

RESEARCH ARTICLE

A randomized adaptive trust region line search method

Saman Babaie-Kafaki* and Saeed Rezaee

Department of Mathematics, Faculty of Mathematics, Statistics and Computer Science, Semnan University, Semnan, Iran
sbk@semnan.ac.ir, s.rezaee@semnan.ac.ir

ARTICLE INFO

Article History:
Received 12 December 2019
Accepted 31 May 2020
Available 27 July 2020

Keywords:
Nonlinear programming
Unconstrained optimization
Trust region method
Line search
Randomized algorithm

AMS Classification 2010:
49M37; 65K05; 68W20

ABSTRACT

Hybridizing the trust region, line search and simulated annealing methods, we develop a heuristic algorithm for solving unconstrained optimization problems. We make some numerical experiments on a set of CUTer test problems to investigate efficiency of the suggested algorithm. The results show that the algorithm is practically promising.



1. Introduction

As the most basic nonlinear optimization problem with continuous variables, unconstrained optimization naturally arises in many disciplines such as regression, image and signal processing, physical systems, optimal control and so on. Even, based on penalization schemes, constrained nonlinear programming problems can be reformulated as unconstrained problems [1]. Generally, the problem can be defined as minimization of an objective function that depends on real variables without any restriction on their values.

Among the efficient tools for solving unconstrained optimization problems there are the trust region (TR) methods and the line search (LS) techniques [1]. In each iteration of a TR method, a neighborhood is defined around the available approximation of the solution, called the trust region, and then, an approximation of the objective function is minimized within the region to achieve the new estimation. The term used for the method originates from the fact that a local approximation is trusted as the predictor of the

objective function behavior. In another guideline, in each iteration of an LS method a search direction is defined at the available approximation of the solution and then, the objective function is minimized along the given direction to achieve the new estimation. As known, an LS method often requires more iterations to find a minimizer of the objective function than does a TR method, while computing the successive approximations of the solution more quickly.

To evaluate acceptability level of the local approximate model of the objective function in an arbitrary iteration of a TR method, a ratio is defined often by dividing the distance of the objective function values to the distance of their local approximations in the recent iterates. When the TR ratio is small, the approximate model is found to be a poor predictor of the objective function behavior. In such situation, the model should be resolved in a smaller region. However, when the TR ratio is large enough, the approximate model is found to be a locally suitable predictor of the

*Corresponding Author

objective function behavior. So, the generated estimation of the solution should be accepted and the region can be enlarged in the next iteration. It is worth noting that to decrease computational cost of the TR methods, the LS techniques can be effectively employed in the case where the TR ratio is small, as an alternative of resolving the approximate model in a reduced neighborhood. A review of the literature reveals an abundance of the studies on the TR methods; see for example [2–4] and the references therein.

Here, based on the simulated annealing strategy, we develop a randomized TR–LS method. The method is discussed in details in the next section. We provide a test bed to shed light on the advantages of our heuristic algorithm in Section 3. Finally, in Section 4 we come out with concluding remarks.

2. A randomized trust region line search algorithm

Consider the unconstrained optimization problem $\min_{x \in \mathbb{R}^n} f(x)$ in which the objective function $f : \mathbb{R}^n \rightarrow \mathbb{R}$ is assumed to be continuously differentiable. Iterative formula of the optimization algorithms is generally in the following form:

$$x_0 \in \mathbb{R}^n, x_{k+1} = x_k + s_k, k = 0, 1, \dots,$$

where s_k is the step taken from x_k . In a TR method, often s_k is an approximate solution of the following subproblem, being a local quadratic approximation of the objective function:

$$\begin{aligned} \min_{s \in \mathbb{R}^n} \quad & m_k(s) = f_k + g_k^T s + \frac{1}{2} s^T B_k s, \\ \text{s.t.} \quad & \|s\| \leq \Delta_k, \end{aligned} \tag{1}$$

where $f_k = f(x_k)$, $g_k = \nabla f(x_k)$, B_k is an approximation of the Hessian $\nabla^2 f(x_k)$, $\Delta_k > 0$ is the TR radius and $\|\cdot\|$ stands for the Euclidean norm. Meanwhile, in an LS method we have $s_k = \alpha_k d_k$ where $d_k \in \mathbb{R}^n$ is a descent search direction and

$$\alpha_k \approx \arg \min_{\alpha > 0} f(x_k + \alpha d_k),$$

is called the step length.

