \square$

Since f is a monotone submodular function, the convex hull of bases $P(f)$ is the following polytope [14]

$$B(f) = \{x; x \geq 0, x(S) \leq f(S), S \subset V, x(V) = f(V)\}$$

We recall that $x + y = d$ (Claim 1) for the bases x and y generated by the greedy algorithm with respect to any linear ordering L of vertices. Thus, $y_v = d_v - x_v$ for all $v \in V$, and hence

$$|x_v - y_v| = |2x_v - d_v|.$$

We now present our original formulation of the problem. By Theorem 1, the maximum cut problem can be formulated as the following special convex program;

$$MaxCut_* = \max\{Cut(x) = \sum_{v \in V} |2x_v - d_v|\} \quad (5)$$

subject to

$$x \in B(f). \qquad (6)$$

In what follows, we propose further formulations for the max-cut problem. To this end, we first state the following lemma. Let

$$f_+(x) = \sum_{v \in V_+(x)} (2x_v - d_v),$$

$$f_-(x) = \sum_{v \in V_-(x)} (d_v - 2x_v),$$

where

$$V_+(x) = \{v \in V; 2x_v - d_v > 0\},$$
$$V_-(x) = \{v \in V; 2x_v - d_v \leq 0\},$$

for any base $x \in B(f)$.

Lemma 1. *For any base* $x \in B(f)$
$$Cut(x) = 2f_+(x) = 2f_-(x).$$

Proof. From the equality

$$f_+(x) - f_-(x) = \sum_{v \in V} (2x_v - d_v) = 0$$

it follows that

$$Cut(x) = f_+(x) + f_-(x) + 0 = 2f_+(x),$$
$$Cut(x) = f_+(x) + f_-(x) - 0 = 2f_-(x).$$
$$\qquad \square$$

By Lemma 1,

$$MaxCut_* = f_+(x^*) + f_-(x^*) = 2f_+(x^*) = 2f_-(x^*),$$

where x^* is an optimal solution to the problem (5)-(6). Since $z = 2x - d = x - y$ for any $z \in EP(f - g)$ and $x \in B(f)$ by Claims 1-3, the vector $z^+ = \{z_v^+; v \in V\}$, where $z_v^+ = \max\{z_v, 0\}$ can be defined with respect to each base $x \in B(f)$. By the equality $Cut(x) = 2f_+(x)$, the following problem

$$\max\{z^+(V); z \in EP(f - g)\}$$

is equivalent to the maximum cut problem (5)-(6). In addition to the above models, by the equality $Cut(x) = f_+(x) + f_-(x)$, the problem

$$\max\{f_+(x) : x \in B(f)\} = c(E)$$
$$- \min\{x(V_-) + y(V_+), x \in B(f), y = d - x\} \quad (7)$$

is also equivalent to the maximum cut problem (5)-(6). The problem in the right hand side of equality (7) can be considered as *dual* of the problem (5)-(6). We note that $z = 2x - d = x - y$ for $x \in B(f)$ and $y = d - x$. We can also define the vector $z^- = \{z_v^-; v \in V\}$, where $z_v^- = \min\{z_v, 0\}$ for any $v \in V$. It is easy to show that the following problem

$$\min\{z^-(V); z \in EP(f - g)\},$$

is equivalent to the above dual problem (7). Thus, the latter problem can be considered as another dual problem of the problem (5)-(6).

Moreover, Lemma 1 says that to solve the maximum cut problem on a given undirected graph, one can find a base $z \in EP(f - g)\}$ for which either $z^+(V)$ is maximum or $z^-(V)$ is minimum. It is well known that the latter problem is used essentially to design polynomial algorithms for minimizing a submodular function. For more details, the reader can refer to [16].

The above models are also useful for solving the maximum cut problem. For example, since the algorithm in [18] defines $O(n)$ bases of $B(f)$ in $O(mn^2)$ time, we can apply it to handle different bases and related strings $Cut(x), V_+(x), V_-(x)$. Let $LIST(Cuts)$ contains strings $Cut(x), V_+(x), V_-(x)$ for each these bases.

Theorem 2. *In $LIST(Cuts)$, if there are strings for some pair of different bases $x^*, x \in B(f)$ such that*

$$Cut(x^*) \geq x(V_+) + y(V_-),$$

for $y = d - x$, $V_+ = V_+(x)$ and $V_- = V_-(x)$, then

$$\frac{MaxCut_*}{2} \geq \frac{3}{4}c(E).$$

Proof. If $LIST(Cuts)$ contains the string for base x, then Claim 3 implies that $x(V_+) = f(V_+)$ and $y(V_-) = f(V_-)$. Let $LIST(Cuts)$ contains the string for the base x^*, too. Then

$$Cut(x^*) \geq f(V_+) + f(V_-)$$
$$= f(V_+) + g(V_-) + \frac{Cut(x)}{2} = c(E) + \frac{Cut(x)}{2}.$$

To define $Cut(x)$, the algorithm in [18] chooses a node $w \neq s$ for which $2x_w - d_w \geq 2x_v - d_v > 0$ for $v \notin V_+$, and sets $V_+ := V_+ + w$, where $V_+ := s$ at the beginning of the algorithm, and $V_- = V \setminus V_+$. This implies that $Cut(x)/2 \geq c(E)/2$. Thus,

$$Cut(x^*) \geq c(E) + \frac{c(E)}{2} = \frac{3}{2}c(E),$$

which completes the proof of the theorem. $\square$

In $LIST(Cuts)$, let
$Cut(x^*) = \max\{Cut(x); Cut(x) \in LIST(Cuts)\}$,
and let $x(V_+) + y(V_-)$ be minimum for a base x ($y = d - x$). In other words, Theorem 2 states that $Cut(x^*)$ is a solution to the maximum cut problem with the approximation ratio at least 0.75. In this case, clearly the graph G has a cut with value at least $3/4c(E)$. If the graph G does not have a cut with value $3/4c(E)$, then Theorem 2 is not true. In this case, we define λ from the equality

$$Cut(x^*) = \lambda(x(V_+) + y(V_-)).$$

Clearly, λ is a positive number and $\lambda \leq 1.3$. By the proof of Theorem 2, it can be shown that

$$\frac{MaxCut_*}{2} \geq \lambda \frac{3}{4}c(E).$$

So, the graph G has a cut with value at least $3/4\lambda c(E)$. In this case, the algorithm in [18] defines a solution to the maximum cut with the approximation ratio at least 0.75λ for some positive number $\lambda < 1$.

As a conclusion of this section, we note that simplicity of the algorithm in [18] allows to solve real practical large problems effectively by Theorem 2. In future, we plan to do some investigations in this direction.

4. Maximum flow model

Now, we formulate the maximum cut problem by another model. Let $x \in B(f)$ be a base generated by the greedy algorithm with respect to a linear ordering L of vertices in V and let $z = 2x - d = x - y \in EP(f - g)$. Since $z(V) = 0$, we can define subsets

$$V_+ = \{v; z_v = 2x_v - d_v > 0, v \in V\}$$

and

$$V_- = \{w; z_w = 2x_w - d_w < 0, w \in V\}.$$

We consider an acyclic oriented graph $G = (V, A)$ obtained after replacing all edges by arcs according to the linear ordering of L. The capacity on each arc vu equals to the given weight of the edge $(v, u) \in E$. We add two new vertices, a source s and a sink r, to the graph $G = (V, E)$. For each vertex $v \in V_+$ and $w \in V_-$, we add arcs sv and wr with capacity z_v and $|z_w|$ to the graph $G = (V, E)$, respectively. In the resulting network $G_z = (V_z, E_z)$, let $\delta_+(S)$ denote the set of entering arcs to the vertices $S \subset V_z$ and $\delta_-(S)$ denote the set of leaving arcs from the vertices of the subset S. Recall that the capacity of the cut separating a subset of S is defined as the sum of the flows on the leaving arcs entering to vertices $v \in S$ minus the sums of the flows on the entering arcs to vertex $v \in S$. A cut with a minimum capacity is called a minimum cut.

Theorem 3. *In the network $G_z = (V_z, E_z)$, any maximum $s - r$ flow (source s, and sink r) saturates all arcs, i.e., on all arcs vu with end node $v, u \in V$, the value of the maximum flow equals to c_{vu}, and on all arcs sv and rw, the value of the maximum flow equals to z_v and $|z_w|$, respectively.*

Proof. Let x be a base generated by the linear ordering of L and $y = d - x$. From the definitions of the capacity of arcs sv and rw, it follows that $c(\delta_+(v)) = c(\delta_-(v))$ in the network G_z. This means that the sums of capacities of arcs in the

sets $\delta_+(v)$ and $\delta_-(v)$ are the same for each vertex $v \in V$. Therefore, to maintain a balance between leaving and entering flows for each vertex $v \in V$, the value of the maximum flow on arcs of the acyclic oriented graph $G = (V, A)$ must be equal to its capacity. In addition, since

$$z(\delta_+(s)) = x(V_+) - y(V_+) = y(V_-) - x(V_-)$$
$$= |z(\delta_-(r))|$$

it follows that z_v and $|z_w|$ are the maximum flow values on the arcs sv and rw, respectively. □

By Theorem 3, since the value of the flow on all arcs is equal to its capacity, that is, any cut separating the source s and the sink r are the minimum cut in the network $G_z = (V_z, E_z)$. Thus, we obtain that $f_+(x)$ is the value of the maximum flow from the source s to the sink r in the constructed network $G_z = (V_z, E_z)$ according to the base $z = 2x - d \in EP(f - g)$.

So, Theorem 3 implies that to solve the maximum cut problem on a given undirected graph $G = (V, E)$, it needs to find a base $z \in EP(f - g)$ such that the capacity of any minimum cut separating the source and sink is maximum in the constructed network $G_z = (V_z, E_z)$. Such a model of the maximum cut problem can have applications for transportation of natural products from time to time in different directions through pipelines of the transport network.

Definition 1. *Let $G_z = (V_z, E_z)$ be a network constructed for the base $z = 2x - d \in EP(f - g)$. The flow on each arc vw is called transit, if vertices v, w are either in $V_+(x)$ or in $V_-(x)$.*

Theorem 4. *A base $z = 2x - d \in EP(f - g)$ is an optimal solution to the problem (5)–(6) if and only if a maximum flow from source to sink has a minimum sum of transit flows in the network $G_z = (V_z, E_z)$ constructed for the base z.*

Proof. Let x be the base generated by the linear ordering of L and the network G_z constructed for $z = 2x - d$ contains the minimum sum of transit flows on the arcs vw. By definition of a cut in the graph G with respect to the bases x and $y = d - x$, if $v, w \in V_+(x)$, then $y(V_+(x))$ if $v, w \in V_+(x)$, then $x(V_-(x))$ are the total number of transit flows on the arcs vw. Therefore, from the dual equality (7), we obtain that $\delta(V_+(x))$ is a maximal cut in the graph G.

If we consider that $y(V_+(x)) + x(V_-(x))$ is the sum of the transit flows in the network G_z constructed for arbitrary bases $z = 2x - d$ and $x \in B(f)$, then

the opposite also follows from the dual equality (7). □

In other words, Theorem 4 states that if the minimum number of variables satisfies inequalities $0 < x_v < d_v$ in solving the problem (5)-(6), then a definite cut for x_v is maximal in the graph G. It is relatively difficult to design an effective algorithm based on this theorem. At a first glance, one might think that the network $G_z = (V_z, E_z)$ should not contain much more transit flows if $x(V_+(x)) = y(V_-(x))$. However, the situation is very complicated, since it is easy to design some small maximum cut problems for which this is not true. Indeed, this theorem states some connection between the maximum independent set and the maximum cut problems, that require new investigations on network flow problems.

5. Concluding remarks

The value of applications of the theory of polymatroids ensures that the optimal solution of many combinatorial optimization problems can be found in polynomial time bounded algorithms. For example, the vector $z \in P(f)$ maximizes cz in polynomial time for the monotonic submodular function f. A deep understanding of this theory makes it possible to use known methods developed for solving the network flows, as a solver of subtasks enumerating in solving optimization problems with a nonlinear objective function over polymatroids structures (see [14]). Considering topological properties of graphs under consideration in solving combinatorial problems over polymatroids leads to a polynomial algorithm as a solver for the maximum cut problem. In [10], topological properties of planar graphs namely the geometric duality is used to develop a polynomial time bounded for finding a maximal cut of these graphs. Since we do not know about unambiguous connections between NP and P, it is difficult to come up with a polynomial time bounded algorithm for solving (5)-(6), only using the above described and other specifics of the problem. But, with respect to the specifics of the objective function (5) and constraints (6), we hope that the next two weaker questions can be solved by a polynomial time algorithm.

(1) Is it possible to design greedy type algorithm by using the subgradient of the objective function at a current point (base) x to compute the next point x^k such that (5) will be strongly increased?

(2) How topological properties of a given graph and the techniques described in the

paper could be combined for finding an exact upper bound of the objective function (5)?

Based on positive answers to these two questions a polynomial time algorithm can be developed for finding an optimal solution to (5)-(6) on graphs with unit edge weights and as a result, we could get $NP = P$.

Acknowledgments

References

[1] Karp, R.M. (1972). Reducibility among combinatorial problems. In: R.E. Miller and J.W. Thatcher, eds. Complexity of Computer Computations. Plenum Press, 85-103.

[2] Garey, M.R., Johnson, D.S., & Stockmeyer, L. (1976). Some simplified NP-complete graph problems. Theoretical Computer Science, 1(3), 237-267.

[3] Garey, M.R., & Johnson, D.S. (1979). Computers and Intractability: A Guide to the Theory of NP-Completeness. W.H. Freeman and Company.

[4] Rendl, F., Rinaldi, G., & Wiegele, A. (2010). Solving MAX-CUT to optimality by intersecting semidefinite and polyhedral relaxations. Mathematical Programming, 121(2), 307-335.

[5] Boros, E., & Hammer, P.L. (2002). Pseudo-Boolean optimization. Discrete Applied Mathematics, 123(1), 155-225.

[6] Goemans, M., & Williamson, D.P. (1995). Improved approximation algorithms for MAX-CUT and satisfiability problems using semidefinite programming. Journal of the ACM, 42(6), 1115-1145.

[7] Bertoni, A., Campadelli, P. & Grossi, G. (2001). An approximation algorithm for the maximum cut problem and its experimental analysis. Discrete Applied Mathematics, 110(1), 3-12.

[8] Ben-Ameur, W., Mahjoub, A.R., & Neto, J. (2014). The maximum cut problem. In: V.T. Paschos, ed. Paradigms of Combinatorial Optimization: Problems and New Approaches, 2nd Edition, J. Wiley and Sons, 131-172.

[9] Orlova, G.I., & Dorfman, Y.G. (1972). Finding the maximum cut in a graph. Engineering Cybernetics, 10(3), 502-506.

[10] Hadlock, F.O. (1975). Finding a maximum cut of a planar graph in polynomial time. SIAM Journal on Computing, 4(3), 221-225.

[11] Grötschel, M., & Pulleyblank, W.R. (1981). Weakly bipartite graphs and the max-cut problem. Operations Research Letters, 1(1), 23-27.

[12] Barahona, F. (1983). The max-cut problem in graphs is not contractible to K5. Operations Research Letters, 2, 107-111.

[13] Chaourar, B. (2017). A Linear Time Algorithm for a Variant of the MAX CUT Problem in Series Parallel Graphs. Advances in Operations Research, 2017, 1-4.

[14] Fujishige, S. (2005). Submodular Function and Optimization. Annals of Discrete Mathematics, 2nd ed. Elsevier Science, Amsterdam.

[15] Nemhauser, G. & Wolsey, L.A. (1998). Combinatorial Optimization. Wiley-Interscience, New York.

[16] Iwata, S. (2008). Submodular function minimization. Mathematical Programming, 112(1), 45-64.

[17] Bazaraa, M.S., Sherali, H.D., & Shetty, C.M. (2006). Nonlinear programming: Theory and Algorithms. 3rd ed. John Wiley and Sons, New York.

[18] Sharifov, F.A. (2018). Finding the maximum cut by the greedy algorithm. Cybernetics and Systems Analysis, 54(5), 737-743.

Hakan Kutucu *received one of his master degree from International Computer Institute in 2004 other from Department of Mathematics at Ege University in 2008. He completed doctoral programme of Department of Mathematics at Ege University in 2011. At the present time, he has focused on network design problems, combinatorial optimization and mathematical modeling. He has been working in the Department of Computer Engineering at Karabuk University in Turkey since 2013.*
ⓘ *http://orcid.org/0000-0001-7144-7246*

Firdovsi Sharifov *is senior researcher at the Department of combinatorial optimization methods and intellect information technologies at the Glushkov institute of Cybernetics of the National Academy of Sciences of Ukraine in Kyiv. He received his Ph.D. and Doctor of Science degrees in Combinatorial optimization in this institute. He was a team leader for grants projects with a joint professor at the National Aviation University. His research interests include combinatorial optimization, computer science and operation research.*
ⓘ *http://orcid.org/0000-0001-8768-3649*

RESEARCH ARTICLE

The complex Ginzburg Landau equation in Kerr and parabolic law media

Esma Ates[*]

Department of Electronics and Communication Engineering, Karadeniz Technical University, Turkey
esmaates@ktu.edu.tr

ARTICLE INFO	ABSTRACT
	This paper study the complex Ginzburg-Landau equation with two different forms of nonlinearity. The Jacobi elliptic ansatz method is used to obtain the optical soliton solutions of this equation in the kerr and parabolic law media. Bright and dark optical soliton solutions are acquired as well as Jacobi elliptic function solutions. The existence criteria of these solutions are also indicated.

1. Introduction

In recent years, studies conducted on findings new analytical solutions of differential equations have attracted attention of scientists from all over the world [1–21]. Especially the dynamics of optical soliton is one of the most fascinating areas of research in the field of mathematical physics. There are a great number of models that studies the dynamics of optical soliton propagation through a large variety of waveguides such as optical fibers, optical couplers, crystals, optical metamaterials and metasurfaces. The complex Ginzburg-Landau equation (CGLE) is one of these models and it is extended kind of the nonlinear Schrodinger equation that is the governing model of this context.The CGLE describes various phenomena including nonlinear optical waves, second-order phase transitions, Rayleigh–Bnard convection superconductivity, superfluidity, Bose–Einstein condensation and liquid crystals [1–4]. It is studied widely all over the world by a variety researchers [1–12]. A wealth of results have been reported in this context. Some

of the integration methods that have been implemented to this model are trial solution approach [7], modified simple equation method [8], first integral method [9], semi-inverse variational pirinciple [10] and others.The current paper will use Jacobi elliptic functions to extract cnoidal and snoidal wave solutions to the model.These will get soliton solutions in the limiting case of the modulus of ellipticity.

2. Mathematical analysis

The dimensions form of CGLE is [5]- [8]

$$iq_t + aq_{xx} + bF\left(|q|^2\right)q = \frac{1}{|q|^2 q^*}\left[\alpha |q|^2 \left(|q|^2\right)_{xx}\right.$$
$$\left. -\beta\left\{\left(|q|^2\right)_x\right\}^2\right] + \gamma q, \qquad (1)$$

where $q(x,t)$ is a complex-valued function which represents the soliton molecule in an optical fiber. The independent variables x and t show spatial and temporal coordinates, respectively. Then a and b represent coefficients of the group velocity dispersion (GVD) and nonlinearity, respectively.

---

*Corresponding Author

Also α and β are additional nonlinear terms and γ stem from the detuning effect [11].

In (1), if we think the complex plane C as a two-dimensional linear space R^2, it can be written

$$F\left(|q|^2\right) q \in \cup_{\ell,n=1}^{\infty} C^k\left((-n,n) \times (-\ell,\ell); R^2\right). \tag{2}$$

The initial hypothesis for (1) is taken by the following form:

$$q(x,t) = u(\xi) e^{i\phi(x,t)}, \tag{3}$$

In (3), u and ϕ represent amplitude and phase component of the soliton respectively and here

$$\xi = x - vt, \tag{4}$$

and

$$\phi = -\kappa x + wt + \theta \tag{5}$$

where v represents the soliton velocity, κ and w represent the frequency and wave number of the soliton respectively and θ is the phase constant.

Substituting (3) into (1) and then decomposing real and imaginary parts, the real part is given

$$(a - 4\beta) u'' - \left(w + a\kappa^2 + \gamma\right) u + F\left(u^2\right) u$$
$$= 2\left(\alpha - 2\beta\right) \frac{(u')^2}{u}. \tag{6}$$

It is also note that $u' = du/d\xi$, $u'' = d^2u/d\xi^2$ and so on. The choice

$$\alpha = 2\beta, \tag{7}$$

Eq. (1) modifies to

$$iq_t + aq_{xx} + F\left(|q|^2\right) q = \frac{\beta}{|q|^2 q^*} \left[2|q|^2 \left(|q|^2\right)_{xx}\right.$$
$$\left. - \left\{\left(|q|^2\right)_x\right\}^2\right] + \gamma q, \tag{8}$$

and the real part reduces

$$(a - 2\alpha) u'' - \left(w + a\kappa^2 + \gamma\right) u + F\left(u^2\right) u = 0, \tag{9}$$

and then imaginary part of the Eq. (1) gives the soliton velocity as:

$$v = -2a\kappa. \tag{10}$$

The velocity of the soliton, given by (10), is independent of the type of nonlinearity. So it stays the same for all forms of fiber in question.

2.1. Kerr law

In this case,

$$F(s) = bs, \tag{11}$$

where b is the real-valued constant. So, Eq. (8) reduces to

$$iq_t + aq_{xx} + \left(b|q|^2\right) q = \frac{\beta}{|q|^2 q^*} \left[2|q|^2 \left(|q|^2\right)_{xx}\right.$$

$$\left. - \left\{\left(|q|^2\right)_x\right\}^2\right] + \gamma q, \tag{12}$$

and the real part equation (9) simplifies to

$$(a - 4\beta) u'' - \left(w + a\kappa^2 + \gamma\right) u + u^3 = 0. \tag{13}$$

We assumed that u is in the form

$$u(\xi) = Asn^\rho\left(B\xi,\ell\right), \quad \xi = x - vt, \tag{14}$$

where ℓ is the modulus of Jacobi elliptic function and $0 < \ell < 1$. Also A represents the amplitude, B is the inverse width of the soliton and unknown index ρ will be determined.

Substituting Eq. (14) and its necessary derivatives in the real part Eq. (13), we have

$$(a - 4\beta)(\rho - 1)\rho AB^2 sn^{\rho-2}(B\xi,\ell)$$
$$- (a - 4\beta)\rho\left[\ell^2(\rho - 1) + \ell + \rho\right] AB^2 sn^\rho(B\xi,\ell)$$
$$+ (a - 4\beta)\ell\rho(\ell\rho + 1) AB^2 sn^{\rho+2}(B\xi,\ell) \tag{15}$$
$$\left(w + a\kappa^2 + \gamma\right) Asn^\rho(B\xi,\ell) + bA^3 sn^{3\rho}(B\xi,\ell) = 0.$$

From Eq.(15), matching the exponents $sn^{\rho+2}(B\xi,\ell)$ and $sn^{3\rho}(B\xi,\ell)$ yields

$$\rho + 2 = 3\rho, \tag{16}$$

which gives

$$\rho = 1. \tag{17}$$

Equating coefficients of them and setting coefficients of $sn^{\rho+j}(B\xi,\ell)$, for $j = -2, 0$, to zero in (15) as these are linearly independent functions yields

$$A = \sqrt{\frac{w + a\kappa^2 + \gamma}{b\ell}}, \tag{18}$$

$$B = \sqrt{\frac{w + a\kappa^2 + \gamma}{(4\beta - a)(\ell + 1)}}, \tag{19}$$

which requires the constraints

$$\left(w + a\kappa^2 + \gamma\right) b > 0, \tag{20}$$

$$\left(w + a\kappa^2 + \gamma\right)(4\beta - a) > 0. \tag{21}$$

So, for Kerr law nonlinearity, the Jacobi elliptic function solution is

$$q(x,t) = \sqrt{\frac{w + a\kappa^2 + \gamma}{b\ell}}.sn\left[\sqrt{\frac{w + a\kappa^2 + \gamma}{(4\beta - a)(\ell + 1)}}\right.$$
$$\left.(x + 2a\kappa t), \ell\right].e^{i(-\kappa x + wt + \theta)}, \tag{22}$$

If the modulus $\ell \to 1$ in Eq. (22), we obtain following dark optical soliton solution

$$q(x,t) = \sqrt{\frac{w + a\kappa^2 + \gamma}{b}}.\tanh\left[\sqrt{\frac{w + a\kappa^2 + \gamma}{2(4\beta - a)}}\right.$$
$$\left.(x + 2a\kappa t)\right].e^{i(-\kappa x + wt + \theta)}. \tag{23}$$

In solutions (22) and (23), $q(x,t)$ represents the soliton molecule in fiber. κ and w are the frequency and wave number of the soliton respectively, θ is the phase constant. Also γ depicts detuning effect, a , b and β are constants.

