An International Journal of Optimization and Control: Theories & Applications
ISSN:2146-0957 eISSN:2146-5703
Vol.11, No.1, pp.41-51 (2021)
http://doi.org/10.11121/ijocta.01.2021.00945

RESEARCH ARTICLE

A fix-and-optimize heuristic for the capacitated multi-item stochastic lot-sizing problem

M. Edib Gurkan[a] and Huseyin Tunc[b*]

[a] *Department of Management Information Systems, Karabuk University, Turkey*
[b] *Department of Management, Social Sciences University of Ankara, Turkey*
edibgurkan@karabuk.edu.tr, huseyin.tunc@asbu.edu.tr

ARTICLE INFO

Article History:
Received 12 March 2020
Accepted 27 August 2020
Available 13 December 2020

Keywords:
Capacitated lot-sizing
Random demand
Inventory
Mixed integer programming
Fix and optimize

AMS Classification 2010:
90-08; 90B05; 90C11

ABSTRACT

This study addresses the stochastic multi-item capacitated lot-sizing problem. Here, it is assumed that all items are produced on a single production resource and unmet demands are backlogged. The literature shows that the deterministic version of this problem is NP-Hard. We consider the case where period demands are time-varying random variables. The objective is to determine the minimum expected cost production plan so as to meet stochastic period demands over the planning horizon. We extend the mixed integer programming formulation introduced in the literature to capture the problem under consideration. Further, we propose a fix-and-optimize heuristic building on an item-period oriented decomposition scheme. We then conduct a numerical study to evaluate the performance of the proposed heuristic as compared to the heuristic introduced by Tempelmeier and Hilger [16]. The results clearly show that the proposed fix-and-optimize heuristic arises as both cost-efficient and time-efficient solution approach as compared to the benchmark heuristic.

1. Introduction

The capacitated lot-sizing problem (CLSP) is concerned with the determination of optimal production schedule and corresponding quantities so as to minimize total cost under given production capacity requirement over a discrete and finite planning horizon. This problem is encountered in many industrial applications and has often been considered as one of the major challenges in production planning process.

Due to its practical relevance, CLSP and its extensions have been extensively investigated in the literature over last several decades (see e.g. [1,2]). Nevertheless, in the majority of studies, period demands over the planning horizon are assumed to be deterministic and time-varying. The uncapacitated lot-sizing problem is shown to be polynomially solvable under mild conditions [3]. Yet, the problem becomes more challenging when a capacity restriction on a production resource is of concern. The single-item CLSP is investigated by Florian et al. [4] and shown that it falls into the class of NP-hard problems. Also, some special cases, where the problem under consideration becomes polynomially solvable, are presented by authors. Then, the multi-item CLSP is also demonstrated to be NP-Hard, even under the conditions where single item counterpart of the problem is polynomially solvable [5]. Hence, the literature provides a variety of heuristic approaches especially for solving multi-item CLSP. For a more detailed discussion on the solution approaches of both deterministic single-item and multi-item CLSP, interested readers are referred to the surveys of Karimi et al [6], Robinson et al. [7], Buschkuhl et al. [8] and Brahimi et al. [9].

[*] Corresponding author

In most of the industrial production practices, period demands are indeed not known with certainty. Instead, decisions are made based on demand forecasts which are subject to uncertainty. As such, neglecting uncertainty and relying on deterministic production planning approaches may lead to cost inefficiency and low degree of customer satisfaction in practice [10]. Therefore, embedding demand uncertainty into lot-sizing problems is essential, nonetheless, it is challenging. A classification on how to capture uncertainty in lot-sizing problems is provided by Bookbinder and Tan [11]. These strategies mainly characterize the inventory control rules as to when production decisions are made. Among these is the static uncertainty strategy where production schedule and corresponding quantities are made at the beginning of the planning horizon. Put in other words, this strategy suggests that the timing and the quantity of replenishments are set once for all, and thus, known before the planning horizon starts. In particular, this strategy may be appealing for production systems where advanced information on production decisions is required (e.g. MRP systems) [12]. In [13], the stochastic CLSP with static uncertainty strategy under β service level (fill rate criterion) constraints is addressed and a heuristic method, referred to as ABC_β heuristic, is developed. Then, a column-generation based heuristic approach is proposed by Tempelmeier [12] for the very problem. It is shown that the column generation based heuristic is superior to the ABC_β heuristic in terms of expected cost criterion. The same problem under δ service level constraints is investigated and two linear mathematical formulations are developed by Helber et al. [14]. The proposed formulations are then solved via the fix-and-optimize heuristic approach introduced in [15]. This heuristic simply aims to decompose a mixed integer programming (MIP) problem with large number of binary variables into a number of tractable sub-problems and to solve the resulting sub-problems in an iterative fashion. Then, the problem under β service level constraints is addressed by Tempelmeier and Hilger [16] once again and a fix-and-optimize heuristic is proposed. The cost performance of the proposed fix-and-optimize heuristic is investigated in a numerical study where the ABC_β heuristic ([13]) and the column generation heuristic ([12]) are used as benchmarks. It is observed that the fix-and-optimize solution approach exhibits a better performance as compared to the ABC_β heuristic in all settings. It is also shown that the fix-and-optimize heuristic outperforms the column generation approach in settings where either production capacity is tight or a limited number of items are of concern. Recently, the stochastic multi-item CLSP considering sequence-dependent changeovers under fill rate constraints is addressed in [17]. Apart from the above mentioned studies, there is a sizeable literature adopting static uncertainty strategy to control inventory/production decisions in various production/manufacturing systems (see e.g. [18], [19], [20], [21], [22], [23], and [24]). Among those, an extended MIP formulation for a single-item, stochastic CLSP under α service level constraints is proposed by Tunc [23]. It is shown that this formulation has a very tight linear relaxation that provides a far superior computational performance as compared to traditional period based formulations present in the literature. It should also be remarked that there are number of recent studies addressing the stochastic multi-item CLSP with setup carryovers ([25]), energy concerns ([26]), and rolling horizon framework under service level constraints ([27], [28]).

