An efficient computational method for a stochastic dynamic lot-sizing problem under service-level constraints

Citation for published version:
Tarim, SA, Dogru, MK, Oezen, U & Rossi, R 2011, 'An efficient computational method for a stochastic dynamic lot-sizing problem under service-level constraints', *European Journal of Operational Research*, vol. 215, no. 3, pp. 563-571. https://doi.org/10.1016/j.ejor.2011.06.034

Digital Object Identifier (DOI):
10.1016/j.ejor.2011.06.034

Link:
Link to publication record in Edinburgh Research Explorer

Document Version:
Early version, also known as pre-print

Published In:
European Journal of Operational Research

An Efficient Computational Method for Non-Stationary (R, S) Inventory Policy with Service Level Constraints

S. Armagan Tarim[a,*,1], Mustafa K. Doğru[b,2], Ulaş Özen[b,2], Roberto Rossi[c]

[a] *Department of Management, Hacettepe University, 06800 Ankara, Turkey*
[b] *Alcatel-Lucent Bell Labs, Blanchardstown Industrial Park, Dublin 15, Ireland*
[c] *Logistics, Decision and Information Sciences Group, Wageningen UR, the Netherlands*

Abstract

This paper provides an efficient computational approach to solve the mixed integer programming (MIP) model developed by Tarim and Kingsman (2004) for calculating the parameters of an (R, S) policy in a finite horizon with non-stationary stochastic demand and service level constraints. Given the replenishment periods, we characterize the optimal order-up-to levels for the MIP model and use it to guide the development of a relaxed MIP model, which can be solved in polynomial time. The effectiveness of the proposed method hinges on three novelties: (i) the proposed relaxation is computationally efficient and yields an optimal solution most of the time, (ii) if the relaxation produces an infeasible solution, this solution can be used as a tight lower bound, and also (iii) this infeasible solution can be modified easily to obtain a feasible solution, which is an upper bound for the optimal solution. In case of infeasibility, the relaxation approach is implemented at each node of the search tree in a simple branch-and-bound procedure to efficiently search for an optimal solution. Extensive numerical tests show that our method dominates the MIP solution approach and can handle real-life size problems in trivial time.

Key words: relaxation, lot sizing, stochastic non-stationary demand, mixed integer programming, service level, replenishment cycle policy

1. Introduction

This paper considers a single stage inventory system facing non-stationary stochastic demand of the customers under periodic review. There is a service

*Corresponding author

Email addresses: armagan.tarim@hacettepe.edu.tr (S. Armagan Tarim), dogru@alcatel-lucent.com (Mustafa K. Doğru), uozen@alcatel-lucent.com (Ulaş Özen), roberto.rossi@wur.nl (Roberto Rossi)

[1] Supported by Hacettepe University (BAB) and the Scientific and Technological Research Council of Turkey (TUBITAK) under Grant No. SOBAG-108K027.

[2] Supported by Industrial Development Agency (IDA) Ireland.

level constraint that a certain probability of no-stock-out has to be achieved in each period in a finite planning horizon of N periods. Costs consist of linear inventory holding cost and fixed cost of placing an order. Any unfulfilled demand is backlogged and satisfied as soon as possible. This system first appeared in [2] and the authors analyzed it under the so-called *static-dynamic uncertainty strategy* where the timing of the replenishment orders (referred to as replenishment schedule) are fixed at the beginning of the planning horizon, but the exact order sizes depend on the demand realizations. This strategy corresponds to the class of (R, S) policy (known as the *replenishment cycle policy*), R denoting the length of a replenishment cycle and S the order-up-to-level, see [1] for more on the (R, S) policy. Due to the non-stationary nature of demand, the replenishment cycles and order-up-to levels in the planning horizon vary, hence, we denote them by R_t and S_t.

Bookbinder and Tan proposes a heuristic method for determining non-stationary R_t and S_t parameters in [2], where the replenishment schedule is determined first and the order-up-to levels are set afterwards. Under the same assumptions, in [3], Tarim and Kingsman provides a mixed-integer programming (MIP) model to compute R_t and S_t optimally, without addressing the computational performance issues. In [4], Tarim and Smith give an equivalent constraint programming (CP) formulation of the same problem with a similar performance.