In order to construct exact solutions for Eq. (12); we use hypothesis in the form

$$u(\xi) = Acn^{\rho}(B\xi, \ell), \qquad (24)$$

From (24), Eq. (13) reduces to

$$(a - 4\beta)(1 - \ell^2)(\rho - 1)\rho AB^2 cn^{\rho-2}(B\xi, \ell)$$

$$+ (a - 4\beta)\rho[\ell^2(2\rho - 1) + \ell - \rho]AB^2 cn^{\rho}(B\xi, \ell)$$

$$- (a - 4\beta)\ell\rho(\ell\rho + 1)AB^2 cn^{\rho+2}(B\xi, \ell) \quad (25)$$

$$- (w + a\kappa^2 + \gamma)Acn^{\rho}(B\xi, \ell) + bA^3 cn^{3\rho}(B\xi, \ell) = 0,$$

Setting the exponents and coefficients of functions $cn^{\rho+2}(B\xi, \ell)$ and $cn^{3\rho}(B\xi, \ell)$ equal to one another, and again setting the coefficients functions of $cn^{\rho+j}(B\xi, \ell)$ to zero for $j = -2, 0$, we acquire the same value of which is in (17) and following equations

$$A = \sqrt{\frac{\ell(\ell+1)(w + a\kappa^2 + \gamma)}{b(\ell^2 + \ell - 1)}}, \qquad (26)$$

$$B = \sqrt{\frac{w + a\kappa^2 + \gamma}{(a - 4\beta)(\ell^2 + \ell - 1)}}, \qquad (27)$$

with the conditions

$$(w + a\kappa^2 + \gamma)b(\ell^2 + \ell - 1) > 0, \qquad (28)$$

$$(w + a\kappa^2 + \gamma)(a - 4\beta)(\ell^2 + \ell - 1) > 0. \qquad (29)$$

Hence, we get the Jacobi elliptic function solution for CGLE with Kerr law nonlinearity as

$$q(x,t) = \sqrt{\frac{\ell(\ell+1)(w + a\kappa^2 + \gamma)}{b(\ell^2 + \ell - 1)}}$$

$$.cn\left[\sqrt{\frac{w + a\kappa^2 + \gamma}{(a - 4\beta)(\ell^2 + \ell - 1)}}(x + 2a\kappa t), \ell\right] \quad (30)$$

$$.e^{i(-\kappa x + wt + \theta)}.$$

When $\ell \to 1$, solution (30) reduces bright optical soliton solution which is given by

$$q(x,t) = \sqrt{\frac{2(w + a\kappa^2 + \gamma)}{b}}$$

$$.\sec h\left[\sqrt{\frac{w + a\kappa^2 + \gamma}{(4\beta - a)}}(x + 2a\kappa t)\right] \qquad (31)$$

$$.e^{i(-\kappa x + wt + \theta)},$$

where κ represents the soliton frequency, while w depicts the wave number of the soliton. θ, a, b and β are constants and so γ arise from the detuning effect.

2.2. Parabolic law

In this case,

$$F(s) = b_1 s + b_2 s^2, \qquad (32)$$

where b_1 and b_2 are constants. So, Eq. (8) reduces to

$$iq_t + aq_{xx} + \left(b_1|q|^2 + b_2|q|^4\right)q = \frac{\beta}{|q|^2 q^*}$$

$$.\left[2|q|^2\left(|q|^2\right)_{xx} - \left\{\left(|q|^2\right)_x\right\}^2\right] + \gamma q, \qquad (33)$$

and the real part Eq. (9) simplifies to

$$(a - 4\beta)u'' - (w + a\kappa^2 + \gamma)u + b_1 u^3 + b_2 u^5 = 0. \qquad (34)$$

The initial hypothesis as given below

$$u(\xi) = A[D + sn(B\xi, \ell)]^{\rho}, \qquad (35)$$

So we get

$$(a - 4\beta)(\rho - 1)\rho AB^2(1 - D^2)(1 - \ell^2 D^2)$$

$$.[D + sn(B\xi, \ell)]^{\rho-2} + (a - 4\beta)\rho\{2\rho(1 - \ell^2 D^2)$$

$$+ \ell(1 - D^2) + \ell^2(3D^2 - 2) - 1\}AB^2 D$$

$$.[D + sn(B\xi, \ell)]^{\rho-1} + (a - 4\beta)\rho$$

$$\{\ell D^2(6\ell\rho - 4\ell D + \ell + 2) + \ell^2(1 - 2D - \rho)$$

$$- \ell - \rho\}AB^2[D + sn(B\xi, \ell)]^{\rho} + (a - 4\beta) \quad (36)$$

$$.\ell\rho(-4\ell\rho + 3\ell - 3)AB^2 D[D + sn(B\xi, \ell)]^{\rho+1}$$

$$+ (a - 4\beta)\ell\rho(\ell\rho + 1)AB^2[D + sn(B\xi, \ell)]^{\rho+2}$$

$$- (w + a\kappa^2 + \gamma)A[D + sn(B\xi, \ell)]^{\rho}$$

$$+ b_1 A^3[D + sn(B\xi, \ell)]^{3\rho} + b_2 A^5[D + sn(B\xi, \ell)]^{5\rho} = 0.$$

Setting the exponents and the coefficients $[D + sn(B\xi, \ell)]^{\rho+1}$ and $[D + sn(B\xi, \ell)]^{3\rho}$ and also $[D + sn(B\xi, \ell)]^{\rho+2}$ and $[D + sn(B\xi, \ell)]^{5\rho}$ equal to one another, again equating the coefficients of $[D + sn(B\xi, \ell)]^{\rho+j}$ to zero, for $j = -2, -1, 0$, in the Eq. (36), yields

$$\rho = \frac{1}{2}, \qquad (37)$$

$$D = \pm 1, \qquad (38)$$

$$A = \sqrt{\frac{b_1(\ell + 2)}{2b_2(\ell - 3)D}}, \qquad (39)$$

$$B = 2\sqrt{\frac{(w + a\kappa^2 + \gamma)}{(a - 4\beta)[4\ell(1 + 2\ell - 2\ell D) + \ell^2(1 - 4D) - 2\ell - 1]}}, \qquad (40)$$

with conditions

$$b_1 b_2(\ell - 3)D > 0, \qquad (41)$$

and

$$(w + a\kappa^2 + \gamma)(a - 4\beta)[4\ell(1 + 2\ell - 2\ell D)$$

$$+ \ell^2(1 - 4D) - 2\ell - 1] > 0. \qquad (42)$$

Thus, the Jacobi elliptic function solution for the CGLE with parabolic law nonlinearity is given by

$$q(x,t) = \sqrt{\frac{b_1(\ell+2)}{2b_2(\ell-3)D}}.$$

$$\left[D + sn\left(2\sqrt{\frac{(w+a\kappa^2+\gamma)}{(a-4\beta)\left[4\ell(1+2\ell-2\ell D)+\ell^2(1-4D)-2\ell-1\right]}} \right.\right.$$

$$\left.\left. (x+2a\kappa t),\ell\right)\right].e^{i(-\kappa x+wt+\theta)}. \tag{43}$$

When the modulus $\ell \to 1$, we obtain following dark optical soliton solution

$$q(x,t) = \frac{b_1}{2}\sqrt{\frac{-3}{b_2 D}}.$$

$$\left[D + \tanh\left(\sqrt{\frac{2(w+a\kappa^2+\gamma)}{(a-4\beta)(5-6D)}} \right.\right. \tag{44}$$

$$\left.\left. (x+2a\kappa t)\right)\right].e^{i(-\kappa x+wt+\theta)}.$$

In solutions (43) and (44), $q(x,t)$ represents the soliton molecule in fiber. κ and w are the frequency and wave number of the soliton respectively, θ is the phase constant. Also γ depicts detuning effect, a, b_1, b_2, D are constants.

Now, if we take the starting assumption as

$$u(\xi) = A\left[D + cn(B\xi,\ell)\right]^\rho, \tag{45}$$

Eq. (34) changes to

$$(a-4\beta)(\rho-1)\rho AB^2(1-D^2)(\ell^2+1)$$

$$.\left[D+cn(B\xi,\ell)\right]^{\rho-2} + (a-4\beta)\rho$$

$$\left\{\left[\ell^2(4\rho-3)+\ell\right](D^2-1)+2\rho-1\right\}$$

$$.AB^2 D\left[D+cn(B\xi,\ell)\right]^{\rho-1} + (a-4\beta)\rho\left\{(1-3D^2)\right.$$

$$\left. (2\ell^2\rho-\ell^2+\ell)-\rho\right\}AB^2\left[D+cn(B\xi,\ell)\right]^\rho \tag{46}$$

$$+ (a-4\beta)\ell\rho(4\ell\rho-\ell+3)AB^2 D\left[D+cn(B\xi,\ell)\right]^{\rho+1}$$

$$- (a-4\beta)\ell\rho(\ell\rho+1)AB^2 D\left[D+cn(B\xi,\ell)\right]^{\rho+2}$$

$$- (w+a\kappa^2+\gamma)A\left[D+cn(B\xi,\ell)\right]^\rho + b_1 A^3$$

$$\left[D+cn(B\xi,\ell)\right]^{3\rho} + b_2 A^5\left[D+cn(B\xi,\ell)\right]^{5\rho} = 0.$$

Doing similar operations, value of the parameters ρ and D obtained the same as Eq.s (37) and (38) respectively and yields

$$A = \sqrt{\frac{-b_1(\ell+2)}{2b_2(\ell+3)D}}, \tag{47}$$

$$B = 2\sqrt{\frac{w+a\kappa^2+\gamma}{(a-4\beta)\left[2\ell(1-3D^2)-1\right]}}, \tag{48}$$

where

$$b_1 b_2 D > 0, \tag{49}$$

$$(w+a\kappa^2+\gamma)(a-4\beta)\left[2\ell(1-3D^2)-1\right] > 0. \tag{50}$$

So, we obtain

$$q(x,t) = \sqrt{\frac{-b_1(\ell+2)}{2b_2(\ell+3)D}}$$

$$\left[D + cn\left(2\sqrt{\frac{w+a\kappa^2+\gamma}{(a-4\beta)\left[2\ell(1-3D^2)-1\right]}} \right.\right. \tag{51}$$

$$\left.\left. (x+2a\kappa t),\ell\right)\right].e^{i(-\kappa x+wt+\theta)}.$$

If the modulus $\ell \to 1$, we get following bright optical soliton solution

$$q(x,t) = \sqrt{\frac{-3b_1}{8b_2 D}}.$$

$$\left[D + \sec h\left(2\sqrt{\frac{w+a\kappa^2+\gamma}{(a-4\beta)\left[1-6D^2\right]}} \right.\right. \tag{52}$$

$$\left.\left. (x+2a\kappa t)\right)\right].e^{i(-\kappa x+wt+\theta)}.$$

where $q(x,t)$ represents the soliton molecule in fiber. κ represents the soliton frequency, while w depicts the wave number of the soliton. θ, a, b and β are constants and so γ arise from the detuning effect.

3. Conclusion

This paper consider CGLE in kerr and parabolic law media. Jacobi elliptic functions are used for the integration scheme here. Bright and dark optical soliton solutions are obtained using two types Jacobi elliptic functions. The existence criteria of these solutions are also indicated. These solutions provide recognise physical phenomena described by the equation. Due to the fact that bright and dark optical soliton solutions always help to address the soliton dynamics in long distance telecommunication system, the results of the paper are useful in the fiber optics communication technology. It can be obtained different solutions of the CGLE using the other Jacobi elliptic functions. This technique is very useful and effective to get soliton solutions of nonlinear partial differential equations in mathematical physics.

References

[1] Aranson, I.S., Kramer, L. (2002). The world of the complex Ginzburg-Landau equation. *Reviews of Modern Physics*, 74(1) 99–143.

[2] Cross, M.C., Hohenberg, p.C. (1993). Pattern formation outside of equilibrium. *Reviews of Modern Physics*, 65(3) 851.

[3] Matsuo, T., Furihata, D. (2001). Dissipative or conservative finite-difference schemes for complex-valued nonlinear partial differential equations. *Journal of Computational Physics*, 171(2) 425–447.

[4] Ginzburg, V.L., Landau, L.D. (1950). On the theory of superconductivity. *Zhurnal Eksperimentalnoi i Teoreticheskoi Fiziki*, 20 1064–1082.

[5] Biswas, A. (2018). Chirp-free bright optical solitons and conservation laws for complex Ginzburg-Landau equation with three nonlinear forms. *Optik - International Journal for Light and Electron Optics*, 174, 207–215.

[6] Mirzazadeh, M., Ekici, M., Sonmezoglu, A., Eslami, M., Zhou, Q., Kara, A.H., Milovic, D., Majid, F.B., Biswas, A., Belic, M. (2016). Optical solitons with complex Ginzburg-Landau equation. *Nonlinear Dynamics*, 85(3),1979-2016.

[7] Biswas, A., Yildirim, Y., Yasar, E., Triki, H., Alshomrani, A. S., Ullah, M. Z., Zhou, Q., Moshokoa, S. p., Belic, M. (2018). Optical soliton perturbation with complex Ginzburg–Landau equation using trial solution approach. *Optik*, 160, 44–60.

[8] Biswas, A., Yildirim, Y., Yasar, E., Triki, H., Alshomrani, A.S., Ullah, M.Z., Belic, M. (2018). Optical soliton perturbation for complex Ginzburg–Landau equation with modified simple equation method. *Optik*, 158, 399–415.

[9] Akram, G., Mahak, N. (2018). Application of the first integral method for solving (1+1) dimensional cubic-quintic complex Ginzburg-Landau equation. *Optik*, 164, 210–217.

[10] Biswas, A., Alqahtani, R.T. (2017). Optical soliton perturbation with complex Ginzburg-Landau equation by semi-inverse variational principle. *Optik*, 147, 77–81.

[11] Shwetanshumala, S. (2008). Temporal solitons of modified complex Ginzburg Landau equation. *Progress In Electromagnetics Research*, 3, 17–24.

[12] Arshed, S. (2018). Soliton solutions of fractional complex Ginzburg-Landau equation with Kerr law and non-Kerr law media. *Optik*, 160, 322-332.

[13] Baskonus, H.M. (2019). Complex Soliton Solutions to the Gilsonpickering Model. *Axioms*, 8(1), 18.

[14] Ilhan, O.A., Esen, A., Bulut H., Baskonus H.M. (2019). Singular Solitons in the pseudo-parabolic Model Arising in Nonlinear Surface Waves. *Results in Physics*, 12, 1712–1715.

[15] Cattani, C., Sulaiman, T.A., Baskonus H.M., Bulut, H. (2018). Solitons in an inhomogeneous Murnaghan's rod. *European Physical Journal Plus*, 133(228), 1-12.

[16] Baskonus, H.M., Sulaiman, T.A., Bulut, H. (2018). Dark, bright and other optical solitons to the decoupled nonlinear Schrödinger equation arising in dual-core optical fibers. *Optical and Quantum Electronics*, 50(4),1-12.

[17] Ciancio A., Baskonus, H.M., Sulaiman, T.A., Bulut, H. (2018). New Structural Dynamics of Isolated Waves Via the Coupled Nonlinear Maccari's System with Complex Structure. *Indian Journal of Physics*, 92(10), 1281–1290.

[18] Ilhan, O.A., Sulaiman, T.A., Baskonus H.M., Bulut, H. (2018). On the New Wave Solutions to a Nonlinear Model Arising in plasma physics. *European Physical Journal Plus*, 133(27), 1-6.

[19] Yel, G., Baskonus H.M., Bulut, H. (2017). Novel archetypes of new coupled Konno–Oono equation by using sine–Gordon expansion method. *Optical and Quantum Electronics*, 49(285), 1-10.

[20] Baskonus H.M., Bulut, H., Sulaiman, T.A. (2017). New complex and hyperbolic function solutions to the generalized double combined Sinh-Cosh-Gordon equation. *AIP Conf. Proc.*, 1798, 1-9.

[21] Baskonus H.M. (2016). New acoustic wave behaviors to the Davey–Stewartson equation with power-law nonlinearity arising in fluid dynamics. *Nonlinear Dynamics*, 86(1), 177–183.

Esma Ates *is an Assistant Professor in the Department of Electronics and Communication Engineering at Karadeniz Technical University. Her main research interest is the exact solutions of nonlinear partial differential equations.*

http://orcid.org/0000-0001-7302-3674

RESEARCH ARTICLE

An improved differential evolution algorithm with a restart technique to solve systems of nonlinear equations

Jeerayut Wetweerapong and Pikul Puphasuk[*]

Deparment of Mathematics, Faculty of Science, Khon Kaen University, Khon Kaen, 40002, Thailand
wjeera@kku.ac.th, ppikul@kku.ac.th

ARTICLE INFO	ABSTRACT
	In this research, an improved differential evolution algorithm with a restart technique (DE-R) is designed for solutions of systems of nonlinear equations which often occurs in solving complex computational problems involving variables of nonlinear models. DE-R adds a new strategy for mutation operation and a restart technique to prevent premature convergence and stagnation during the evolutionary search to the basic DE algorithm. The proposed method is evaluated on various real world and synthetic problems and compared with the recently developed methods in the literature. Experiment results show that DE-R can successfully solve all the test problems with fast convergence speed and give high quality solutions. It also outperforms the compared methods.

1. Introduction

Difficult nonlinear problems arise in a variety of fields in science and engineering, which demands for the effective computational methods and leads to the development of various intelligence methods [1, 2] such as genetic algorithm, particle swarm optimization, and differential evolution algorithm. These methods are recognized as the alternative approaches to the analytical ones and have found applications in many areas. For example, they have been applied to data clustering problems in data science [3, 4], the weight training of artificial neural networks [5,6], and numerical treatments of nonlinear systems [7–11]. This research focuses on solving nonlinear systems of equations which is common and important both in theoretical and application aspects [12–16]. It has been used as a key part of solving complex problems involving decision variables of nonlinear models. New systems of nonlinear equations often emerge from computational problems and can range from moderate problems with a few variables to the hard ones with many variables.

For example, physical models that are expressed as nonlinear partial differential equations become large systems of nonlinear equations when discretized [17]. The systems may also have strongly dependent variables or a large number of local solutions, which makes them much more difficult to solve. So the analytical solution approach aiming to get the closed form solution is usually impractical or impossible.

There are two practical approaches to solving a system of nonlinear equations: the local methods that directly solve the original system and the global methods that transform the system into an optimization problem (with box constraints of the variables) and solve that equivalent optimization problem instead. The local methods consist of iterative procedures that require an approximate solution as a starting vector point and use the local information (the derivatives or gradients) from the equational functions of the system to compute a new better approximate solution point for the next iteration. They include all various Newton-type methods [12] and can produce solutions with good computational speed and solution quality if

---

*Corresponding Author

the functions satisfy some analytical properties and an approximate solution sufficiently close to a real solution is given. Since these critical requirements of the local methods are often not fullfilled, the derivative-free global solution methods are needed.

1.1. Related work

There are a few global methods proposed for solving nonlinear systems in the literature. In 1998, Karr et al. [18] presented a hybrid scheme using a genetic algorithm to locate initial guesses of solutions, which are then supplied to a Newton method. Their results on one selected test problem of finding the nodes and weights for Gauss-Legendre quadrature showed that the genetic algorithm can effectively locate an initial guess that allows the Newton method to converge to an accurate solution. Later, Grosan and Abraham [19] applied a multi-objective optimization approach to solve nonlinear systems in 2008. They used a genetic algorithm and considered the nondominated solutions stored in an external set. Several problems are tested and the obtained solutions of various qualities both in number of different solutions and accuracy are reported. In 2009, Hirsch et al. [20] used the continuous GRASP optimization method, a multi-start local search procedure, to find all roots of nonlinear systems. In order to find different solutions, the objective function is adaptively modified to create an area of repulsion (or penalty region) around solutions that have already been found, and the continuous GRASP is run multiple times. The method showed promising results on four selected nonlinear systems from the literature. Jaberipour et al. [21] used a particle swarm optimization algorithm to solve nonlinear systems in 2011. They proposed a new way of updating each particle and a mechanism to replace some of the worst particles. Several test problems including both nonlinear systems of a single equation and a few equations are tested, and the solutions and their accuracies are reported. In 2012, Pourjafari and Mojallali [22] proposed a hybrid scheme of a two-phase root finder for a nonlinear system using an invasive weed optimization algorithm and a clustering technique. They also aimed to locate all roots of the system. The approach gives successful results on several constructed nonlinear equations in single variable that have many local solutions and on three real world problems of small size that have a few different solutions. Oliveira and Petraglia [23] proposed a stochastic optimization method known as fuzzy adaptive simulated annealing (fuzzy ASA) to find many solutions of

a nonlinear system in 2013. The fuzzy ASA is run several times to explore different regions during different activations. The method is stopped when the solutions with the predefined accuracy are found. Several test problems are tested and the obtained high accuracy solutions are reported. In 2016, Raja et al. [24] presented a memetic algorithm (GA-SQP), a hybrid of a genetic algorithm and a local search method based on a sequential quadratic programming (SQP) technique, for solving nonlinear systems. Several variants of GA-SQPs are proposed and tested on six different application problems. The results showed that the hybrid approaches give higher precision solutions and their proposed methods outperform several methods: simulated annealing (SA), pattern-search (PS), Nelder–Mead (NM) and Levenberg-Marquardt (LM) algorithm.

Recently in 2018, Zhang et al. [25] proposed a modified cuckoo search algorithm (CSA) for solving nonlinear equations and nonlinear systems. Four application systems are used to evaluate the CSA performance. By setting high precision tolerance as the termination condition, solutions with high accuracies are obtained and reported. They have shown that the CSA gives more accurate solutions than those obtained by GA-SQPs. And also in 2018, Raja et al. [10] have presented the particle swarm optimization hybrid with Nelder-Mead method (PSO-NMM) to solve nonlinear benchmark models. PSO-NMM exploits the strength of PSO as an efficient global search method to find good initial solutions and then applies the Nelder-Mead simplex method to refine the solutions for rapid local convergence. They have shown that for moderate to difficult nonlinear system problems, the hybridization of NMM can enhance the convergence and give quality solutions with high precision.

1.2. Innovative contribution

From the development of computational intelligence and evolutionary optimization methods to solve various complex real world and synthetic test problems of nonlinear systems, it is clear that the global methods become the essential tools and need more researchers attentions. This approach will also make solving nonlinear systems an important field of global optimization since it can supply plenty of challenging test problems. In addition, research contributions in this direction still lack common ground of how to compare and establish the obtained results.

In this research, we aim to apply the differential

evolution algorithm (DE) to solve nonlinear systems. DE is a popular and efficient global optimization method introduced by Storn and Price during the years 1995-1997 [26, 27] and it is surprising that no explicit research contributions on using DE to solve the problems are found among our extensive related literature reviews. We propose an improved differential evolution algorithm with a restart technique (DE-R) for solving the problems and offer a basic ground of applying DE in this direction. The main features of the proposed algorithm can be summarized as follows:

- The mixing strategy of the xbest-mutation to the basic DE mutation that utilizes the current best vector solution to enhance the search and the convergence to an optimal solution.
- The incorporating of a restart technique to prevent premature convergence and stagnation during the evolutionary search.
- Its performance both in term of the convergence speed and the achievement of high precision solutions are tested on several nonlinear benchmark problems of varying difficulty.