Here, we investigate a finite horizon, periodic review, stochastic multi-item CLSP under static uncertainty strategy. More specifically, we consider a multi-item production system where items share a single production resource with a finite capacity. For each item, a production firm faces time-varying stochastic period demands over the planning horizon. The aim is to determine the production schedule and corresponding production quantities of each item under the maximum available production capacity so as to minimize the total expected cost over a finite planning horizon. The total expected cost is comprised of the fixed setup cost, variable production cost, inventory holding cost and backlogging cost.

The contribution of the current study is multi-fold as provided in the following:

- We extend the MIP models proposed in [23] and [29] so as to capture multi-item case and backlogging cost of stockouts.
- We propose a fix-and-optimize heuristic essentially built on that of Tempelmeier and Hilger [16]. In particular, we introduce and adopt a novel decomposition scheme within the proposed fix-and-optimize heuristic. The proposed heuristic relies on the extended MIP formulation whereas that of Tempelmeier and Hilger employs the MIP formulation developed earlier in the literature.
- We conduct an extensive numerical study to evaluate the computational performance of the proposed heuristic against the fix-and-optimize heuristic presented in

Tempelmeier and Hilger [16]. The numerical study demonstrates that the proposed heuristic is more time-efficient and yields better cost solutions as compared to the solutions obtained with the heuristic provided by Tempelmeier and Hilger [16] in all instances tested.

The remainder of the paper is structured as follows. In the following section, we define the problem under consideration. Section 3 presents the complete MIP model. In Section 4, we explain the fix-and-optimize heuristic proposed for the problem. Next, we provide the results obtained by the proposed heuristic and compare it to the results of [16]. Finally, In Section 6, we summarize our main findings and provide directions for future research.

2. Problem definition

We consider a firm producing N items on a single production resource over a finite planning horizon of T discrete time periods. For each period t, the total production quantity of all items is limited with the maximum available capacity C_t. We assume that period demands are independent, but not necessarily identically distributed, random variables over the planning horizon. We let d_{pt} denote the random demand of an item p in period t and assume that d_{pt} follows a known probability distribution function. For convenience, total random demand up to and including period t of an item p, i.e. $\sum_{i=1}^{t} d_{pi}$, is denoted as D_{pt}. A fixed setup cost K incurred in each production period for each item. Demand not satisfied is assumed to be backlogged with a unit backlogging cost b. Similarly, there is a holding cost h charged for each unit of excess production over demand at the end of each period. For the sake of simplicity, we assume that production quantities are received instantaneously and unit production costs are negligible. The problem aims to determine production periods and corresponding production quantities for each item so as to minimize the total expected cost. Also, these decisions are made at the beginning of the planning horizon following the static uncertainty strategy.

Now, let us define interval $[i, j)$ as production cycle for item p if i and j are consecutive production periods for item p. That is, a production order takes place in i and j but not in between. Then, we can derive an expression for the expected total cost belonging to item p within this production cycle as follows;

$$K + \sum_{t=i}^{j-1} \left[h(y - \mathbb{E}\, D_{pt}) + (h+b)\, \mathbb{E}(y - D_{pt})^{-} \right] \tag{1}$$

where y is the cumulative production quantity of item p in interval $[1, i]$, $\mathbb{E}$ is an expectation operator, and $\mathbb{E}(y - D_{pt})^{-}$ represents the expected magnitude of stockouts of item p in period t. It is also well-known that $\mathbb{E}(y - D_{pt})^{-}$ is referred to as the first-order loss function (see e.g. [30]).

Here, we remark that Eq. (1) is non-linear since it includes the first-order loss function which is known to be a decreasing, non-negative valued, and convex function [31]. Therefore, one needs to tackle this non-linearity while building its MIP formulation. A common way to handle this issue in relevant literature is to use a piecewise linear approximation of the original loss function (see e.g. [32], [33], [34]). To do so, one is required to have a finite number of linear functions whose pointwise maximum could provide a piecewise linear approximation of the convex first-order loss function. Now, consider a set of intercept and slope pairs, i.e. $Y_t = \{(\alpha_1, \beta_1), (\alpha_2, \beta_2), \ldots, (\alpha_m, \beta_m)\}$ in order to define m linear functions. Then, a piecewise linear approximation of the first-order loss function for an item p in any period t can be represented as $\mathbb{E}(y - D_{pt})^{-} \approx \max_{(\alpha,\beta)\in Y_t} \{\alpha + \beta(y - \mathbb{E}\, D_{pt})\}$. We hereby assume that the number of linear segments are given and corresponding slope and intercept values are available. In order to demonstrate the piecewise linear approximation approach given above, we give the following illustrative example. We also refer the interested readers to [35], [36], [37] for more detailed discussion on the loss function and its piecewise linearization.

Example. We now illustrate the approximation of an expected total cost of an item over a production cycle by replacing the first-order loss function in Eq (1) with its piecewise linear approximation. More specifically, we consider a production cycle comprising 3 discrete time periods in the illustrative example considered. It is assumed that demands in each period are normally distributed with mean values $\mathbb{E}\, d_t \in \{30, 100, 50\}$ and fixed coefficient of variation $\rho = 0.2$. Other cost parameters are $K = 250$, $h = 1$, $b = 10$. In Fig. 1, we show the approximation of a cost function with four line segments. The expected total cost is equal to 534.696 whereas the approximated cost is 510.183. Note that the piecewise linearization yields better results when the number of line segments are increased.

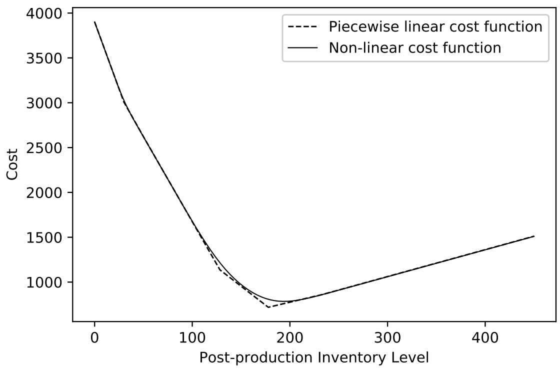

Figure 1. Piecewise linearization of the expected production cycle cost.