To describe our randomization scheme, we use the framework of the TR–LS algorithm proposed in [2]. Firstly, we adopt the adaptive choice of the TR radius suggested in [5], that is

$$\Delta_k = -\frac{g_k^T q_k}{q_k^T B_k q_k} \|q_k\|, \tag{2}$$

in which B_k is a positive definite quasi-Newton approximation of the Hessian and $q_k \in \mathbb{R}^n$ is a vector parameter satisfying the angle condition

[1], i.e.

$$-\frac{g_k^T q_k}{\|g_k\| \|q_k\|} \geq \tau, \tag{3}$$

for some constant $\tau \in (0, 1]$. To evaluate local consistency between the objective function and the quadratic model (1), we apply the following traditional TR ratio [1]:

$$\rho_k = \frac{f_k - f(x_k + s_k)}{m_k(0) - m_k(s_k)}. \tag{4}$$

Now, for a prespecified constant $\mu \in (0, 1)$, if ρ_k is large enough in the sense that $\rho_k > \mu$, then we set $x_{k+1} = x_k + s_k$. Otherwise, to avoid resolving the TR subproblem (1), we set $x_{k+1} = x_k + s_k$ with a specific probability which depends on the value of ρ_k , or (similar to the approach of [2]) we use the Armijo-type LS procedure proposed by Wan et al. [6] as follows:

Line search 2.1. Let L_k be an approximation of the Lipschitz constant of the gradient and set $\beta_k = -\frac{g_k^T s_k}{L_k \|s_k\|^2}$. The step length α_k is the largest quantity in $\{t^i \beta_k\}_{i=0}^\infty$ which satisfies the following inequality:

$$f(x_k + \alpha_k s_k) \leq f_k + \sigma \alpha_k (g_k^T s_k - \frac{1}{2} \alpha_k r L_k \|s_k\|^2),$$

where $t \in (0, 1)$, $\sigma \in (0, 1/2)$, and $r \in [0, +\infty)$ are real constants.

As seen, the distinct feature of our algorithm is that we may accept a trial step s_k even when $\rho_k < \mu$, despite the classical TR algorithms for which such trial steps are rejected and the subproblem (1) is resolved with a smaller radius, or an LS strategy is employed. So, we need to define a reasonable probability for the mentioned randomized part of the algorithm. In this context, we apply the probabilistic approach of the simulated annealing (SA) strategy.

Among the earliest and most popular metaheuristic techniques of optimization, there is the simulated annealing (SA) algorithm. The method originates from the successful annealing process of the materials which involves the cautious control of the cooling schedule [7]. SA is a local search algorithm capable of escaping from local optima by use of random hill-climbing moves in the search process [8, 9]. It is very efficient in practice [9, 10] and well-developed in theory [11, 12].

To provide a detailed description of the SA method [8], note that similar to the TR technique, at the iteration t of the method a neighborhood \mathcal{N}_t is defined around the iterate x_t . Then, a neighbor $y \in \mathcal{N}_t$ is randomly selected. If y is better than x_t (often in the cost function point of view, i.e. $f(y) < f(x_t)$), then we move to y in the sense

that we set $x_{t+1} = y$. However, when x_t is better than y , we move to y with the probability

$$p_t = e^{-\frac{d(x_t, y)}{T}}, \quad (5)$$

and stay in x_t otherwise, where T is a positive constant commonly called the temperature and $d(x_t, y)$ is a nonnegative function which demonstrates the measure of unfitness of the feasible solution y in contrast to x_t .

The temperature T controls the likelihood of cost increases in the sense that when T is small, cost increases are highly unlikely while when T is large, the value of $d(x_t, y)$ has an insignificant effect on the probability p_t and any particular transition. In order to guarantee the global convergence with probability one, the temperature needs to be decreased logarithmically with the iteration number t [13], making the process too slow. In practice, the temperature is usually updated by

$$T \leftarrow \lambda T, \quad (6)$$

with a prespecified constant $0 \ll \lambda < 1$ [11].