The remainder of the paper is organized as follows. The basic differential algorithm is presented in the next section and the proposed algorithm is described in Section 3. Section 4 gives details of the experimental design and lists all the benchmark problems. In Section 5, the performances of the DE-R method are compared with those of the basic DE algorithms based on the setting of the value to reach, and with those of the PSO and PSO-NMM methods [10] based on the setting of the maximum number of function evaluations. Finally, the conclusion are given in the last section.

2. The differential evolution algorithm (DE)

The basic or *classic* DE algorithm [26,27] for solving a continuous optimization problem is a stochastic search method using a population of real vectors with four main operations: initialization, mutation, crossover and selection. The pseudo code of the basic DE is illustrated in Table 1. Its main features are the differential mutation, the combined binomial crossover and the greedy selection. First, the initial population is generated uniformly in feasible region. For each generation and each target vector x_i, three different random

population vectors x_{r1}, x_{r2}, x_{r3}, which are also different from the target vector, are used to generate a mutant vector $xm = x_{r1} + F(x_{r2} - x_{r3})$ by adding the scaled difference of two vectors to another one with the scaling factor F. Then, some components of the target vector are exchanged with those of the mutant vector according to the crossover rate CR to produce the trial vector. The trial vector will replace the target vector in the selection process if it is fitter.

For more than two decades, DE has been shown to be one of the most efficient methods for continuous optimization problems [28–30]. However, DE's performances depend on five main factors: the population size NP, the control parameters F and CR, the dimension D, the objective function f to be solved, and the amount of computations allowed [31,32]. There are numerous research contributions on modifying DE to improve the performances in solving practical problems. These include the main approaches of adapting the control parameters and adjusting the basic mutation operation [33–42].

3. An improved differential evolution algorithm with a restart technique (DE-R)

We propose an improved differential evolution algorithm with a restart technique (DE-R) as a general purpose method for solving nonlinear systems through their equivalent transformed objective functions. Let the form of a nonlinear system be

$$
\begin{cases}
f_1(x_1, x_2, ..., x_n) & = 0 \\
f_2(x_1, x_2, ..., x_n) & = 0 \\
\quad\quad\quad\quad\quad\vdots \\
f_m(x_1, x_2, ..., x_n) & = 0
\end{cases}
$$

where $f_i : [LB, UB]^n \subseteq R^n \to R$ for $i = 1, ..., m$ are nonlinear equations (including linear functions) and $x = (x_1, x_2, ..., x_n)$ is a real vector. We want to find a solution x^* such that $f_i(x^*) = 0$ simultaneously for all i. The problem is transformed into the corresponding optimization problem by defining the objective function f as the sum of the absolutes or the squares of all f_i. For the smoothness of f, we use the sum of the squares. So the objective function f is as follows:

$$
f(x) = \sum_{i=1}^{m} f_i^2(x). \tag{1}
$$

The DE algorithm

(1) **Inputs:**
Objective function to be minimize (f), problem dimension (D), and population size (NP).

(2) **Initialization:**
(2.1) Randomly initialize all (row) vectors of the population matrix $P = [x_{ij}]$ of size $NP \times D$ with real values between the lower and upper bounds LB and UB.
(2.2) Calculate the fitness (the objective function value) for each population vector and record the best vector $xbest$ and the best value $fbest$.

(3) **Mutation:**
For each target population vector x_i, construct the mutant vector xm by

$$xm = x_{r1} + F(x_{r2} - x_{r3})$$

where $r1$, $r2$ and $r3$ are randomly generated distinct indices (in the range of 1 to NP) which are also different from the target index i, and F is a scaling factor.

(4) **Crossover:**
Construct the trial vector (the candidate vector) xc by replacing some components of x_i with the corresponding components of xm as follows:

$$xc_j = \begin{cases} xm_j & ; rand() < CR \text{ or } j = IC, \\ x_{ij} & ; \text{otherwise} \end{cases}$$

where $rand()$ is a uniform random number in $[0, 1]$, CR is a crossover rate, and IC is a randomly fixed index from 1 to D.

(5) **Selection:**
Apply the greedy selection and check for an update of $xbest$.
(5.1) Greedy selection:
If $f(xc) < f(x_i)$, then replace the target vector x_i with xc.
(5.2) Updating $xbest$:
If $f(xc) < fbest$, then update $xbest$ with xc and update $fbest$ with $f(xc)$.

(6) **Stopping condition:**
Repeat all the steps (3) - (5) until $fbest$ is less than the value to reach (VTR) or the maximum number of function evaluations ($maxnf$) is reached. Then report the obtained best solution.

To be able to optimize the objective function by using small number of function evaluations, we tend to use small population size. But optimizing with small populations will tend to loss diversity and lead to premature convergence or stagnation easily [31, 32, 43, 44]. We can also accelerate the convergence by utilizing the information of the population $xbest$ but this again increases the chance of those convergence problems. DE-R utilizes the $xbest$ information and at the same time prevents premature convergence and stagnation by incorporating an xbest-mutation and a restart technique.

The pseudo code and the flowchart of the proposed DE-R method are presented in Table 2 and Figure 1, respectively. After initialization, the DE-R creates a mutant vector by using the mixing scheme of two mutation operations: the basic mutation

$$xm = x_{r1} + F(x_{r2} - x_{r3}) \qquad (2)$$

and the xbest-mutation [28] that uses the $xbest$ information

$$xm = xbest + F_1(x_{r1} - x_{r2}) + F_2(x_{r3} - x_{r4}) \qquad (3)$$

where x_{r1}, x_{r2}, x_{r3}, and x_{r4} are different random vectors from the population and different from the target vector x_i. Each mutation operation is randomly chosen and applied with the proportion of $50\% : 50\%$. And instead of using a fixed value for scaling factors, the DE-R algorithm uses random values in the range of $[0.5, 0.7]$ for F, F_1 and F_2. These basic proportion and range of scaling factors are chosen from the preliminary experiments. For the crossover operation, the fixed value of crossover rate $CR = 0.9$ is used to generate the trial vector. Then the same greedy selection as in basic DE is applied. To prevent premature convergence or stagnation, the restart technique randomly restart PR of the population vectors by replacing them with the new generated vectors as in the initialization for every NRS generations. This incorporated restart operation periodically supplies small amount of new contents to the evolving population. Note that the $xbest$ vector is kept, updated and used in the xbest-mutation along the entire optimization process.

Table 2. Pseudo code of the improved DE algorithm with a restart (DE-R).

The improved DE algorithm with a restart (DE-R)

(1) **Inputs and control parameters:**
Objective function to be minimize : f
Problem dimension: D
Lower and upper bounds of the problem: LB, UB
Population size: $NP = 50$
The value to reach: $VTR = 10^{-20}$
Maximum number of function evaluations: $maxnf$
Scaling factors: F, F_1, F_2 in the range of $[0.5, 0.7]$
Crossover rate: $CR = 0.9$
Mixing rate of basic mutation and xbest-mutation: 0.5
The period to apply a restart (number of generations): $NRS = 200$
Restart rate (the percentage to restart the population vectors): $PR = 0.2$

(2) **Initialization:**
(2.1) Randomly initialize all (row) vectors of the population matrix $P = [x_{ij}]$ of size $NP \times D$ with real values between the lower and upper bounds LB and UB.
(2.2) Calculate the fitness (the objective function value) for each population vector and record the best vector $xbest$ and the best value $fbest$.

(3) **Mutation:**
(3.1) For each target population vector x_i, generate a uniform random number u in $[0, 1]$. If $u < 0.5$, apply the DE basic mutation in (3.2); otherwise, apply the xbest- mutation in (3.3).
(3.2) Basic mutation:
Randomly generate distinct indices $r1$, $r2$ and $r3$ (in the range of 1 to NP) which are also different from the target index i. Construct the mutant vector xm by

$$xm = x_{r1} + F(x_{r2} - x_{r3})$$

where F is randomly generated in $[0.5, 0.7]$.
(3.3) The xbest-mutation:
Randomly generate distinct indices $r1$, $r2$, $r3$, and $r4$ (in the range of 1 to NP) which are also different from the target index i. Construct the mutant vector xm by

$$xm = xbest + F_1(x_{r1} - x_{r2}) + F_2(x_{r3} - x_{r4})$$

where $xbest$ is the current best solution and F_1, F_2 are randomly generated in $[0.5, 0.7]$.

(4) **Crossover:**
Construct the trial vector (the candidate vector) xc by replacing some components of x_i with the corresponding components of xm as follows:

$$xc_j = \begin{cases} xm_j & ; rand() < CR \text{ or } j = IC, \\ x_{ij} & ; \text{otherwise} \end{cases}$$

where xm is the mutant vector from the step (3), CR is the crossover rate, $rand()$ is a uniform random number in $[0, 1]$, and IC is a randomly fixed index from 1 to D.

(5) **Selection:**
Apply the greedy selection and check for an update of $xbest$.
(5.1) Greedy selection:
If $f(xc) < f(x_i)$, then replace the target vector x_i with xc.
(5.2) Updating $xbest$:
If $f(xc) < fbest$, then update $xbest$ with xc and update $fbest$ with $f(xc)$.

(6) **Restart:**
Apply the restart technique every NRS generations by randomly choosing $PR \times NP$ distinct population vectors to be re-initialized as in the initialization where PR is the restart rate.

(7) **Stopping condition:**
Repeat all the steps (3) - (6) until $fbest$ is less than VTR or $maxnf$ is reached. Then report the obtained best solution.
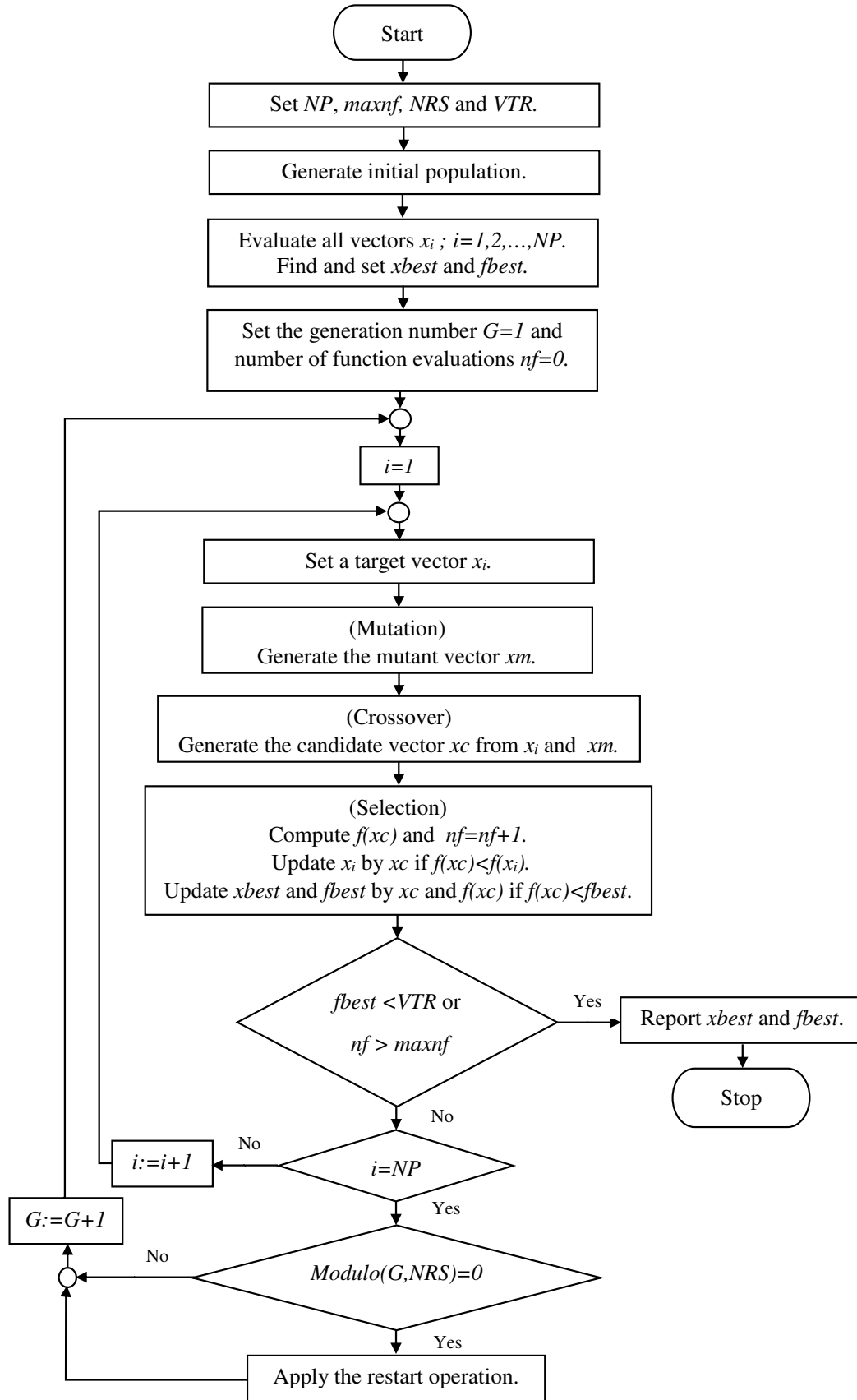
Figure 1. Flowchart of the proposed DE-R method.

4. Experimental design

To evaluate the performance of the DE-R algorithm, two experiments with different settings and measurements are performed. The first experiment assesses the proposed DE-R algorithm against the basic DE algorithms on 10 nonlinear benchmark models by setting the value to reach (VTR) while the second experiment compares the DE-R algorithm to the PSO and PSO-NMM methods [10] by setting the maximum number of function evaluations ($maxnf$).

4.1. Experiment 1: Performance comparison of the DE-R with the basic DE algorithms

The DE-R algorithm and the basic DE algorithms are tested on 10 selected nonlinear systems consisting of 6 real world problems (case study 1-6) and 4 synthetic problems (case study 7-10). Their definitions, parameters, variable bounds and the solutions (for the case of synthetic problems) are listed as follows.

Case study 1: *Neurophysiology application* [23, 45]. The system consists of the following six equations in six variables

$$\begin{cases} f_1(x) = & x_1^2 + x_3^2 - 1 = 0 \\ f_2(x) = & x_2^2 + x_4^2 - 1 = 0 \\ f_3(x) = & x_5 x_3^3 + x_6 x_4^3 - c_1 = 0 \\ f_4(x) = & x_5 x_1^3 + x_6 x_2^3 - c_2 = 0 \\ f_5(x) = & x_5 x_1 x_3^2 + x_6 x_4^2 x_2 - c_3 = 0 \\ f_6(x) = & x_5 x_1^2 x_3 + x_6 x_2^2 x_4 - c_4 = 0 \end{cases}$$

where $c_i = 0$ for all i as in [23] and $-10 \le x_i \le 10$.

Case study 2: *Robot kinematics application* [46]. This problem concerns the indirect-position problem which is to find the desired position and orientation of the robot hand and the relative joint displacements. This problem can be reduced to the following system of eight equations in eight variables

$$\begin{cases} f_1(x) = & 4.731 \times 10^{-3} x_1 x_3 - 0.3578 x_2 x_3 \\ & -0.1238 x_1 + x_7 - 1.637 \times 10^{-3} x_2 \\ & -0.9338 x_4 - 0.3571 = 0 \\ f_2(x) = & 0.2238 x_1 x_3 + 0.7623 x_2 x_3 + 0.2638 x_1 \\ & -0.07745 x_2 - 0.6734 x_4 - 0.6022 = 0 \\ f_3(x) = & x_6 x_8 + 0.3578 x_1 + 4.731 \times 10^{-3} x_2 = 0 \\ f_4(x) = & -0.7623 x_1 + 0.2238 x_2 + 0.3461 = 0 \\ f_5(x) = & x_1^2 + x_2^2 - 1 = 0 \\ f_6(x) = & x_3^2 + x_4^2 - 1 = 0 \\ f_7(x) = & x_5^2 + x_6^2 - 1 = 0 \\ f_8(x) = & x_7^2 + x_8^2 - 1 = 0 \end{cases}$$

where $-1 \le x_i \le 1$.

Case study 3: *Automative steering application* [20, 47]. This problem describes the kinematic synthesis of a trailing six-member mechanism for automotive steering. The system contains three equations in three unknown as follows

$$\begin{cases} f_i(x) = & [E_i(x_2 \sin \phi_i - x_3) - F_i(x_2 \sin \psi_i - x_3)]^2 \\ & +[F_i(1 + x_2 \cos \psi_i) - E_i(x_2 \cos \phi_i - 1)]^2 \\ & -[(1 + x_2 \cos \psi_i)(x_2 \sin \phi_i - x_3)x_1 \\ & -(x_2 \sin \psi_i - x_3)(x_2 \cos \phi_i - x_3)x_1]^2 = 0, \\ & i = 1, 2, 3 \end{cases}$$

where $0 \le x_i \le 1$ and

$$\begin{aligned} E_i = & \; x_2(\cos \psi_i - \cos \psi_0) - x_2 x_3 (\sin \psi_i - \sin \psi_0 \\ & -(x_2 \sin \psi_i - x_3)x_1 \\ F_i = & \; -x_2 \cos \phi_i - x_2 x_3 \sin \phi_i + x_2 \cos \phi_0 \\ & +x_1 x_3 + (x_3 - x_1)x_2 \sin \phi_0. \end{aligned}$$

The constants ϕ_i and ψ_i are given as follows

$$\begin{aligned} \phi_0 = & \; 1.3954170041747090114, \\ \phi_1 = & \; 1.7444828545735749268, \\ \phi_2 = & \; 2.0656234369405315689, \\ \phi_3 = & \; 2.4600678478912500533, \\ \psi_0 = & \; 1.7461756494150842271, \\ \psi_1 = & \; 2.0364691127919609051, \\ \psi_2 = & \; 2.2390977868265978920, \\ \psi_3 = & \; 2.4600678409809344550. \end{aligned}$$

Case study 4 : *Economics modeling application* [23, 45]. This problem arises in economics modeling. It can be extended for general dimensions n as follows

$$\begin{cases} f_i(x) = & (x_i + \sum_{j=1}^{n-i-1} x_j x_{j+i})x_n - c_i = 0, \\ & i = 1, 2, ..., n - 1 \\ f_n(x) = & \sum_{j=1}^{n-1} x_j + 1 = 0 \end{cases}$$

where $c_i = 0$ for all i as in [23] and $-10 \le x_i \le 10$.

Case study 5 : *Chemical equilibrium application* [23, 46, 48]. This problem describes a chemical equilibrium system. It concerns the combustion of propane in air to form ten products, which are transformed to ten equations in ten variables. To solve this problem, the system can be reduced to the following systems of five equations in five

variables

$$\begin{cases} f_1(x) = & x_1 x_2 + x_1 - 3x_5 = 0 \\ f_2(x) = & 2x_1 x_2 + x_1 + x_2 x_3^2 + R_5 x_2 - R_1 x_5 \\ & +2R_7 x_2^2 + R_4 x_2 x_3 + R_6 x_2 x_4 = 0 \\ f_3(x) = & 2x_2 x_3^2 + 2R_2 x_3^2 - 8x_5 + R_3 x_3 \\ & +R_4 x_2 x_3 = 0 \\ f_4(x) = & R_6 x_2 x_4 + 2x_4^2 - 4R_1 x_5 = 0 \\ f_5(x) = & x_1(x_2 + 1) + R_7 x_2^2 + x_2 x_3^2 + R_5 x_2 \\ & +R_2 x_3^2 + x_4^2 - 1 + R_3 x_3 + R_4 x_2 x_3 \\ & +R_6 x_2 x_4 = 0 \end{cases}$$

where $-100 \leq x_i \leq 100$ and the constants used in this system are

$$\begin{aligned} R_1 = & \ 10, R_2 = 0.193, R_3 = 0.002597/\sqrt{40}, \\ R_4 = & \ 0.003448/\sqrt{40}, R_5 = 0.00001799/40, \\ R_6 = & \ 0.0002155/\sqrt{40}, R_7 = 0.00003846/40. \end{aligned}$$

Case study 6 : *Combustion application* [23, 45]. This problem is a typical chemical equilibrium problem which represents a combustion problem. The system consists of ten equations in ten unknowns as follows

$$\begin{cases} f_1(x) = & x_2 + 2x_6 + x_9 + 2x_{10} - 10^{-5} = 0 \\ f_2(x) = & x_3 + x_8 - 3 \times 10^{-5} = 0 \\ f_3(x) = & x_1 + x_3 + 2x_5 + 2x_8 + x_9 + x_{10} \\ & -5 \times 10^{-5} = 0 \\ f_4(x) = & x_4 + 2x_7 - 10^{-5} = 0 \\ f_5(x) = & 0.5140437 \times 10^{-7} x_5 - x_1^2 = 0 \\ f_6(x) = & 0.1006932 \times 10^{-6} x_6 - 2x_2^2 = 0 \\ f_7(x) = & 0.7816278 \times 10^{-15} x_7 - x_4^2 = 0 \\ f_8(x) = & 0.1496236 \times 10^{-6} x_8 - x_1 x_3 = 0 \\ f_9(x) = & 0.6194411 \times 10^{-7} x_9 - x_1 x_2 = 0 \\ f_{10}(x) = & 0.2089296 \times 10^{-14} x_{10} - x_1 x_2^2 = 0 \end{cases}$$

where $-20 \leq x_i \leq 20$.

Case study 7 : *Rosenbrock function* [49]. This function is well-known for testing the performance of the optimization methods. The two-dimensional function has the global minimum inside a long, narrow, parabolic shaped flat valley. It is unimodal for dimensions 2 and 3, and has 2 minima for higher dimensions. The high-dimensional function is highly nonseparable and is used as one of the difficult test functions. Rosenbrock function can be written in the form of system of nonlinear equations in general dimensions n as follows

$$\begin{cases} f_{2i-1}(x) = 10(x_{i+1} - x_i^2) = 0, \ i = 1, 2, ..., n-1 \\ f_{2i}(x) = 1 - x_i = 0, \ i = 1, 2, ..., n-1. \end{cases}$$

In this work, we set $n = 10$ and $-100 \leq x_i \leq 100$. The global solution is $(1, 1, ..., 1)$.

Case study 8 : *SINQUAD function* [50]. This function is multimodal and nonseparable and it is one of the test functions from CUTE: Constrained and Unconstrained Testing Environment. It can be scaled up to arbitrary dimension n and can be written in the form of the following system

$$\begin{cases} f_1(x) = (x_1 - 1)^2 = 0 \\ f_i(x) = \sin(x_i - x_n) - x_1^2 + x_i^2 = 0, \ i = 1, ..., n-1 \\ f_n(x) = x_n^2 - x_1^2 = 0. \end{cases}$$

We set $n = 10$ and $-100 \leq x_i \leq 100$. There are $2^{n-1} + 1$ solutions in the forms $(1, 1, ..., 1, 1)$ and $(1, a_1, a_2, ..., a_8, -1)$ where $a_j \in \{0.2357835607, -1\}$.

Case study 9 : *Proposed function 1.* This function can be written in the form of the system of three nonlinear equations in n variables where $n \geq 3$. The first two equations give the intersection of two n-spheres which is an $(n-1)$-sphere. The last equation chooses the two intersection points where $x_1 = 0.05$. This system is expected to be highly nonseparable since the searching points must lie in the $(n-1)$-sphere. It is shown as follows

$$\begin{cases} f_1(x) = & x_1^2 + x_2^2 + \cdots + x_n^2 - 100 = 0 \\ f_2(x) = & (x_1 - 0.1)^2 + x_2^2 + \cdots + x_n^2 - 100 = 0 \\ f_3(x) = & x_1^2 + (x_2 - x_3)^2 + (x_3 - x_4)^2 + \cdots \\ & +(x_{n-1} - x_n)^2 - 0.0025 = 0. \end{cases}$$

We set $n = 10$ and $-100 \leq x_i \leq 100$. There are two solutions which are $(0.05, a, a, ..., a)$ and $(0.05, -a, -a, ..., -a)$ where $a = \sqrt{(100 - 0.05^2)/(n-1)}$.

