Heuristic Methods for The Capacitated Stochastic Lot-Sizing Problem Under The Static-Dynamic Uncertainty Strategy

A. Cem Randa
University of Chicago, Booth School of Business, 5807 South Woodlawn Avenue Chicago, IL 60637,USA , randa@chicagobooth.edu

Mustafa K. Doğru
Verizon Wireless, One Verizon Way, Basking Ridge, NJ 07920, USA, mkdacademic@gmail.com

Cem Iyigun*
Department of Industrial Engineering, Middle East Technical University, 06531 Ankara, Turkey, iyigun@metu.edu.tr

Ulaş Özen
Ozyegin University, Faculty of Business, Cekmekoy Campus, Nisantepe District, Orman Street, 34794 Cekmekoy, Istanbul, Turkey, ulas.ozen@ozyegin.edu.tr

We consider a lot-sizing problem in a single-item single-stage production system facing non-stationary stochastic demand in a finite planning horizon. Motivated by practice, the set-up times need to be determined and frozen once and for all at the beginning of the horizon while decisions on the exact lot sizes can be deferred until the setup epochs. This operating scheme is referred to as the static-dynamic uncertainty strategy in the literature. It has been shown that a modified base stock policy is optimal for a capacitated system with minimum lot size restrictions under the static-dynamic uncertainty strategy. However, the optimal policy parameters require an exhaustive search for which the computational time grows exponentially in the number of periods in the planning horizon. In order to alleviate the computational burden for real-life size problems, we develop and test seven different heuristics for computational efficiency and solution quality. Our extensive numerical experiments show that average optimality gaps less than 0.1% and maximum optimality gaps below 4% can be attained in reasonable running times by using a combination of these heuristics.

Key words: stochastic lot-sizing problem, static-dynamic uncertainty, non-stationary, heuristic, penalty cost

1. Introduction

This paper considers a multi-period single-item stochastic capacitated lot-sizing problem. Costs are non-stationary and they consist of linear production, inventory holding and penalty costs, and fixed set-up cost. Any unfulfilled demand is backlogged and satisfied as soon as possible. There exists restrictions on minimum and maximum number of products that can be produced in any period. The objective is to find the periods in which production will take place and the corresponding production quantities such that the total expected cost is minimized.

Contrary to its deterministic counterpart (i.e., deterministic lot sizing problems), which has been studied extensively in the literature, the stochastic lot sizing problem poses different alternatives depending on how production decisions are made and executed. Bookbinder and Tan (1988) mention three strategies: static uncertainty strategy, dynamic uncertainty strategy and static-dynamic uncertainty strategy. Under static uncertainty strategy, the timing and quantities are fixed at the beginning of the planning horizon once and for all. The practical benefit of following this strategy

* Corresponding Author

is reduced system nervousness due to the fixed production plan. However, this strategy is inflexible and cannot respond to the demand realizations. A practical remedy is to solve the problem in a rolling horizon fashion in pre-set frequencies to partially alleviate this inflexibility without creating excessive nervousness in the system. Under dynamic uncertainty strategy, neither the timing nor the quantities are fixed at the beginning of the planning horizon, and determined as time evolves. This strategy allows flexibility for production/inventory management by adjusting the inventory position at the beginning of each period and reduces inventory holding and backlog costs. However, it is bound to create higher system nervousness. Finally, in static-dynamic uncertainty strategy, the timing of the production is fixed at the beginning of the planning horizon but the exact quantities are determined later in the horizon. Bookbinder and Tan (1988) consider this strategy a more accurate representation of industrial practice since freezing schedules is a common practice to decrease system nervousness. Inderfurth (1994) notes that nervousness due to "deviations in delivery timing" requests is considered to be more severe in practice compared to "deviations in quantities". Note that only "deviations in quantities" are present in static-dynamic uncertainty strategy.

In a recent study, Özen et al. (2012) introduced an exact formulation for the problem under static-dynamic uncertainty strategy and showed that the optimal policy is a modified based stock policy. They proposed two dynamic programming based heuristics that are showed to perform well for uncapacitated problems. In this paper, we focus on the capacitated version of the problem with minimum production quantities and capacity limits. We developed two groups of heuristic solutions. The first group of heuristics is modifications of the dynamic programming based (DP-based) heuristics by Özen et al. (2012) while the second group is search based. We evaluated all of our heuristic procedures in terms of optimality gap and computational time. Interestingly, our extensive numerical experiments show that these two types of heuristics show complementary performance. DP-based heuristics perform better in the scenarios where capacity constraints are not restrictive (i.e., high maximum lot size regime) whereas search-based heuristics achieve better results when capacity restrictions are tight (i.e., low maximum lot size regime). We statistically tested which combinations of these two groups of heuristics dominate the others. The best performing combinations can achieve less than 1% optimality gap under certain parameter regions. We also observed that computational time of our heuristics grow polynomially in the number of periods.

Remaining of the paper is organized as follows. In Section 2, we provide a literature review. Section 3 introduces the model in detail and the notation used. Section 4 discusses the optimal policy and how optimal policy parameters can be calculated. Sections 5 and 6 are dedicated to our dynamic programming and search-based heuristics, respectively. We present the numerical study in Section 7 with a brief discussion of the findings. Finally, we conclude in Section 8. Additional discussion on lower bounds and detailed results of the numerical study are presented in the Online Appendix.

2. Literature Review

The literature on deterministic lot sizing problem is very extensive. We refer to Brahimi et al. (2006) for a comprehensive review on single-item lot-sizing problem. In this section, we review the stochastic lot-sizing literature in three groups based on the strategy used (i.e., static uncertainty, dynamic uncertainty and static-dynamic uncertainty).

Considering static uncertainty strategy, Vargas (2009) studies the stochastic version of the classical lot-sizing problem by Wagner and Whitin (1958), and determines the optimal solution by constructing an acyclic network and solving for a shortest path. Sox (1997) solves Wagner-Whitin model with random demand and backlogging, and constructs a mixed integer nonlinear program with non-stationary costs. Sox and Muckstadt (1997) extend this model to multi-item, capacitated case, and proposes a near optimal procedure through solving for Lagrangian dual. Tempelmeier

(2011) studies multi item capacitated lot sizing problem with random demand under fill rate constraints and develops an approximate set partitioning model, which is then solved by an heuristic based on column generation. We refer to Sox et al. (1999) for a review of stochastic multi-item lot-sizing problems, i.e., stochastic lot scheduling problems.

One standard approach of modelling dynamic uncertainty strategy is using stochastic dynamic programming. Scarf (1959) introduces the concept of K-convexity and proves that (s, S) policy is optimal for such problems. Veinott and Wagner (1965) extend the Wagner-Within model by considering stochastic demand and positive lead times. They develop a computational approach for finding optimal (s, S) inventory policy parameters. Sethi and Cheng (1997) consider a generalised model with depended Markovian demand and prove that (s, S) policy is optimal for the generalised model and its several extensions. Another common approach for modeling uncertainty in demand is to use scenario trees. Guan and Miller (2008) model the stochastic lot-sizing problem with a scenario based demand distribution and suggest a polynomial time algorithm making use of optimality conditions. Following the same scenario tree based approach, Huang and Kucukyavuz (2008) consider stochastic and dependent costs, demands and order lead times, and develop a polynomial time algorithm in tree size for finding an optimal solution.

The static-dynamic uncertainty strategy is first described by Bookbinder and Tan (1988) who proposed a two-step heuristic solution method using a Wagner-Whitin type model. Tarim and Kingsman (2004) address the same problem and provide a mixed integer programming (MIP) model that solves simultaneously for the exact timing of future replenishments and corresponding order quantities. In a separate study, Tarim et al. (2009) focus on the computational issues and provide an efficient computation approach to solve the MIP model by Tarim and Kingsman (2004). Tempelmeier (2007) studies the same problem under fill rate constraint. In all of the works above, the common approach is to formulate the stochastic problem using certainty equivalent mathematical programs and analyze the resulting deterministic problems. Özen et al. (2012) take a different approach and provide an exact formulation for the problem under static-dynamic uncertainty strategy. After showing that the optimal policy is a modified based stock policy, they develop two dynamic programming based heuristics. Their main focus is the uncapacitated problem and the proposed heuristics are shown to perform well for this regime. In this paper, we extend their analysis to problems with minimum production quantities and capacity limits, which increase the complexity of the problem considerably. We remark that deterministic lot-sizing problems are a subclass of stochastic lot-sizing problems under static dynamic uncertainty strategy. Therefore, the complexity results of deterministic capacitated lot-sizing problems, which are known to be hard (see Bitran and Yanase (1982) and Brahimi et al. (2006)), applies to the problem we study here. The main contribution of our study is the development of effective and efficient heuristic solutions to this complex problem.

3. Model and Notation

In this section, we introduce the model and the basic assumptions. We follow the same notation used in Özen et al. (2012).

Consider a single-stage single-item production system in a finite horizon under periodic review. The planning horizon consists of equal length time periods (days, weeks, etc.). Let $t \in \{1, 2, ..., N\}$ be the index for periods with N being the last period in the planning horizon. Let D_t denote non-negative random demand in period t, which is independent but not necessarily identical for each period. At the beginning of period t, a fixed cost of A_t is incurred if production is scheduled for period t. We assume that the production environment is capacitated. If there is production in period t, then the production quantity should be greater than or equal to a minimum lot size u_t and less than or equal to a capacity level (maximum lot size) o_t. The unit production cost in period t is denoted by c_t. If there is unfulfilled demand in period t, then it is backlogged to be satisfied

in the following periods, and a unit penalty cost of b_t is incurred at the end of period t. Similarly, if there is inventory on hand at the end of a period t, holding cost of h_t per unit is charged. We assume that all cost and lot size parameters are nonnegative: $c_t, h_t, b_t, A_t, u_t, o_t \geq 0$.