3. Model formulation

In the context of the static-uncertainty strategy, we aim to determine all decisions regarding to the production of N items at the beginning of the planning horizon. Therefore, the solution of this problem corresponds to a production schedule and respective production quantities for each item. Here, we present the following decision variables in order to construct the MIP model.

- x_{pij}: binary variable that takes the value of 1 if $[i, j)$ is a production cycle for item p and 0, otherwise;
- q_{pij}: expected cumulative production quantity up to ith period (including period i) if $[i, j)$ is a production cycle for item p and 0, otherwise
- H_{pijt}: approximate loss function value of an item p at period t if $[i, j)$ ($i \leq t$ and $j > t$) is a production cycle for item p and 0, otherwise.

Now, we provide the complete MIP formulation of the problem below.

$$\min \sum_{p=1}^{N}\sum_{i=1}^{T}\sum_{j=i+1}^{T+1}\left(Kx_{pij} + \sum_{t=i}^{j-1}\left(h\left(q_{pij} - \mathbb{E}\,D_{pt}x_{pij}\right) + (h+b)H_{pijt}\right)\right) \tag{2a}$$

subject to,

$$\sum_{i=1}^{t-1} x_{pit} = \sum_{j=t+1}^{T+1} x_{ptj}, \quad p \in [1,N], t \in [2,T], \tag{2b}$$

$$\sum_{j=2}^{T+1} x_{p1j} = 1, \quad p \in [1,N], \tag{2c}$$

$$\sum_{i=1}^{T} x_{pi,T+1} = 1, \quad p \in [1,N], \tag{2d}$$

$$q_{pij} \leq \left(\sum_{t=1}^{i} C_t\right) x_{pij}, \quad p \in [1,N], i \in [1,T], j \in [i+1, T+1], \tag{2e}$$

$$\sum_{p=1}^{N}\sum_{j=2}^{T+1} q_{pij} \leq C_1, \tag{2f}$$

$$\sum_{p=1}^{N}\left(\sum_{j=t+1}^{T+1} q_{ptj} - \sum_{i=1}^{t-1} q_{pit}\right) \leq C_t, \quad t \in [2,T], \tag{2g}$$

$$\sum_{i=1}^{t-1} q_{pit} \leq \sum_{j=t+1}^{T+1} q_{ptj}, \quad p \in [1,N], t \in [2,T], \tag{2h}$$

$$H_{pijt} \geq \alpha x_{pij} + \beta\left(q_{pij} - \mathbb{E}\,D_{pt}x_{pij}\right), \quad p \in [1,N], i \in [1,T], j \in [i+1,T+1], t \in [i, j-1], (\alpha,\beta) \in Y_{it}, \tag{2i}$$

$$H_{pijt} \geq 0, q_{pij} \geq 0, x_{pij} \in \{0,1\}. \tag{2j}$$

The objective function (2a) minimizes the sum of expected total costs over the planning horizon for each item. Notice that the total expected cost expression vanishes if the corresponding $[i, j)$ is not a production cycle for item p. As such, this expression indeed sum up the total expected costs of valid production cycles, i.e. corresponding $x_{pij} = 1$. Constraints (2b)–(2d) are the conventional flow conversation equations. In particular, constraint (2b) guarantees that, for any item p, if a production cycle ends at any period t then the subsequent cycle must start in t. For a given item, constraint (2c) states that there should be a production cycle starting at the very first period. In a similar manner, (2d) ensures that the last production cycle of any item has to end when the planning horizon is over. The constraint (2e) ensures that q_{pij} takes a positive value only when $[i, j)$ is a production cycle for item p. Constraints (2f) and (2g) state that the total production quantity of all items in period t is capacitated and can not exceed the given capacity limit C_t. Constraint (2h) guarantees that the cumulative production quantity of an item must be non-decreasing throughout the planning horizon. As discussed in the earlier section, a lower bound on the approximate loss function value H_{pijt} is set with constraint (2i). Notice that the lower bound is established only if the $[i, j)$ is a production cycle, whereas the expression vanishes, otherwise.

4. Fix-and-optimize heuristic

As we mentioned earlier, single-item CLSP is a challenging problem even when period demands are assumed to be deterministic. Naturally, the problem becomes even more challenging as stochastic demands and multiple items are of concern. The difficulty in solving the proposed mathematical model mainly springs from the abundance of the binary setup variables in use. At this point, the fix-and-optimize approach arises as a promising heuristic as it aims to find a near-optimal solution to an MIP formulation by solving series of sub-problems in an iterative fashion. A sub-problem is the reduced form of the original problem where a set of decision variables are fixed in value whereas the remaining set are left to be optimized. As such, fix-and-optimize heuristics often provide high quality solutions especially when the binary decision variables are vast. Also, it is known for its ease of implementation, and thus, has been commonly adopted in various lot-sizing problems over the last decade (see e.g. [15], [14], [16]). We hereby employ the variant of the fix-and-optimize heuristic proposed by Sahling et al. [38]. Furthermore, we provide a novel item-period oriented decomposition scheme employed within the adopted fix-and-optimize heuristic.

We can briefly summarize how the adopted fix-and-optimize heuristic works as follows. In each iteration, the binary decision variables are divided into two sets, i.e. set of variables fixed (SF) and set of variables to be optimized (SO). Then, all continuous variables and SO are optimized with a set of fixed values for SF obtained from previous iterations. The sub-problems generated in each iteration are formulated as MIP models and solved by off-the-shelf MIP solvers. In the next iteration, SF and SO are updated and then the same steps are followed until the heuristic reaches to a termination criteria. For more information on the fix-and-optimize heuristic, the reader can be referred to the studies of Pochet and Wolsey [39] and Sahling et al. [38]. In what follows, we provide details of the proposed fix-and-optimize heuristic.