Case study 10 : *Proposed function 2.* This function can be written in the form of the system of three nonlinear equations in n variables where n is even. The system has one obvious solution at $(n, n, ..., n)$. It is presented as follows

$$\begin{cases} f_1(x) = & x_1 + x_2 + \cdots + x_n - n^2 = 0 \\ f_2(x) = & x_1^2 + x_2^2 + \cdots + x_n^2 - n^3 = 0 \\ f_3(x) = & x_1^2 - x_2^2 + x_3^2 - x_4^2 + \cdots + x_{n-1}^2 - x_n^2 = 0. \end{cases}$$

We set $n = 10$ and $-100 \leq x_i \leq 100$. For the first experiment, the proposed DE-R is tested and compared with two basic DE algorithms. The basic DEs follow the settings as recommended in [27] to use NP in the range of $5 \times D$ to $10 \times D$, $F = 0.5$ and $CR = 0.9$. Since the maximum D of all test problems is 10, $NP = 50$ and $NP = 100$ are chosen. So we denote these basic DEs as DE59-50 and DE59-100, respectively.

The DE-R method, denoted by DE59-50-R, uses $NP = 50$, F in $[0.5, 0, 7]$ and the same $CR = 0.9$. The settings and features of DE59-50, DE59-100 and DE59-50-R are summarized in Table 3. Each algorithm is run 30 times for each problem. The maximum number of function evaluations $maxnf$ is set to 1000000 and the VTR (value to reach) for $fbest$ is set to 10^{-20} to guarantee that each $f_i(xbest)$ is less than 10^{-10}. If the $fbest$ value is less than the VTR before reaching the $maxnf$, the successful run and the number of function evaluations used (nf) are recorded. We report the number of successful runs (NS), the mean of function evaluations (Mean nf) and the percentage of standard derivation (%SD).

4.2. Experiment 2: Performance comparison of DE-R with PSO and PSO-NMM methods

The second experiment compares the performance of the DE-R algorithm with those of the PSO and PSO-NMM using the same setting as in [10]. The objective function f is the mean square defined by

$$f(x) = \frac{1}{m} \sum_{i=1}^{m} f_i(x)^2. \tag{4}$$

We choose the following PSO and PSO-NMM variants that obtain good results in [10]: PSO-12, PSO-15, PSO-16, PSO-NMM-12, PSO-NMM-15, and PSO-NMM-16. They are compared on the case studies 1, 4, 5, and 6. Note that for the case study 4, the parameters $c_i = 1$ for $i = 1, 2, ..., n-1$ where $n = 5$. Each algorithm conducts 100 independent runs for each problem. The Min, Mean and SD values of $fbest$ of DE-R at each $maxnf$ are compared with those of PSO and PSO-NMM. The PSO-12 and PSO-15 methods use $maxnf = 50000$ while PSO-16 uses $maxnf = 100000$. The PSO-NMM-12, PSO-NMM-15 and PSO-NMM-16 variants extend the corresponding PSO variants respectively. For each PSO-NMM method, if its PSO phase obtains the solution that reaches the tolerance 10^{-15} or uses up all the $maxnf$ without reaching the tolerance, then it enters the NMM phase by applying the local search using the Nelder-Mead method with the additional function evaluations of 200000. To compare the DE-R with each PSO-NMM method on each test problem, the $maxnf$ for DE-R needs to be adjusted properly. For the former case, Neurophysiology and Economics modeling applications, the DE-R method uses only 200000 as the $maxnf$. For the latter case, Chemical equilibrium and Combustion applications, DE-R uses the sum of the function evaluations of both two phases.

5. Results and discussion

This section shows performance comparison of the DE-R with the basic DE algorithms and the performance comparison of DE-R with PSO and PSO-NMM methods. In each report table, the best values are indicated in bold. The discussion for each experiment is given as follows.

5.1. Performance comparison of the DE-R with the basic DE algorithms

The performance comparison of DE59-50, DE59-100 and DE59-50-R are shown in Table 4, Figure 2 and Figure 3. The table presents the NS, Mean and %SD of each method for all 10 test systems. For the ability and stability of solving each problem, the number of successful runs out of the total 30 runs is considered first. From Table 4, the results show that the DE59-50 algorithm can successfully solve 6 out of 10 problems. It cannot solve the problems 6, 8, 9 and 10, and gives no successful runs out of total 30 runs for these problems. It can solve problem 5 (Chemical Equilibrium Applications) and problem 7 (Rosenbrock problem), but gives high Means and %SDs. However, DE59-50 gives the best results for the first four problems which appear to be relatively easy problems. We can conclude that DE59-50 is good for easy problems but cannot be recommended for solving difficult problems. This is because of its too small population size.

The DE59-100 algorithm can solve all 10 problems. This shows that the increased population size of 100 can increase the solving ability of the DE algorithm with $F = 0.5$ and $CR = 0.9$. However, DE59-100 gives very high Mean values. This shows that its speed of convergence is rather slow. The proposed DE59-50-R algorithm can successfully solve all 10 problems. It also gives the best results (smallest Means) for the last 6 problems which appear to be difficult problems. For the first four easy problems, DE59-50-R gives the second best results. This shows that the incorporated new mutation strategy and restart technique can prevent the premature convergence and stagnation, and still gives fast convergence speeds for difficult problems as clearly shown in Figure 2 and Figure 3.

Some solutions obtained by DE59-50-R for all application systems and SINQUAD function are reported. We show only 4 different solutions for each problem. Since we set the $VTR = 10^{-20}$, the absolute values of each $f_i(x)$ must be less than 10^{-10}. Table 5 shows some solutions of Neurophysiology application. Our method gives 30

Table 3. The settings and features of DE59-50, DE59-100 and DE59-50-R.

DE algorithm	NP	F	CR	Mutation and other feature
DE59-50	50	0.5	0.9	basic mutation
DE59-100	100	0.5	0.9	basic mutation
DE59-50-R	50	$[0.5, 0.7]$	0.9	basic mutation & xbest-mutation and a restart technique

Table 4. Performance comparison of DE59-50, DE59-100 and DE59-50-R using NS, Mean nf, %SD values at $VTR = 10^{-20}$ averaged over 30 independent runs.

Systems	Methods	NS	Mean nf	%SD
1: Neurophysiology application	DE59-50	30	**27272.70**	16.98
	DE59-100	30	76787.67	19.51
	DE59-50-R	30	40233.67	16.99
2: Robot kinematics application	DE59-50	30	**24942.53**	28.27
	DE59-100	30	57355.60	18.13
	DE59-50-R	30	34721.30	17.70
3: Automative steering application	DE59-50	30	**2303.30**	12.34
	DE59-100	30	4307.83	8.89
	DE59-50-R	30	2682.10	12.03
4: Economics modeling application	DE59-50	30	**12780.60**	5.96
	DE59-100	30	30193.97	4.63
	DE59-50-R	30	21831.93	7.96
5: Chemical equilibrium application	DE59-50	30	190249.03	47.06
	DE59-100	30	124793.33	5.70
	DE59-50-R	30	**30582.23**	3.95
6: Combustion application	DE59-50	0	-	-
	DE59-100	30	95503.70	3.79
	DE59-50-R	30	**59380.20**	4.13
7: Rosenbrock function	DE59-50	30	193992.90	50.51
	DE59-100	30	110388.83	3.62
	DE59-50-R	30	**59565.40**	2.52
8: SINQUAD function	DE59-50	0	-	-
	DE59-100	30	173259.07	12.29
	DE59-50-R	30	**81755.37**	8.90
9: Proposed function 1	DE59-50	0	-	-
	DE59-100	30	109782.20	6.95
	DE59-50-R	30	**65107.80**	5.72
10: Proposed function 2	DE59-50	0	-	-
	DE59-100	30	638354.80	14.20
	DE59-50-R	30	**160827.47**	12.69

different solutions for 30 runs. The values of each variable can be positive or negative. There are two trends of solutions. First, their magnitudes are pairwise equal as $|x_1| = |x_2|$, $|x_3| = |x_4|$ and $|x_5| = |x_6|$. For another trend, all their components are different. We notice that the absolute values of x_5 and x_6 are quite smaller than other components.

Some solutions of Robot kinematics application are presented in Table 6. The authors in [46] claimed that there are 16 solutions. Our results show that the proposed method gives 10 different solutions in 30 runs. All $|x_i|$ are not equal but have roughly the same order. Table 7 shows some solutions of Automative steering application. Our method gives all 30 different solutions for 30 runs. The absolute values of x_1 are bigger than others and the values of x_2 and x_3 have quite the same order. For the Economics modeling application, 30 different solutions are obtained. Some of them are reported in Table 8. Each variable can be positive or negative. The absolute values of x_{10} are much smaller than others which have the same order.
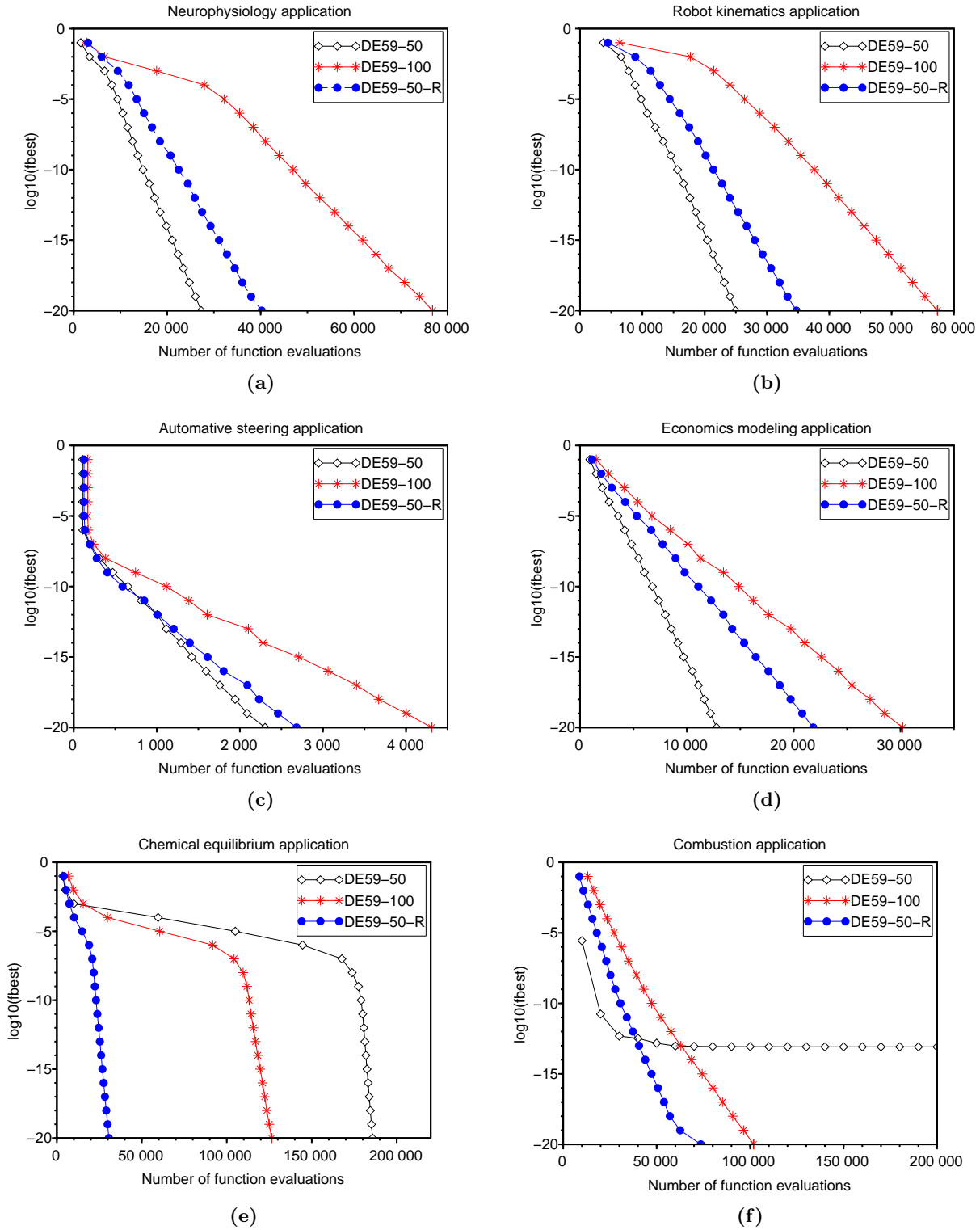
Figure 2. Convergence graphs of DE59-50, DE59-100 and DE59-50-R for systems 1 - 6.

There are four real solutions of the Chemical equilibrium application reported in [48]. Our method gives all four solutions as shown in Table 9. For the Combustion application, the proposed method gives 30 different solutions in 30 runs. Some solutions are presented in Table 10. It shows that the absolute values of x_5, x_6, x_9 and x_{10} are bigger than those of other components

which are rather small.

In case of the synthetic test problems, the numbers of solutions for each of Rosenbrock function, proposed functions 1 and 2 are at most 2 and all these solutions are found. Thus we show only the solutions of SINQUAD function which has $2^{n-1} + 1$ solutions. Some of them are shown in Table 11. After running the proposed method 30
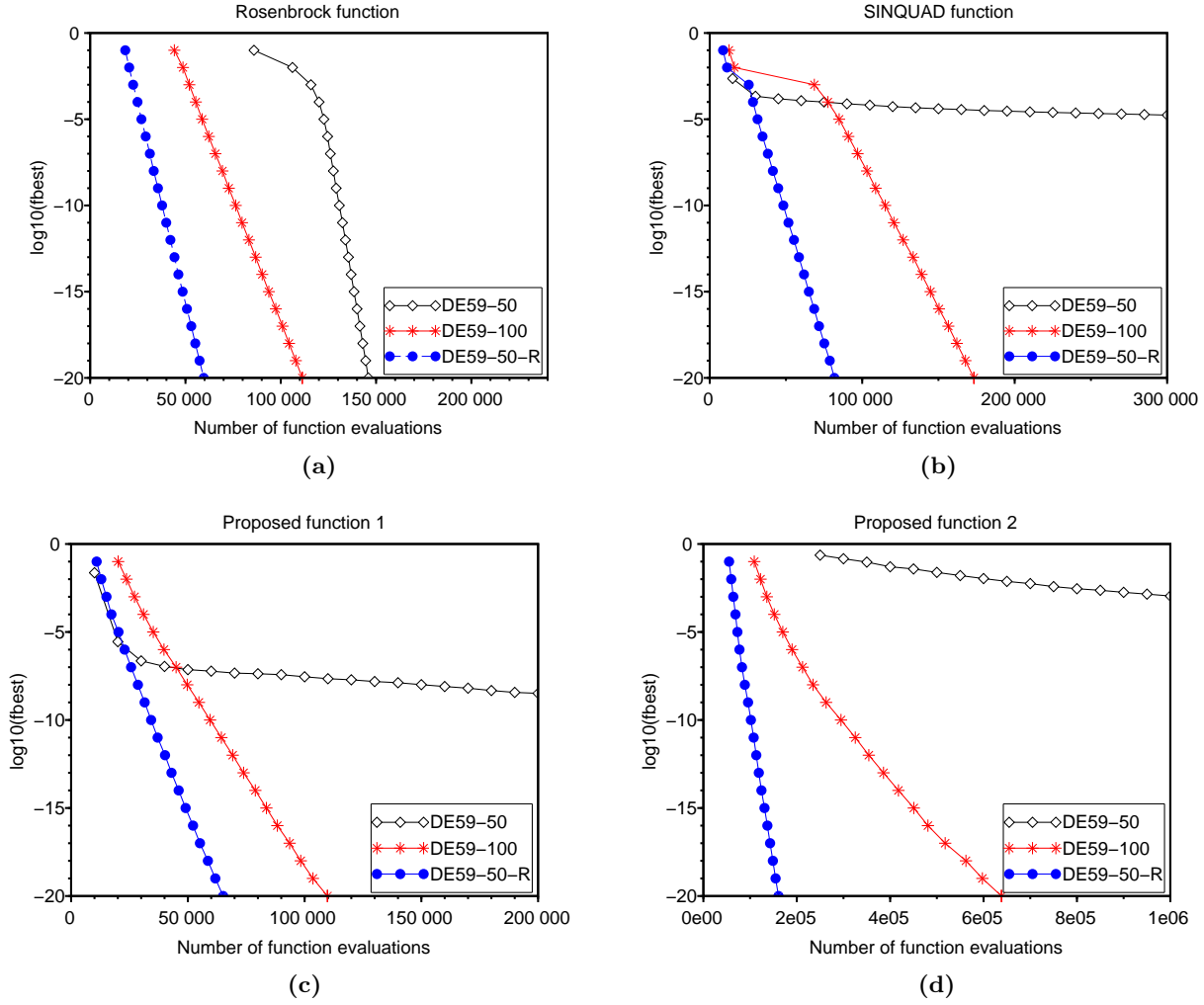
Figure 3. Convergence graphs of DE59-50, DE59-100 and DE59-50-R for systems 7 - 10.

Table 5. Some solutions of Neurophysiology application.

	Solution 1	Solution 2	Solution 3	Solution 4
x_1	9.7749269097E-01	- 4.2695388140E-02	9.2272247439E-02	- 8.2491292630E-01
x_2	- 9.7749277453E-01	4.2695288767E-02	- 1.1674880868E-01	7.9424101299E-01
x_3	- 2.1096928480E-01	9.9908813614E-01	9.9573381602E-01	5.6525981992E-01
x_4	2.1096889745E-01	- 9.9908814042E-01	- 9.9316147514E-01	6.0760284168E-01
x_5	- 2.9012525772E-05	1.0549841840E-04	1.0199560579E-09	- 4.1050316028E-11
x_6	- 2.9012444215E-05	1.0549834938E-04	1.0098557221E-09	4.2217134911E-11
$f_1(x)$	1.9671153595E-11	- 5.6388893555E-11	1.6900481015E-11	- 1.7983725620E-11
$f_2(x)$	- 4.6815662458E-11	1.6239010137E-11	3.8434810889E-11	- 7.0937811181E-11
$f_3(x)$	2.2663307018E-12	6.7481831402E-11	1.7678400883E-11	2.0558188234E-12
$f_4(x)$	- 6.9223284026E-11	- 6.2704301665E-14	- 8.0570377011E-13	4.4194815939E-11
$f_5(x)$	- 8.0752967592E-12	- 1.3367709799E-11	- 2.2980110987E-11	2.3198700376E-11
$f_6(x)$	2.6177939401E-11	1.0192828755E-12	- 5.0234619236E-12	3.9133806914E-13

times, it can give 27 different solutions.
To show that the proposed method can also give high quality solutions, it is applied to solve two difficult application systems and one difficult synthetic function (Neurophysiology and Combustion systems, and SINQUAD function) by setting 3 different levels of the VTR values: 10^{-20}, 10^{-30}

and 10^{-40}. The solutions once reaching each VTR level are recorded in the same run in order to investigate their behaviors and accuracies. The results are shown in tables 12, 13, and 14, respectively. One set of three solutions (each one at each accuracy level) from the same run is reported for each problem.

Table 6. Some solutions of Robot kinematics application.

	Solution 1	Solution 2	Solution 3	Solution 4
x_1	1.6443166583E-01	1.6443166582E-01	6.7155426177E-01	6.7155426182E-01
x_2	- 9.8638847688E-01	- 9.8638847685E-01	7.4095537891E-01	7.4095537883E-01
x_3	- 9.5472843449E-01	2.3961601716E-01	- 2.3961165919E-01	9.5472976979E-01
x_4	2.9747876626E-01	- 9.7086773778E-01	- 9.7086881339E-01	2.9747448069E-01
x_5	- 9.1115479620E-01	- 9.9763539824E-01	9.5791710189E-01	- 1.2877823744E-01
x_6	4.1206423943E-01	- 6.8728539911E-02	- 2.8704498935E-01	9.9167341678E-01
x_7	9.9132241509E-01	- 6.1550840708E-01	- 5.2790902637E-01	9.6931180772E-01
x_8	- 1.3145291671E-01	7.8813031971E-01	8.4930092421E-01	- 2.4583453656E-01
$f_1(x)$	- 2.4255042419E-12	4.6582404600E-11	2.1869284161E-11	- 1.5662637853E-11
$f_2(x)$	3.1039615322E-11	4.5871084708E-12	- 1.3820167233E-11	5.0422999109E-12
$f_3(x)$	2.1743830694E-12	1.2673250470E-11	1.4527121259E-11	- 5.7471936498E-11
$f_4(x)$	1.5060341862E-11	2.7537028213E-11	5.3715809578E-11	- 2.9148350400E-12
$f_5(x)$	4.0200731632E-11	- 1.2329803845E-11	3.2028157904E-11	- 1.9571233523E-11
$f_6(x)$	- 9.7184482684E-12	- 6.6788574671E-11	3.0616620350E-11	- 2.4448998381E-11
$f_7(x)$	5.4400040028E-11	1.0798473227E-11	- 5.2414739216E-12	- 1.9850676658E-11
$f_8(x)$	- 2.2364221586E-11	2.5351054589E-11	- 1.3006706823E-11	- 3.8787306700E-11

Table 7. Some solutions of Automative steering application.

	Solution 1	Solution 2	Solution 3	Solution 4
x_1	1.1192696492E-01	1.4669784820E-01	2.1001674043E-02	1.8459769700E-02
x_2	3.8819470790E-05	2.5842964204E-05	1.0358260865E-05	2.5086298124E-05
x_3	1.3969968025E-05	9.5393185049E-06	9.4604945191E-05	4.0484643474E-06
$f_1(x)$	- 2.4953389394E-12	5.1027141958E-13	1.5509436820E-11	1.0749400610E-12
$f_2(x)$	6.2113867145E-12	- 1.8836413824E-12	2.9312085336E-11	1.9709660895E-11
$f_3(x)$	9.1878352205E-11	3.1456991936E-11	4.8961749305E-11	6.6461220923E-11

Table 8. Some solutions of Economics modeling application.

	Solution 1	Solution 2	Solution 3	Solution 4
x_1	- 6.1626101672E+00	6.0154692495E+00	5.1444125822E+00	3.5697160902E-01
x_2	8.4423418690E+00	2.4451383666E+00	- 8.8035624273E-01	- 3.2209182844E+00
x_3	- 6.0135423035E+00	- 4.3358334024E+00	1.8536153932E+00	6.4891799317E+00
x_4	6.6724322251E+00	9.7533235680E-01	9.9392766203E-01	- 8.4197837365E+00
x_5	1.4648933274E+00	- 6.7099720313E+00	8.8882743353E-01	3.8711746522E+00
x_6	- 9.4952931192E+00	2.9005275369E+00	8.0171844852E-01	5.6239280280E+00
x_7	- 1.8950537683E+00	- 2.0062068336E+00	- 3.9152883109E+00	6.0633634547E-01
x_8	2.5753259373E+00	3.0111923766E+00	- 5.5692586733E-01	1.0345842326E+00
x_9	3.4115059994E+00	- 3.2956476191E+00	- 5.3299310986E+00	- 7.3414727781E+00
x_{10}	- 2.1904782760E-13	- 5.8674381000E-14	- 6.1325687030E-13	4.1496727200E-14
$f_1(x)$	2.8765042169E-11	2.4583005293E-12	- 2.7179403893E-12	- 3.7642815954E-12
$f_2(x)$	8.8232434540E-13	- 2.3163223689E-12	- 1.6660038822E-11	3.1943021197E-13
$f_3(x)$	4.1467842031E-12	3.4719167082E-12	6.0666202644E-13	- 6.1741234660E-13
$f_4(x)$	1.0720305561E-11	- 8.5259062950E-14	4.7145635625E-12	- 2.4212746954E-12
$f_5(x)$	- 1.1227997919E-11	6.1242977433E-13	- 1.3063316353E-12	3.0065629487E-12
$f_6(x)$	- 7.4689413357E-13	- 7.3251520844E-13	1.7618556611E-11	- 1.8728334542E-12
$f_7(x)$	- 2.4172620758E-12	- 4.7228264171E-13	1.2805457393E-12	1.0217298215E-12
$f_8(x)$	4.0410941261E-12	9.8653194558E-13	1.7156652338E-11	- 6.5818503348E-14
$f_9(x)$	- 7.4728297801E-13	1.9337008404E-13	3.2686168644E-12	- 3.0464709312E-13
$f_{10}(x)$	4.9998227780E-11	- 8.8817841970E-16	- 8.0000006619E-11	- 9.9991126490E-12

From tables 12 and 13, the solution for both application problems at $VTR = 10^{-20}$ are quite different from those at $VTR = 10^{-30}$ and 10^{-40} where at these higher accuracies we obtained similar solutions with more accurate solutions at $VTR = 10^{-40}$. For the solutions of SINQUAD function as in Table 14, we can see the same trend

Table 9. All solutions of Chemical equilibrium application.