The system evolves as follows. The *production periods* are determined at the beginning of period 1 in accordance with static-dynamic uncertainty strategy. The function δ_t indicates these production periods: $\delta_t = 1$ if there is production in period t, and $\delta_t = 0$ otherwise. The consecutive periods which are to be covered by the same lot produced, i.e. the periods between two consecutive production epochs, is called a production cycle. Index $i \in \{1, \dots m\}$ denotes these production cycles where m is the last cycle. Let $(m, \mathbf{r})$ be a *production schedule*, where m denotes the number of lots to be produced in the planning horizon ($m \leq N$) and $\mathbf{r} = (\mathbf{r_1}, \mathbf{r_2}, \dots, \mathbf{r_m})$ is an ordered vector of periods in which there is production such that $\delta_{r_i} = 1$ for $i = 1, \dots, m$. We assume that production is scheduled at the beginning of the planning horizon, i.e. $\delta_1 = 1$. The following sequence of events takes place in a given production period t (ie., $\delta_t = 1$):

- At the beginning of the period, fixed cost A_t is incurred.
- The inventory level is observed. The current cycle's lot size x_i is determined, $o_t \geq x_i \geq u_t$. Then instantaneous production takes place and the production cost $c_t x_i$ is incurred (*at the beginning of the period*);
- Demand occurs within the period, D_t;
- Holding and penalty costs are incurred (*at the end of the period*).

If period t is not a production period (ie. $\delta_t = 0$), the sequence of events is slightly different:

- Inventory level is observed at the beginning of the period. No lot is produced;
- Demand occurs within the period, D_t;
- Holding and penalty costs are incurred (*at the end of the period*).

Consider the production cycle i. Let I_i and x_i denote the net inventory level at the beginning of the cycle and the lot size, respectively. Then, $y_i = I_i + x_i$ denotes the inventory level just after the production in cycle i, and the expected holding and penalty costs incurred during the cycle given y_i is :

$$\mathcal{L}_i(y_i) = \mathsf{E}\left[\sum_{k=r_i}^{r_{i+1}-1}\left(h_k\Big(y_i - \sum_{j=r_i}^{k} D_j\Big)^+ + b_k\Big(y_i - \sum_{j=r_i}^{k} D_j\Big)^-\right)\right], \tag{1}$$

where $w^+ = \max\{w, 0\}$ and $w^- = \max\{-w, 0\}$ for any $w \in \mathbb{R}$. Moreover, the starting inventory level at each production cycle is given by the following inventory balance equations:

$$I_{i+1} = I_i + x_i - \sum_{k=r_i}^{r_{i+1}-1} D_k, \qquad i = 1, \dots, m, \tag{2}$$

where I_1 and I_{m+1} are the net inventory at the beginning and the end of the planning horizon, respectively, and $r_{m+1} = N + 1$.

Our objective is to find the production schedule $(m, \mathbf{r})$ and the inventory levels just after production (y_i) in the corresponding production cycle i ($\equiv$ determining lot size x_i) for all $i = 1, \dots, m$ such that the expected total cost over the planning horizon is minimized, or equivalently, to solve the following problem:

Problem P:

$$\min_{(m,\mathbf{r})}\left\{\sum_{k=1}^{N} A_k\delta_k + \min_{y_1: I_1+o_{r_1} \geq y_1 \geq I_1+u_{r_1}}\left\{c_{r_1}(y_1 - I_1) + \mathcal{L}_1(y_1) + \mathsf{E}_{D_{r_1},\dots,D_{r_2-1}}\Big[\min_{y_2: I_2+o_{r_2} \geq y_2 \geq I_2+u_{r_2}}\Big\{\right.\right.$$
$$c_{r_2}(y_2 - I_2) + \mathcal{L}_2(y_2) + \mathsf{E}_{D_{r_2},\dots,D_{r_3-1}}\Big[\cdots\Big[\min_{y_m: I_m+o_{r_m} \geq y_m \geq I_m+u_{r_m}}\big\{c_{r_m}(y_m - I_m) + \mathcal{L}_m(y_m) +$$

$$\mathsf{E}_{D_{r_m},\ldots,D_{r_{m+1}-1}}\big[B(I_{m+1})\big]\}]\cdots\Big]\Big\}\Big]\Big\}\Big\}.$$

We assume that the anticipated cost of inventory at the end of the planning horizon is a linear function of the ending inventory level i.e. $B(q) = -c_N q$, $\forall q \in \mathbb{R}$. In other words, if there is on hand inventory at the end of the horizon, it will be sold at the production cost of the last period; if there is backlog, it will be cleared at the same cost. Finally, we assume $c_t + \sum_{i=t}^{N} h_i > c_N \;\; \forall t \in \{1,2,...,N\}$ in order to prevent speculative ordering. If this condition is not satisfied, then it would be more beneficial to produce items as much as possible in a certain period, knowing that clearing the excess at the end of the horizon will be cost effective.

Next, we discuss the structure of the optimal policy for Problem P and present an approach for finding optimal policy parameters.

4. Optimal Policy for Problem P

Consider Problem P under a given production schedule $(m, \mathbf{r})$. For a given starting inventory level I_i, let $y_i^*(I_i)$ be the optimal inventory level just after production in cycle i for $i = 1,\ldots,m$. The optimal inventory levels $(y_1^*(I_1),\ldots,y_m^*(I_m))$ can be found by the solving a backward dynamic program with m stages where a stage is a production cycle. The recursive function is given by

$$f_{m+1}(I_{m+1}) = -c_N I_{m+1} \tag{3}$$

$$f_i(I_i) = \min_{y_i : I_i + o_{r_i} \geq y_i \geq I_i + u_{r_i}} G_i(y_i) \tag{4}$$

for $i = 1,\ldots,m$ where $f_i(I_i)$ is the minimum expected cost of the system over production cycles $i, i+1,\ldots,m$ (i.e. over periods $r_i, r_i+1,\ldots,N$) given the net inventory level at the beginning of production cycle i, I_i, and

$$G_i(y_i) := A_{r_i} + c_{r_i}(y_i - I_i) + \mathcal{L}_i(y_i) + \mathsf{E}\Big[f_{i+1}\Big(y_i - \sum_{j=r_i}^{r_{i+1}-1} D_j\Big)\Big]. \tag{5}$$

Starting from $f_m(I_m)$ and solving for $f_1(I_1)$ by backward recursion gives $\mathbf{y}^* = (y_1^*(I_1),\ldots,y_m^*(I_m))$ for a given production schedule $(m, \mathbf{r})$ and opening net inventory level I_1.

A modified base stock policy with base stock levels $\mathbf{S} = (\mathbf{S_1}, \mathbf{S_2}, \ldots, \mathbf{S_m})$ is such that the inventory level just after production in cycle i is

$$y_i(I_i) = \begin{cases} I_i + u_i & \text{if} \quad S_i - I_i < u_{r_i} \\ I_i + o_i & \text{if} \quad S_i - I_i > o_{r_i} \\ S_i & \text{otherwise.} \end{cases} \tag{6}$$

The following theorem by Özen et al. (2012) characterizes the optimal policy for Problem P given a production schedule.

THEOREM 1. *Optimal policy for Problem P under a given production schedule* $(m, \mathbf{r})$ *is a modified base stock policy with base stock levels* $\mathbf{S}^* = (\mathbf{S_1^*}, \mathbf{S_2^*}, \ldots, \mathbf{S_m^*})$ *that are given by*

$$S_i^* = \arg\min_{y_i}\{G_i(y_i)\}, \quad i = 1,2,\ldots,m.$$

Since $G_i(.)$ is a convex function in y_i, a simple binary search on the interval $[0,\infty)$ is sufficient to find S_i^* value for all $i = 1,\ldots,m$.

Even though finding the optimal policy parameters for a given production schedule can determined efficiently, every possible production schedule should be evaluated individually in order to find the optimal solution for Problem P. This results in an exponentially growing solution space in the length of the planning horizon N. Nevertheless, tight lower bounds can help to eliminate the effort to solve for optimal base stock levels for certain production schedules, which is discussed next.

4.1. Computing the Parameters of an Optimal Policy

Determining the optimal base stock levels for a given production schedule is a relatively easy task. The main difficulty in solving Problem P is determining the optimal production schedule. The best know approach is to search greedily over 2^{N-1} different production schedules, and the optimal solution is the one with the minimum cost. The efficiency of this search can be improved by tight lower bounds. Özen et al. (2012) introduced two lower bounds, called LB_1 and LB_2, for the uncapacitated version of Problem P ($o_t = \infty$ and $u_t = 0$ for all t), and showed the effectiveness of these lower bounds with numerical experiments. However, we observed that the quality of these lower bounds do deteriorate in tighter capacity and minimum order quantity regimes. Therefore, we developed a third lower bound, called LB_3, based on linear approximation of cycle cost functions. Although finding LB_3 requires solving an additional DP that is computationally expensive, a combined application of these three lower bounds reduces the overall solution time for finding the optimal solution. We discuss the details of LB_1, LB_2 and LB_3 in Online Appendix 1. When evaluating a certain production schedule, we utilize the following procedure for computing the parameters of an optimal policy.

Procedure 1 Computing the Parameters of an Optimal Policy

Input: Best cost obtained so far ($BestCostSoFar$)
Compute LB_1
if $LB_1 <$ BestCostSoFar **then**
  Compute LB_2
  if $LB_2 <$ BestCostSoFar **then**
    Compute LB_3
    if $LB_3 <$ BestCostSoFar **then**
      Solve for the optimal parameters. Let the optimal cost for the current production schedule be C.
      if $C < BestCostSoFar$ **then**
        $BestCostSoFar := C$. Continue with the next production schedule.
      end if
    end if
  end if
end if

We evaluate all 2^{N-1} possible production schedules using the procedure above to compute the optimal policy parameters. Nevertheless, enumerating all feasible production schedules is an exponentially growing search and is not suitable for problems with longer planning horizons (large N). Our numerical experiments demonstrate that even for $N = 18$, the computation times can be in the order of days. Due to this scalability issue in search by enumeration, we develop heuristics for Problem P. While the first group of heuristics utilize different DP-based approaches that approximate the true cost-to-go function, the second group uses greedy search among possible replenishment schedules.

5. Dynamic Programming Based Heuristics

This section presents the heuristic procedures that are based on dynamic programming. We start with introducing a heuristic procedure by Özen et al. (2012), which simultaneously determines the production schedule and base stock levels. However, performance of this heuristic deteriorates with tighter capacity and minimum order quantity constraints. Therefore, we introduce two new modified heuristics to improve the performance.

5.1. Approximation Heuristic (AH)

The main obstacle in finding an optimal solution for Problem P emanates from the combinatorial nature of the outmost minimization problem. As a remedy, Özen et al. (2012) introduces a heuristic based on a backward recursion that selects the replenishment schedule and the corresponding base

stock levels simultaneously. AH heuristic solves the following problem for each period $t = N, \ldots, 1$ in backward fashion.