4.1. Item-period oriented decomposition scheme

We propose an item-period oriented decomposition scheme for generating the sub problems (i.e. SF and SO) in each iteration. This decomposition scheme focuses on both items and time periods. In particular, it decomposes the entire set of items and the complete planning horizon into subsets of items and time windows. Here, the decomposed items and/or periods set the binary variables to be optimized. The decomposition scheme is characterized by the following input parameters; the number of items to be optimized (wit) in each iteration, the number of periods considered in each time window $(wper)$, and the rate of overlapping time periods to be optimized within two consecutive iterations (ort). Subsets of items and time windows are generated once the input parameters are given. For simplicity, we do not consider all possible subsets of items in our item decomposition scheme and restrict our attention solely on the subsets of items with consecutive indices where overlapping is not allowed. For the sake of brevity, we refer subsets of items as item windows and denote them as intervals – $[i, j]$ for any given $i \leq j$. For instance, let us consider an illustrative toy example where the length of the planning horizon is 4 periods and the number of items is 4. Also, let the decomposition scheme be $wit = 2, wper = 2$ and $ort = 0.5$. Then, we have three different time windows as, $[1, 2]$, $[2, 3]$, and $[3, 4]$, and two different subsets of items as $[1, 2]$ and $[3, 4]$.

In this decomposition scheme, item and time windows are combined as follows. Let $\mathcal{N}$ and $\mathcal{K}$ respectively denote the ordered set of item windows and the ordered set of time windows. Then, we also let the ordered set $\mathcal{NK}$ denote the cartesian product of the associated sets, i.e. $\mathcal{NK} = \mathcal{N} \times \mathcal{K}$.

Each item and time window combination $(\gamma, \tau) \in \mathcal{NK}$ defines a sub-problem to be solved in each iteration of the proposed fix-and-optimize heuristic. Furthermore, for any given sub-problem (γ, τ), we define a set $w_{opt}^{\gamma,\tau}$ that contains item and production cycle triplets (p, i, j) that is related to both item window γ and time window τ. That is, $w_{opt}^{\gamma,\tau} = \{(p, i, j) : p \in \gamma, i \in \tau, j - 1 \in \tau,\ i < j\}$. For convenience, we also define the complement of

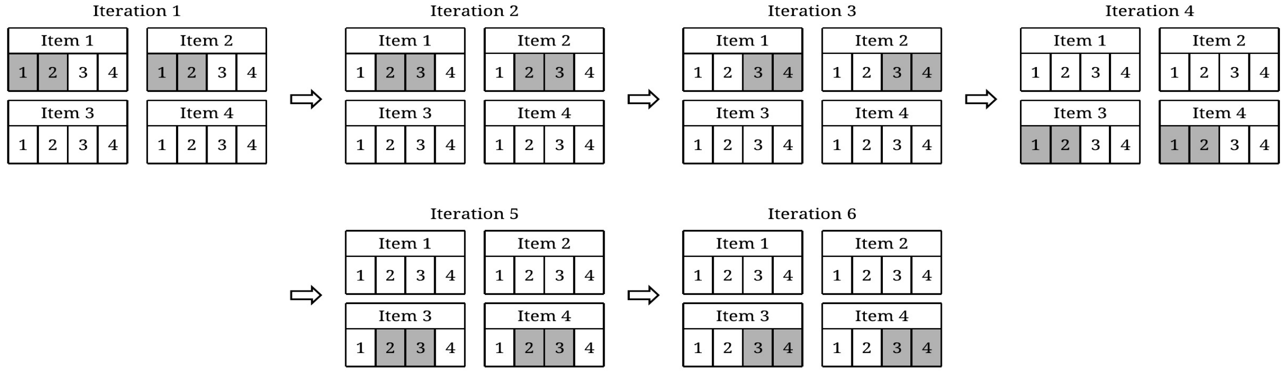

Figure 2. Example – fix-and-optimize heuristic with item-period oriented decomposition scheme.

$w_{opt}^{\gamma,\tau}$ as $w_{fix}^{\gamma,\tau}$. The latter shows which binary variables are fixed, whereas the former refers which variables are to be optimized. For any given iteration, one can formulate sub-problem (γ, τ) by embedding the following constraints into the MIP model given in Section 3

$$x_{pij} = \hat{x}_{pij} \quad \forall (p, i, j) \in w_{fix}^{\gamma,\tau} \tag{3}$$

where $\hat{x}_{pij}$ denotes the value of the binary setup variable x_{pij} in the best solution ever achieved in earlier iterations of the algorithm. After fixing the binary variables as shown in Eq. (3), a limited number of binary variables provided in the set $w_{opt}^{\gamma,\tau}$ are optimized by means of the off-the-shelf MIP solvers and then the minimum cost setup pattern is obtained. Then, the next iteration starts.

To better illustrate how the item-period oriented decomposition scheme works, we provide Fig. 2 for the toy example given above. Here, for each iteration, we have four boxes representing items; and for each item, we have four boxes representing periods. Also, for any item, the gray boxes stand for the periods where setup decisions are optimized, whereas the rest represent the periods in which setup decisions are fixed to the minimum cost setup pattern obtained in earlier iterations for the corresponding item.

4.2. The algorithm

We provide the basic structure of the adopted fix-and-optimize heuristic with item-period oriented decomposition in Algorithm 1. The algorithm starts with an initial solution. This solution can be found using one of the construction heuristics in the literature. Nevertheless, since the aim is to reduce the computational time, we use lot-for-lot setup pattern as a starting point as it is the case in [16]. Therefore, we first let $x_{pii+1} = 1$ for all $p \in [1, N]$ and $i \in [1, T]$, and refer this partial solution as x^{init}. Given this setup pattern, the initial problem is solved using function `SolveInitMIP()`. This function receives x^{init} as an input parameter and determines the optimal lot sizes and then returns the objective function value Z^{init} (line 2). Respectively in lines 3 and 4, the initial solution x^{init} and the objective function value Z^{init} are set as the best setup pattern x^* and the best objective value Z^* achieved so far.