	Solution 1	Solution 2	Solution 3	Solution 4
x_1	3.1141022831E-03	2.7571773851E-03	2.4710000144E-03	2.1533077099E-03
x_2	3.4597924347E+01	3.9242289252E+01	4.3879222733E+01	5.0549570315E+01
x_3	6.5041778861E-02	- 6.1387603945E-02	5.7784455215E-02	- 5.4144807517E-02
x_4	8.5937805056E-01	8.5972442500E-01	- 8.6020547295E-01	- 8.6067132299E-01
x_5	3.6951859146E-02	3.6985043297E-02	3.6965520015E-02	3.7000695742E-02
$f_1(x)$	4.4613771011E-11	- 4.0732806017E-11	- 2.5496216249E-11	- 2.4678356580E-11
$f_2(x)$	7.7566347442E-12	- 3.0284536164E-11	5.4004083757E-11	2.1900292344E-11
$f_3(x)$	- 5.7297743156E-11	- 7.1773320794E-12	- 2.5416176759E-11	8.9192338411E-12
$f_4(x)$	1.4782175484E-11	- 2.1605606193E-11	- 7.5870421057E-11	- 2.4946711363E-11
$f_5(x)$	- 4.8531411962E-11	- 6.2299256465E-11	- 2.2982240893E-11	- 3.8321561636E-11

Table 10. Some solutions of Combustion application.

	Solution 1	Solution 2	Solution 3	Solution 4
x_1	- 2.1256693800E-07	- 1.2076437010E-06	- 8.2052168644E-08	- 2.9426051511E-07
x_2	- 8.1757590664E-06	4.2644337543E-06	- 1.4477162217E-07	- 4.7401492243E-06
x_3	- 6.7527163990E-04	- 4.2090672635E-05	5.1227964677E-05	- 6.0923109793E-05
x_4	- 4.1833078103E-06	6.4157325619E-06	2.1289152054E-06	3.0404307410E-06
x_5	1.6567014001E-04	4.5312194101E-04	- 2.2149062432E-04	3.2378004814E-04
x_6	1.2934173578E-03	8.0745848233E-04	- 2.3204194677E-04	1.6392254306E-04
x_7	7.0916610888E-06	1.7921487387E-06	3.9355599926E-06	3.4797831273E-06
x_8	7.0527161222E-04	7.2090667874E-05	- 2.1227993570E-05	9.0923084400E-05
x_9	5.3586029742E-04	- 3.0507240444E-04	4.9435402384E-04	- 1.1232728214E-03
x_{10}	- 1.5522596511E-03	- 6.5205449512E-04	- 1.0062660059E-05	4.0508395318E-04
$f_1(x)$	- 4.8273876190E-11	3.7401625020E-12	3.8557848129E-11	2.1883831795E-11
$f_2(x)$	- 2.7678047280E-11	- 4.7604545021E-12	- 2.8892893645E-11	- 2.5393269464E-11
$f_3(x)$	- 5.6092310138E-11	1.8660651569E-12	4.0505311148E-11	2.6580721802E-11
$f_4(x)$	1.4367258554E-11	3.0039278904E-11	3.5190617200E-11	- 3.0043345588E-12
$f_5(x)$	8.4709844719E-12	2.1834044602E-11	- 1.1392318563E-11	1.6557120143E-11
$f_6(x)$	- 3.4477399301E-12	4.4934787964E-11	- 2.3406963800E-11	- 2.8432143924E-11
$f_7(x)$	- 1.7500064230E-11	- 4.1161624304E-11	- 4.5322799486E-12	- 9.2442190882E-12
$f_8(x)$	- 3.8015147210E-11	- 4.0044070424E-11	1.0271567782E-12	- 4.3230264589E-12
$f_9(x)$	3.1455493138E-11	- 1.3747522017E-11	3.0610441206E-11	- 7.0974973960E-11
$f_{10}(x)$	1.0965489675E-17	2.0599143169E-17	- 1.9304158566E-20	7.4580841134E-18

Table 11. Some solutions of SINQUAD function.

	Solution 1	Solution 2	Solution 3	Solution 4
x_1	1.0000013135E+00	1.0000020233E+00	9.9999578687E-01	9.9999635081E-01
x_2	- 1.0000013135E+00	2.3578778570E-01	- 9.9999578691E-01	- 9.9999635080E-01
x_3	2.3578630346E-01	2.3578778573E-01	2.3577476333E-01	- 9.9999635077E-01
x_4	2.3578630350E-01	2.3578778572E-01	- 9.9999578687E-01	- 9.9999635077E-01
x_5	2.3578630340E-01	- 1.0000020233E+00	- 9.9999578688E-01	- 9.9999635083E-01
x_6	- 1.0000013135E+00	- 1.0000020234E+00	2.3577476328E-01	- 9.9999635075E-01
x_7	2.3578630343E-01	2.3578778581E-01	- 9.9999578683E-01	- 9.9999635080E-01
x_8	2.3578630341E-01	2.3578778575E-01	2.3577476333E-01	- 9.9999635085E-01
x_9	- 1.0000013135E+00	2.3578778572E-01	- 9.9999578684E-01	- 9.9999635081E-01
x_{10}	- 1.0000013135E+00	- 1.0000020233E+00	- 9.9999578686E-01	- 9.9999635083E-01
$f_1(x)$	1.7252114825E-12	4.0939003788E-12	1.7750473715E-11	1.3316562876E-11
$f_2(x)$	2.0599966177E-11	- 2.9519581224E-11	2.7519320156E-11	7.3162587100E-12
$f_3(x)$	1.2524037363E-11	- 3.2059285782E-12	3.5025565781E-11	- 2.5635049639E-11
$f_4(x)$	4.9083563602E-11	- 1.0487832824E-11	- 1.0218825786E-11	- 2.1720181209E-11
$f_5(x)$	- 2.9645889910E-11	- 1.5809797915E-11	- 4.2824632729E-12	3.1798563782E-11
$f_6(x)$	5.6868731946E-11	- 7.3832051584E-12	- 1.0347889212E-11	- 4.7247761259E-11
$f_7(x)$	- 6.3057475908E-12	6.1809182772E-11	- 4.7515213986E-11	9.0893959026E-13
$f_8(x)$	- 2.4836653567E-11	1.0494015379E-11	3.5985284197E-11	5.4105275815E-11
$f_9(x)$	- 1.9166890297E-12	- 1.3320324010E-11	- 4.3021475271E-11	1.6147416737E-11
$f_{10}(x)$	1.9149126729E-12	- 5.2741366829E-11	- 2.2285395751E-11	3.4775182733E-11

Table 12. Solutions of Neurophysiology application at $VTR = 10^{-20}$, 10^{-30} and 10^{-40}.

	Solution at $VTR = 10^{-20}$	Solution at $VTR = 10^{-30}$	Solution at $VTR = 10^{-40}$
x_1	9.774926909651037E-01	9.774910633733470E-01	9.774910827638724E-01
x_2	- 9.774927745309928E-01	- 9.774910647465765E-01	- 9.774910827654670E-01
x_3	- 2.109692848010614E-01	- 2.109768258014221E-01	- 2.109767359618408E-01
x_4	2.109688974538331E-01	2.109768194390176E-01	2.109767359544526E-01
x_5	- 2.901252577197261E-05	- 9.810715906270790E-08	2.444699263961000E-10
x_6	- 2.901244421520325E-05	- 9.810715863387080E-08	2.444699263882000E-10
$f_1(x)$	1.967115359491345E-11	2.220446049250313E-16	0
$f_2(x)$	- 4.681566245778868E-11	- 2.220446049250313E-16	0
$f_3(x)$	2.266330701846849E-12	8.737850539388192E-17	- 3.153751293361295E-22
$f_4(x)$	- 6.922328402630745E-11	- 1.434521764580082E-17	6.260981755202151E-21
$f_5(x)$	- 8.075296759227116E-12	- 2.701156197524962E-16	1.071345911685355E-21
$f_6(x)$	2.617793940060294E-11	6.272936219045604E-16	- 3.157529343873903E-21
Mean nf	40233.67	57500.57	77651.60
%SD	16.99	12.58	5.00

Table 13. Solutions of Combustion application at $VTR = 10^{-20}$, 10^{-30} and 10^{-40}.

	Solution at $VTR = 10^{-20}$	Solution at $VTR = 10^{-30}$	Solution at $VTR = 10^{-40}$
x_1	- 2.125669379957560E-07	1.371784251667994E-07	1.379796690717610E-07
x_2	- 8.175759066442083E-06	- 9.619976447815970E-08	- 1.024640702937120E-07
x_3	- 6.752716398985501E-04	1.565137776437351E-05	1.560729129475898E-05
x_4	- 4.183307810306110E-06	7.224452410444500E-09	6.565809411140000E-11
x_5	1.656701400100495E-04	3.763180720674607E-07	3.703652200388360E-07
x_6	1.293417357824663E-03	1.844925109281104E-07	2.085321794461283E-07
x_7	7.091661088782333E-06	4.996387774070343E-06	4.999967170952946E-06
x_8	7.052716122205029E-04	1.434862223559040E-05	1.439270870524103E-05
x_9	5.358602974230222E-04	- 2.040883524405975E-07	- 2.282373401823797E-07
x_{10}	- 1.552259651139891E-03	4.965651547579558E-06	4.956818525791918E-06
$f_1(x)$	- 4.827387619045422E-11	9.657869664593532E-17	0
$f_2(x)$	- 2.767804727950166E-11	- 3.609037981661123E-17	0
$f_3(x)$	- 5.609231013833449E-11	- 5.014435047745458E-18	0
$f_4(x)$	1.436725855439561E-11	5.511287633945886E-16	1.694065894508601E-21
$f_5(x)$	8.470984471899497E-12	5.264730829992297E-16	1.728855061170187E-21
$f_6(x)$	- 3.447739930109454E-12	6.835193007961348E-17	1.049095379878782E-21
$f_7(x)$	- 1.750006423022506E-11	- 5.218880731519356E-17	- 4.028719824372849E-22
$f_8(x)$	- 3.801514721037196E-11	- 1.388394783354915E-16	2.272081369646687E-21
$f_9(x)$	3.145549313765976E-11	5.544608392318181E-16	- 3.954927090396494E-22
$f_{10}(x)$	1.096548967520139E-17	9.105212626901849E-21	8.907628343995374E-21
Mean nf	59380.20	93172.65	129106.13
%SD	4.13	4.11	4.23

of behaviors at all 3 different VTR values. From these 3 tables, we can conclude that the proposed method DE-R can give more accurate solutions by setting higher precision values of VTR. From $VTR = 10^{-20}$ to 10^{-30} and from 10^{-30} to 10^{-40}, DE-R requires greater numbers of function evaluations. The increased percentages of Means are 35.83% and 35.04% for Neurophysiology application. For Combustion application, they are 59.91% and 60.51%. And for SINQUAD function, they are 41.08% and 58.25%, respectively.

5.2. Performance comparison of DE-R with PSO and PSO-NMM methods

The performance comparison of DE-R, PSO variants, and PSO-NMM variants on 4 nonlinear benchmark problems are presented in Table 15 and Table 16. The DE-R is the same as DE59-50-R in the first experiment except that it uses the setting of $maxnf$ as described in Section 4.2.

Both tables report Min, Mean and SD of $fbest$ based on the corresponding $maxnf$'s. From the results, the first two problems are relatively easy problems whereas the last two problems are more difficult problems with the Chemical equilibrium

Table 14. Solutions of SINQUAD function at $VTR = 10^{-20}$, 10^{-30} and 10^{-40}.

	Solution at $VTR = 10^{-20}$	Solution at $VTR = 10^{-30}$	Solution at $VTR = 10^{-40}$
x_1	1.000001313473061E+00	1.000000000741824E+00	9.999999999921714E-01
x_2	- 1.000001313492704E+00	- 1.000000000741824E+00	- 9.999999999921714E-01
x_3	2.357863034574390E-01	2.357835623069449E-01	2.357835607415836E-01
x_4	2.357863035031185E-01	2.357835623069448E-01	2.357835607415836E-01
x_5	2.357863034047495E-01	2.357835623069443E-01	2.357835607415836E-01
x_6	- 1.000001313528972E+00	- 1.000000000741824E+00	- 9.999999999921714E-01
x_7	2.357863034339120E-01	2.357835623069443E-01	2.357835607415836E-01
x_8	2.357863034107586E-01	2.357835623069438E-01	2.357835607415836E-01
x_9	- 1.000001313470187E+00	- 1.000000000741824E+00	- 9.999999999921714E-01
x_{10}	- 1.000001313474019E+00	- 1.000000000741824E+00	- 9.999999999921714E-01
$f_1(x)$	1.725211482508887E-12	5.503030808840126E-19	6.128739500341215E-23
$f_2(x)$	2.059996617731485E-11	0	0
$f_3(x)$	1.252403736273777E-11	4.232725281383409E-16	0
$f_4(x)$	4.908356360244781E-11	3.677613769070831E-16	0
$f_5(x)$	- 2.964588990961303E-11	- 1.179611963664229E-16	0
$f_6(x)$	5.686873194576947E-11	0	0
$f_7(x)$	- 6.305747590751309E-12	- 1.179611963664229E-16	0
$f_8(x)$	- 2.483665356711739E-11	- 4.371503159461554E-16	0
$f_9(x)$	- 1.916689029712870E-12	0	0
$f_{10}(x)$	1.914912672873470E-12	0	0
Mean nf	81755.37	115341.27	182524.47
%SD	8.90	6.03	18.83

Table 15. Performance comparison of PSO, PSO-NMM and DE-R methods on Neurophysiology and Economics modeling applications averaged over 100 independent runs.

Problems	$maxnf$	Methods	Min	Mean	SD
Neurophysiology application	50000	PSO-12	1.01E-29	1.41E-11	1.11E-10
		PSO-15	0	1.48E-11	1.39E-10
		DE-R	3.75E-36	**1.06E-13**	1.03E-12
	100000	PSO-16	0	2.37E-10	2.34E-09
		DE-R	0	**1.52E-37**	1.49E-36
	200000	PSO-NMM-12	0	6.14E-30	3.71E-29
		PSO-NMM-15	0	3.09E-32	1.27E-31
		PSO-NMM-16	0	2.36E-32	1.16E-31
		DE-R	0	**5.39E-96**	5.39E-95
Economics modeling application	50000	PSO-12	2.77E-30	2.52E-27	4.08E-27
		PSO-15	2.47E-33	**1.03E-32**	4.56E-33
		DE-R	7.40E-33	1.36E-32	4.30E-33
	100000	PSO-16	2.47E-33	**9.69E-33**	4.32E-33
		DE-R	7.40E-33	1.35D-32	4.41E-33
	200000	PSO-NMM-12	4.93E-33	4.91E-32	4.13E-32
		PSO-NMM-15	2.47E-33	1.02E-32	4.46E-33
		PSO-NMM-16	2.47E-33	**9.60E-33**	4.24E-33
		DE-R	7.40E-33	1.37E-32	4.15E-33

application as the most difficult one. The DE-R gives the best Mean values for all cases of Neurophysiology application, Chemical equilibrium application, and Combustion application. It shows much better performances on Chemical equilibrium application and Combustion application, especially on Chemical equilibrium application where it gives the Mean values in the order of 10^{-33} while the PSO and PSO-NMM give the values in the order of 10^{-4}. For the Economics modeling application, all methods produce nearly the same results with PSO-15, PSO-16 and PSO-NMM-16 giving only slightly better results. Thus, we can conclude that the DE-R outperforms all the compared methods.

6. Conclusions

In this paper, we have proposed an efficient improvement of the differential evolution algorithm

Table 16. Performance comparison of PSO, PSO-NMM and DE-R methods on Chemical equilibrium and Combustion applications averaged over 100 independent runs.

Problems	$maxnf$	Methods	Min	Mean	SD
Chemical equilibrium application	50000	PSO-12	9.84E-06	5.79E-04	3.04E-03
		PSO-15	3.55E-06	1.61E-03	5.10E-03
		DE-R	1.54E-34	**1.05E-33**	1.34E-33
	100000	PSO-16	4.16E-07	5.73E-04	3.05E-03
		DE-R	1.54E-34	**1.08E-33**	1.33E-33
	250000	PSO-NMM-12	2.58E-34	5.33E-04	3.04E-03
		PSO-NMM-15	1.22E-34	1.60E-03	5.11E-03
		DE-R	1.54E-34	**9.19E-34**	1.26E-33
	300000	PSO-NMM-16	5.93E-34	5.33E-04	3.04E-03
		DE-R	1.54E-34	**1.19E-33**	1.60E-33
Combustion application	50000	PSO-12	3.08E-11	2.76E-08	4.34E-08
		PSO-15	1.01E-11	4.88E-09	8.68E-09
		DE-R	3.68E-21	**3.56E-18**	9.62E-18
	100000	PSO-16	3.99E-12	2.22E-09	4.21E-09
		DE-R	9.83E-37	**4.41E-22**	1.87E-21
	250000	PSO-NMM-12	1.45E-33	4.69E-17	1.04E-16
		PSO-NMM-15	1.28E-33	6.91E-17	2.37E-16
		DE-R	1.29E-42	**2.43E-23**	1.17E-22
	300000	PSO-NMM-16	4.69E-34	5.05E-17	1.09E-16
		DE-R	2.49E-44	**7.80E-24**	3.34E-23

by using a mixing scheme of two mutation operations and a restart technique for solving the nonlinear systems. The designed algorithm has the advantage of integrating both the global and local search techniques to balance the exploration and exploitation. It can successfully solve all ten selected test problems with varying degrees of difficulty and outperforms the two basic differential evolution algorithms using the recommended setting from the literature. It also outperforms the compared methods recently developed in the literature. This performance results from the proper modification to the basic DE algorithms and shows that the DE algorithm with the restart technique is a promising tool for solving complex systems of nonlinear equations. Future study could investigate on designing and applying the differential evolution algorithms to more complicated nonlinear problems in high dimensions such as those derived from difficult nonlinear ODEs and PDEs, and those from learning models of artificial neural networks.

Acknowledgements

References

[1] Boussaïd, I., Lepagnot, J., & Siarry, P. (2013). A survey on optimization metaheuristics. *Information sciences*, 237, 82–117 .

[2] Siddique, N., & Adeli, H. (2015). Nature inspired computing: An overview and some future directions. *Cogn Comput*, 7, 706–714.

[3] Nanda, S. J., & Panda, G. (2014). A survey on nature inspired metaheuristic algorithms for partitional clustering. *Swarm and Evolutionary Computation*, 16, 1—18.

[4] José-García, A., & Gómez-Flores, W. (2016). Automatic clustering using nature-inspired metaheuristics: A survey. *Applied Soft Computing*, 41, 192—213.

[5] Hamm, L., Brorsen, B. W., & Hagan, M. T. (2007). Comparison of Stochastic Global Optimization Methods to Estimate Neural Network Weights. *Neural Process Lett*, 26, 145—158.

[6] Piotrowski, A. P. (2014). Differential evolution algorithms applied to neural network training suffer from stagnation. *Applied Soft Computing*, 21, 382—406.

[7] Raja, M. A. Z., Umar, M., Sabir, Z., Khan, J. A., & Baleanu, D. (2018). A new stochastic computing paradigm for the dynamics of nonlinear singular heat conduction model of the human head. *Eur. Phys. J. Plus*, 133(364), DOI 10.1140/epjp/i2018-12153-4.

[8] Sabir, Z., Manzar, M. A., Raja, M. A. Z., Sheraz, M., & Wazwaz, A. M. (2018). Neuroheuristics for nonlinear singular Thomas-Fermi systems. *Applied Soft Computing*, 65, 152-169.

[9] Raja, M. A. Z., Shah, Z., Manzar, M. A., Ahmad, I., Awais, M.,& Baleanu, D. (2018). A new stochastic computing paradigm for nonlinear Painlevé II systems in applications of random matrix theory. *Eur. Phys. J. Plus*, 133(254), DOI 10.1140/epjp/i2018-12080-4.

[10] Raja, M. A. Z., Zameer, A., Kiani, A. K., Shehzad, A., & Khan, A. R. (2018). Nature-inspired computational intelligence integration with Nelder-Mead method to solve nonlinear benchmark models. *Neural Comput & Applic*, 29, 1169-1193.

[11] Ahmad, I., Zahid, H., Ahmad, F., Raja, M. A. Z., & Baleanu, D. (2019). Design of computational intelligent procedure for thermal analysis of porous fin model. *Chinese Journal of Physics*, 59, 641-655.

[12] Broyden, C. G. (1965). A class of methods for solving nonlinear simultaneous equations. *Mathematics of computation*, 19(92), 577–593.

[13] Martínez, J. M. (1994). Algorithms for solving nonlinear systems of equations. *In* : E. Spedicato, ed. *Algorithms for Continuous Optimization-The state of the art.* Kluwer Academic Publishers, London, 81–108.

[14] Kelley, C. T. (1995). *Iterative methods for solving linear and nonlinear equations.* SIAM, Philadelphia.

[15] Dennis, J. E., & Schnabel, R. B. (1996). *Numerical methods for unconstrained optimization and nonlinear equations.* SIAM, Philadelphia.

[16] Ortega, J. M., & Rheinboldt, W. C. (2000). *Iterative solution of nonlinear equations in several variables.* SIAM, Philadelphia.

[17] Kelley, C. T. (2003). *Solving nonlinear equations with Newton's method.* SIAM, Philadelphia.

[18] Karr, C. L., Weck, B., & Freeman, L. M. (1998). Solutions to systems of nonlinear equations via a genetic algorithm. *Eng. Appl. Artif. Intell.*, 11(3), 369–375.

[19] Grosan, C., & Abraham, A. (2008). A new approach for solving nonlinear equations systems. *IEEE Transactions on Systems Man and Cybernetics-Part A: Systems and Humans*, 38(3), 698–714.

[20] Hirsch M. J., Pardalos, P. M., & Resende, M. G. C. (2009). Solving systems of nonlinear equations with continuous GRASP. *Nonlinear Analysis: Real World Applications*, 10, 2000–2006.

[21] Jaberipour, M., Khorram, E., & Karimi, B. (2011). Particle swarm algorithm for solving systems of nonlinear equations. *Computers and Mathematics with Applications*, 62(2), 566–576.

[22] Pourjafari, E., & Mojallali, H. (2012). Solving nonlinear equations systems with a new approach based on invasive weed optimization algorithm and clustering. *Swarm and Evolutionary Computation*, 4, 33–43.

[23] Oliveira, H. A., & Petraglia, A. (2013). Solving nonlinear systems of functional equations with fuzzy adaptive simulated annealing. *Applied Soft Computing*, 13, 4349–4357.

[24] Raja, M. A. Z., Kiani, A. K., Shehzad, A., & Zameer, A. (2016). Memetic computing through bio-inspired heuristics integration with sequential quadratic programming for nonlinear systems arising in different physical model. *SpringerPlus*, 5:2063, DOI 10.1186/s40064-016-3750-8.

[25] Zhang, X., Wan, Q., & Fan, Y. (2019). Applying modified cuckoo search algorithm for solving systems of nonlinear equations. *Neural Comput & Applic*, 31, 553–576.

[26] Storn, R., & Price, K. (1995). *Differential evolution—a simple and efficient adaptive scheme for global optimization over continuous spaces.* Technical Report TR-95-012, ICSI, Berkeley.

[27] Storn, R., & Price, K. (1997). Differential evolution: A simple and efficient heuristic for global optimization over continuous spaces. *J Glob Optim*, 11(4), 341–359.

[28] Storn, R. (2008). Differential evolution research-Trends and open questions. *In* : U. K. Chakraborty, ed. *Advances in Differential Evolution.* Springer, Berlin, 1–31.

[29] Neri, F., & Tirronen, V. (2010). Recent advances in differential evolution: a survey and experimental analysis. *Artif Intell Rev*, 33, 61–106.