In order to allow probable moves to some inferior solutions as well as to reduce the effect of unsuccessful iterations (with $\rho_k < \mu$), we apply the SA scheme in our algorithm. In this context, when at the k th iteration of the algorithm the TR ratio is negative or a small positive number near to zero, we may accept the trial step s_k . More exactly, if $\rho_k < \mu$, then we set $x_{k+1} = x_k + s_k$ with the following probability:

$$p_{\rho_k} = e^{-\frac{\mu - \rho_k}{T}}, \quad (7)$$

and stay in x_k otherwise, where T is the temperature. Considering (4) and (5), here we set $d(x_k, y) = \mu - \rho_k$ with $y = x_k + s_k$. As seen, the given probability is small when $\rho_k \ll \mu$ or the temperature T is small.

Here, based on the above preliminaries, we are in a position to describe the algorithm in details.

Algorithm 2.1. (A randomized trust region line search algorithm (RTRLS))

Step 0: {Initialization} Choose an initial point $x_0 \in \mathbb{R}^n$, a symmetric positive definite matrix $B_0 \in \mathbb{R}^{n \times n}$, and the constants $t \in (0, 1)$, $\sigma \in (0, 1/2)$, $r \in [0, +\infty)$, $\mu \in (0, 1]$, $L_0 > 0$, $\epsilon > 0$, and $T_0 > 0$ as the initial temperature. Compute f_0 , and set $k = 0$ and $T = T_0$.

Step 1: {Stopping criterion} **If** $\|g_k\| < \epsilon$, **then stop**.

Step 2: Choose q_k satisfying (3) and compute Δ_k by (2).

Step 3: Solve the subproblem (1) to find the trial step s_k .

Step 4: Compute ρ_k by (4). **If** $\rho_k \geq \mu$, **then** set $x_{k+1} = x_k + s_k$, and **goto** Step 6; **otherwise**, with the probability p_{ρ_k} given by (7) set $x_{k+1} = x_k + s_k$ and **goto** Step 6.

Step 5: Find the step length α_k using Line search 2.1 and set $x_{k+1} = x_k + \alpha_k s_k$.

Step 6: Compute the new Hessian approximation B_{k+1} by a quasi-Newton updating formula. Set $k = k + 1$, decrease the temperature T and **goto** Step 1.

Note that if the temperature is decreased logarithmically, then, based on the classical convergence properties of the SA [13] and the convergence analysis conducted in [5], with probability one Algorithm 2.1 can be globally convergent.

3. Numerical experiments

Here, we present some numerical results obtained by applying MATLAB 7.14.0.739 (R2012a) implementations of RTRLS (Algorithm 2.1) and the efficient accelerated nonmonotone TR-LS algorithm proposed in [2] (in which Andrei's initial choice of the step length is employed [14]), here called AccTRLS. The runs were performed on a set of 84 unconstrained optimization test problems of the CUTER collection [15] with the minimum dimension being equal to 50, as specified in [3], using a computer Intel(R) Core(TM)2 Duo CPU 2.00 GHz with 1.50 GB of RAM.

For both algorithms, we adopted the parameter values suggested in [2] as well as the same stopping criteria. In addition, for RTRLS we set $T_0 = \|g_0\|$ and in Step 4, we decreased T by (6) with $\lambda = 0.9$, found to be appropriate. Among the wide scope of the choices of q_k satisfying (3), here we set $q_k = -B_k^{-1}g_k$. Similar to the approach of [2], to compute the Hessian approximation we used the scaled memoryless DFP formula where its inverse can be effectively determined in a memoryless form [1]. Also, we used the double Dogleg method [1] to solve the subproblem (1).

Efficiency comparisons were drawn using the Dolan-Moré performance profile [16] on the number of iterations, number of objective function evaluations, number of gradient evaluations and the running time. Performance profile gives, for every $\omega \geq 1$, the proportion $p(\omega)$ of the test problems that each considered algorithmic variant has a performance within a factor of ω of the best. Figures 1–4 illustrate the results of comparisons. As seen, generally RTRLS outperforms AccTRLS. It is worth noting that in 64% of the iterations RTRLS achieves the solution faster than AccTRLS. Thus, in general our randomized strategy

based on the SA method turns out to be practically promising. Especially, it can be employed as an alternative of the acceleration/nonmonotone schemes used in the TR-LS algorithms.