After obtaining the initial solution, item-period oriented decomposition scheme is applied through function `decompose()` as described in Section 4.1. Specifically, this function first determines the all item and time windows based on the given input parameters (i.e. *wit*, *wper*, *ort*). Then, it generates a set $\mathcal{NK}$ that is comprised of all item and time window combinations. Here, each element in $\mathcal{NK}$ corresponds to a sub-problem and we iteratively solve those (lines 6–13). In each sub-problem, initially the associated sets $w_{fix}^{\gamma,\tau}$ and $w_{opt}^{\gamma,\tau}$ are defined (line 7). Subsequently, the current sub-problem is solved using the function `SolveMIP()` based on the input parameters x^*, $w_{opt}^{\gamma,\tau}$ and $w_{fix}^{\gamma,\tau}$. This function operates as follows. It first fixes the value of binary setup variables associated with $w_{fix}^{\gamma,\tau}$ to the values in the best solution achieved so far, i.e. x^*. Then, the resulting sub-problem is solved via an MIP solver. Finally, this function returns the optimal setup pattern $\tilde{x}$ and the corresponding objective function value $\tilde{Z}$ for the sub-problem under consideration (line 8). In the inner loop (lines 9–12), the best objective value Z^* and the corresponding setup pattern x^* is updated if the objective value obtained in the current sub-problem $\tilde{Z}$ is less than Z^*. In the final step of the each iteration, all variables previously fixed are set free. The algorithm terminates when all sub-problems are solved.

Algorithm 1: Fix-and-optimize heuristic

```
Input  : Problem parameters
Output: $x^*, Z^* \leftarrow$ best solution and the
        best objective function value
        found to the model given in
        Section 3
Initialize $x^{init}$;
$Z^{init} \leftarrow$ SolveInitMIP($x^{init}$);
$x^* = x^{init}$;
$Z^* = Z^{init}$;
$\mathcal{NK} \leftarrow$ decompose($wit, wper, ort$);
for $(\gamma, \tau) \in \mathcal{NK}$ do
    determine the sets $w_{fix}^{\gamma,\tau}$ and $w_{opt}^{\gamma,\tau}$;
    $(\tilde{x}, \tilde{Z}) \leftarrow$ SolveMIP($x^*, w_{fix}^{\gamma,\tau}, w_{opt}^{\gamma,\tau}$);
    if $\tilde{Z} \leq Z^*$ then
        $Z^* = \tilde{Z}$
        $x^* = \tilde{x}$
    end
    Unfix all previously fixed variables
end
```

5. Numerical study

In this section, we aim to conduct a numerical study in order to show the performance of the proposed heuristic – referred to as PH as compared to that Tempelmeier and Hilger [16] – referred to as TH. In what follows, we first present the design of the numerical study and then provide a discussion regarding our results and main findings.

In our test instances, we consider 32 discrete time periods and 32 items. For each item, period demands over the planning horizon are assumed to be normally distributed random variables with fixed coefficient of variation $\rho = 0.2$. The mean demand values of each item are drawn from a discrete uniform distribution on interval $[1, 100]$. We have generated 3 random instances. We use the same holding cost $h = 1$ and backlogging cost $b = 10$ in all instances. Also, we consider four fixed production costs $K \in \{250, 500, 1000, 2000\}$. Following [16], we use three capacity factors $\psi \in \{1.1, 1.5, 2\}$ in order to specify the capacity level over the planning horizon. More specifically, the capacity level can be calculated as follows.

$$C = \psi \times \frac{\sum_{k=1}^{N} \sum_{t=1}^{T} \mathbb{E}\, d_{kt}}{T}$$

Note that the capacity level C gets tighter as the value of capacity factor ψ decreases.

It should be remarked that the performance of a fix-and-optimize heuristic strongly depends on the decomposition scheme as it determines both the size and number of sub-problems to be solved. One would easily expect that the optimal solution of a sub-problem would approach to the optimal solution of the original problem as the size of the sub-problem increases. As such, the solution quality would improve as the size of sub-problem increases. Yet, it could be very challenging to solve large sub-problems from the computational point of view. To be able to demonstrate a detailed picture of how the proposed fix-and-optimize heuristic performs with varying decomposition schemes, we conduct a search on an extensive range of decomposition parameters. As stated earlier, we characterize the decomposition scheme with parameters $(wit, wper, ort)$. More specifically, as wit and/or $wper$ increases, the size of sub-problems increases. To control the growth of the sub-problem size, we define a set on $wit \times wper \in \{8, 16, 32, 64, 128, 256, 512, 1024\}$. In this set, we have excluded instances such as 2 and 4 since the resulting sub-problems are rather trivial and we predict that the solution quality of these cases would be poor. For values of wit and $wper$, we use orders of 2 where we omit cases larger than 32 (i.e. 2^5) due to the required excessive computational complexity in resulting sub-problems. Finally, we use the constant overlap rate $ort = 0.5$. This leads to 26 different $(wit, wper, ort)$ triplets in total. Table 1 presents all the values of decomposition scheme used in the numerical study.

Table 1. Parameters of the decomposition scheme.

$(wit, wper)$	(1,8), (2,4), (4,2), (1,16), (2,8), (4,4), (8,2), (2,16), (4,8), (8,4), (16,2), (2,32), (4,16), (8,8), (16,4), (32,2), (4,32), (8,16), (16,8), (32,4),(8,32), (16,16), (32,8), (16,32), (32,16), (32,32)
ort	0.5

The fix-and-optimize heuristics are implemented in Python where we use Gurobi v6.5.2 as MIP solver. All numerical experiments are performed on an Intel Core i7-5500 CPU with 8 GB RAM. Since we may not be able to solve each sub-problem to optimality within a reasonable time, we impose a time limit for each-sub-problem. In particular, the computational time required to

Table 2. Results of the numerical study.