[30] Das, S., & Suganthan, P. N. (2011). Differential evolution: a survey of the state-of-the-art. *IEEE Trans Evol Comput*, 15(1), 4–31.

[31] Lampinen, J., & Zelinka, I. (2000). On stagnation of the differential evolution algorithm. *Proceedings of the 6th international Mendel conference on soft computing*, 76–83.

[32] Gämperle, R., Müller, S. D., & Koumoutsakos, P. (2002). A parameter study for differential evolution. *Proceedings of the conference in neural networks and applications (NNA), fuzzy sets and fuzzy systems (FSFS) and evolutionary computation (EC), WSEAS*, 293–298.

[33] Fan, H. Y., & Lampinen, J. (2003). A trigonometric mutation operation to differential evolution. *J Glob Optim*, 27(1), 105–129.

[34] Das, S., Konar, A., & Chakraborty, U. K. (2005). Two improved differential evolution schemes for faster global search. *ACM-SIGEVO Proceedings of genetic and evolutionary computation conference*, 991–998.

[35] Kaelo, P., & Ali, M. M. (2007). Differential evolution algorithms using hybrid mutation. *Comput Optim Appl*, 37, 231–246.

[36] Das, S., Abraham, A., Chakraborty, U. K., & Konar, A. (2009). Differential evolution with a neighborhood-based mutation operator. *IEEE Trans Evol Comput*, 13(3), 526–553.

[37] Neri, F., & Tirronen, V. (2009). Scale factor local search in differential evolution. *Memet Comput J*, 1(2), 153–171.

[38] Qin, A. K., & Suganthan, P. N. (2005). Self-adaptive differential evolution algorithm for numerical optimization. *Proceedings of the IEEE congress on evolutionary computation*, 1785–1791.

[39] Salman, A., Engelbrecht, A. P., & Omran, M. G. (2007). Empirical analysis of self-adaptive differential evolution. *Eur J Oper Res*, 183(2), 785–804.

[40] Soliman, O. S., & Bui, L. T. (2008). A self-adaptive strategy for controlling parameters in differential evolution. *Proceedings of the IEEE congress on evolutionary computation*, 2837–2842.

[41] Yang, Z., Tang, K., & Yao, X. (2008). Self-adaptive differential evolution with neighborhood search. *Proceedings of the world congress on computational intelligence*, 1110–1116.

[42] Qin, A. K., Huang, V. L., & Suganthan, P. N. (2009). Differential evolution algorithm with strategy adaptation for global numerical optimization. *IEEE Trans Evol Comput*, 13(2), 398–417.

[43] Zaharie, D. (2002). Critical values for control parameters of differential evolution algorithm. *Proceedings of the 8th international Mendel conference on soft computing*, 62–67.

[44] Zaharie, D. (2003). Control of population diversity and adaptation in differential evolution algorithms. *Proceedings of the 9th international Mendel conference on soft computing*, 41–46.

[45] Verschelde, J., Verlinden, P., & Cools, R. (1994). Homotopies exploiting newton polytopes for solving sparse polynomial systems. *SIAM J. Numer. Anal.*, 31, 915–930.

[46] Morgan, A., & Shapiro, V. (1987). Box-bisection for solving second-degree systems and the problem of clustering. *ACM Transaction on Mathematical Software*, 13, 152–167.

[47] Pramanik, S. (2002). Kinematic synthesis of a six-member mechanism for automotive steering. *ASME Journal of Mechanical Design*, 124, 642–645.

[48] Meintjes, K., & Morgan, A. (1990). Chemical equilibrium systems as numerical test problems. *ACM Transaction on Mathematical Software*, 16, 143–151.

[49] More, J., Garbow, B., & Hillstrom, K. (1981). Testing unconstrained optimization software. *ACM Transaction on Mathematical Software*, 7, 17–41.

[50] Bongartz, I., Conn, A., Gould, N., & Toint, Ph. (1995). CUTE: constrained and unconstrained testing environment. *ACM Transactions on Mathematical Software*, 21, 123–160.

Jeerayut Wetweerapong *completed M.Sc. degree in Mathematics from West Virginia University, US in 1995 and Ph.D. degree in Mathematics from Khon Kaen University, Thailand in 2012. He has been teaching and doing research in field of scientific computing and optimization at Department of Mathematics, Khon Kaen University.*
🆔 *http://orcid.org/0000-0001-5053-3989*

Pikul Puphasuk *completed M.Sc. degree in Mathematics from Khon Kaen University, Thailand in 2002 and Ph.D. degree in Applied Mathematics from Suranaree University of Technology, Thailand in 2009. She is an assistant professor at Department of Mathematics, Khon Kaen University. Her research areas include computational sciences, numerical analysis and optimization.*
🆔 *http://orcid.org/0000-0001-9069-1703*

RESEARCH ARTICLE

Using genetic algorithms for estimating Weibull parameters with application to wind speed

Melih Burak Koca [iD], Muhammet Burak Kılıç *[iD], Yusuf Şahin [iD]

Department of Business Administration, Burdur Mehmet Akif Ersoy University, Turkey
mbkoca@mehmetakif.edu.tr, mburak@mehmetakif.edu.tr, ysahin@mehmetakif.edu.tr

ARTICLE INFO	ABSTRACT
	Renewable energy has become a prominent subject for researchers since fossil fuel reserves have been decreasing and are not promising to meet the energy demand of the future. Wind takes an important place in renewable energy resources and there is extensive research on wind speed modeling. Herein, one of the most commonly used distributions for wind speed modeling is the Weibull distribution with its simplicity and flexibility. Maximum likelihood (ML) method is the most frequently used technique in Weibull parameter estimation. Iterative techniques such as Newton-Raphson (NR) use random initial values to obtain the ML estimators of the parameters of the Weibull distribution. Therefore, the success of the iterative techniques highly depends on the initial value selection. In order to deliver a solution to the initial value problem, genetic algorithm (GA) is considered to obtain the estimators of the model parameters. The ML estimators obtained using the GA and NR techniques are compared with the method of moments (MoM) estimators via Monte Carlo simulation and wind speed applications. The results show that the ML estimators obtained using GA present superiority over MoM and the ML estimators obtained using NR.

1. Introduction

The increase in population and the inadequacy of existing energy resources put the human being into the search of alternative energy resources over the course of human history. In the last decades, there is extensive research on renewable energy due to the decrease in fossil fuel reserves and the increase in environmental awareness. As a clean and never-ending resource, the wind has become an important energy resource and distinguished among the other renewable energy forms such as geothermal energy, hydro energy, solar energy, and biomass energy.

Converting the kinetic energy carried by wind to electrical energy is a clean and economical way to produce energy. Once the wind plant is set up, the maintenance cost is relatively low compared to other energy plants. However, the wind turbines and installation costs are high, therefore the wind energy potential of a region should be carefully estimated to determine the proper turbine type. Wind speed is the key factor in determining the wind energy potential of a region [1-4]. Statistical distributions are used to model wind speed and estimate energy potential. The

Weibull distribution is one of the most commonly used distributions in wind energy studies due to its simplicity and flexibility [1, 2, 4-10].

There are various techniques used in Weibull parameter estimation. Sohoni et al. [2] estimated the Weibull parameters using the method of moments (MoM). Seguro and Lambert [5] employed MoM, maximum likelihood (ML) method and modified maximum likelihood (MML) methods. They found that the ML method is more appropriate for the data sets in time series format. For the data sets in frequency distribution format, they recommended using MML method. Akgül et al. [6] compared the least square method, ML method and MML method. Although they found that ML is the most efficient method in overall, they mentioned that ML and MML has a similar efficiency for the large data sets, however, MML has less computational complexity. Arslan et al. [8] compared MoM, L-Moments (L-Mom) method and ML method, and showed that L-Mom method is more efficient for small data sets where ML method is more efficient for larger data sets. Kaplan [10] found that graphical method provides more efficiency than MoM in Weibull

---

*Corresponding author

parameter estimation. Kollu et al. [11], Akpınar and Akpınar [12] used the ML method to estimate Weibull parameters in their studies. Teimouri et al. [13] compared their proposed L-moment estimator with several methods including the ML method, method of logarithmic moment, percentile method and MoM. They found that their proposed method and the ML method are the most efficient estimators. Akdağ and Dinler [14] proposed the power density method. They found it superior to commonly used methods including MoM and ML method. Saleh et al. [15] compared five different methods and recommended the mean wind speed method and the ML method for fitting Weibull distribution. Azad et al. [16] found MoM and ML method more efficient among several methods. Recently, Usta et al. [17] proposed a new estimation approach based on moments for estimating the Weibull parameters.

It is seen from the previous studies that the ML method is one of the most frequently used parameter estimation methods for the Weibull distribution. Due to the nonlinear nature of the log-likelihood function of the Weibull distribution, numerical methods such as Newton-Raphson (NR) should be employed. However, when the iterative techniques are employed, the success of the technique highly depends on the initial value selection. This study departs from the literature by delivering a solution to the initial value problem by using genetic algorithms (GA), which is a heuristic search algorithm and uses a set of solution (search space) instead of single points, for ML estimation of the Weibull parameters. GA is a useful approach in the solution of optimization problems and applied in various studies such as signal control optimization [18] or optimization of mixture parameters of high-performance concrete [19]. In parameter estimation, GA was previously used for negative binomial gamma mixture distribution [20], skew-normal distribution [21] and nonlinear regression [22]. Parameter estimation of Weibull distribution using GA was introduced by Thomas et al. [23] for breakdown times of insulating fluid dataset. GA presented a comparable good performance based on the maximization of the log-likelihood function. With this motivation, the applicability of GA is used in wind speed data modeling. To the best our knowledge, this is the first time GA is used to estimate the parameters of Weibull distribution in wind speed distribution modeling. Observations were obtained from an existing wind farm and different meteorological stations. The efficiency of ML method estimation using GA was compared with ML estimation using NR, and MoM. Mean absolute error (MAE), bias and Kolmogorov-Smirnov (K-S) test were used as decision criteria. The remainder of this paper is structured as follows: Section 2 gives basic information about the Weibull distribution, Section 3 gives detailed information about the parameter estimation methods, Section 4 presents the simulation experiments and wind speed data analysiṣ. Section 5 includes the conclusion.

2. Weibull distribution

The probability density function (pdf) and cumulative distribution function (cdf) of Weibull distribution are respectively given by:

$$f(v; k, c) = \frac{k}{c}\left(\frac{v}{c}\right)^{k-1} \exp\left[-\left(\frac{v}{c}\right)^k\right], \ v, k, c > 0 \quad (1)$$

and

$$F(v; k, c) = 1 - \exp\left[-\left(\frac{v}{c}\right)^k\right], \ v, k, c > 0 \quad (2)$$

where v is the wind speed, k and c are the Weibull shape and scale (dimensionless) parameters respectively. Probability density plots for some different parameter values are given in Figure 1.
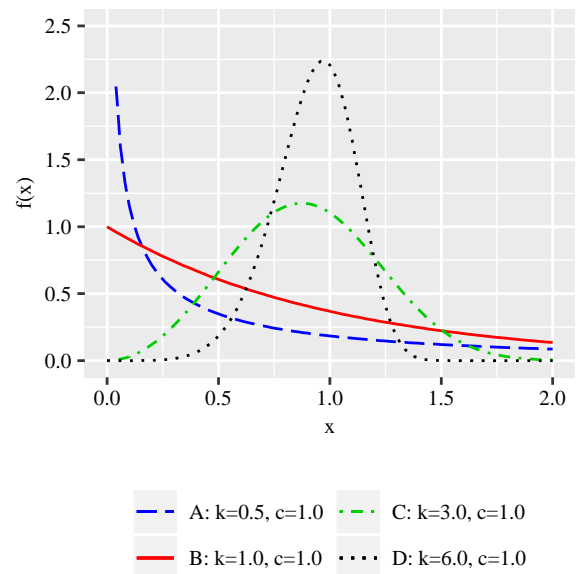

Figure 1. Probability density plots of the Weibull distribution for different parameters.

3. Parameter estimation methods

3.1. Method of moments estimation

MoM is based on equating sample moments with theoretical moments of respective distribution. To estimate the parameters of the Weibull distribution, coefficient of variation of the sample should be calculated and set equal to the theoretical coefficient of variation as follows [8]:

$$\widehat{CV}_{MoM} = \left[\frac{(\sum_{i=1}^n v_i^2)n}{(\sum_{i=1}^n v_i)^2} - 1\right] = \left[\frac{\Gamma\left(\frac{2}{k}+1\right)}{\left[\Gamma\left(\frac{1}{k}+1\right)\right]^2} - 1\right] \quad (3)$$

where n is the number of data points, Γ is the gamma function. When the shape parameter k is obtained from the Equation (3), scale parameter c can be calculated by:

$$\hat{c} = \left[\frac{\frac{1}{n}(\sum_{i=1}^n v_i)}{\Gamma\left(\frac{1}{k}+1\right)}\right]. \quad (4)$$

3.2. Maximum likelihood estimation

The ML method is based on the maximization of the log-likelihood function of the underlying distribution. The log-likelihood function of the Weibull distribution is given as follows:

$$\ln L(v; k, c) = n \ln k - nk \ln c$$
$$+ (k-1) \sum_{i=1}^{n} \ln v_i \qquad (5)$$
$$- c^{-k} \sum_{i=1}^{n} v_i^{k}.$$

By maximizing the log-likelihood function, taking derivative respect to each of the parameters and equating them to zero, the ML estimators of the shape and the scale parameters will be obtained as follows:

$$\hat{k} = \left[\frac{\sum_{i=1}^{n} v_i^{\hat{k}} \ln v_i}{\sum_{i=1}^{n} v_i^{\hat{k}}} - \frac{\sum_{i=1}^{n} \ln v_i}{n} \right]^{-1} \qquad (6)$$

and

$$\hat{c} = \left[\frac{1}{n} \sum_{i=1}^{n} v_i^{\hat{k}} \right]^{\frac{1}{\hat{k}}}. \qquad (7)$$

ML estimator of the shape parameter k includes nonlinear function, therefore, it can be solved by numerical techniques such as NR algorithm, Nelder-Mead algorithm, simulated annealing algorithm or GA. In this study, we used the NR algorithm and the GA in the maximization of the log-likelihood function given in Equation (5).

3.2.1. Newton-Raphson algorithm

The steps of the NR algorithm are summarized in [21] as follows:

1. Determine the initial values $k^{(0)}$ and $c^{(0)}$ for k and c.

2. Compute the vector $U(k^{(m)}, c^{(m)})$ and $V(k^{(m)}, c^{(m)})$ for $m = 0, 1, \dots$ where U and V are defined by:

$$U = \left(\frac{\partial \ln L}{\partial k}, \frac{\partial \ln L}{\partial c} \right)$$

and

$$V = \begin{bmatrix} \frac{\partial^2 \ln L}{\partial k^2} & \frac{\partial^2 \ln L}{\partial k \partial c} \\ \frac{\partial^2 \ln L}{\partial k \partial c} & \frac{\partial^2 \ln L}{\partial c^2} \end{bmatrix}.$$

3. Compute the values of k and c at $(m+1)$th iteration by using the following equation:

$$\begin{bmatrix} k^{(m+1)} \\ c^{(m+1)} \end{bmatrix}$$
$$= \begin{bmatrix} k^{(m)} \\ c^{(m)} \end{bmatrix} - V^{-1}(k^{(m)}, c^{(m)}) U(k^{(m)}, c^{(m)})$$

4. Repeat the iterations until the convergence criterion is satisfied.

NR is a fast-converging powerful algorithm, however, it is dependent on the initial guess. Therefore, we considered the GA in the maximization of the log-likelihood function of the Weibull distribution.

3.2.2. Genetic algorithm

GA is a heuristic search algorithm motivated by the principles of biological evolution of species, to obtain the estimators of the model parameters. Unlike the conventional optimization techniques, GA uses a set of initial solutions which are called as chromosome. A flowchart of GA is presented in Figure 2. The steps of the GA in this study are summarized as follows:

1. A range of possible solutions (search space) was defined as arbitrarily for both shape and scale parameters. A sensitivity analysis was carried out to determine the initial population size where it was taken 6, 10, 15, and 20 respectively. Most efficient outcomes were obtained when the initial population size was set to 6, therefore, initial population size was set to 6.

2. Each set of possible solutions is evaluated using the fitness function. The log-likelihood function of the Weibull distribution is the fitness function in this study.

3. The best solution in each iteration is kept as parent chromosome.

4. New offsprings are reproduced by crossover and mutation with the rate of 0.8 and 0.1 respectively. The size of the population including original parents, crossover and mutation offsprings is equal to the initial population size in step 1.

5. New population is evaluated as in step 2. Steps 3-5 are repeated.

The algorithm stops if the decision criterion is satisfied or the maximum number of iterations is achieved. A flowchart of the study is given in Figure 2.
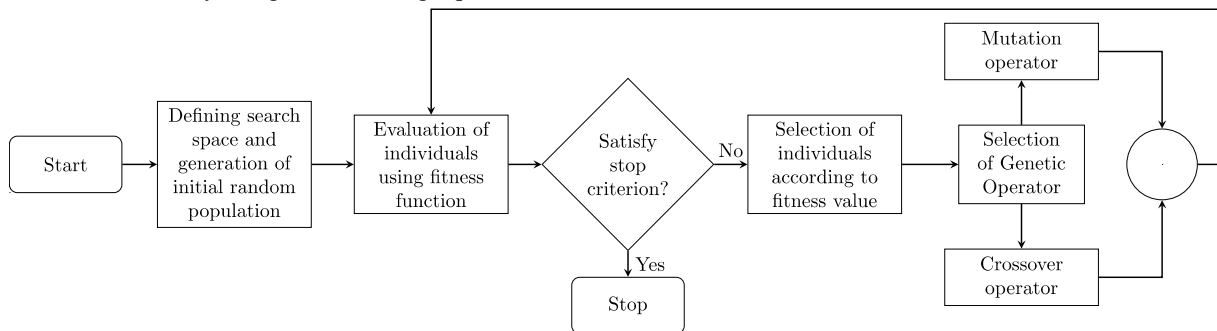

Figure 2. Flowchart of the GA used in this study.

Table 1. Parameter estimations, MAE and bias values for different simulation scenarios.

n	k	Method	$\widehat{k}$ Mean	MAE	Bias	$\widehat{c}$ Mean	MAE	Bias
20	0.5	MoM	0.6510	0.1637	0.1510	1.3549	0.5207	0.3549
		NR	0.5412	0.0840	0.0412	1.0823	0.3938	0.0823
		GA	0.5399	**0.0829**	**0.0399**	1.0570	**0.3685**	**0.0570**
	1	MoM	1.1172	0.1905	0.1172	1.0221	**0.1912**	0.0221
		NR	1.0823	0.1680	0.0823	1.0114	0.1915	0.0114
		GA	1.0822	**0.1679**	**0.0822**	1.0114	0.1915	**0.0114**
	3	MoM	3.2191	0.4729	0.2191	0.9975	0.0623	**-0.0025**
		NR	3.2470	0.5039	0.2470	0.9973	0.0640	-0.0027
		GA	3.1153	**0.3413**	**0.1153**	0.9949	**0.0622**	-0.0051
	6	MoM	6.4952	1.0341	0.4952	0.9977	0.0312	-0.0023
		NR	6.4939	1.0077	0.4939	0.9978	0.0321	**-0.0022**
		GA	6.2305	**0.6825**	**0.2305**	0.9967	**0.0312**	-0.0033
50	0.5	MoM	0.5818	0.1019	0.0818	1.2071	0.3412	0.2071
		NR	0.5172	0.0485	0.0172	1.0449	0.2514	0.0449
		GA	0.5171	**0.0484**	**0.0171**	1.0422	**0.2486**	**0.0422**
	1	MoM	1.0532	0.1144	0.0532	1.0105	**0.1221**	0.0105
		NR	1.0345	0.0971	0.0345	1.0101	0.1235	0.0101
		GA	1.0344	**0.0970**	**0.0344**	1.0097	0.1231	**0.0097**
	3	MoM	3.0866	0.2665	0.0866	0.9993	**0.0391**	-0.0007
		NR	3.1034	0.2912	0.1034	1.0007	0.0411	0.0007
		GA	3.0763	**0.2593**	**0.0763**	1.0000	0.0405	**0.0000**
	6	MoM	6.1854	0.5747	0.1854	0.9993	**0.0196**	-0.0007
		NR	6.2069	0.5824	0.2069	1.0000	0.0205	**0.0000**
		GA	6.1526	**0.5185**	**0.1526**	0.9997	0.0203	-0.0003
100	0.5	MoM	0.5495	0.0748	0.0495	1.1224	0.2469	0.1224
		NR	0.5086	0.0331	0.0086	1.0182	0.1675	0.0182
		GA	0.5085	**0.0330**	**0.0085**	1.0155	**0.1645**	**0.0155**
	1	MoM	1.0282	0.0839	0.0282	1.0034	0.0870	0.0034
		NR	1.0172	0.0662	0.0172	1.0034	0.0832	0.0034
		GA	1.0172	**0.0662**	**0.0172**	1.0034	**0.0832**	**0.0034**
	3	MoM	3.0470	0.1915	0.0470	0.9992	0.0280	-0.0008
		NR	3.0516	0.1985	0.0516	0.9999	0.0277	**-0.0001**
		GA	3.0112	**0.1808**	**0.0112**	0.9860	**0.0140**	-0.0140
	6	MoM	6.0963	0.4120	0.0963	0.9994	0.0140	-0.0006
		NR	6.1032	0.3971	0.1032	0.9998	0.0139	**-0.0002**
		GA	6.0939	**0.3868**	**0.0939**	0.9997	**0.0138**	-0.0003
500	0.5	MoM	0.5151	0.0369	0.0151	1.0339	0.1245	0.0339
		NR	0.5015	0.0140	0.0015	1.0029	0.0752	0.0029
		GA	0.5015	**0.0140**	**0.0015**	1.0029	**0.0752**	**0.0029**
	1	MoM	1.0063	0.0366	0.0063	0.9974	0.0387	-0.0026
		NR	1.0031	0.0280	0.0031	1.0003	0.0376	0.0003
		GA	1.0031	**0.0280**	**0.0031**	1.0003	**0.0376**	**0.0003**
	3	MoM	3.0090	0.0877	**0.0090**	0.9986	0.0126	-0.0014
		NR	3.0092	0.0840	0.0092	0.9999	0.0125	-0.0001
		GA	3.0092	**0.0840**	0.0092	0.9999	**0.0125**	**-0.0001**
	6	MoM	6.0193	0.1887	0.0193	0.9993	0.0063	-0.0007
		NR	6.0184	0.1680	0.0184	0.9999	0.0063	-0.0001
		GA	6.0176	**0.1666**	**0.0176**	0.9999	**0.0062**	**-0.0001**

4. Application

4.1. Monte Carlo simulations

In order to compare the parameter estimation methods for the Weibull distribution, a Monte Carlo simulation was conducted where the shape parameter is taken 0.5, 1, 3 and 6 and the scale parameter was fixed to 1. The parameter sets used in the simulation can also be seen in Figure 1. The simulation was repeated 1000 times for each of the sample sizes of 20, 50, 100 and 500. MoM estimations were considered as the initial values for the NR. For the GA, the population size was chosen 6, mutation rate and crossover rate were fixed to 0.8 and 0.1 respectively. ML estimations using NR and GA were obtained via "maxLik" [24] and "GA" [25]

packages of R software. Mean absolute error (MAE) and bias are chosen as goodness-of-fit criteria for comparing the efficiencies of the parameter estimation methods. MAE and bias for the parameters k and c are given by:

$$MAE(\hat{k}) = \frac{1}{n} \sum_{i=1}^{n} |\hat{k}_i - k|$$

$$bias(\hat{k}) = \frac{1}{n} \sum_{i=1}^{n} (\hat{k}_i - k)$$

(8)

and

$$MAE(\hat{c}) = \frac{1}{n} \sum_{i=1}^{n} |\hat{c}_i - c|$$

$$bias(\hat{c}) = \frac{1}{n} \sum_{i=1}^{n} (\hat{c}_i - c).$$

(9)

Smaller values the absolute value of the bias and MAE indicate higher efficiency. Parameter estimations, absolute value of the bias and MAE for each parameter estimation method can be seen in Table 1. Accordingly,

best results are highlighted in bold.