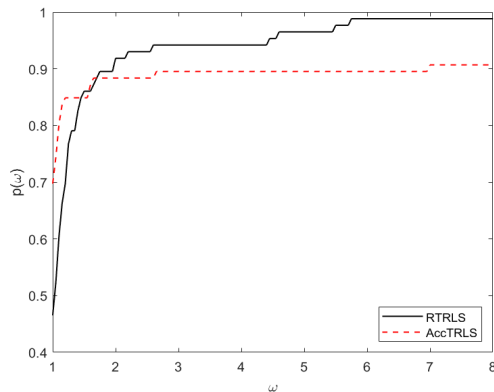


Figure 1. Number of iterations performance profiles

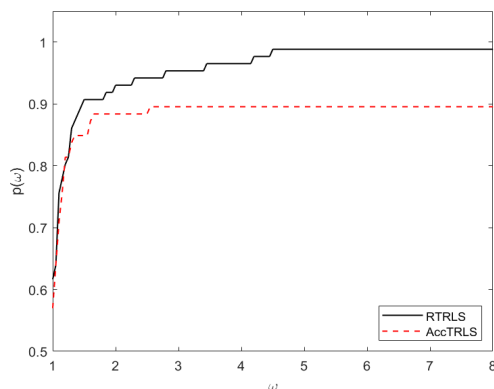


Figure 2. Number of objective function evaluations performance profiles

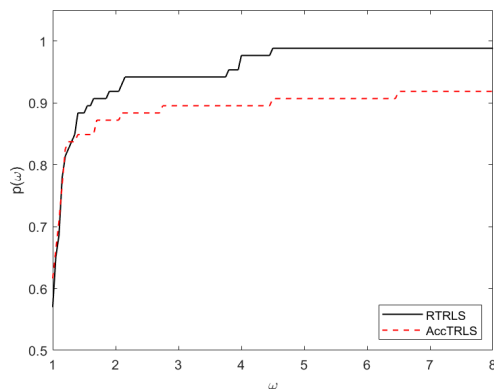


Figure 3. Number of gradient evaluations performance profiles

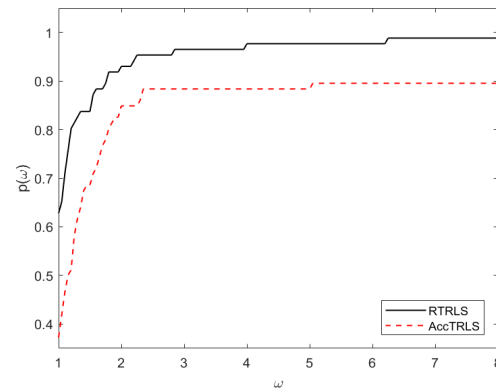


Figure 4. CPU time performance profiles

4. Conclusions

Employing the simulated annealing aspects in a recent adaptive trust region line search method, a heuristic algorithm has been suggested to be used in unconstrained optimization. The method can also be considered as a randomized version of the trust region line search algorithm. Numerical experiments showed that the proposed randomization scheme can enhance efficiency of the classical trust region line search algorithms; especially, it can serve as an alternative of the acceleration/nonmonotone approaches used in the algorithms.

As a future work, one can investigate possible employing of other metaheuristic algorithms in the trust region line search methods. In addition, effect of such randomized schemes on the backtracking line search techniques can be studied.

Acknowledgements


The authors thank the Research Council of Semnan University for its support. They are also grateful to the anonymous reviewer for his/her valuable comments and suggestions helped to improve the quality of this work.

References


- [1] Sun, W., & Yuan, Y.X. (2006). Optimization Theory and Methods: Nonlinear Programming. Springer, New York.
- [2] Babaie-Kafaki, S., & Rezaee, S. (2018). Two accelerated nonmonotone adaptive trust region line search methods. *Numerical Algorithms*, 78(3), 911–928.
- [3] Rezaee, S., & Babaie-Kafaki, S. (2019). An adaptive nonmonotone trust region algorithm. *Optimization Methods and Software*, 34(2), 264–277.