Instance			TH				PH				GAP_3
#	ψ	K	DPAR	TIME	CTIME	GAP_1	DPAR	TIME	CTIME	GAP_2	
1	1.1	250	(2,8)	276.2	281.3	0.04	(32,8)	157.9	600.4	-4.22	-0.56
		500	(1,16)	265.0	289.6	0.00	(32,8)	301.5	600.5	-7.71	-0.90
		1000	(1,16)	263.2	328.2	-0.17	(16,8)	223.4	600.6	-8.14	-1.53
		2000	(1,16)	207.8	371.8	0.31	(8,8)	203.9	601.1	-5.65	-0.66
	1.5	250	(1,16)	160.4	203.7	0.00	(32,16)	129.9	600.3	-4.68	-0.46
		500	(1,16)	138.2	210.4	0.00	(32,16)	182.6	600.2	-6.38	-0.99
		1000	(1,16)	109.8	220.5	0.00	(32,16)	207.7	600.2	-7.96	-2.31
		2000	(1,16)	80.5	224.0	0.00	(16,16)	218.9	600.4	-7.05	-2.43
	2	250	(1,16)	151.2	201.0	0.00	(32,16)	103.2	600.2	-3.09	-0.62
		500	(1,16)	130.4	200.6	0.00	(32,16)	115.3	600.2	-5.11	-1.10
		1000	(1,16)	100.2	205.1	0.00	(32,16)	127.1	600.2	-6.17	-1.62
		2000	(1,16)	71.6	205.8	0.00	(32,16)	157.3	600.2	-6.34	-2.13
2	1.1	250	(1,16)	246.1	255.4	0.00	(16,8)	96.2	600.7	-4.22	-0.76
		500	(1,16)	232.2	274.5	-0.01	(16,8)	161.3	600.6	-6.92	-1.27
		1000	(1,16)	289.1	296.5	0.16	(8,16)	399.8	600.7	-8.33	-1.36
		2000	(4,8)	587.9	205.7	-0.74	(8,8)	198.0	601.1	-5.96	-0.94
	1.5	250	(1,16)	168.1	205.5	0.00	(32,16)	145.7	600.2	-3.93	-0.82
		500	(1,16)	149.9	209.8	0.00	(32,16)	153.6	600.2	-6.17	-1.44
		1000	(1,16)	115.3	219.8	0.00	(32,16)	171.6	600.2	-8.26	-2.19
		2000	(8,8)	601.3	85.1	-0.17	(8,16)	189.8	600.7	-4.74	-1.51
	2	250	(1,16)	158.5	200.2	0.00	(32,16)	123.6	600.3	-3.91	-0.51
		500	(1,16)	134.6	202.1	0.00	(32,16)	117.4	600.2	-4.55	-0.81
		1000	(1,16)	104.8	207.4	0.00	(32,16)	116.9	600.2	-6.28	-1.90
		2000	(1,16)	74.8	214.1	0.00	(32,16)	146.3	600.2	-6.90	-2.18
3	1.1	250	(2,8)	289.7	282.2	0.00	(32,8)	157.8	600.4	-5.41	-0.85
		500	(1,16)	270.9	296.7	-0.01	(32,8)	298.0	600.4	-8.27	-1.26
		1000	(1,16)	249.0	325.9	0.00	(8,8)	196.0	601.1	-5.44	-0.53
		2000	(1,16)	221.2	363.0	-0.03	(8,8)	208.1	601.1	-6.28	-0.23
	1.5	250	(1,16)	169.6	205.7	0.00	(32,16)	126.4	600.2	-3.56	-0.44
		500	(1,16)	140.7	210.3	0.00	(32,16)	151.6	600.2	-5.90	-0.83
		1000	(1,16)	113.9	217.9	0.00	(32,16)	190.6	600.2	-7.90	-1.76
		2000	(2,16)	600.8	212.9	-0.41	(8,16)	165.1	600.7	-5.03	-1.29
	2	250	(1,16)	163.0	201.9	0.00	(32,16)	111.9	600.2	-3.90	-0.27
		500	(1,16)	133.2	203.8	0.00	(32,16)	138.6	600.2	-5.73	-0.94
		1000	(1,16)	100.8	204.3	0.00	(32,16)	120.1	600.2	-6.59	-1.69
		2000	(1,16)	71.1	210.0	0.00	(32,16)	151.2	600.2	-8.06	-1.96

solve any sub-problem is limited with $600/\ell$ seconds, where ℓ refers to the total number of subproblems generated by the corresponding decomposition scheme. It should also be noted that while employing piecewise linearization approach, we have used 11 pieces (see e.g. [37]) to approximate the objective function values of formulation in Eq (2) and the model presented in Tempelmeier and Hilger [16].

For each test instance, we obtain the cost figures and solution times for both heuristics using the decomposition parameters given in Table 1. That is, we run PH and TH 936 times ($3 \times 3 \times 4 \times 26$). In Table 2, we present the results of the best solutions obtained by using PH and TH with varying decomposition parameters for each problem instance. In Table 2, we report the following statistics. DPAR represents the decomposition parameters $(wit, wper)$ used to yield the best objective value. TIME reports the computation time of the associated heuristic with the pivot decomposition parameters (in seconds), whereas CTIME corresponds to that of the other heuristic. GAP_1 and GAP_2 show the percentage objective value difference between PH and TH with the same pivot DPAR values. As such, GAP_1 (GAP_2) measures how far off PH (TH) from TH (PH) where TH (PH) reaches its best. Finally, GAP_3 shows the percentage cost gap between the best solutions obtained by PH and TH. Let us respectively denote the cost figures obtained by PH and TH as $C^{(PH)}$ and $C^{(TH)}$. Then, all GAP statistics are calculated as follows.

$$100 \times \frac{C^{(PH)} - C^{(TH)}}{C^{(TH)}}$$

Here, we remark that a negative GAP value implies that PH exhibits better performance as compared to TH in the instance under consideration.