It is seen from the simulation results that the GA approach was more efficient than NR and MoM in the estimation of the shape and scale parameters according to MAE and bias criteria. For the sample size of 20, 50 and 100, the GA approach provided the best efficiency for the shape parameter in each simulation scenario in terms of MAE and bias. For the sample size of 500, GA also provided the best efficiency for the shape parameter in each simulation scenario according to MAE.

In the estimation of scale parameter for the sample sizes of 20,50 and 100, GA provided the highest efficiency according to at least one of the decision criteria in almost each simulation scenario. For the sample size of 100, GA was the most efficient method in each simulation scenario according to MAE and bias. In overall, it can be said that GA is a very efficient method for small, moderate and large sample sizes. MAE and absolute values of the biases are also presented in Figures 3-6.
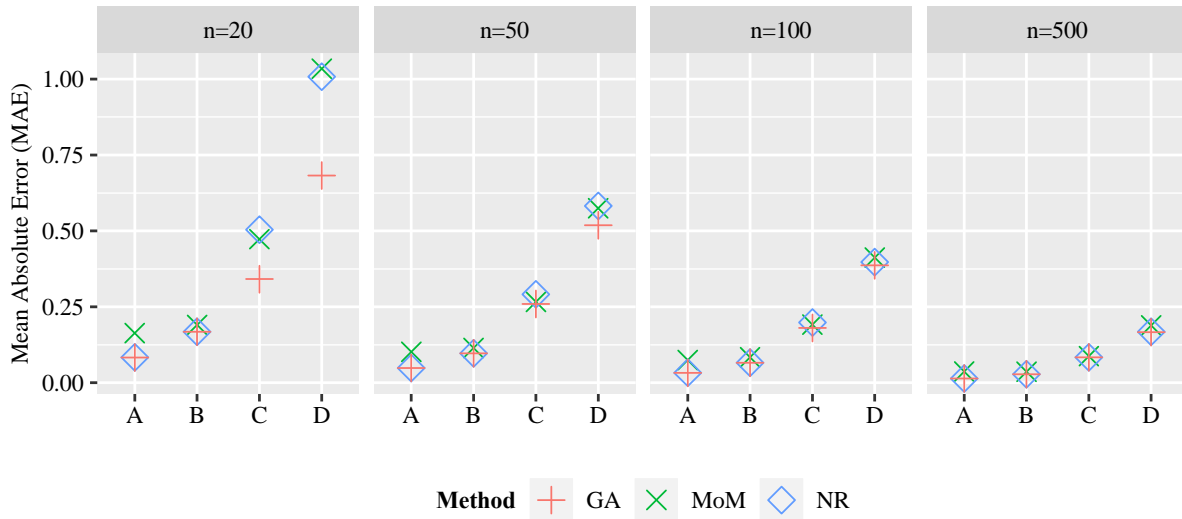

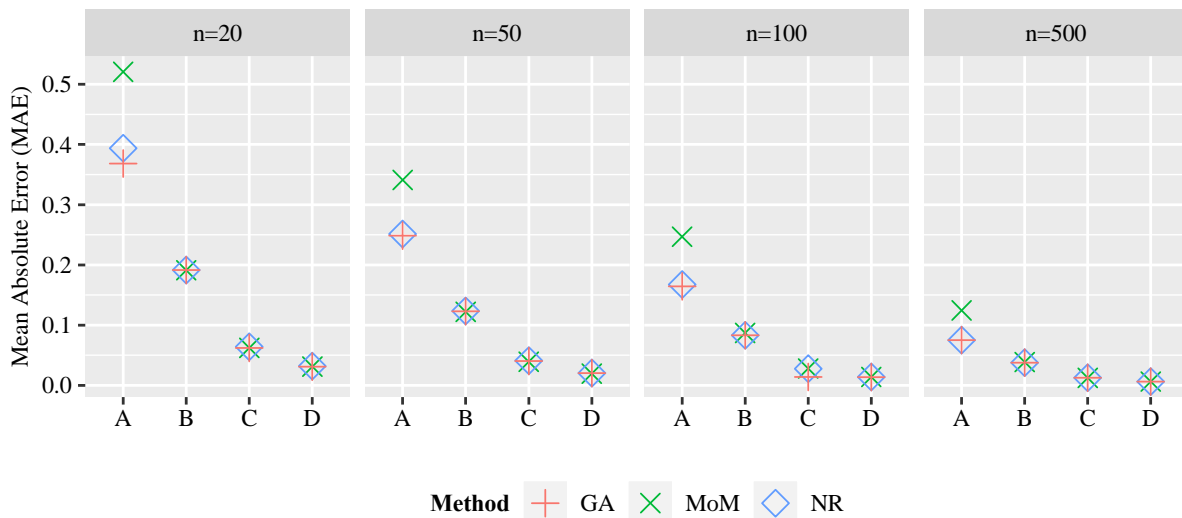**Figure 3.** Comparison of the parameter estimation methods for k according to MAE criterion.

Figure 4. Comparison of the parameter estimation methods for c according to MAE criterion.
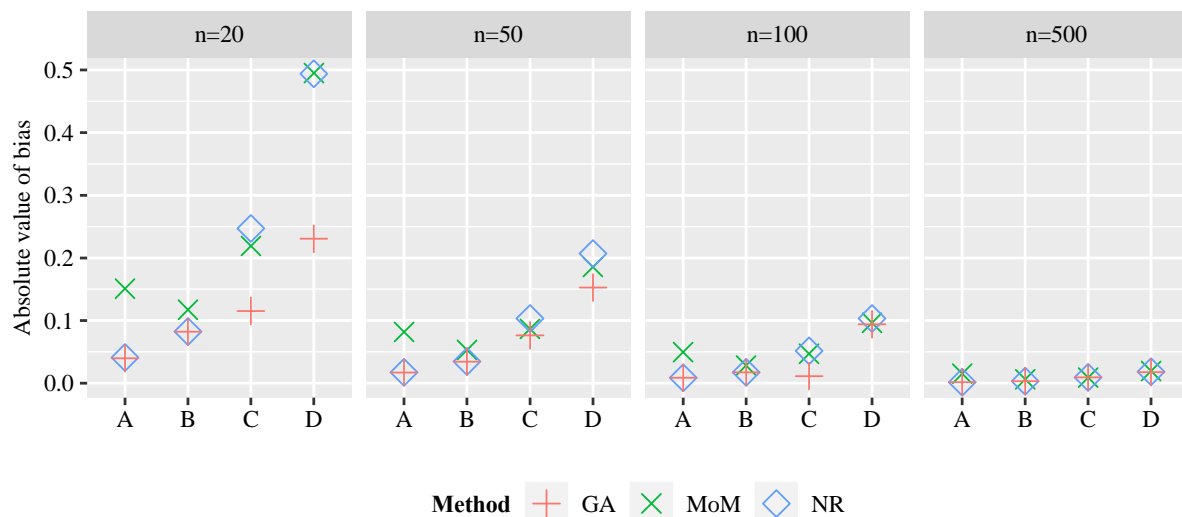
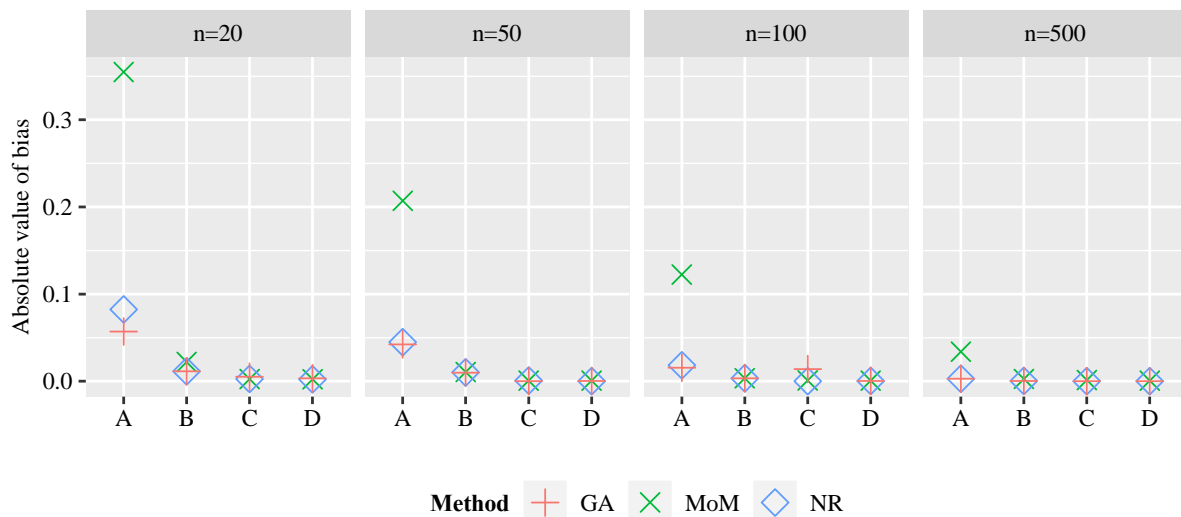**Figure 5.** Comparison of the parameter estimation methods for k according to bias criterion

Figure 6. Comparison of the parameter estimation methods for c according to bias criterion.

Figure 3 presents the MAE values for the shape parameter k. GA presented more efficiency than NR and MoM in all simulation scenarios. NR was the second-best method. MAE values were decreased when the sample size was increased. However, when the value of the shape parameter was increased, MAE values were also increased.

Figure 4 shows the MAE values for the scale parameter c. GA was the most efficient method for the sample sizes of 20, 100 and 500. MoM was the most efficient for the sample size of 50. MAE values were decreased when the value of the shape parameter was increased. Similarly, MAE values were also decreased when the sample size was increased.

Figure 5 presents the absolute value of bias for the shape parameter k. GA presented the most efficient results. MoM presented better results than NR on some occasions. Similar to the MAE values, absolute values of the bias were decreased when the sample size was increased. However, when the value of the shape

parameter was increased, the absolute values of the bias were also increased

Figure 6 shows the absolute values of bias for the scale parameter c. GA was more efficient than other methods for most of the time. NR was the second-best method. With the increase in the value of shape parameter and sample size, absolute values of bias were decreased.

4.2. Wind speed analysis

Wind speed observations obtained from three different locations, namely Belen Wind Farm (Belen), Gökçeada Meteorological Station (Gökçeada) and Datça/Deveboynu Feneri Meteorological Station (Datça) were used for the comparison of the parameter estimation methods. Belen data set was provided by Belen Electric Generation Co. Inc. Gökçeada and Datça data sets were provided by the Turkish State Meteorological Service. Information about the geographical coordinates of the stations, elevation, selected period of observations and collection process

are presented in Table 2. Accordingly, the wind speed data for the Belen Station were observed in 10-min basis. The wind speed data observed at other stations were collected on hourly basis. The descriptive statistics including mean, standard deviation, minimum and maximum for the data sets used in this study are presented in Table 3. It can be seen from Table 3 that the average and the maximum wind speed were observed at Datça station.

Table 2. Geographical coordinates of the stations, selected period of observations and data collection process.

Station	Period of Observations	Collection Basis	Height	Latitude	Longitude	Elevation
Belen	01 Jan 2013 – 31 Dec 2014	10-min	80 m	36°28'42.2"N	36°12'45.0"E	744
Gökçeada	01 Jan 2010 – 31 Dec 2017	Hourly	10 m	40°11'27.6"N	25°54'27.0"E	79
Datça	01 Jan 2014 – 30 Apr 2017	Hourly	10 m	36°41'12.1"N	27°21'47.9"E	28

Table 3. Descriptive statistics for the data sets.

Station	Year	Mean	Std. Dev.	Min.	Max.
Belen	2013	7.2949	3.3345	0.4	24.9
	2014	7.3790	3.2631	0.4	24.9
Gökçeada	2010	4.3956	3.0779	0.1	16.9
	2011	4.5014	2.7362	0.1	16.6
	2012	4.1776	2.6393	0.2	14.9
	2013	3.6880	2.4716	0.1	16.9
	2014	3.7678	2.5150	0.1	14.9
	2015	4.3323	2.6961	0.1	18.0
	2016	4.2892	2.8234	0.1	16.7
	2017	3.8563	2.6384	0.1	15.4
Datça	2014	6.9937	4.5638	0.3	28.9
	2015	7.6496	4.9330	0.4	28.9
	2016	7.8554	5.1781	0.2	33.7
	2017	7.3715	4.8864	0.2	24.2

Weibull distribution is fitted at the monthly base for the Belen, Gökçeada and Datça data sets. To statistically test that monthly data sets come from Weibull distribution, the K-S test is separately applied to each data set.

K-S test is used for testing if a sample distribution belongs to a population with a specific distribution. K-S test statistic is the maximum difference between the empirical distribution $F_0(x)$ and theoretical distribution $S_N(x)$ [26].

$$d = \max |F_0(x) - S_N(x)| \qquad (10)$$

After the K-S test process, monthly distributions that come from Weibull distribution are selected for further analysis (p-value>0.05). The parameter estimates and K-S test results for Belen, Gökçeada and Datça data sets are presented in Tables 4-6 respectively.

Table 4. Parameter estimations and K-S goodness-of-fit test results for Belen data set.

Date	Method	$\hat{k}$	$\hat{c}$	K-S	p-value
2014 - Feb	MoM	2.1104	6.7724	0.0182	0.1381
	NR	2.1191	6.7779	0.0197	0.0875
	GA	2.0984	6.7646	**0.0161**	**0.2438**

Table 4 shows that GA provides the best fit in terms of the K-S test for Belen data set.

Table 5. Parameter estimations and K-S goodness-of-fit test results for Gökçeada data set.

Date	Method	$\hat{k}$	$\hat{c}$	K-S	p-value
2010 - Oct	MoM	1.5527	4.8650	0.0443	0.1074
	NR	1.5571	4.8752	0.0453	0.0947
	GA	1.4954	4.8088	**0.0394**	**0.1990**
2011 - Apr	MoM	1.9337	5.7312	0.0517	0.0423
	NR	1.9002	5.7122	0.0501	0.0537
	GA	1.9023	5.7247	**0.0485**	**0.0673**
2011 - Nov	MoM	1.7368	5.6574	0.0363	0.2998
	NR	1.7307	5.6551	0.0363	0.3004
	GA	1.7299	5.6507	**0.0359**	**0.3144**
2012 - Dec	MoM	1.8973	5.4681	0.0440	0.1123
	NR	1.8750	5.4574	0.0414	0.1562
	GA	1.8737	5.4526	**0.0411**	**0.1615**
2013 - May	MoM	1.6149	4.4635	0.0463	0.0829
	NR	1.6212	4.4682	0.0472	0.0724
	GA	1.6148	4.4237	**0.0449**	**0.1001**
2015 - Jan	MoM	1.5037	5.6700	0.0611	0.0401
	NR	1.5681	5.7215	0.0591	0.0518
	GA	1.5673	5.7129	**0.0584**	**0.0562**
2015 - Feb	MoM	1.7919	6.6341	0.0480	0.0982
	NR	1.7502	6.6019	0.0484	0.0937
	GA	1.7509	6.6086	**0.0478**	**0.1016**
2015 - Apr	MoM	1.4922	5.0242	0.0370	0.3096
	NR	1.4876	5.0285	0.0365	0.3268
	GA	1.4863	5.0104	**0.0358**	**0.3493**
2015 - May	MoM	1.6151	4.0105	0.0524	0.0919
	NR	1.5916	4.0010	0.0488	0.1374
	GA	1.5905	3.9947	**0.0484**	**0.1435**
2015 - Oct	MoM	1.9727	4.9659	0.0514	0.0485
	NR	1.9312	4.9455	0.0487	0.0711
	GA	1.9301	4.9435	**0.0486**	**0.0721**
2015 - Nov	MoM	1.5010	5.2991	**0.0329**	**0.6387**
	NR	1.5066	5.3106	0.0343	0.5861
	GA	1.5034	5.3074	0.0338	0.6027
2016 - Jan	MoM	1.3632	5.1459	**0.0432**	**0.1321**
	NR	1.3710	5.1638	0.0453	0.1004
	GA	1.3683	5.1604	0.0447	0.1089
2016 - Mar	MoM	1.5731	5.4328	0.0466	0.1599
	NR	1.5608	5.4299	0.0448	0.1946
	GA	1.5582	5.4213	**0.0441**	**0.2078**
2016 - May	MoM	1.6266	4.0770	0.0457	0.3642
	NR	1.6111	4.0731	0.0430	0.4399
	GA	1.6109	4.0723	**0.0430**	**0.4422**

Table 5. *(continued)*

	Method	$\hat{k}$	$\hat{c}$	K-S	p-value
2016 - Jun	MoM	1.8951	3.0578	**0.0877**	**0.1657**
	NR	1.8507	3.0452	0.0947	0.1096
	GA	1.8535	3.0461	0.0942	0.1128
2016 - Nov	MoM	1.5976	5.3758	0.0532	0.0355
	NR	1.5571	5.3534	0.0474	0.0810
	GA	1.5569	5.3533	**0.0474**	**0.0814**
2017 - Dec	MoM	1.9966	6.8246	0.0516	0.1153
	NR	1.9906	6.8205	0.0513	0.1186
	GA	1.9909	6.8219	**0.0513**	**0.1197**

Table 6. Parameter estimations and K-S goodness-of-fit test results for Datça data set.

Date	Method	$\hat{k}$	$\hat{c}$	K-S	p-value
2014 - Jan	MoM	1.7162	6.0504	0.0570	0.0756
	NR	1.7787	6.0924	0.0566	0.0790
	GA	1.7672	5.9995	**0.0490**	**0.1783**
2014 - Aug	MoM	1.8009	8.4971	**0.0446**	**0.1031**
	NR	1.8379	8.5336	0.0461	0.0842
	GA	1.8373	8.5273	0.0457	0.0898
2015 - Jan	MoM	1.7307	10.8387	0.0431	0.1255
	NR	1.7483	10.8725	0.0437	0.1172
	GA	1.7501	10.9125	**0.0422**	**0.1408**
2015 - Mar	MoM	1.8631	9.6927	0.0409	0.1662
	NR	1.8842	9.7204	0.0418	0.1491
	GA	1.8683	9.7017	**0.0409**	**0.1660**
2015 - Jun	MoM	1.8422	7.4869	**0.0348**	**0.3482**
	NR	1.8781	7.5196	0.0418	0.1619
	GA	1.8051	7.5222	0.0363	0.2982
2015 - Oct	MoM	2.0635	8.5650	0.0463	0.0828
	NR	2.0843	8.5813	0.0493	0.0539
	GA	2.0452	8.5486	**0.0435**	**0.1198**
2015 - Nov	MoM	1.8870	9.2486	0.0440	0.1237
	NR	1.9013	9.2654	0.0467	0.0863
	GA	1.8814	9.2433	**0.0430**	**0.1398**
2015 - Dec	MoM	2.1032	10.2284	0.0383	0.2301
	NR	2.1241	10.2526	0.0407	0.1748
	GA	2.1181	10.2101	**0.0372**	**0.2585**
2016 - Jan	MoM	1.8773	10.8451	0.0279	0.6078
	NR	1.8728	10.8427	0.0278	0.6138
	GA	1.8451	10.8504	**0.0257**	**0.7112**
2016 - Feb	MoM	1.2831	9.1734	0.0370	0.3162
	NR	1.3318	9.2773	0.0442	0.1439
	GA	1.2793	9.1544	**0.0359**	**0.3519**
2016 - May	MoM	1.6948	8.4204	0.0314	0.4729
	NR	1.7220	8.4530	0.0365	0.2905
	GA	1.7033	8.3964	**0.0308**	**0.4997**
2016 - Sep	MoM	1.6449	8.8944	0.0478	0.0829
	NR	1.6563	8.9189	0.0505	0.0576
	GA	1.6342	8.7143	**0.0433**	**0.1469**
2016 - Oct	MoM	1.7423	9.4553	0.0403	0.1784
	NR	1.7605	9.4816	0.0442	0.1096
	GA	1.7254	9.4345	**0.0368**	**0.2652**
2017 - Mar	MoM	1.6872	7.9891	0.0400	0.1956
	NR	1.7152	8.0254	0.0425	0.1457
	GA	1.6913	8.0015	**0.0399**	**0.1990**

It can be seen from Table 5 that GA provides the highest efficiency in 14 of 17 months in terms of the K-S test results in Gökçeada data set. MoM provides the best fit in 3 months.

Table 6 shows that GA provides the best fit in 12 of 14 months. MoM is the second-best estimator and has the highest efficiency in 2 months for Datça dataset.

5. Conclusion

In this paper, we have obtained the ML estimators of the parameters of Weibull distribution using GA and NR techniques, and compared them with MoM. The efficiencies of the parameter estimation methods are evaluated based on MAE, bias and K-S test criteria. Results of the Monte Carlo simulation and real wind speed data analysis show that ML estimator using GA is more efficient than ML estimator using NR and MoM estimator in Weibull parameter estimation. Furthermore, it can be said that all data sets were observed in different geographical regions with different weather characteristics. GA showed superiority on these data sets including different types of weather conditions. Finally, arbitrary search spaces were used in this study which can be seen as a limitation. In the future works, we will focus on developing a data-based search space in GA for Weibull parameter estimation.

References

[1] Hou, Y., Peng, Y., Johnson, A.L. & Shi, J. (2012). Empirical analysis of wind power potential at multiple heights for North Dakota wind observation sites. *Energy Science and Technology,* 4(1), 1-9. DOI:10.3968/j.est.1923847920120401.289

[2] Sohoni, V., Gupta, S. & Nema, R. (2016). A comparative analysis of wind speed probability distributions for wind power assessment of four sites. *Turkish Journal of Electrical Engineering & Computer Sciences,* 24(6), 4724-4735. DOI:10.3906/elk-1412-207

[3] Turkan, Y.S., Aydogmus, H.Y. & Erdal, H. (2016). The prediction of the wind speed at different heights by machine learning methods. *An International Journal of Optimization and Control: Theories & Applications,* 6(2), 179-197. DOI:10.11121/ijocta.01.2016.00315

[4] Koca, M.B., Kılıç, M.B. & Şahin, Y. Assessing wind energy potential using finite mixture distributions. *Turkish Journal of Electrical Engineering & Computer Sciences,* 27(3), 2276-2294. DOI:10.3906/elk-1802-109

[5] Seguro, J.V., & Lambert, T.W. (2000). Modern estimation of the parameters of the Weibull wind speed distribution for wind energy analysis. *Journal of Wind Engineering and Industrial Aerodynamics*, 85, 75-84. DOI:10.1016/S0167-6105(99)00122-1

[6] Akgül, F.G., Şenoğlu, B., & Arslan, T. (2016). An alternative distribution to Weibull for modeling the wind speed data: Inverse Weibull distribution. *Energy Conversion and Management*, 114, 234-240. DOI:10.1016/j.enconman.2016.02.026

[7] Lun, I.Y.F., & Lam, J.C. (2000). A study of Weibull parameters using long-term wind observations. *Renewable Energy*, 20, 145-153. DOI:10.1016/s0960-1481(99)00103-2

[8] Arslan, T., Bulut, Y.M., & Yavuz, A.A. (2014). Comparative study of numerical methods for determining Weibull parameters for wind speed modeling. *Renewable and Sustainable Energy Reviews*, 40, 820-825. DOI:10.1016/j.ser/2014.08.009

[9] Safari, B. (2011). Modeling wind speed and wind power distributions in Rwanda. *Renewable and Sustainable Energy Reviews*, 15, 925-935. DOI:10.1016/j.ser.2010.11.001

[10] Kaplan, Y.A. (2016). The evaluating of wind energy potential of Osmaniye region with using Weibull and Rayleigh distributions. *Süleyman Demirel University Journal of Natural and Applied Sciences*, 20(1), 62-71. DOI:10.19113/sdufbed.63806

[11] Kollu, R., Rayapudi, S.R., Narasimham, S.V.L., & Pakkurthi, K.M. (2012). Mixture probability distribution functions to model wind speed distributions. *International Journal of Energy and Environmental Engineering*, 3(27). DOI:10.1186/2251-6832-3-27

[12] Akpınar, E.K., & Akpınar, S. (2004). Determination of the wind energy potential for Maden-Elazig, Turkey. *Energy Conversion and Management,* 45, 2901-2914. DOI:10.1016/j.enconman.2003.12.016

[13] Teimouri, M., Hoseini, S.M., & Nadarajah, S. (2013). Comparison of estimation methods for the Weibull distribution. *Statistics,* 47(1), 93-109. DOI:10.1080/02331888.2011.559657

[14] Akdağ, S.A. & Dinler, A. (2009). A new method to estimate Weibull parameters for wind energy applications. *Energy Conversion and Management,* 50, 1761-1766. DOI:10.1016/j.enconman.2009.03.020

[15] Saleh, H., Abou El-Azm Aly, A. & Abdel-Hady, S. (2012). Assessment of different methods used to estimate Weibull distribution parameters for wind speed in Zafarana wind farm, Suez Gold, Egypt. *Energy,* 44, 710-719. DOI:10.1016/j.energy.2012.05.021

[16] Azad. A.K., Rasul, M. G. & Yusaf, T. (2014). Statistical diagnosis of the best Weibull methods for wind power assessment for agricultural applications. *Energies,* 7, 3056-3085. DOI:10.3390/en7053056

[17] Usta, I., Arik, I., Yenilmez, I. & Kantar, Y.M. (2018). A new estimation approach based on moments for estimating Weibull parameters in wind speed power applications. *Energy Conversion and Management,* 164, 570-578. DOI:10.1016/j.enconman.2018.03.033

[18] Tu, T.V. & Sano, K. (2013). Genetic algorithm for optimization in adaptive bus signal priority control. An International Journal of Optimization and Control: Theories & Applications, 3(1), 35-43. DOI:10.11121/ijocta.01.2013.00138

[19] Şimşek, B. & Şimşek, E.H. (2017). Assessment and optimization of thermal and fluidity properties of high strength concrete via genetic algorithm. An International Journal of Optimization and Control: Theories & Applications, 7(1), 90-97. DOI:10.11121/ijocta.01.2017.00345

[20] Gençtürk, Y., & Yiğiter, A. (2016). Modelling claim number using a new mixture model: negative binomial gamma distribution. *Journal of Statistical Computation and Simulation,* 86, 1829-1839. DOI:10.1080/00949655.2015.1085987

[21] Yalçınkaya, A., Şenoğlu, B., & Yolcu, U. (2018). Maximum likelihood estimation for the parameters of skew normal distribution using genetic algorithm. *Swarm and Evolutionary Computation,* 38, 127-138. DOI:10.1016/j.swevo.2017.07.007

[22] Altunkaynak, B., & Esin, A. (2004). The genetic algorithm method for parameters estimation in nonlinear regression. *Gazi University Journal of Science,* 17(2), 43-51.