Consider period t. Define

- $\hat{y}_t$: inventory level at the beginning of period t just after production
- $\hat{a}_t$: number of periods covered by the lot produced in t

In period t, only the current periods decisions, $\hat{a}_t$ and $\hat{y}_t$, are optimized by solving the following minimization problem:

$$\min_{\substack{\hat{y}_t:\hat{y}_t\geq 0\\ \hat{a}_t:N-t+1\geq\hat{a}_t\geq 1}} \Big\{A_t + \hat{y}_t c_t + \hat{\mathcal{L}}_t(\hat{a}_t,\hat{y}_t) + \mathsf{E}\Big[\varphi_{t+\hat{a}_t}(\hat{y}_t - \sum_{j=t}^{t+\hat{a}_t-1} D_j)\Big]\Big\}, \tag{7}$$

where

$$\hat{\mathcal{L}}_t(\hat{a}_t,\hat{y}_t) = \mathsf{E}\Big[\sum_{k=t}^{t+\hat{a}_t-1} h_k\Big(\hat{y}_t - \sum_{j=t}^{k} D_j\Big)^+ + b_k\Big(\hat{y}_t - \sum_{j=t}^{k} D_j\Big)^-\Big] \tag{8}$$

$\varphi_l(.)$ is the recursive function defined for any $N \geq l > t$ as

$$\varphi_l(q) = A_l + c_l(p_l(q,\hat{S}_l^*) - q) + \hat{\mathcal{L}}_l(\hat{a}_l^*, p_l(q,\hat{S}_l^*)) + \mathsf{E}\Big[\varphi_{l+\hat{a}_l^*}(p_l(q,\hat{S}_l^*) - \sum_{j=l}^{l+\hat{a}_l^*-1} D_j)\Big] \tag{9}$$

where $\varphi_{N+1}(q) = c_N q$, $p_l(q,\hat{S}_l^*)$ is the production function that determines the inventory position after production with the following definition

$$p_l(q_1,q_2) = \begin{cases} q_1 + u_l & \text{if } q_2 - q_1 \leq u_l \\ q_1 + o_l & \text{if } q_2 - q_1 \geq o_l \\ q_2 & \text{otherwise,} \end{cases}$$

and $(\hat{a}_l^*, \hat{S}_l^*)$ is the optimal solution to the problem (7) for period $l > t$:

$$(\hat{a}_l^*, \hat{S}_l^*) = \arg \min_{\substack{\hat{a}_l:N-l+1\geq\hat{a}_l\geq 1\\ \hat{y}_l:\hat{y}_l\geq 0}} \Big\{A_l + \hat{y}_l c_l + \hat{\mathcal{L}}_l(\hat{a}_l,\hat{y}_l) + \mathsf{E}\Big[\varphi_{l+\hat{a}_l}(\hat{y}_l - \sum_{j=l}^{l+\hat{a}_l-1} D_j)\Big]\Big\},$$

which is already known when the optimization problem (7) is solved for period t.

Solving (7) for each period $t = N, \ldots, 1$ in backward fashion provides a solution for Problem P with setups placed in periods $r_1 = 1$, $r_2 = 1 + \hat{a}_1^*$,$r_3 = 1 + \hat{a}_1^* + \hat{a}_{1+\hat{a}_1^*}^*$ and so on, and corresponding base stock levels $\hat{S}_{r_1}^*$, $\hat{S}_{r_2}^*$, $\hat{S}_{r_3}^*$ and so on.

AH is an heuristic procedure and does not guarantee optimal solution for Problem P since it assumes zero starting inventory level while solving the problem in (7) for period t, i.e., $\hat{y}_t > 0$, and this solution is used while solving the problem for $t-1$. Hence, AH neglects the impact of the actions that would be taken in the prior periods on the opening inventory level. Nevertheless AH is a polynomial time procedure with complexity of $O(N^2)$ in terms of number of cycles.

5.2. Approximation Heuristic Extension I (AH I)

AH is based on the assumption that the starting inventory of each period will be zero. However, this assumption would not be accurate in the presence of limits on production quantities. In order to decide about the current period accurately, we require information about starting inventory distribution that is determined by the decisions likely to be taken in the prior periods. In order to tackle this problem, we consider to solve nested DP's simultaneously which will still eliminate the exponential growth of the problem in number of periods.

This procedure works as follows. Consider period t. Define the following additional decision variables:

• $\bar{y}_k$: inventory level at the beginning of period k just after production such that period t is the ending period and $\hat{a}_t$ is the number of periods to cover from period t.
• $\bar{a}_k$: number of periods covered by the setup scheduled in k just after the production such that period t is the ending period and $\hat{a}_t$ is the number of periods to cover from period t.
Then, for period t, we solve the following problem.

$$\min_{\substack{\hat{y}_t:\hat{y}_t\geq 0\\ \hat{a}_t:N-t+1\geq\hat{a}_t\geq 1}} \{Q_1^t(\hat{a}_t,\hat{y}_t)\} \tag{10}$$

where

$$Q_k^t(\hat{a}_t,\hat{y}_t) = \min_{\substack{\bar{y}_k:\bar{y}_k\geq 0\\ \bar{a}_k:t-k\geq\bar{a}_k\geq 1}} \Big\{A_k+\bar{y}_k c_k+\hat{\mathcal{L}}_k(\bar{a}_k,\bar{y}_k)+\mathsf{E}\Big[\phi_{k+\bar{a}_k}(\bar{y}_k-\sum_{j=k}^{k+\bar{a}_k-1}D_j)\Big]\Big\} \tag{11}$$

The definition of $\phi_k(.)$ is given as follows:

$$\phi_k(q) = A_k+(p_k(q,\bar{S}_k^*)-q)c_k+\hat{\mathcal{L}}_k(\bar{a}_k^*,p_k(q,\bar{S}_k^*))+\mathsf{E}\Big[\phi_{k+\bar{a}_k^*}(p_k(q,\bar{S}_k^*)-\sum_{j=k}^{k+\bar{a}_k^*-1}D_j)\Big] \quad \forall k<t$$

and

$$\phi_t(q) = A_t+(p_t(q,\hat{y}_t)-q)c_t+\hat{\mathcal{L}}_t(\hat{a}_t,p_t(q,\hat{y}_t))+\mathsf{E}\Big[\varphi_{t+\hat{a}_t}(p_t(q,\hat{y}_t)-\sum_{j=t}^{t+\hat{a}_t-1}D_j)\Big],$$

where

$$(\bar{a}_k^*,\bar{S}_k^*) = \arg\min_{\substack{\bar{y}_k:\bar{y}_k\geq 0\\ \bar{a}_k:t-k\geq\bar{a}_k\geq 1}} \Big\{A_k+\bar{y}_k c_k+\hat{\mathcal{L}}_k(\bar{a}_k,\bar{y}_k)+\mathsf{E}\Big[\phi_{k+\bar{a}_k}(\bar{y}_k-\sum_{j=k}^{k+\bar{a}_k-1}D_j)\Big]\Big\}$$

and $\varphi_l(.)$ is defined as in (9) with slight difference, that is the optimal decisions fixed for future periods, $l=t+1,...N$ are given by

$$(\hat{a}_l^*,\hat{S}_l^*) = \arg\min_{\substack{\hat{y}_l:\hat{y}_l\geq 0\\ \hat{a}_l:N-l+1\geq\hat{a}_l\geq 1}} \{Q_1^l(\hat{a}_l,\hat{y}_l)\}$$

Similarly as in AH, a solution to Problem P is obtained by solving (10) in a backward fashion for $t=N,\ldots,1$. AH I and AH follow a very similar procedure. The main difference is that in AH I while determining $(\hat{a}_t^*,\hat{S}_t^*)$, we don't assume zero starting inventory as in AH but extend the DP procedure by solving (11) for $k=t,...,1$ in backward fashion to get a solution as production decisions taken in the prior periods. This solution is incorporated as a better estimate of the starting inventory distribution in the cost calculation of decision $(\hat{a}_t,\hat{y}_t)$ for period t while solving (10).

5.3. Approximation Heuristic Extension II (AH II)

Although non-exponential, AH I has a high order polynomial complexity as it develops a solution of production decisions for all prior periods. Another alternative is to consider a fixed number of prior periods to get an estimate of starting inventory distribution while determining the decision parameters for period t. This procedure extends AH as follows: Consider period t. Let n be the fixed number of prior periods to be considered. Define the following decision variables:

- $\bar{n} = \begin{cases} n \text{ if } n \leq t \\ t \text{ otherwise} \end{cases}$
- $\bar{m}$: number of setups in the prior periods
- $\bar{\mathbf{r}} = \{\bar{r}_1, \ldots, \bar{r}_{\bar{m}}\}$: the schedule of setups in prior periods where $\bar{r}_1 = t - \bar{n}$
- $\bar{y}_j$: base stock level of production cycle j in prior periods
- $\bar{\delta}_k$: indicator function of production periods in prior periods $k = t - \bar{n}, \ldots, t-1$ such that $\bar{\delta}_{\bar{r}_j} = 1, \forall j = 1, \ldots, \bar{m}$ and $\bar{\delta}_k = 0$ otw.

Moreover, consider the following inventory balance equations:

$$\bar{I}_{j+1} = \bar{I}_j + x_j - \sum_{k=\bar{r}_j}^{\bar{r}_{j+1}-1} D_k, \qquad j = 1, \ldots, \bar{m}, \tag{12}$$

where $\bar{I}_1 = 0$ and $\bar{I}_{\bar{m}+1}$ are the net inventory at the beginning and the end of the planning horizon, respectively, and $\bar{r}_{\bar{m}+1} = t$.