This paper presents a new efficient method for solving the MIP model proposed in [3] for computing R_t and S_t parameters of a non-stationary (R, S) policy and does not require the use of any MIP or CP commercial solver. The relaxation essentially overlooks the fact that two consecutive replenishment cycles are dependent.

Our computational procedure works as follows. First, an optimal solution to the relaxed model is found by solving an equivalent shortest path problem, which is done in polynomial time with the "reaching algorithm" having a complexity of $O(N^2)$, see [5]. The optimal cost of the relaxed model is a lower bound on the optimal cost of the original model. Next, the feasibility of this solution for the original model is checked in polynomial time. If the solution is feasible, then it is also optimal for the original model, hence, the procedure is terminated. Otherwise, the replenishment schedule of this solution is used to obtain a feasible solution for the original model in polynomial time, the cost of which serves as an upper bound. Next, a branch-and-bound procedure is initiated to search efficiently for an optimal solution to the original model. The relaxation approach described above is implemented at each node of the search tree until all the nodes have been explored or pruned. The numerical tests over a wide range of randomly selected system parameters reveal that our procedure terminates after solving the shortest path problem in the majority of the cases. For the rest, the branch-and-bound procedure enhanced with the upper and lower bounds works efficiently. Overall, the results show that the computational performance has been increased by many orders of magnitude, now rendering it possible to solve any practically relevant instance in trivial time.

The paper is organized as follows. In §2, we set the notation and introduce the MIP formulation by Tarim and Kingsman. §3 is dedicated to the development of the relaxed model and the theory behind the computational method we propose for calculating R_t and S_t. The computational procedure is presented in §4. Numerical tests and results are reported in §5. Finally, in §6 some concluding remarks are given.

2. Notation and the MIP Model

Consider the following MIP model developed in [3] for calculating the policy parameters of a non-stationary (R, S) policy.

$$\min \mathsf{E}[TC] = \sum_{t=1}^{N} \left(a\delta_t + h\tilde{I}_t \right) \tag{1}$$

s.t.

$$\tilde{I}_t = \tilde{S}_t - \tilde{d}_t \qquad t = 1, \ldots, N \tag{2}$$

$$\tilde{S}_t \geq \tilde{I}_{t-1} \qquad t = 1, \ldots, N \tag{3}$$

$$\tilde{S}_t - \tilde{I}_{t-1} \leq M\delta_t \qquad t = 1, \ldots, N \tag{4}$$

$$\tilde{S}_t - \tilde{I}_{t-1} \geq -M\delta_t \qquad t = 1, \ldots, N \tag{5}$$

$$\tilde{I}_t \geq \sum_{j=1}^{t} \left(G^{-1}_{d_{t-j+1}+d_{t-j+2}+\ldots+d_t}(\alpha) - \sum_{k=t-j+1}^{t} \tilde{d}_k \right) P_{tj} \qquad t = 1, \ldots, N \tag{6}$$

$$\sum_{j=1}^{t} P_{tj} = 1 \qquad t = 1, \ldots, N \tag{7}$$

$$\tilde{I}_t \geq 0 \qquad t = 1, \ldots, N \tag{8}$$

$$P_{tj} \geq \delta_{t-j+1} - \sum_{k=t-j+2}^{t} \delta_k \qquad j = 1, \ldots, t \quad t = 1, \ldots, N \tag{9}$$

$$\delta_t, P_{tj} \in \{0, 1\} \qquad j = 1, \ldots, t \quad t = 1, \ldots, N, \tag{10}$$