- [4] Yuan, Y.X. (2015). Recent advances in trust region algorithms. *Mathematical Programming*, 151(1, Ser. B), 249–281.
- [5] Shi, Z.J., & Guo, J.H. (2008). A new trust region method for unconstrained optimization. *Journal of Computational and Applied Mathematics*, 213(1), 509–520.
- [6] Wan, W., Huang, S., & Zheng, X.D. (2012). New cautious BFGS algorithm based on modified Armijo-type line search. *Journal of Inequalities and Applications*, 2012(1), 1–10.
- [7] Yang, X.S. (2014). *Nature-Inspired Optimization Algorithms*. Elsevier, London.
- [8] Bertsimas, D., & Tsitsiklis, J.N. (1997). *Introduction to Linear Optimization*. Athena Scientific, Massachusetts.
- [9] Henderson, D., Jacobson, S.H., & Johnson, A.W. (2003). The theory and practice of simulated annealing. In: F.W. Glover and G.A. Kochenberger, eds. *Handbook of Metaheuristics*, volume 57 of *International Series in Operations Research and Management Science*. Kluwer Academic Publishers, Boston, MA, 287–319.
- [10] Reeves, C.R. (1996). Modern heuristic techniques. In: V.J. Rayward-Smith, ed. *Modern Heuristic Search Methods*. John Wiley and Sons, Chichester, 1–24.
- [11] Babaie-Kafaki, S., Ghanbari, R., & Mahdavi-Amiri, N. (2012). An efficient and practically robust hybrid metaheuristic algorithm for solving fuzzy bus terminal location problems. *Asia-Pacific Journal of Operational Research*, 29(2), 1–25.
- [12] Aards, E. and Korst, J., & van Laarhoven, P. (1997). Simulated annealing. In: E.H.L. Aarts and J.K. Lenstra, eds. *Local Search in Combinatorial Optimization*. John Wiley and Sons, Chichester, 91–121.
- [13] Hajek, B. (1988). Cooling schedules for optimal annealing. *Mathematics of Operations Research*, 13(2), 311–329.
- [14] Andrei, N. (2006). An acceleration of gradient descent algorithm with backtracking for unconstrained optimization. *Numerical Algorithms*, 42(1), 63–73.
- [15] Gould, N.I.M., Orban, D., & Toint, Ph.L. (2006). CUTER: a constrained and unconstrained testing environment, revisited. *ACM Transactions on Mathematical Software*, 29(4), 373–394.
- [16] Dolan, E.D., & Moré, J.J. (2002). Benchmarking optimization software with performance profiles. *Mathematical Programming*, 91(2, Ser. A), 201–213.
- [17] Hager, W.W., & Zhang, H. (2002). Algorithm 851: CG-Descent, a conjugate gradient method with guaranteed descent. *ACM Transactions on Mathematical Software*, 32(1), 113–137.

Saman Babaie-Kafaki is a Professor in Department of Mathematics of Semnan University, Iran. He received his B.Sc. in Applied Mathematics from Mazandaran University, Iran, in 2003, and his M.Sc. and Ph.D. in Applied Mathematics from Sharif University of Technology, Iran, in 2005 and 2010, respectively, under supervision of Professor Nezam Mahdavi-Amiri. His research interests lie within numerical continuous optimization, numerical linear algebra and heuristic algorithms.

 <http://orcid.org/0000-0003-0122-8384>

Saeed Rezaee received his B.Sc. in Applied Mathematics from University of Hormozgan, Iran, in 2005, his M.Sc. in Applied Mathematics from Sharif University of Technology, Iran, in 2010, and his Ph.D. in Applied Mathematics (Operational Research) from Semnan University, Semnan, Iran, in 2017. His M.Sc. supervisor was Professor Nezam Mahdavi-Amiri and his Ph.D. supervisor was Professor Saman Babaie-Kafaki. His research interests lie within nonlinear programming modelling and algorithms.

 <http://orcid.org/0000-0002-2636-3868>