The results of our numerical study show that PH dominates TH in expected cost performance for all instances tested. GAP_3 clearly demonstrates that PH always provide lower cost solutions in their respective best settings. Even in the settings favorable for TH, GAP_1 reveals that PH performs almost always better (or even) in expected cost. Obviously, GAP_2 becomes more dramatic in settings favorable for PH. In line with earlier expectations, the results show that high quality solutions are always obtained in settings having relatively large size sub-problems ($wit \times wper$). In most of the instances considered, TH yields its best solutions with decomposition parameters $(1, 16)$. If we turn our attention to the settings favorable for PH, it is observed that TH fails to solve any sub-problem to optimality within a given time limit when larger size of sub-problems are of concern. This result confirms the earlier findings indicating that the total number of binary variables to be optimized in a sub-problem should be 40 at most while TH is considered (see e.g. [16]). On the other hand, PH obtains its best solutions mostly with decomposition parameters $(32, 16)$. This implies that PH can solve sub-problems larger in size as compared to TH. From the computational point of view, the search conducted on the decomposition parameters reveals that the computation time of PH mostly depend on the $wper$ value chosen. That is because the set of binary variables to be optimized, $w_{opt}^{\gamma,\tau}$, grows quadratically with $wper$, whereas it is linearly dependent on wit.

If we just simply look at TIME and CTIME statistics, it appears that PH is more time-efficient although we should admit that there exists some instances where TH terminates faster. If we take a closer look to the instances tested, it can be observed (not very consistent though) that the cost gap between two heuristics reach higher levels as capacity factor and fixed ordering cost increase. Consequently, we can conclude that the PH arises as the promising MIP-based heuristic that can solve large-sized realistic problem instances within reasonable computational times.

6. Conclusion and future research directions

In this paper, we consider a capacitated multi-item stochastic lot-sizing problem. In particular, we extend the MIP models provided in [23] and [29] to the setting in which multi-item and backlogging cost are of concern. Nevertheless, the MIP models are often fail to solve the problem under consideration within a reasonable time as is the case in the proposed formulation. Therefore, we propose an item-period oriented decomposition scheme and combine it with a fix-and-optimize heuristic approach in order to obtain fast and high quality solutions. We then compare the performance of the proposed heuristic with the heuristic presented in Tempelmeier and Hilger [16]. The results clearly show that the proposed heuristic yields lower cost solutions as compared to the benchmark heuristic. We also demonstrate that the proposed heuristic performs very well for large-size problems as well. The results further reveal that the proposed heuristic exhibits a very competitive performance even for decomposition schemes where benchmark heuristic reaches its best.

Several research directions could be considered for future work. The extension of the model considered in this study to the case of setup carry-overs appears to be very promising. Also, we use a lot-for-lot setup pattern so as to obtain the initial solution. Therefore, it would also be of interest to combine the proposed heuristic with other well-known construction heuristics. Another promising research avenue would be to analyse a production system with multiple parallel machines instead of assuming single production resource. Finally, extending the mathematical model and the heuristic proposed in this study to the multi-level production systems may also be of interest.

Acknowledgments

We would like to thank two anonymous reviewers for their valuable feedback. Huseyin Tunc is supported by Social Sciences University of Ankara (ASBU-BAB) under Grant no SBF-2020-192.

References

[1] Axsäter, S. (2015). *Inventory control*. Vol. 225. Springer.

[2] Ramya, R., Rajendran, C., Ziegler, H., Mohapatra, S., & Ganesh, K. (2019). *Capacitated Lot Sizing Problems in Process Industries*. Springer.

[3] Wagner, H. M., & Whitin, T. M. (1958). Dynamic version of the economic lot size model. *Management science*, 5(1), 89-96.

[4] Florian, M., Lenstra, J. K., & Rinnooy Kan, A. H. G. (1980). Deterministic production planning: Algorithms and complexity. *Management science*, 26(7), 669-679.

[5] Bitran, G. R., & Yanasse, H. H. (1982). Computational complexity of the capacitated lot size problem. *Management Science*, 28(10), 1174-1186.

[6] Karimi, B., Ghomi, S. F., & Wilson, J. M. (2003). The capacitated lot sizing problem: a review of models and algorithms. *Omega*, 31(5), 365-378.

[7] Robinson, P., Narayanan, A., & Sahin, F. (2009). Coordinated deterministic dynamic demand lot-sizing problem: A review of models and algorithms. *Omega*, 37(1), 3-15.

[8] Buschkühl, L., Sahling, F., Helber, S., & Tempelmeier, H. (2010). Dynamic capacitated lot-sizing problems: a classification and review of solution approaches. *Or Spectrum*, 32(2), 231-261.

[9] Brahimi, N., Absi, N., Dauzère-Pérès, S., & Nordli, A. (2017). Single-item dynamic lot-sizing problems: An updated survey. *European Journal of Operational Research*, 263(3), 838-863.

[10] Hu, Z., & Hu, G. (2016). A two-stage stochastic programming model for lot-sizing and scheduling under uncertainty. *International Journal of Production Economics*, 180, 198-207.

[11] Bookbinder, J. H., & Tan, J. Y. (1988). Strategies for the probabilistic lot-sizing problem with service-level constraints. *Management Science*, 34(9), 1096-1108.

[12] Tempelmeier, H. (2011). A column generation heuristic for dynamic capacitated lot sizing with random demand under a fill rate constraint. *Omega*, 39(6), 627-633.

[13] Tempelmeier, H., & Herpers, S. (2010). ABC β–a heuristic for dynamic capacitated lot sizing with random demand under a fill rate constraint. *International Journal of Production Research*, 48(17), 5181-5193.

[14] Helber, S., Sahling, F., & Schimmelpfeng, K. (2013). Dynamic capacitated lot sizing with random demand and dynamic safety stocks. *OR Spectrum*, 35(1), 75-105.