where

TC	:	total holding and ordering/set-up cost of the system over N periods;
a	:	fixed ordering/set-up cost;
h	:	proportional inventory holding cost per period;
α	:	probability that the closing inventory in a period is non-negative (type I service level);
d_t	:	demand in period t, a non-negative random variable with probability density function, $g_t(d_t)$;
δ_t	:	a binary variable that takes the value of 1 if a replenishment occurs in period t and 0 otherwise;
$\tilde{I}_t$	:	expected inventory level at the end of period t;
$\tilde{I}_0$	:	the inventory level at the beginning of the planning horizon;
$\tilde{S}_t$	:	the order-up-to-level for period t if there is any replenishment, otherwise expected opening inventory level;

and "$\sim$" denotes the expectation operator, ie., $\tilde{Y} = \mathsf{E}[Y]$ for any random variable Y. M is some large positive number and $G_{d_i+d_{i+1}+\ldots+d_j}(\cdot)$ is the cumulative probability distribution function of $d_i + d_{i+1} + \ldots + d_j$. It is assumed that G is strictly increasing, therefore G^{-1} is uniquely defined, and

$$d_t \geq 0 \quad t = 1, \ldots, N. \tag{11}$$

Without loss of generality, we make the following mild assumption for the sake of simplicity: the beginning inventory ($\tilde{I}_0$), the demand distribution in period 1 (d_1) and the service level (α) ensure a replenishment order being placed in period 1, ie., $\delta_1 = 1$.

The objective is to minimize the expected inventory holding and ordering costs in a horizon of N periods, see (1). The expected closing inventory for a period is simply the expected opening inventory minus the expected demand for the period, which is given in (2). The order-up-to level in period t (or the expected opening inventory level if no replenishment is scheduled), $\tilde{S}_t$, is greater than (equal to) the expected closing inventory level of the previous period, $\tilde{I}_{t-1}$; see (3). If $\tilde{S}_t - \tilde{I}_{t-1} > 0$ then a replenishment is scheduled in period t, so $\delta_t = 1$. Otherwise, $\tilde{S}_t - \tilde{I}_{t-1} = 0$, no replenishment is scheduled for period t, hence, $\delta_t = 0$. Constraint (5) is not part of the MIP model developed in [3], but added by us as a redundant constraint. Note that whether δ_1 is 0 or 1, constraint (3) is tighter than (5). The need for this constraint is purely technical and will be clear in the proof of Lemma 2.

The main idea behind this MIP formulation is the introduction of binary variables P_{tj} for $t = 1, \ldots, N$ and $j \leq t$, which takes the value 1 if an order is placed in period $t - j + 1$ to cover the demand of periods $t - j + 1, \ldots, t$. If $P_{tj} = 1$, then through (6), the order size should ensure the service level of α in period t. Due to the assumption of strictly increasing G, service level α translates into a safety stock level

$$G^{-1}_{d_{t-j+1}+d_{t-j+2}+\ldots+d_t}(\alpha) - \sum_{k=t-j+1}^{t} \tilde{d}_k \tag{12}$$

for period t, which is required to be achieved at the minimal. Hence, safety stock levels are calculated off-line using (12) and substituted into the MIP formulation to find the optimal ordering and inventory decisions. For a detailed discussion of the MIP model, we refer to [3].

Next, we consider a variation of the MIP model, which takes the replenishment periods as inputs. Define disjoint sets $\mathfrak{T}$ and $\overline{\mathfrak{T}}$ such that $\mathfrak{T} \cup \overline{\mathfrak{T}} = \{1, \ldots, N\}$, $\delta_i = 1$ for all $i \in \mathfrak{T}$ and $\delta_i = 0$ for all $i \in \overline{\mathfrak{T}}$. In plain words, we partition the set of periods into two subsets where $\mathfrak{T}$ is the set of periods with replenishment orders and $\overline{\mathfrak{T}}$ is the set with no orders. Recall that $\delta_1 = 1$, so $1 \in \mathfrak{T}$.