We solve the following problem:

$$\min_{\substack{\hat{y}_t : \hat{y}_t \geq 0 \\ \hat{a}_t : N-t+1 \geq \hat{a}_t \geq 1}} \{Q^t(\hat{a}_t, \hat{y}_t)\} \tag{13}$$

such that

$$\begin{aligned} Q^t(\hat{a}_t, \hat{y}_t) = \min_{\substack{(\bar{m}, \bar{\mathbf{r}}) \\ \bar{m} : \bar{m} \leq \bar{n}}} \Bigg\{ \sum_{k=t-\bar{n}}^{t-1} A_k \bar{\delta}_k + \min_{\substack{y_1 : \bar{I}_1 + o_{\bar{r}_1} \geq \bar{y}_1 \\ \bar{y}_1 \geq \bar{I}_1 + u_{\bar{r}_1}}} \Bigg\{ c_{\bar{r}_1}(\bar{y}_1 - \bar{I}_1) + \mathcal{L}_1(\bar{y}_1) + \mathsf{E}_{D_{\bar{r}_1}, \ldots, D_{\bar{r}_2 - 1}} \Bigg[\\ \min_{\substack{\bar{y}_2 : \bar{I}_2 + o_{\bar{r}_2} \geq \bar{y}_2 \\ \bar{y}_2 \geq \bar{I}_2 + u_{\bar{r}_2}}} \Bigg\{ c_{\bar{r}_2}(\bar{y}_2 - \bar{I}_2) + \mathcal{L}_2(\bar{y}_2) + \mathsf{E}_{D_{\bar{r}_2}, \ldots, D_{\bar{r}_3 - 1}} \Bigg[\cdots + \Bigg[\min_{\substack{\bar{y}_{\bar{m}} : \bar{I}_{\bar{m}} + o_{\bar{r}_{\bar{m}}} \geq \bar{y}_{\bar{m}} \\ \bar{y}_{\bar{m}} \geq \bar{I}_{\bar{m}} + u_{\bar{r}_{\bar{m}}}}} \Bigg\{ \\ c_{\bar{r}_{\bar{m}}}(\bar{y}_{\bar{m}} - \bar{I}_{\bar{m}}) + \mathcal{L}_{\bar{m}}(\bar{y}_{\bar{m}}) + \mathsf{E}_{D_{\bar{r}_{\bar{m}}}, \ldots, D_{\bar{r}_{\bar{m}+1} - 1}} \Big[\varphi_t(\bar{I}_{\bar{m}+1}) \Big] \Bigg\} \Bigg] \ldots \Bigg] \Bigg\} \Bigg] \Bigg\} \Bigg\} \Bigg\}, \end{aligned} \tag{14}$$

where $\mathcal{L}_i$ is defined as in (1) for production schedule $(\bar{m}, \bar{\mathbf{r}})$ and $\varphi_l(.)$ is defined as in (9) with slight difference, that is the optimal decisions fixed for future periods, $l = t+1, \ldots N$ are given by

$$(\hat{a}_l^*, \hat{S}_l^*) = \arg\min_{\hat{a}_l, \hat{y}_l} \{Q^l(\hat{a}_l, \hat{y}_l)\}.$$

Similarly as in AH and AH I, a solution to Problem P is obtained by solving (13) in a backward fashion for $t = N, \ldots, 1$. AH II extends AH as follows. In AH II, we assume that the production horizon is started n periods before t with zero inventory and we solve for the optimal production schedule of the prior periods by enumeration while calculating the cost of decision $(\hat{a}_t, \hat{y}_t)$ in (13). This way, we incorporate a better estimate of the starting inventory distribution while determining $(\hat{a}_t^*, \hat{S}_t^*)$. Since we consider a fixed number of prior periods, it increases the computational burden by constant factor 2^n where n can be chosen accordingly.

6. Search Based Heuristics

This section introduces our new heuristic procedures for solving Problem P. The basic idea of these heuristics is to perform a search on production schedules in greedy fashion by considering three operations: merging, dividing and switching.

1. Merging: In a given production schedule, merging operation combines two consecutive production cycles and forms a production schedule with one setup less by simply removing one setup from the schedule.

Procedure 2 Merging

Input: a production schedule $(m, \mathbf{r})$
Input: $l \in \mathbb{Z}$, $l < m$
Output: a production schedule with one cycle less
Output: Optimal cost C
Let $(l+1)^{th}$ 1 in vector δ be 0.
$m \leftarrow m - 1$
Solve $(m, \mathbf{r})$ for optimality. Let C the optimal cost of this schedule.
return $(m, \mathbf{r})$ and C

2. Dividing: As an opposite move to merging, dividing operation separates a production cycle into two by adding a new setup to the current schedule.

Procedure 3 Dividing

Input: a production schedule $(m, \mathbf{r})$
Input: $l \in \mathbb{Z}$, $l < N$
Output: a production schedule
Output: Optimal cost C
if $\delta_l = 0$ **then**
    $\delta_l \leftarrow 1$
    $m \leftarrow m + 1$
end if
Solve $(m, \mathbf{r})$ for optimality. Let C the optimal cost of this schedule.
return $(m, \mathbf{r})$ and C

3. Switching: Even though the number of setups are fixed, the location of the setups can affect the total cost incurred. Switching operation moves location of a setup one period backward or forward and obtains a new production schedule.

Procedure 4 Switching

Input: a production schedule $(m, \mathbf{r})$
Input: $l \in \mathbb{Z}$, $l \leq m$
Input: $s \in \{-1, 1\}$
Output: a production schedule
Output: Optimal cost C
if $r_l + s = 0$ or $r_l + s = N + 1$ **then**
    Solve $(m, \mathbf{r})$ for optimality. Let C the optimal cost of this schedule.
    return $(m, \mathbf{r})$ and C
end if
if $\delta_{r_l+s} = 1$ **then**
    $m \leftarrow m - 1$
end if
$\delta_{r_l+s} \leftarrow 1$
$\delta_{r_l} \leftarrow 0$
Solve $(m, \mathbf{r})$ for optimality. Let C the optimal cost of this schedule.
return $(m, \mathbf{r})$ and C

The following heuristic procedures make use of these operations in different combinations and orders while searching for a new production schedule with lower cost. First two algorithms use merging and switching operations while the other two algorithms combine dividing and switching operations.

6.1. Merging Method I (MM I)

For a planning horizon of N periods; suppose, we start with a production schedule $(m, \mathbf{r})$ where $m = N$. Which means there are scheduled setups at the beginning every period where every period corresponds to a cycle. At this point, we can start our local search by merging these small cycles one by one, and calculating optimal base stock levels and costs of new production schedules. Among all possible schedules obtained by merging operation, we pick the one providing the most cost improvement as our next iterate and obtain a production schedule with $N-1$ setups. Then we perform merging operations on the new production schedule obtained. We continue merging cycles until no improvement is present. Then, we apply switching operation by 1 period forward and backwards on the current periods of setups. We again pick the production schedule with the best improvement and obtain a production schedule with same number of setups where one of the setups is in a different period. And we continue with switching operation until no improve is observed. Afterwards, we apply merging on the our new current production schedule again until cost no longer improves. And then again, we apply the switching operation. We apply these two procedures repeatedly one after the other until the total expected cost cannot be improved by neither of these two. The following algorithm formulates the MM I method:

Algorithm 5 Merging Method I

Input: production schedule $(m, \mathbf{r})$
Output: Best cost schedule
Find the optimal base stock levels for schedule $(m, \mathbf{r})$. Let C the optimal cost of this schedule. Let $BestSoFar \leftarrow C$. Let $min1 \leftarrow \infty, min2 \leftarrow BestSoFar$
for $min1 > min2$ **do**
  $\hat{\mathbf{r}} \leftarrow \mathbf{r}$, $\bar{\mathbf{r}} \leftarrow \mathbf{r}$, $l \leftarrow 1$, $n \leftarrow m$, $imp \leftarrow 0$
  for $l < n$ **do**
    Merge($(m, \hat{\mathbf{r}})$,l)
    if $C < BestSoFar$ **then**
      Let $BestSoFar \leftarrow C, \bar{\mathbf{r}} \leftarrow \hat{\mathbf{r}}, imp \leftarrow 1$.
    end if
    $\hat{\mathbf{r}} \leftarrow \mathbf{r}, l \leftarrow l + 1$.
  end for
  $\mathbf{r} \leftarrow \bar{\mathbf{r}}$, $\hat{\mathbf{r}} \leftarrow \mathbf{r}$
  if $imp = 1$ **then**
    Go to 3
  end if
  $l \leftarrow 1$, $n \leftarrow m$, $imp \leftarrow 0$, $s \leftarrow -1$
  for $s \leq 1$ **do**
    for $l \leq n$ **do**
      Switch($(m, \hat{\mathbf{r}})$,l,s)
      if $C < BestSoFar$ **then**
        Let $BestSoFar \leftarrow C, \bar{\mathbf{r}} \leftarrow \hat{\mathbf{r}}, imp \leftarrow 1$.
      end if
      $\hat{\mathbf{r}} \leftarrow \mathbf{r}, l \leftarrow l + 1$.
    end for
    $s \leftarrow s + 2$, $l \leftarrow 1$
  end for
  $\mathbf{r} \leftarrow \bar{\mathbf{r}}$, $\hat{\mathbf{r}} \leftarrow \mathbf{r}$
  if $imp = 1$ **then**
    Go to 15
  end if
  $min1 \leftarrow min2$, $min2 \leftarrow BestSoFar$
end for
return $(m, \mathbf{r})$ and C

6.2. Merging Method II (MM II)

MM I can be seen as a depth first method. In MM II, we make a small modification in the following way. Again, we start with replenishment schedule where there is a setup in all periods. We start merging two cycles which results in the best cost improvement compared to merging the other cycles. After applying merging operation once, we immediately apply switching operation to

the new production schedule and proceed with the production schedule with the best cost. Then again, we apply merging to the current production schedule followed by switching. We continue on applying these operations one immediately after the other until no improvement is observed. Following algorithm summarizes MM II:

Algorithm 6 Merging Method II

Input: production schedule $(m, \mathbf{r})$
Output: Best cost schedule
Find the optimal base stock levels for schedule $(m, \mathbf{r})$. Let C the optimal cost of this schedule. Let $BestSoFar \leftarrow C$. Let $min1 \leftarrow \infty, min2 \leftarrow BestSoFar$
for $min1 > min2$ **do**
  $\hat{\mathbf{r}} \leftarrow \mathbf{r}$, $\bar{\mathbf{r}} \leftarrow \mathbf{r}$, $l \leftarrow 1$, $n \leftarrow m$
  for $l < n$ **do**
    Merge($(m, \hat{\mathbf{r}})$,l)
    if $C < BestSoFar$ **then**
      Let $BestSoFar \leftarrow C, \bar{\mathbf{r}} \leftarrow \hat{\mathbf{r}}$.
    end if
    $\hat{\mathbf{r}} \leftarrow \mathbf{r}, l \leftarrow l + 1$.
  end for
  $\mathbf{r} \leftarrow \bar{\mathbf{r}}$, $\hat{\mathbf{r}} \leftarrow \mathbf{r}$
  $l \leftarrow 1$, $n \leftarrow m$, $s \leftarrow -1$
  for $s \leq 1$ **do**
    for $l \leq n$ **do**
      Switch($(m, \hat{\mathbf{r}})$,l,s)
      if $C < BestSoFar$ **then**
        Let $BestSoFar \leftarrow C, \bar{\mathbf{r}} \leftarrow \hat{\mathbf{r}}$.
      end if
      $\hat{\mathbf{r}} \leftarrow \mathbf{r}, l \leftarrow l + 1$.
    end for
    $s \leftarrow s + 2$, $l \leftarrow 1$
  end for
  $\mathbf{r} \leftarrow \bar{\mathbf{r}}$, $\hat{\mathbf{r}} \leftarrow \mathbf{r}$
  $min1 \leftarrow min2$, $min2 \leftarrow BestSoFar$
end for
return $(m, \mathbf{r})$ and C

6.3. Dividing Method I (DM I)

Following the idea in MM I, DM I considers a similar method proceeding backwards. We start with a production schedule $(m, \mathbf{r})$ where $m = 1$. Which implies that in this production schedule, there is only one setup and one cycle containing all of the periods. Starting from this one-cycle production schedule, we apply dividing operation. As we simply insert a setup in the one cycle schedule, we proceed with the production schedule that provides the best cost improvement where there are two production cycles. Then on this new production schedule, we apply the dividing operation once more. We keep dividing until placing another setup does not improve the cost. Then, we apply switching on the resulting production schedule as we did in MM I. Then we continue to apply dividing and switching operations switching one from the other in the same fashion as in MM I. We stop when neither of these procedures can improve the cost. The following algorithm summarizes DM I:

Algorithm 7 Dividing Method I

Input: production schedule $(m, \mathbf{r})$
Output: Best cost schedule
1: Find the optimal base stock levels for schedule $(m, \mathbf{r})$. Let C the optimal cost of this schedule. Let $BestSoFar \leftarrow C$. Let $min1 \leftarrow \infty, min2 \leftarrow BestSoFar$
2: **for** $min1 > min2$ **do**
3: $\hat{\mathbf{r}} \leftarrow \mathbf{r}$, $\bar{\mathbf{r}} \leftarrow \mathbf{r}$, $l \leftarrow 1$, $n \leftarrow m$, $imp \leftarrow 0$
4: **for** $l \leq N$ **do**
5: Divide($(m, \hat{\mathbf{r}})$,l)
6: **if** $C < BestSoFar$ **then**
7: Let $BestSoFar \leftarrow C, \bar{\mathbf{r}} \leftarrow \hat{\mathbf{r}}, imp \leftarrow 1$.
8: **end if**
9: $\hat{\mathbf{r}} \leftarrow \mathbf{r}, l \leftarrow l + 1$.
10: **end for**
11: $\mathbf{r} \leftarrow \bar{\mathbf{r}}$, $\hat{\mathbf{r}} \leftarrow \mathbf{r}$
12: **if** $imp = 1$ **then**
13: Go to 3
14: **end if**
15: $l \leftarrow 1$, $n \leftarrow m$, $imp \leftarrow 0$, $s \leftarrow -1$
16: **for** $s \leq 1$ **do**
17: **for** $l \leq n$ **do**
18: Switch($(m, \hat{\mathbf{r}})$,l,s)
19: **if** $C < BestSoFar$ **then**
20: Let $BestSoFar \leftarrow C, \bar{\mathbf{r}} \leftarrow \hat{\mathbf{r}}, imp \leftarrow 1$.
21: **end if**
22: $\hat{\mathbf{r}} \leftarrow \mathbf{r}, l \leftarrow l + 1$.
23: **end for**
24: $s \leftarrow s + 2$, $l \leftarrow 1$
25: **end for**
26: $\mathbf{r} \leftarrow \bar{\mathbf{r}}$, $\hat{\mathbf{r}} \leftarrow \mathbf{r}$
27: **if** $imp = 1$ **then**
28: Go to 15
29: **end if**
30: $min1 \leftarrow min2$, $min2 \leftarrow BestSoFar$
31: **end for**
32: **return** $(m, \mathbf{r})$ and C

6.4. Dividing Method II (DM II)

We remark that DM I is obtained by replacing merging operation in MM I with diving operation. We replace the merging operation in MM II with dividing operation to obtain DM II. The following algorithm summarizes DM II:

Algorithm 8 Dividing Method II (DM II)

Input: Replenishment schedule $(m, \mathbf{r})$
Output: Best cost schedule
Find the optimal base stock levels for schedule $(m, \mathbf{r})$. Let C the optimal cost of this schedule. Let $BestSoFar \leftarrow C$. Let $min1 \leftarrow \infty, min2 \leftarrow BestSoFar$
for $min1 > min2$ **do**
  $\hat{\mathbf{r}} \leftarrow \mathbf{r}$, $\bar{\mathbf{r}} \leftarrow \mathbf{r}$, $l \leftarrow 1$, $n \leftarrow m$
  for $l < n$ **do**
    Divide($(m, \hat{\mathbf{r}})$,l)
    if $C < BestSoFar$ **then**
      Let $BestSoFar \leftarrow C, \bar{\mathbf{r}} \leftarrow \hat{\mathbf{r}}$.
    end if
    $\hat{\mathbf{r}} \leftarrow \mathbf{r}, l \leftarrow l + 1$.
  end for
  $\mathbf{r} \leftarrow \bar{\mathbf{r}}$, $\hat{\mathbf{r}} \leftarrow \mathbf{r}$
  $l \leftarrow 1$, $n \leftarrow m$, $s \leftarrow -1$
  for $s \leq 1$ **do**
    for $l \leq n$ **do**
      Switch($(m, \hat{\mathbf{r}})$,l,s)
      if $C < BestSoFar$ **then**
        Let $BestSoFar \leftarrow C, \bar{\mathbf{r}} \leftarrow \hat{\mathbf{r}}$.
      end if
      $\hat{\mathbf{r}} \leftarrow \mathbf{r}, l \leftarrow l + 1$.
    end for
    $s \leftarrow s + 2$, $l \leftarrow 1$
  end for
  $\mathbf{r} \leftarrow \bar{\mathbf{r}}$, $\hat{\mathbf{r}} \leftarrow \mathbf{r}$
  $min1 \leftarrow min2$, $min2 \leftarrow BestSoFar$
end for
return $(m, \mathbf{r})$ and C

7. Numerical Study

In this section, we evaluate the performance of the heuristics developed in Sections 5 and 6. We first show the effectiveness of the lower bounds used in the procedure described in 4.1. Next, we test the performance of the heuristics in terms of the quality of the solution (optimality gap) and the computational time.

We start our experiments with stationary cost parameters and lot size restrictions, i.e., values do not change from period to period, and create the test bed as follows. For each model parameter, we select two or three levels namely low, medium, and high. We use set up costs of $A \in \{2, 20, 50, 200\}$, unit production costs of $c \in \{1, 5\}$, and unit penalty costs of $b \in \{2, 8, 32\}$. Unit holding cost is a linear function of the unit production cost such that $h = 0.1 \times c$. We consider all cost parameter combinations, but exclude the ones with $c = 5$ and $b = 2$ since $b < c$ is not realistic. We select lot size parameters from $u \in \{0, 5, 10\}$ and $o \in \{10, 20, 40\}$ and exclude any cases with $(u, o) = (5, 10)$, $(10, 10)$ and $(10, 20)$. The planning horizon is fixed at 12 periods, $N = 12$. We assume that demand in a period follows a Poisson distribution, and we consider six different demand patterns: static (P1), increasing (P2), decreasing (P3), hectic (P4), product life cycle (P5), and seasonal (P6). Mean demands for periods where $N = 12$ are summarized in the Table 1. The above specifications result in 720 problem instances which will be referred to as cases or scenarios. We solve each case using the heuristics of Sections 5 and 6, and the procedure for finding an optimal solution in Section 4. Following subsections discuss the results.

7.1. Lower Bound Effectiveness

Our procedure for finding an optimal solution for Problem P utilizes three lower bounds, LB_1, LB_2 and LB_3, in the order of increasing tightness and computational complexity. In this section, we will evaluate the effectiveness of this procedure and the lower bounds used.

The implementation of LB_3 requires the number of line segments for approximation to be set (k). After experimenting with different k values, we decided to use $k = 3$ for all computations in

Table 1 Parameters for λ_t

Period	λ_t (Poisson Distribution)					
	P1	P2	P3	P4	P5	P6
1	5	1.62	8.38	2	7.5	3.52
2	5	2.23	7.77	1	9.33	7.04
3	5	2.85	7.15	23.5	10	7.04
4	5	3.46	6.54	1	9.33	7.04
5	5	4.08	5.92	2	7.5	7.04
6	5	4.69	5.31	1	5	7.04
7	5	5.31	4.69	2	2.5	6.04
8	5	5.92	4.08	21	0.67	5.04
9	5	6.54	3.46	2	0	4.04
10	5	7.15	2.85	1	0.67	3.04
11	5	7.77	2.23	2	2.5	2.04
12	5	8.38	1.62	1.5	5	1.08
$\sum$	60	60	60	60	60	60

this paper. (See Online Appendix A.3 for the details of the analysis.)

To test the effectiveness of the lower bounds, we first solved the 720 cases in the test bed for optimality only with LB_1, LB_2 or LB_3, and calculated the average number of production schedules eliminated by each lower bound out of 2048 schedules for each case. Then we solved the problems again using $LB_1\&LB_2$ and $LB_1\&LB_2\&LB_3$. Table 2 summarizes the results. Since the motivation behind LB_3 is our observation that LB_1 and LB_2 become inaccurate as the system gets highly capacitated, we organized the table to provide the average number of cases eliminated for each maximum lot size value. In the last row, we present the CPU time to obtain all the solutions for the test bed using a machine with Intel Core i5 2.67 GHz processor and 4GB of memory running on 64-bit Windows 7 operating system.