[15] Helber, S., & Sahling, F. (2010). A fix-and-optimize approach for the multi-level capacitated lot sizing problem. *International Journal of Production Economics*, 123(2), 247-256.

[16] Tempelmeier, H., & Hilger, T. (2015). Linear programming models for a stochastic dynamic capacitated lot sizing problem. *Computers & Operations Research*, 59, 119-125.

[17] De Smet, N., Minner, S., Aghezzaf, E. H., & Desmet, B. (2020). A linearisation approach to the stochastic dynamic capacitated lotsizing problem with sequence-dependent changeovers. *International Journal of Production Research*, 58(16), 4980-5005.

[18] Choudhary, D., Shankar, R., Tiwari, M. K., & Purohit, A. K. (2016). VMI versus information sharing: an analysis under static uncertainty strategy with fill rate constraints. *International Journal of Production Research*, 54(13), 3978-3993.

[19] Hilger, T., Sahling, F., & Tempelmeier, H. (2016). Capacitated dynamic production and remanufacturing planning under demand and return uncertainty. *OR Spectrum*, 38(4), 849-876.

[20] Koca, E., Yaman, H., & Aktürk, M. S. (2015). Stochastic lot sizing problem with controllable processing times. *Omega*, 53, 1-10.

[21] Liu, K., & Zhang, Z. H. (2018). Capacitated disassembly scheduling under stochastic yield and demand. *European Journal of Operational Research*, 269(1), 244-257.

[22] Pauls-Worm, K. G., Hendrix, E. M., Alcoba, A. G., & Haijema, R. (2016). Order quantities for perishable inventory control with nonstationary demand and a fill rate constraint. *International Journal of Production Economics*, 181, 238-246.

[23] Tunc, H (2019). The capacitated stochastic lot sizing problem with convex processing time compression cost. *Working paper.*

[24] Tunc, H., Kilic, O. A., Tarim, S. A., & Eksioglu, B. (2013). A simple approach for assessing the cost of system nervousness. *International Journal of Production Economics*, 141(2), 619-625.

[25] Li, L., Song, S., Wu, C., & Wang, R. (2017). Fix-and-optimize and variable neighborhood search approaches for stochastic multi-item

capacitated lot-sizing problems. *Mathematical Problems in Engineering*, 2017.

[26] Liang, J., Wang, Y., Zhang, Z. H., & Sun, Y. (2019). Energy efficient production planning and scheduling problem with processing technology selection. *Computers & Industrial Engineering*, 132, 260-270.

[27] Meistering, M., & Stadtler, H. (2017). Stabilized-Cycle Strategy for Capacitated Lot Sizing with Multiple Products: Fill-Rate Constraints in Rolling Schedules. *Production and Operations Management*, 26(12), 2247-2265.

[28] Tavaghof-Gigloo, D., & Minner, S. (2020). Planning approaches for stochastic capacitated lot-sizing with service level constraints. *International Journal of Production Research.*

[29] Tunc, H., Kilic, O. A., Tarim, S. A., & Eksioglu, B. (2014). A reformulation for the stochastic lot sizing problem with service-level constraints. *Operations Research Letters*, 42(2), 161-165.

[30] Porteus, E. L. (2002). *Foundations of stochastic inventory theory.* Stanford University Press.

[31] Silver, E. A., Pyke, D. F., & Peterson, R. (1998). *Inventory management and production planning and scheduling.* Vol. 3. New York: Wiley.

[32] Tarim, S. A., & Kingsman, B. G. (2006). Modelling and computing (R_n, S_n) policies for inventory systems with non-stationary stochastic demand. *European Journal of Operational Research*, 174(1), 581-599.

[33] Tunc, H., Kilic, O. A., Tarim, S. A., & Eksioglu, B. (2016). The stochastic lot sizing problem with piecewise linear concave ordering costs. *Computers & Operations Research*, 65, 104-110.

[34] Tunc, H., Kilic, O. A., Tarim, S. A., & Rossi, R. (2018). An extended mixed-integer programming formulation and dynamic cut generation approach for the stochastic lot-sizing problem. *INFORMS Journal on Computing*, 30(3), 492-506.

[35] Frenzen, C. L., Sasao, T., & Butler, J. T. (2010). On the number of segments needed in a piecewise linear approximation. *Journal of Computational and Applied mathematics*, 234(2), 437-446.

[36] Gavrilović, M. M. (1975). Optimal approximation of convex curves by functions which are piecewise linear. *Journal of Mathematical Analysis and Applications*, 52(2), 260-282.

[37] Rossi, R., Tarim, S. A., Prestwich, S., & Hnich, B. (2014). Piecewise linear lower and upper bounds for the standard normal first order loss function. *Applied Mathematics and Computation*, 231, 489-502.

[38] Sahling, F., Buschkühl, L., Tempelmeier, H., & Helber, S. (2009). Solving a multi-level capacitated lot sizing problem with multi-period setup carry-over via a fix-and-optimize heuristic. *Computers & Operations Research*, 36(9), 2546-2553.

[39] Pochet, Y., & Wolsey, L. A. (2006). *Production planning by mixed integer programming.* Springer Science & Business Media.

M. Edib Gurkan *is Assistant Professor in the Department of Management Information Systems at Karabuk University. He recieved his Ph.D. degree (2016) in Business Administration from Hacettepe University. His research interests are inventory control/optimization, mathematical programming and supply chain management.*
https://orcid.org/0000-0003-1961-0215

Huseyin Tunc *is Assistant Professor in the Department of Business Administration at Social Sciences University of Ankara. He received his Ph.D. in Industrial and Systems Engineering from Mississippi State University in 2012. He joined the Social Sciences University of Ankara in 2019. Prior to joining SSUA, he worked at Hacettepe University, Swiss Federal Institute of Technology in Lausanne (EPFL), and Mississippi State University.*
https://orcid.org/0000-0001-5508-3702