Lemma 1. *Given any $\mathfrak{T}$ and $\overline{\mathfrak{T}}$, the optimal solution for the MIP model has*

$$\tilde{S}_t = \begin{cases} \max\left\{\tilde{S}_{t-1} - \tilde{d}_{t-1}, G^{-1}_{d_t + d_{t+1} + \ldots + d_{\bar{t}-1}}(\alpha), \sum_{k=t}^{\bar{t}-1} \tilde{d}_k\right\} & \textit{for} \quad t \in \mathfrak{T} \\ \\ \tilde{S}_{t-1} - \tilde{d}_{t-1} & \textit{for} \quad t \in \overline{\mathfrak{T}}, \end{cases} \tag{13}$$

where $\tilde{S}_0 = \tilde{I}_0$, $\tilde{d}_0 = 0$ and

$$\bar{t} \quad = \quad \begin{cases} N+1 & \textit{if} \qquad t = N \\ N+1 & \textit{if} \qquad \delta_k = 0 \textit{ for all } k \geq t+1 \\ \min\{k|\delta_k = 1, k > t\} & \textit{o/w.} \end{cases} \tag{14}$$

Proof. The main line of thinking behind the proof is that we introduce new tighter feasible cuts, which allow us to eliminate some of the constraints in the original formulation and simplify further.

Given $\mathfrak{T}$ and $\overline{\mathfrak{T}}$, for any $t \in \mathfrak{T} \cup \overline{\mathfrak{T}} = \{1, \ldots, N\}$, let $\underline{t}$ be the first period at or before t with a replenishment order scheduled:

$$\underline{t} \quad = \quad \max\{k|\delta_k = 1, k \leq t\}. \tag{15}$$

Given a replenishment schedule, $\mathfrak{T}$ and $\overline{\mathfrak{T}}$, the MIP model reduces to the following linear program (LP):

$$\min \sum_{t=1}^{N} h(\tilde{S}_t - \tilde{d}_t) \tag{16}$$

s.t.

$$\tilde{S}_t \geq \tilde{S}_{t-1} - \tilde{d}_{t-1} \qquad t \in \mathfrak{T} \cup \overline{\mathfrak{T}} \tag{17}$$

$$\tilde{S}_t \leq \tilde{S}_{t-1} - \tilde{d}_{t-1} \qquad t \in \overline{\mathfrak{T}} \tag{18}$$

$$\tilde{S}_t \geq \tilde{S}_{t-1} - \tilde{d}_{t-1} \qquad t \in \overline{\mathfrak{T}} \tag{19}$$

$$\tilde{S}_t - \tilde{d}_t \geq G^{-1}_{d_{\underline{t}} + d_{\underline{t}+1} + \ldots + d_t}(\alpha) - \sum_{k=\underline{t}}^{t} \tilde{d}_k \qquad t \in \mathfrak{T} \cup \overline{\mathfrak{T}} \tag{20}$$

$$\tilde{S}_t \geq \tilde{d}_t \qquad t \in \mathfrak{T} \cup \overline{\mathfrak{T}} \tag{21}$$

$$\tilde{S}_0 = \tilde{I}_0, \ \tilde{d}_0 = 0, \tag{22}$$

which follows from substituting $\tilde{S}_t - \tilde{d}_t$ into $\tilde{I}_t$ in (3)-(6) and (8), and rewriting the righthand side of constraint (6) using the notation introduced in (15). The fixed cost of ordering, $\sum_{i=1}^{N} a\delta_i$, is constant, so it is dropped from the objective function. Since all δ variables are fixed, (3) is redundant for all t with $\delta_t = 1$. Thus, (4) reduces to (18). Similarly, (5) is redundant for all t with $\delta_t = 1$, leading to (19).

Combining (18) and (19) yields

$$\tilde{S}_t = \tilde{S}_{t-1} - \tilde{d}_{t-1} \qquad t \in \overline{\mathfrak{T}}. \tag{23}$$

Further, (17) implies

$$\tilde{S}_t \geq \tilde{S}_{t-1} - \tilde{d}_{t-1} \qquad t \in \mathfrak{T}. \tag{24}$$

We can add (23) and (24) as feasible cuts to the LP above. Note that constraints (17)-(19) become redundant in the presence of (23) and (24), hence, they can be left out from the LP formulation.