Table 2 Lower Bound Effectiveness Summary

	# of Cases	LB_1	LB_2	LB_3	$LB_1\&LB_2$	$LB_1\&LB_2\&LB_3$
o=10	120	250.03	692.43	1817.43	692.43	1839.58
o=20	240	746.88	1907.08	1771.56	1907.08	1957.46
o=40	360	946.31	1978.27	1859.04	1978.27	2020.69
Overall	720	763.79	1740.23	1822.94	1740.23	1969.42
Time (min)	720	137.71	86.22	100.60	81.55	62.18

Note that LB_1 is outperformed by all other lower bound combinations. If one wants to implement a single lower bound, it is not clear to pick one between LB_2 and LB_3. For $o = 20$ and $o = 40$, LB_2 is somewhat better if the average number of eliminated schedules is considered. However, the performance of LB_2 deteriorates significantly for $o = 10$. Overall LB_3 prunes more schedules, but this comes at a cost of increased computation time.

The most significant performance improvement is observed when the bounds are utilized together. The best results for both the number of schedules eliminated and the computation time are obtained by $LB_1\&LB_2\&LB_3$, which is the procedure explained in Section 4.1. By using all three lower bounds, we eliminate on the average more than 95% of the total schedules to be searched for a problem instance, and we find an optimal solution in about 5.2 seconds (average).

7.2. Performance Comparison

We use percentage gap between the expected cost of a heuristic and the optimal cost, *optimality gap*, as the performance measure of our comparisons. The optimality gap is defined as

$$optimality\ gap\ (\%) = \frac{(cost\ of\ schedule\ (\bar{m}, \bar{\mathbf{r}})) - (cost\ of\ schedule\ (m^*, \mathbf{r}^*))}{(cost\ of schedule\ (m^*, \mathbf{r}^*))}$$

where $(\bar{m}, \bar{\mathbf{r}})$ is the production schedule of the heuristic method and $(m^*, \mathbf{r}^*)$ is the optimal production schedule for Problem P.

7.2.1. Test Bed with Stationary Parameters After solving all cases of the test bed for optimality using the procedure developed in Section 4.1, we solved each scenario with the heuristic procedures. Table 3 summarizes the results. First row shows the number of cases (out of 720) solved optimally by the heuristic. We also present the number of cases for which the optimality gap is less than 1, 2 and 5 percent. As we observe from the table:

1. The performance of DP-based heuristics show a wide range. While AH achieves optimality in 440 cases, AH I solves 565 scenarios to optimality. Similar observations also hold for other measures in the table. Among DP-based heuristics, AH I and AH II (n=4) outperform the others in terms of the number of cases solved optimally and the number cases with optimality gap less than 1, 2 and 5 percent. Looking at the average and maximum optimality gap figures, AH II (n=4) can be considered the best performing heuristic in its class.

2. Search-based heuristics perform well in this test bed and unlike for the DP-based heuristics, the performance is consistent across the board. At least 635 cases (88%) are solved optimally by this class of heuristics. Note that each heuristic achieves an optimality gap below 5% for the entire test bed. Moreover, any heuristic in this class achieves less than 1% optimality gap for more than 99% of the cases considered.

3. Search-based heuristics have a commanding lead over the DP-based procedures. Specifically, DM II stands out among the four search-based heuristics. DM II solves 718 cases within 1% of the optimal cost. Even for the scenarios with $> 1\%$ gap, the maximum optimality gap is 1.42%.

Table 3 Summary of Results for the Test Bed with Stationary Parameters ($N = 12$)

			AH II				Merge		Divide	
	AH	AH I	n=1	n=2	n=3	n=4	MM I	MM II	DM I	DM II
optimal	440	565	508	524	537	562	649	637	635	640
$< 1\%$	541	662	627	650	647	662	716	713	714	718
$< 2\%$	611	690	675	682	677	692	718	716	719	720
$< 5\%$	655	710	702	701	706	712	720	720	720	720
ave (%)	1.36	0.35	0.50	0.51	0.41	0.27	0.05	0.06	0.06	0.05
max (%)	31.80	25.66	14.17	24.91	14.42	10.20	3.27	3.28	2.19	1.42

Next, we take a deep dive into the optimality gap distributions of the heuristics in order to get more insights about the parameter regions leading to good/bad performance. We analyzed the data by comparing performance of these two classes of heuristics one parameter at a time. Interestingly, we were able to find a region where DP-based heuristics perform better than the search-based ones. Table 4 shows the number of cases solved for optimality, minimum and maximum optimality gap, and 50th, 75th, 90th, 95th and 99th percentiles. of the optimality gap distribution. The table is organized such that each row corresponds to a specific maximum lot size (o) level. (We only give the results for AH II for n=4 since it this value of n performs the best.)

First of all, we observe the dramatic performance improvement for all DP-based heuristics as the maximum lot size increases. For example, AH II can only achieve optimality in 32% of the cases (38 out of 120) for $o = 10$, the percentage jumps to 98% (351 out of 360) for $o = 40$. The optimality gap distribution also supports this observation. For $o = 40$, 95th percentile is still 0 and the maximum gap is 0.54%. For $o = 10$, even the 50th percentile is positive and maximum optimality gap is above 7.64%. Similar observations also hold for AH and AH I. We see that the trend is reversed for search-based heuristics such that the performance considerably as the maximum lot size increases. DM I and DM II achieves optimality in 98% of the cases for o=10, but this number drops to 84% for o=40. The shift is also observed in the optimality gap distributions. For example, for DM I, 90th

percentiles are 0.00%, 0.03% and 0.38% for o=10, 20 and 40, respectively. We observe a similar behaviour at the 95th percentile such that the corresponding values are 0.00% 0.25% 0.72% for o=10, 20 and 40, respectively.

Table 4 Optimality Gap Distribution for Different Maximum Lot Size Levels (o)

	Maximum lot size (o)	Total # of Cases	optimal	min	50th	75th	90th	95th	99th	max
AH	10	120	28	0.00%	1.59%	4.12%	10.41%	14.27%	19.09%	31.80%
	20	240	84	0.00%	0.41%	2.28%	7.04%	9.65%	19.55%	22.05%
	40	360	328	0.00%	0.00%	0.00%	0.00%	0.59%	1.48%	1.61%
AH I	10	120	39	0.00%	0.29%	1.82%	4.32%	5.63%	20.59%	25.66%
	20	240	181	0.00%	0.00%	0.00%	0.31%	1.11%	4.04%	10.74%
	40	360	345	0.00%	0.00%	0.00%	0.00%	0.00%	0.48%	1.27%
AH II (n=4)	10	120	38	0.00%	0.48%	1.61%	3.77%	5.01%	7.05%	7.64%
	20	240	173	0.00%	0.00%	0.06%	0.48%	0.75%	1.55%	10.20%
	40	360	351	0.00%	0.00%	0.00%	0.00%	0.00%	0.35%	0.54%
MM I	10	120	113	0.00%	0.00%	0.00%	0.00%	0.03%	0.43%	0.65%
	20	240	214	0.00%	0.00%	0.00%	0.02%	0.39%	0.95%	1.77%
	40	360	322	0.00%	0.00%	0.00%	0.03%	0.41%	0.89%	3.27%
MM II	10	120	111	0.00%	0.00%	0.00%	0.00%	0.08%	0.60%	0.97%
	20	240	210	0.00%	0.00%	0.00%	0.13%	0.48%	0.95%	1.15%
	40	360	316	0.00%	0.00%	0.00%	0.18%	0.57%	1.60%	3.28%
DM I	10	120	118	0.00%	0.00%	0.00%	0.00%	0.00%	0.06%	0.23%
	20	240	213	0.00%	0.00%	0.00%	0.03%	0.25%	0.95%	1.39%
	40	360	304	0.00%	0.00%	0.00%	0.38%	0.72%	0.99%	2.19%
DM II	10	120	117	0.00%	0.00%	0.00%	0.00%	0.00%	0.20%	0.44%
	20	240	222	0.00%	0.00%	0.00%	0.00%	0.12%	0.71%	1.42%
	40	360	301	0.00%	0.00%	0.00%	0.30%	0.63%	0.87%	0.97%

Based on these findings, we conclude that the performance of the DP-based heuristics improve as the capacity is less constrained (lower maximum lot size implies tighter capacity). On the contrary, the search-based heuristics perform better in tight capacity regimes. These conclusions gave us an idea to exploit the complementary behavior we see in these two classes of heuristics. Pairing a DP-based heuristic with a search-based one, solving the same problem with both heuristics, and selecting the solution with the minimum cost as the final solution may lead to a numerical procedure with higher accuracy. To test this idea, we considered all 12 pairwise matchings between three DP-based heuristics of AH, AH I and AH II with n=4, and four search-based heuristics of MM I, MM II, DM I and DM II. The results are given in Table 5, which is organized in a similar way to Table 3. The improvement in performance is remarkable: all combinations achieve less than 1% optimality gap for the entire test bed. Even the worst performing DP-based heuristic, AH, achieves optimality in at least 687 cases when coupled with a search-based heuristic; compare this against 440 cases in Table 3. Finally, note that MM II & AH and DM II & AH II pairs obtain respective maximum gaps of 0.40% and 0.46%, which happen to be the best performers.

Table 5 Summary of Results for Pairwise Heuristic Combinations (AH II refers to AH II with n=4)

	MM I			MM II			DM I			DM II		
	AH	AH I	AH II	AH	AH I	AH II	AH	AH I	AH II	AH	AH I	AH II
optimal	687	702	705	710	698	703	697	710	710	698	705	709
$< 1\%$	720	720	720	720	720	720	720	720	720	720	720	720
ave (%)	0.01	0.01	0.00	0.00	0.01	0.00	0.01	0.00	0.00	0.01	0.01	0.00
max (%)	0.82	0.76	0.76	0.40	0.76	0.76	0.79	0.93	0.76	0.72	0.93	0.46

7.2.2. Dynamic Capacity The results of the test bed with stationary parameters suggest that capacity (maximum lot size) is an important factor with a clear impact on heuristic performance. The capacity in a production environment is expected to vary over time, thus, we need to expose our heuristic procedures to dynamic capacity regimes and stress test their performance. The test bed for dynamic capacity is created as follows. We first generate capacity patterns of $\mathbf{e}$ vectors with entries $\{-1, 0, 1\}$ such that $\sum_t e_t = 0$. The patterns considered for $N = 12$ are given in Table 6.