[23] Thomas, G.M., Gerth, R., Velasco, T., & Rabelo, L.C. (1995). Using real-coded genetic algorithms for Weibull parameter estimation. *Computers & Industrial Engineering,* 29, 377-381. DOI:10.1016/0360-8352(95)00102-7

[24] Henningsen, A., & Toomet, O. (2010). maxLik: A package for maximum likelihood estimation in R. *Computational Statistics,* 26, 443-458. DOI: 10.1007/s00180-010-0217-1

[25] Scrucca, L. (2013). GA: A package for genetic algorithms in R. *Journal of Statistical Software,* 53(4), 1-37. DOI: 10.18637/jss.v053.i04

[26] Massey JR, F.J. (1951). The Kolmogorov-Smirnov test for goodness of fit. *Journal of American Statistical Association,* 46, 68-78. DOI:10.1080/01621459.1951.10500769

Melih Burak Koca *received his M.S. degree in Applied Statistics from Purdue University in 2014. He is currently doing his Ph.D. in Quantitative Methods at Burdur Mehmet Akif Ersoy University.*
  http://orcid.org/0000-0003-2737-3643

Muhammet Burak Kılıç *received his M.S. degree in Statistics from Fırat University in 2011 and his Ph.D. in Statistics from Middle East Technical University in 2015. His research interests are directional statistics, Bayesian*

models, and computational methods.
ⓘⅮ *http://orcid.org/0000-0002-9597-1576*

Yusuf Şahin *received the M.S. degree in industrial engineering from Pamukkale University in 2009 and a PhD degree in business administration from Suleyman Demirel University in 2014. He has been an assistant professor of business administration at Burdur Mehmet Akif Ersoy University since 2014. His field of study includes operations research, logistics, warehouse management, vehicle routing, meta-heuristics, and quantitative models.*
ⓘⅮ *http://orcid.org/0000-0002-3862-6485*

An International Journal of Optimization and Control: Theories & Applications (http://ijocta.balikesir.edu.tr)

INSTRUCTIONS FOR AUTHORS

Aims and Scope

This journal shares the research carried out through different disciplines in regards to optimization, control and their applications.

The basic fields of this journal are linear, nonlinear, stochastic, parametric, discrete and dynamic programming; heuristic algorithms in optimization, control theory, game theory and their applications. Problems such as managerial decisions, time minimization, profit maximizations and other related topics are also shared in this journal.

Besides the research articles expository papers, which are hard to express or model, conference proceedings, book reviews and announcements are also welcome.

Journal Topics

- Applied Mathematics,
- Financial Mathematics,
- Control Theory,
- Game Theory,
- Fractional Calculus,
- Fractional Control,
- Modeling of Bio-systems for Optimization and Control,
- Linear Programming,
- Nonlinear Programming,
- Stochastic Programming,
- Parametric Programming,
- Conic Programming,
- Discrete Programming,
- Dynamic Programming,
- Optimization with Artificial Intelligence,
- Operational Research in Life and Human Sciences,
- Heuristic Algorithms in Optimization,
- Applications Related to Optimization on Engineering.

Submission of Manuscripts

New Submissions

Solicited and contributed manuscripts should be submitted to IJOCTA via the journal's online submission system. You need to make registration prior to submitting a new manuscript (please click here to register and do not forget to define yourself as an "Author" in doing so). You may then click on the "New Submission" link on your User Home.

IMPORTANT: If you already have an account, please click here to login. It is likely that you will have created an account if you have reviewed or authored for the journal in the past.

On the submission page, enter data and answer questions as prompted. Click on the "Next" button on each screen to save your work and advance to the next screen. The names and contact details of at least four internationally recognized experts who can review your manuscript should be entered in the "Comments for the Editor" box.

You will be prompted to upload your files: Click on the "Browse" button and locate the file on your computer. Select the description of the file in the drop down next to the Browse button. When you have selected all files you wish to upload, click the "Upload" button. Review your submission before sending to the Editors. Click the "Submit" button when you are done reviewing. Authors are responsible for verifying all files have uploaded correctly.

You may stop a submission at any phase and save it to submit later. Acknowledgment of receipt of the manuscript by IJOCTA Online Submission System will be sent to the corresponding author, including an assigned manuscript number that should be included in all subsequent correspondence. You can also log-

on to submission web page of IJOCTA any time to check the status of your manuscript. You will receive an e-mail once a decision has been made on your manuscript.

Each manuscript must be accompanied by a statement that it has not been published elsewhere and that it has not been submitted simultaneously for publication elsewhere.

Manuscripts can be prepared using LaTeX (.tex) or MSWord (.docx). However, manuscripts with heavy mathematical content will only be accepted as LaTeX files.

Preferred first submission format (for reviewing purpose only) is Portable Document File (.pdf). Please find below the templates for first submission.

Click here to download Word template for first submission (.docx)

Click here to download LaTeX template for first submission (.tex)

Revised Manuscripts

Revised manuscripts should be submitted via IJOCTA online system to ensure that they are linked to the original submission. It is also necessary to attach a separate file in which a point-by-point explanation is given to the specific points/questions raised by the referees and the corresponding changes made in the revised version.

To upload your revised manuscript, please go to your author page and click on the related manuscript title. Navigate to the "Review" link on the top left and scroll down the page. Click on the "Choose File" button under the "Editor Decision" title, choose the revised article (in pdf format) that you want to submit, and click on the "Upload" button to upload the author version. Repeat the same steps to upload the "Responses to Reviewers/Editor" file and make sure that you click the "Upload" button again.

To avoid any delay in making the article available freely online, the authors also need to upload the source files (Word or LaTex) when submitting revised manuscripts. Files can be compressed if necessary. The two-column final submission templates are as follows:

Click here to download Word template for final submission (.docx)

Click here to download LaTeX template for final submission (.tex)

Authors are responsible for obtaining permission to reproduce copyrighted material from other sources and are required to sign an agreement for the transfer of copyright to IJOCTA.

Article Processing Charges

There are no charges for submission and/or publication.

English Editing

Papers must be in English. Both British and American spelling is acceptable, provided usage is consistent within the manuscript. Manuscripts that are written in English that is ambiguous or incomprehensible, in the opinion of the Editor, will be returned to the authors with a request to resubmit once the language issues have been improved. This policy does not imply that all papers must be written in "perfect" English, whatever that may mean. Rather, the criteria require that the intended meaning of the authors must be clearly understandable, i.e., not obscured by language problems, by referees who have agreed to review the paper.

Presentation of Papers

Manuscript style

Use a standard font of the **11-point type: Times New Roman** is preferred. It is necessary to single line space your manuscript. Normally manuscripts are expected not to exceed 25 single-spaced pages including text, tables, figures and bibliography. All illustrations, figures, and tables are placed within the text at the appropriate points, rather than at the end.

During the submission process you must enter: (1) the full title, (2) names and affiliations of all authors and (3) the full address, including email, telephone and fax of the author who is to check the proofs. Supply a brief **biography** of each author at the end of the manuscript after references.

- Include the name(s) of any **sponsor(s)** of the research contained in the paper, along with **grant number(s).**
- Enter an **abstract** of no more than 250 words for all articles.

Keywords

Authors should prepare no more than 5 keywords for their manuscript.

Maximum five **AMS Classification number** (http://www.ams.org/mathscinet/msc/msc2010.html) of the study should be specified after keywords.

Writing Abstract

An abstract is a concise summary of the whole paper, not just the conclusions. The abstract should be no more than 250 words and convey the following:

1. An introduction to the work. This should be accessible by scientists in any field and express the necessity of the experiments executed.
2. Some scientific detail regarding the background to the problem.
3. A summary of the main result.
4. The implications of the result.
5. A broader perspective of the results, once again understandable across scientific disciplines.

It is crucial that the abstract conveys the importance of the work and be understandable without reference to the rest of the manuscript to a multidisciplinary audience. Abstracts should not contain any citation to other published works.

Reference Style

Reference citations in the text should be identified by numbers in square brackets "[ ]". All references must be complete and accurate. Please ensure that every reference cited in the text is also present in the reference list (and vice versa). Online citations should include date of access. References should be listed in the following style:

Journal article

Author, A.A., & Author, B. (Year). Title of article. Title of Journal, Vol(Issue), pages.

Castles, F.G., Curtin, J.C., & Vowles, J. (2006). Public policy in Australia and New Zealand: The new global context. Australian Journal of Political Science, 41(2), 131–143.

Book

Author, A. (Year). Title of book. Publisher, Place of Publication.

Mercer, P.A., & Smith, G. (1993). Private Viewdata in the UK. 2nd ed. Longman, London.

Chapter

Author, A. (Year). Title of chapter. In: A. Editor and B. Editor, eds. Title of book. Publisher, Place of publication, pages.

Bantz, C.R. (1995). Social dimensions of software development. In: J.A. Anderson, ed. Annual review of software management and development. CA: Sage, Newbury Park, 502–510.

Internet document

Author, A. (Year). Title of document [online]. Source. Available from: URL [Accessed (date)].

Holland, M. (2004). Guide to citing Internet sources [online]. Poole, Bournemouth University. Available from: http://www.bournemouth.ac.uk/library/using/guide_to_citing_internet_sourc.html [Accessed 4 November 2004].

Newspaper article

Author, A. (or Title of Newspaper) (Year). Title of article. Title of Newspaper, day Month, page, column.

Independent (1992). Picking up the bills. Independent, 4 June, p. 28a.

Thesis

Author, A. (Year). Title of thesis. Type of thesis (degree). Name of University.

Agutter, A.J. (1995). The linguistic significance of current British slang. PhD Thesis. Edinburgh University.

Illustrations

Illustrations submitted (line drawings, halftones, photos, photomicrographs, etc.) should be clean originals or digital files. Digital files are recommended for highest quality reproduction and should follow these guidelines:

- 300 dpi or higher
- Sized to fit on journal page
- TIFF or JPEG format only
- Embedded in text files and submitted as separate files (if required)

Tables and Figures

Tables and figures (illustrations) should be embedded in the text at the appropriate points, rather than at the end. A short descriptive title should appear above each table with a clear legend and any footnotes suitably identified below.

Proofs

Page proofs are sent to the designated author using IJOCTA EProof system. They must be carefully checked and returned within 48 hours of receipt.

Offprints/Reprints

Each corresponding author of an article will receive a PDF file of the article via email. This file is for personal use only and may not be copied and disseminated in any form without prior written permission from IJOCTA.

Submission Preparation Checklist

As part of the submission process, authors are required to check off their submission's compliance with all of the following items, and submissions may be returned to authors that do not adhere to these guidelines.

1. The submission has not been previously published, nor is it before another journal for consideration (or an explanation has been provided in Comments for the Editor).
2. The submission file is in Portable Document Format (.pdf).
3. The ORCID profile numbers of "all" authors are ready to enter in the step of Article Metadata (visit https://orcid.org for more details).
4. The text is single line spaced; uses a 11-point font; employs italics, rather than underlining (except with URL addresses); and all illustrations, figures, and tables are placed within the text at the appropriate points, rather than at the end.
5. The text adheres to the stylistic and bibliographic requirements outlined in the Author Guidelines, which is found in "About the Journal".
6. Maximum five AMS Classification number (http://www.ams.org/mathscinet/msc/msc2010.html) of the study have been provided after keywords.
7. After the acceptance of manuscript (before copy editing), Word (.docx) or LaTeX (.tex) version of the paper will be presented.
8. The names and email addresses of at least four (4) possible reviewers have been indicated in "Comments for the Editor" box in Paper Submission Step 1. Please note that at least two of the recommendations should be from different countries. Avoid suggesting reviewers who are at arms-length from you or your co-authors. This includes graduate advisors, people in your current department, or any others with a conflict of interest.

Peer Review Process

All contributions, prepared according to the author guidelines and submitted via IJOCTA online submission system are evaluated according to the criteria of originality and quality of their scientific content. The corresponding author will receive a confirmation e-mail with a reference number assigned to the paper, which he/she is asked to quote in all subsequent correspondence.

All manuscripts are first checked by the Technical Editor using plagiarism detection software (iThenticate) to verify originality and ensure the quality of the written work. If the result is not satisfactory (i.e. exceeding the limit of 30% of overlapping), the submission is rejected and the author is notified.

After the plagiarism check, the manuscripts are evaluated by the Editor-in-Chief and can be rejected without reviewing if considered not of sufficient interest or novelty, too preliminary or out of the scope of the journal. If the manuscript is considered suitable for further evaluation, it is first sent to the Area Editor. Based on his/her opinion the paper is then sent to at least two independent reviewers. Each reviewer is allowed up to four weeks to return his/her feedback but this duration may be extended based on his/her availability. IJOCTA has instituted a blind peer review process where the reviewers' identities are not known to authors. When the reviews are received, the Area Editor gives a decision and lets the author know it together with the reviewer comments and any supplementary files.

Should the reviews be positive, the authors are expected to submit the revised version usually within two months the editor decision is sent (this period can be extended when the authors contact to the editor and let him/her know that they need extra time for resubmission). If a revised paper is not resubmitted within the deadline, it is considered as a new submission after all the changes requested by reviewers have been made. Authors are required to submit a new cover letter, a response to reviewers letter and the revised manuscript (which ideally shows the revisions made in a different color or highlighted). If a change in authorship (addition or removal of author) has occurred during the revision, authors are requested to clarify the reason for change, and all authors (including the removed/added ones) need to submit a written consent for the change. The revised version is evaluated by the Area editor and/or reviewers and the Editor-in-Chief brings a decision about final acceptance based on their suggestions. If necessary, further revision can be asked for to fulfil all the requirements of the reviewers.

When a manuscript is accepted for publication, an acceptance letter is sent to the corresponding author and the authors are asked to submit the source file of the manuscript conforming to the IJOCTA two-column final submission template. After that stage, changes of authors of the manuscript are not possible. The manuscript is sent to the Copyeditor and a linguistic, metrological and technical revision is made, at which stage the authors are asked to make the final corrections in no more than a week. The layout editor prepares the galleys and the authors receive the galley proof for final check before printing. The authors are expected to correct only typographical errors on the proofs and return the proofs within 48 hours. After the final check by the layout editor and the proofreader, the manuscript is assigned a DOI number, made publicly available and listed in the forthcoming journal issue. After printing the issue, the corresponding metadata and files published in this issue are sent to the databases for indexing.
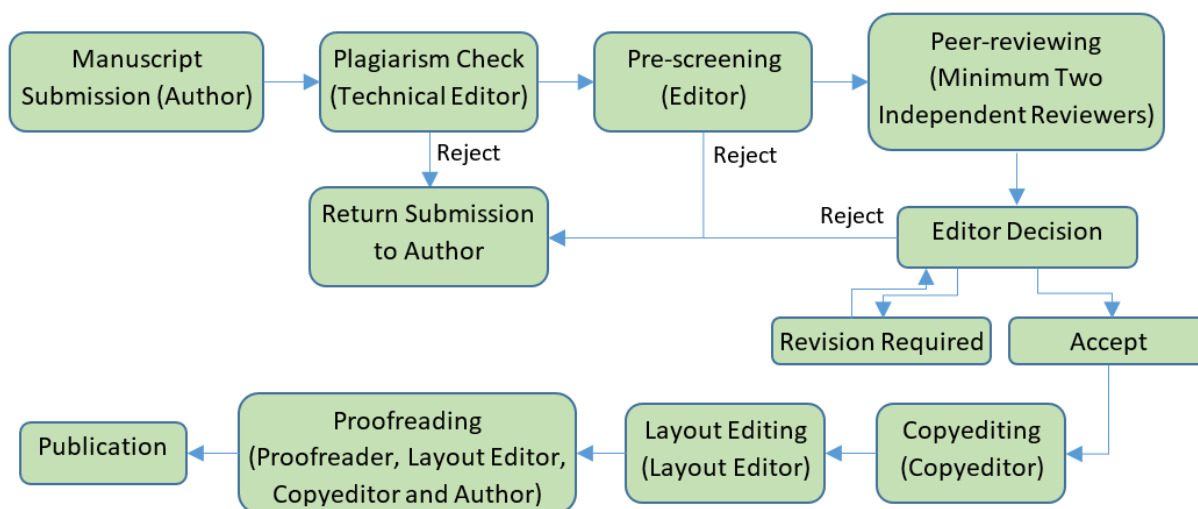
Publication Ethics and Malpractice Statement

IJOCTA is committed to ensuring ethics in publication and quality of articles. Conforming to standards of expected ethical behavior is therefore necessary for all parties (the author, the editor(s), the peer reviewer) involved in the act of publishing.

International Standards for Editors

The editors of the IJOCTA are responsible for deciding which of the articles submitted to the journal should be published considering their intellectual content without regard to race, gender, sexual orientation, religious belief, ethnic origin, citizenship, or political philosophy of the authors. The editors may be guided by the policies of the journal's editorial board and constrained by such legal requirements as shall then be in force regarding libel, copyright infringement and plagiarism. The editors may confer with other editors or reviewers in making this decision. As guardians and stewards of the research record, editors should encourage authors to strive for, and adhere themselves to, the highest standards of publication ethics. Furthermore, editors are in a unique position to indirectly foster responsible conduct of research through their policies and processes.

To achieve the maximum effect within the research community, ideally all editors should adhere to universal standards and good practices.

- Editors are accountable and should take responsibility for everything they publish.
- Editors should make fair and unbiased decisions independent from commercial consideration and ensure a fair and appropriate peer review process.
- Editors should adopt editorial policies that encourage maximum transparency and complete, honest reporting.
- Editors should guard the integrity of the published record by issuing corrections and retractions when needed and pursuing suspected or alleged research and publication misconduct.
- Editors should pursue reviewer and editorial misconduct.
- Editors should critically assess the ethical conduct of studies in humans and animals.
- Peer reviewers and authors should be told what is expected of them.
- Editors should have appropriate policies in place for handling editorial conflicts of interest.

Reference:

Kleinert S & Wager E (2011). Responsible research publication: international standards for editors. A position statement developed at the 2nd World Conference on Research Integrity, Singapore, July 22-24, 2010. Chapter 51 in: Mayer T & Steneck N (eds) Promoting Research Integrity in a Global Environment. Imperial College Press / World Scientific Publishing, Singapore (pp 317-28). (ISBN 978-981-4340-97-7) [Link].

International Standards for Authors

Publication is the final stage of research and therefore a responsibility for all researchers. Scholarly publications are expected to provide a detailed and permanent record of research. Because publications form the basis for both new research and the application of findings, they can affect not only the research community but also, indirectly, society at large. Researchers therefore have a responsibility to ensure that their publications are honest, clear, accurate, complete and balanced, and should avoid misleading, selective or ambiguous reporting. Journal editors also have responsibilities for ensuring the integrity of the research literature and these are set out in companion guidelines.

- The research being reported should have been conducted in an ethical and responsible manner and should comply with all relevant legislation.
- Researchers should present their results clearly, honestly, and without fabrication, falsification or inappropriate data manipulation.
- Researchers should strive to describe their methods clearly and unambiguously so that their findings can be confirmed by others.
- Researchers should adhere to publication requirements that submitted work is original, is not plagiarised, and has not been published elsewhere.
- Authors should take collective responsibility for submitted and published work.
- The authorship of research publications should accurately reflect individuals' contributions to the work and its reporting.

- Funding sources and relevant conflicts of interest should be disclosed.
- When an author discovers a significant error or inaccuracy in his/her own published work, it is the author's obligation to promptly notify the journal's Editor-in-Chief and cooperate with them to either retract the paper or to publish an appropriate erratum.

Reference:

Wager E & Kleinert S (2011) Responsible research publication: international standards for authors. A position statement developed at the 2nd World Conference on Research Integrity, Singapore, July 22-24, 2010. Chapter 50 in: Mayer T & Steneck N (eds) Promoting Research Integrity in a Global Environment. Imperial College Press / World Scientific Publishing, Singapore (pp 309-16). (ISBN 978-981-4340-97-7) [Link].

Basic principles to which peer reviewers should adhere

Peer review in all its forms plays an important role in ensuring the integrity of the scholarly record. The process depends to a large extent on trust and requires that everyone involved behaves responsibly and ethically. Peer reviewers play a central and critical part in the peer-review process as the peer review assists the Editors in making editorial decisions and, through the editorial communication with the author, may also assist the author in improving the manuscript.

Peer reviewers should:

- respect the confidentiality of peer review and not reveal any details of a manuscript or its review, during or after the peer-review process, beyond those that are released by the journal;
- not use information obtained during the peer-review process for their own or any other person's or organization's advantage, or to disadvantage or discredit others;
- only agree to review manuscripts for which they have the subject expertise required to carry out a proper assessment and which they can assess within a reasonable time-frame;
- declare all potential conflicting interests, seeking advice from the journal if they are unsure whether something constitutes a relevant conflict;
- not allow their reviews to be influenced by the origins of a manuscript, by the nationality, religion, political beliefs, gender or other characteristics of the authors, or by commercial considerations;
- be objective and constructive in their reviews, refraining from being hostile or inflammatory and from making libellous or derogatory personal comments;
- acknowledge that peer review is largely a reciprocal endeavour and undertake to carry out their fair share of reviewing, in a timely manner;
- provide personal and professional information that is accurate and a true representation of their expertise when creating or updating journal accounts.

Reference:

Homes I (2013). COPE Ethical Guidelines for Peer Reviewers, March 2013, v1 [Link].

Copyright Notice

IJOCTA

An International Journal of
Optimization and Control:
Theories & Applications
2010

CONTENTS

crossref