Consider any $t \in \mathfrak{T} \cup \overline{\mathfrak{T}}$ and $m \in \{\underline{t}+1, \ldots, \overline{t}-1\}$. The equations below follow from (23):

$$\begin{array}{rcl} \tilde{S}_{\underline{t}+1} & = & \tilde{S}_{\underline{t}} - \tilde{d}_{\underline{t}} \\ \tilde{S}_{\underline{t}+2} & = & \tilde{S}_{\underline{t}+1} - \tilde{d}_{\underline{t}+1} \\ \vdots & & \vdots \\ \tilde{S}_m & = & \tilde{S}_{m-1} - \tilde{d}_{m-1}. \end{array}$$

They imply

$$\tilde{S}_m = \tilde{S}_{\underline{t}} - \sum_{k=\underline{t}}^{m-1} \tilde{d}_k, \tag{25}$$

which follows from adding both sides of the equations and canceling equal terms. Using (20) and the equality above

$$\tilde{S}_m - \tilde{d}_m = \left(\tilde{S}_{\underline{t}} - \sum_{k=\underline{t}}^{m-1} \tilde{d}_k \right) - \tilde{d}_m \geq G^{-1}_{d_{\underline{t}}+\ldots+d_m}(\alpha) - \sum_{k=\underline{t}}^{m} \tilde{d}_k,$$

which simplifies to

$$\tilde{S}_{\underline{t}} \geq G^{-1}_{d_{\underline{t}}+\ldots+d_m}(\alpha). \tag{26}$$

For $\underline{t}$, the constraint (20) is

$$\begin{array}{rcl} \tilde{S}_{\underline{t}} - \tilde{d}_{\underline{t}} & \geq & G^{-1}_{d_{\underline{t}}}(\alpha) - \tilde{d}_{\underline{t}} \\ \tilde{S}_{\underline{t}} & \geq & G^{-1}_{d_{\underline{t}}}(\alpha). \end{array} \tag{27}$$

The inequalities in (26) and (27) can be rewritten as

$$\tilde{S}_{\underline{t}} \geq \max\left\{G^{-1}_{d_{\underline{t}}}(\alpha), G^{-1}_{d_{\underline{t}}+d_{\underline{t}+1}}(\alpha), \ldots, G^{-1}_{d_{\underline{t}}+\ldots+d_{\bar{t}-1}}(\alpha)\right\} = G^{-1}_{d_{\underline{t}}+\ldots+d_{\bar{t}-1}}(\alpha), \quad (28)$$

because of non-negative period demands and strictly increasing G. Hence, (28) for all $t \in \mathfrak{T}$ can be added to the LP formulation as a feasible cut. In the presence of (23) and (28), constraint (20) becomes redundant, thus, can be excluded.

Replacing the first four constraints in the LP with (23), (24) and (28) yields

$$\begin{aligned}
&\min \ \sum_{t=1}^{N} h(\tilde{S}_t - \tilde{d}_t) && \\
&\text{s.t.} && \\
&\tilde{S}_t \geq \tilde{S}_{t-1} - \tilde{d}_{t-1} && t \in \mathfrak{T} \\
&\tilde{S}_t = \tilde{S}_{t-1} - \tilde{d}_{t-1} && t \in \overline{\mathfrak{T}} \qquad (29)\\
&\tilde{S}_t \geq G^{-1}_{d_t+\ldots+d_{\bar{t}-1}}(\alpha) && t \in \mathfrak{T} \\
&\tilde{S}_t \geq \tilde{d}_t && t \in \mathfrak{T} \cup \overline{\mathfrak{T}}, \qquad (30)
\end{aligned}$$

where $\tilde{S}_0 = 0$, $\tilde{d}_0 = 0$, and $\bar{t}$ is defined in (14). Consider any $t \in \mathfrak{T} \cup \overline{\mathfrak{T}}$. From (30)

$$\tilde{S}_{\underline{t}} \geq \tilde{d}_{\underline{t}}. \quad (31)$$

Take any $m \in \{\underline{t}+1, \ldots, \bar{t}-1\}$. Substituting (25) into (30) leads to