Table 6 Patterns for Dynamic Capacity Scenarios

Period	C1	C2	C3	C4	C5	C6	C7	C8	C9	C10	C11	C12
1	-1	-1	1	-1	-1	-1	-1	1	0	1	-1	0
2	1	-1	1	0	1	0	0	-1	1	-1	-1	0
3	0	-1	1	1	-1	1	-1	0	1	-1	1	-1
4	-1	-1	1	0	1	-1	1	-1	-1	-1	-1	0
5	-1	0	0	-1	-1	0	-1	1	-1	1	1	0
6	0	0	0	0	1	1	-1	-1	0	1	-1	1
7	1	0	0	1	-1	-1	1	0	1	0	1	1
8	-1	0	0	0	1	0	1	1	0	1	1	0
9	1	1	-1	-1	-1	1	1	1	-1	1	1	0
10	0	1	-1	0	1	-1	0	0	-1	0	-1	1
11	0	1	-1	1	-1	0	-1	-1	1	-1	-1	-1
12	1	1	-1	0	1	1	1	0	0	-1	1	-1

We calculate the upper bound on lot sizes (maximum lot size) using the following rule

$$o_t = \alpha \times (10 + \beta \times e_t), \qquad t = 1, 2, \ldots, 12,$$

where $\beta \in \{1, 3\}$ represents the amplitude of deviation from the average capacity in a given period, and $\alpha \in \{0.75, 1, 3\}$ is a multiplier. Three values of α correspond to low, moderate, and high capacity levels. This way we are able to generate numerous capacity scenarios with different degrees of variability. For this test bed, we excluded the lower bound on lot sizes by simply setting $u_t = 0$ for all t. The rest of the parameters take the values as in the test bed of subsection 7.2.1. This yields a total of 8640 cases. As in the analysis for the test bed with stationary parameters, we solve each problem instance for optimality and use the optimal cost as a bench mark for the performance of the heuristics. The results are summarized in Table 7.

Table 7 Summary of Results for the Test Bed with Dynamic Capacity (N=12)

			AH II				Merge		Divide	
	AH	AH I	n=1	n=2	n=3	n=4	MM I	MM II	DM I	DM II
optimal	3624	5409	4639	4983	5242	5514	7569	7369	7869	7912
$< 1\%$	4794	6664	6054	6245	6475	6706	8197	8056	8530	8544
$< 2\%$	5443	7225	6683	6816	7054	7262	8342	8211	8609	8609
$< 5\%$	6498	8075	7598	7763	8018	8157	8517	8441	8638	8638
ave (%)	3.70	1.18	1.94	1.52	1.12	0.98	0.24	0.37	0.05	0.05
max (%)	50.78	92.74	44.26	34.47	26.00	55.83	23.43	35.77	6.66	6.66

Once again, we observe the poor performance of the DP-based heuristics. Except the maximum optimality gap, AH II with n=4 outperforms other heuristics in its class achieving optimality for 5514 cases, which represents only 64% of the entire test bed. While the average gap is 0.98%, the maximum gap observed is a staggering 55.83% indicating a poor worst case performance, which is a behavior exhibited by all the heuristics in this class. This clearly indicates that applying DP-based heuristics for dynamic capacity scenarios may fail to provide good solutions.

In comparison, search-based heuristics perform relatively better. Unlike in the test bed with stationary parameters, there is a clear performance difference between Divide and Merge heuristics.

DM I and DM II solves more problems to optimality (7869 and 7912 respectively) and achieves smaller optimality gaps. They yield optimality gaps of less than 1% in more than 98% of the cases in the test bed. We observe that in only 2 cases the optimality gap is beyond 5% with a maximum of 6.66%. Next, we will apply the same approach as before to test whether we can leverage the complementary behavior between the two classes of heuristics in order to improve the worst case behavior.

Table 8 shows the results by pairing a DP-based heuristic with a search-based taking the better solution for each problem instance. We see an improvement across the board. Specifically, we observe that when Divide heuristics (DM I and DM II) are coupled with AH I or AH II, they outperform other pairs. Average optimality gap drops down to 0.01% and the maximum falls below 5% mark. More than 97% of the total number of cases are solved to optimality.

Table 8 Summary of Results for Pairwise Heuristic Combinations (AH II refers to AH II with n=4)

	MM I			MM II			DM I			DM II		
	AH	AH I	AH II	AH	AH I	AH II	AH	AH I	AH II	AH	AH I	AH II
optimal	7960	8267	8256	8336	8191	8173	8267	8420	8432	8295	8432	8441
$< 1\%$	8366	8519	8508	8543	8480	8469	8577	8613	8612	8578	8610	8612
$< 2\%$	8462	8566	8557	8581	8546	8537	8619	8634	8633	8619	8634	8633
$< 5\%$	8578	8620	8618	8627	8620	8618	8638	8640	8640	8638	8640	8640
ave (%)	0.14	0.05	0.06	0.04	0.06	0.07	0.03	0.01	0.01	0.02	0.01	0.01
max (%)	23.43	8.62	9.13	8.24	8.62	9.13	6.66	3.23	3.23	6.66	3.23	3.23

7.3. Computational Times

In this subsection, we compare the computation times of our heuristics to see how they scale as the planning horizon extends. We plotted the log of the computation times in minutes against the number of periods in the planning horizon (N) for the test bed with stationary parameters (total of 720 cases) in Figure 1. The plot on the left shows the times for DP based heuristics and the procedure for finding an optimal solution, and the one on the right shows the times for search-based heuristics. As we discussed in Section (4.1), optimal solution requires a search procedure that scales exponentially in N. The computation times for the search procedure are 63.57 and 5782.63 minutes for $N = 12$ and $N = 18$, respectively. Note that 5782.63 minutes is approximately 4 days, so computational times are likely to exceed weeks, or even months for $N \geq 24$ on a desktop machine. Thus we did not attempt to solve the problems for optimally for N =24, 30, 36.

We observe that all DP-based heuristics scale nicely indicating a polynomial time complexity. AH has the lowest computation times among all heuristics, but recall its poor performance from an accuracy point of view. As expected, computational time for AH II increases slightly as n is increased. The one heuristic that stands out is AH I. Although the plot of the times exhibits a polynomial behavior, its order is higher than all the other heuristics. Due to the long computation times (in the order of days), we decided to exclude $N = 36$ for AH I. Search-based heuristics also scale satisfactorily and unlike the DP-based heuristics, they have comparable computation times. The difference between Divide and Merge heuristics becomes only apparent for higher N.

Overall, all of our heuristics show polynomial time behavior. Except AH I, they all scale nicely as N is increased. Finally, AH I is not recommended for longer planning horizons due to it excessive solution times.

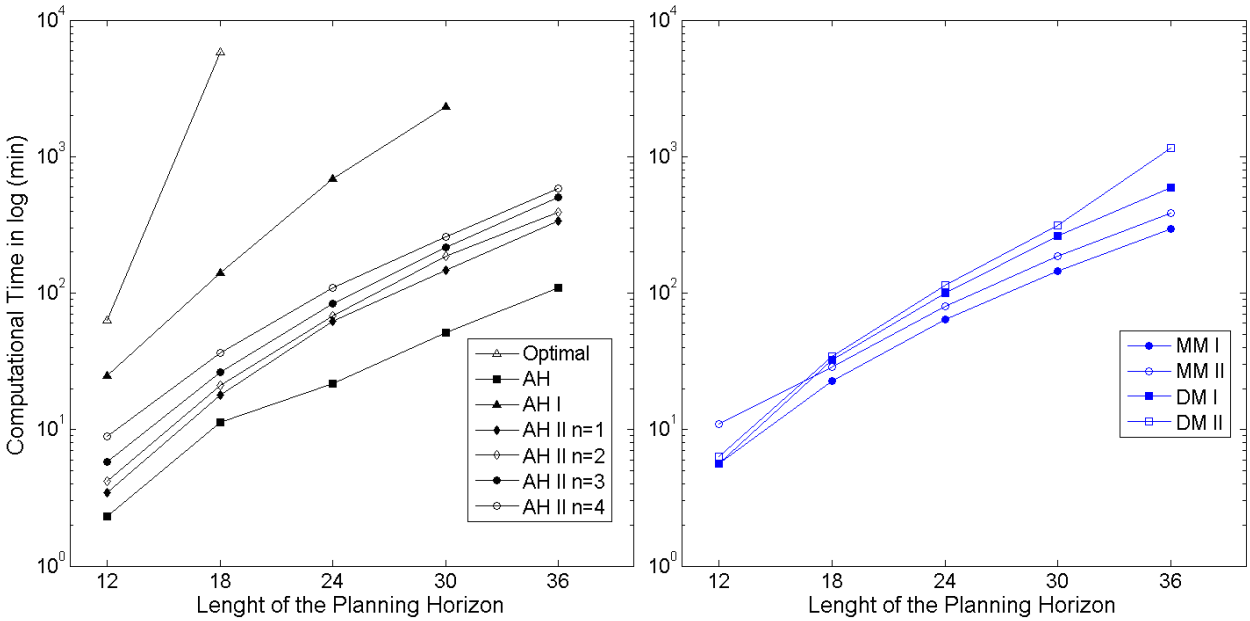

Figure 1 Computational Times versus N

7.4. Statistical Analysis

Our conclusions so far have been based on the trends observed in the results. We will close out the numerical study with a statistical analysis to explore the dominance relationships between the heuristics. We conduct paired t-tests with 95% confidence between the optimality gaps of heuristics for each problem solved. The details of the t-tests are in Online Appendix C.

Stationary Parameters: Out of the 720 problems in the test bed with stationary parameters for $N = 12$, we take the optimality gap of each heuristic as an observation and compare whether the sample mean optimality gap of a procedure differs from the others. Since we apply the heuristics to the same problems, we can pair the observations with respect to the heuristics. The relationship between the heuristics is found to be

$$\text{AH} < \text{AH II}_{n=1} \equiv \text{AH II}_{n=2} < \text{AH I} < \text{AH II}_{n=3} < \text{AH II}_{n=4} < \text{MM I} \equiv \text{MM II} \equiv \text{DM I} < \text{DM II}$$

where "$a < b$" represents that heuristic a provides significantly less accuracy compared to heuristic b, "$a \equiv b$" denotes there is no significant difference. For the test bed with stationary parameters, DM II outperforms the other heuristics – a statistical support for our conclusion based on the results in Table 3.

Next, we apply the same tests for the pairwise combinations of the heuristics. The results indicate the following relationships (AH II is AH II with $n = 4$):

$$\text{MM I \& AH} < \text{MM I \& AH I} \equiv \text{MM I \&AH II} \equiv \text{MM II \& AH I} < \text{MM II \& AH} \equiv \text{MM II \& AH II} \equiv \text{DM I \& AH} < \text{DM I \& AH I} \equiv \text{DM I \& AH II} \equiv \text{DM II \& AH} \equiv \text{DM II \& AH I} \equiv \text{DM II \& AH II},$$

which simply suggest that there is no statistical evidence for DM I & AH I, DM I & AH II, DM II & AH, DM II & AH I, and DM II & AH I pairs to have different sample means. Note that this is a statement on the sample mean. From Table 5, we already know DM II & AH II pair achieves the most number of optimal solutions and has the lowest maximum optimality gap. Hence, we still would recommend DM II & AH II pair for this test bed.

Dynamic Capacity: The results of the statistical analysis for the dynamic capacity test bed with a sample size of 8640 problem instances yield the following relationship:

$$\text{AH} < \text{AH II}_{n=1} < \text{AH II}_{n=2} < \text{AH I} \equiv \text{AH II}_{n=3} < \text{AH II}_{n=4} < \text{MM I} < \text{MM II} < \text{DM I} < \text{DM II}$$

Note that the line up of the heuristics is the same as in the test bed with stationary parameters, but some relationships are different. Except one relationship (between AH I and AH II with $n = 3$),

there is significant statistical evidence that the mean optimality gaps are different suggesting a clear ranking in terms of accuracy. DM II outperforms all other individual heuristics which is in line with the results in Table 7.

For the analysis of paired heuristics, we follow the organization of Table 8 and look at the search-based heuristics one at a time:

$$\begin{array}{ccccc} \text{MM I \& AH} & < & \text{MM I \& AH I} & \equiv & \text{MM I \& AH II} \\ \text{MM II \& AH I} & < & \text{MM II \& AH} & \equiv & \text{MM II \& AH II} \\ \text{DM I \& AH} & < & \text{DM I \& AH I} & \equiv & \text{DM I \& AH II} \\ \text{DM II \& AH} & < & \text{DM II \& AH I} & \equiv & \text{DM II \& AH II.} \end{array}$$

Moreover, any Merge heuristic (MM I or MM II) and a given DP-based heuristic pair is found to be dominated by a Divide heuristic (DM I or DM II) and the same DP-based heuristic pair in the t-tests. For example, let us take AH II:

$$\begin{array}{ccc} \text{MM I \& AH II} & < & \text{DM I \& AH II} \\ \text{MM I \& AH II} & < & \text{DM II \& AH II} \\ \text{MM II \& AH II} & < & \text{DM I \& AH II} \\ \text{MM II \& AH II} & < & \text{DM II \& AH II.} \end{array}$$

These results support our earlier observations based on the results in Table 8. However, we could not find statistical evidence indicating distinct sample means of Divide heuristic pairs. The results show that

$$\text{DM I \& AH I} \equiv \text{DM I \& AH II} \equiv \text{DM II \& AH I} \equiv \text{DM II \& AH II.}$$

It is not surprising that the results above corroborate our earlier conclusions.

7.5. Discussion

Our extensive numerical experiments showed that capacity (maximum lot size o) is the factor that clearly differentiates the performance of the two classes of heuristics we have developed. The handicap of DP-based heuristics is the assumption of zero inventory level for solving each stage's problem in the backward recursion. Opening inventory level is a random variable and the discrepancy between the actual inventory level for an optimal solution and the zero level assumption gets larger resulting in selecting base stock levels lower than they should be. This optimistic behavior yields higher backlogs and increased overall cost. Even for the best performing DP-based heuristic, AH II, we have observed a maximum optimality gap above 10% for the test bed with stationary capacity. When the capacity is dynamic, this figure goes beyond 25% mark.

Overall, search-based heuristics perform better than DP-based ones. The search starts with either having 1 (Divide heuristics) or N (Merge heuristics) production cycles. The procedure moves to a production schedule with one cycle less or more, and it keeps on adding/reducing cycles until no more improvement in cost is observed. The major shortcoming is to end up with a non-optimal production schedule due to the greedy search these procedures follows. We observed in our experiments that this drawback yields unsatisfactory results only for the scenarios with high capacity for which it is likely to have a few setups at optimality. This can also be used to explain the performance difference between Merge and Divide heuristics. MM I and MM II start the search further away from an optimal production schedule in terms of the number of setups. Therefore, it is more likely for them to get stuck in a production schedule further away from an optimal solution. On the contrary, DM I and DM II start off with a single production schedule, which is less likely to deviate from an optimal schedule that has a small number of setups. The numerical results clearly support the commanding lead of the Divide heuristics, which has considerably lower average and maximum optimality gaps, and better gap distributions. We also found statistical evidence that Divide heuristics dominate all other procedures for average optimality gap.

Revealing the shortcomings of both classes of heuristics and observing the complementary behavior led us to the idea of pairing DP-based heuristics with search-based heuristics. This approach works well having accuracy levels that cannot be achieved by individual heuristics. Considering both the accuracy and the computation time, we recommend the use of DM I & AH $II_{n=4}$ or DM II & AH $II_{n=4}$ for Problem P. These pairs yield 0.01% average optimality gap and 3.23% maximum optimality gap, and solve more than 97% of the total number of problem instances to optimality.

8. Conclusion

We consider the the stochastic lot sizing problem under static dynamic uncertainty strategy. Özen et al. (2012) have shown that a modified base stock policy is optimal, but finding an optimal solution requires an exhaustive search that grows exponentially in the length of the planing horizon. They proposed dynamic programming (DP) based heuristics for the uncapacitated problem and provided numerical evidence on their good performance. In this paper, we study the capacitated problem and develop two classes of heuristics. The first class, DP-based heuristics, consists of the best performing heuristic of Özen et al. (2012) and two variations (AH I and AH II) that we developed in this paper. Our extensive numerical experiments show that DP-based heuristics performs well in slack capacity regimes, which supports the earlier findings of Özen et al. (2012), but performs poorly if the capacity is tight. The second class, search-based heuristics, performs better than DP-based heuristics in general, but the performance degrades as the capacity becomes slack. By exploiting the complementary behavior of these two classes of heuristics, we tested all pairwise combinations of DP- and search-based heuristics and found that Divide heuristics (DM I or DM II –both of which are search-based procedures) coupled with one specific DP-based heuristic (AH II) outperforms all other combinations and have reasonable computation times. Over a total of 9000 problem instances considered, we report average and maximum optimality gaps of 0.01% and 3.23%, respectively, and more than 97% cases being solved to optimality with these two procedures. Moreover, applying paired t-test, we show statistical evidence that DM I & AH II and DM II & AH II pairs have smaller average optimality gaps than other pairs.

We plan to tackle the multi-item version of the problem under consideration as a topic for future research. This is a harder problem because coordinating the setups between items adds another layer of complexity. The learnings from this study would guide the heuristic development for the multi-item problem.

References

Bitran, G.R., H.H. Yanase. 1982. Computational complexity of the capacitated lot size problem. *Management Science* **28**(10) 1174–1186.

Bookbinder, J. H., J. Y. Tan. 1988. Strategies for the probabilistic lot-sizing problem with service-level constraints. *Management Science* **34** 1096–1108.

Brahimi, N., S. Dauzere-Peres, N. M. Najid, A. Nordli. 2006. Single item lot sizing problems. *European Journal of Operational Research* **168**(1) 1–16.

Guan, Y., A.J. Miller. 2008. Polynomial-time algorithms for stochastic uncapacitated lot-sizing problems. *Operations Research* **56**(5) 1172–1183.

Huang, K., S. Kucukyavuz. 2008. On stochastic lot-sizing problems with random lead times. *Operations Research Letters* **36** 303–308.

Inderfurth, K. 1994. Nervousness in inventory control. *OR Spektrum* **16** 113–123.

Özen, U., M.K. Dogru, S.A. Tarim. 2012. Static dynamic uncertainty strategy for single-item stochastic inventory control problem. *Omega* **30** 348–357.

Scarf, H. 1959. Optimality of (s,s) in the dynamic inventory problem. Tech. rep., Applied Mathematics and Statistics Laboratory Stanford University.

Sethi, S. P., F. Cheng. 1997. Optimality of (s, s) policies in inventory models with markovian demand. *Operations Research* **45** 931–939.

Sox, C. R. 1997. Dynamic lot sizing with random demand and non-stationary costs. *Operations Research Letters* **20** 155–164.

Sox, C.R., P. L. Jackson, A. Bowmanc, J.A. Muckstadt. 1999. A review of the stochastic lot scheduling problem. *International Journal of Production Economics* **62**(3) 181–200.

Sox, C.R., J.A. Muckstadt. 1997. Optimization-based planning for the stochastic lot-scheduling problem. *IIE Transactions* **29** 349–357.

Tarim, S. A., M. K. Doğru, U. Özen, R. Rossi. 2009. An efficient computational method for non-stationary (r, s) inventory policy with service level constraints. *Working Paper*, Alcatel-Lucent Bell Labs.

Tarim, S. A., B. G. Kingsman. 2004. The stochastic dynamic production/inventory lot-sizing problem with service-level constraints. *International Journal of Production Economics* **88** 105–119.

Tempelmeier, H. 2007. On the stochastic uncapacitated dynamic single-item lotsizing problem with service level constraints. *European Journal of Operational Research* **181** 184–194.

Tempelmeier, H. 2011. *Inventory management in supply networks : problems, models, solutions*. Norderstedt : Books on Demand GmbH.

Vargas, V. 2009. An optimal solution for the stochastic version of the wagner whitin dynamic lot-size model. *European Journal of Operational Research* **198** 447–451.

Veinott, A.T., H. Wagner. 1965. Computing optimal (s, s) inventory policies. *Management Science* **11**(5) 525–552.

Wagner, H.M., T.M. Whitin. 1958. Dynamic version of the economic lot size model. *Management Science* **5**(1) 88–96.