$$\tilde{S}_m = \tilde{S}_{\underline{t}} - \sum_{k=\underline{t}}^{m-1} \tilde{d}_k \geq \tilde{d}_m,$$

which is equivalent to

$$\tilde{S}_{\underline{t}} \geq \sum_{k=\underline{t}}^{m} \tilde{d}_k.$$

The inequality above and (31) can be rewritten as

$$\tilde{S}_{\underline{t}} \geq \max\left\{\tilde{d}_{\underline{t}}, \tilde{d}_{\underline{t}} + \tilde{d}_{\underline{t}+1}, \ldots, \sum_{k=\underline{t}}^{\bar{t}-1} \tilde{d}_k\right\} = \sum_{k=\underline{t}}^{\bar{t}-1} \tilde{d}_k.$$

Hence, for any t, (30) implies the inequality above. The following constraint can be added as a feasible cut to the LP

$$\tilde{S}_t \geq \sum_{k=t}^{\bar{t}-1} \tilde{d}_k \qquad t \in \mathfrak{T}.$$

In the presence of (29) and the inequality above, (30) becomes redundant, so can be omitted. Since $\sum_{t=1}^{N} \tilde{d}_t$ is constant, the objective function of the LP simplifies to minimizing $\sum_{t=1}^{N} \tilde{S}_t$. Thus, the LP reduces to

$$\begin{aligned}
&\min \ \sum_{t=1}^{N} \tilde{S}_t \\
&\text{s.t.} \\
&\tilde{S}_t \geq \tilde{S}_{t-1} - \tilde{d}_{t-1} && t \in \mathfrak{T} \\
&\tilde{S}_t = \tilde{S}_{t-1} - \tilde{d}_{t-1} && t \in \overline{\mathfrak{T}} \\
&\tilde{S}_t \geq G^{-1}_{d_t+\ldots+d_{\bar{t}-1}}(\alpha) && t \in \mathfrak{T} \\
&\tilde{S}_t \geq \sum_{k=t}^{\bar{t}-1} \tilde{d}_k && t \in \mathfrak{T},
\end{aligned}$$

the optimal solution of which is

$$\tilde{S}_t = \begin{cases} \max\left\{\tilde{S}_{t-1} - \tilde{d}_{t-1}, G^{-1}_{d_t+\ldots+d_{\bar{t}-1}}(\alpha), \sum_{k=t}^{\bar{t}-1} \tilde{d}_k\right\} & \text{for} \quad t \in \mathfrak{T} \\ \\ \tilde{S}_{t-1} - \tilde{d}_{t-1} & \text{for} \quad t \in \overline{\mathfrak{T}}. \end{cases}$$

This completes the proof. □

Lemma 1 gives the form of the replenishment quantities for a given replenishment schedule. The implication of this is that the ordering quantities can easily be computed once the optimal replenishment periods are known: Start with $\tilde{S}_1$ first, next $\tilde{S}_2$, ..., and last $\tilde{S}_N$. Hence, the optimization problem formulated by the MIP model can equivalently be expressed as

$$\min_{\delta_1,\ldots,\delta_N} \mathsf{E}[TC] = \sum_{t=1}^{N} \Big(a\delta_t + h(\tilde{S}_t - \tilde{d}_t)\Big), \tag{32}$$

where $\tilde{S}_t$ for $t = 1, \ldots, N$ are given in (13), $\tilde{S}_0 = \tilde{I}_0$ and $\tilde{d}_0 = 0$. Even though $\sum_{t=1}^{N} h\tilde{d}_t$ is constant and does not effect the optimization problem, we include it to reflect the equivalence of the objective functions of the MIP model and (32).

3. Relaxation of the MIP Model

Consider the relaxation of constraints (3) and (8) in the MIP model discussed in §2. We refer to this new mixed integer program as the *relaxed MIP model* here after. In order to differentiate the relaxed MIP model from the original given in (1)-(10), we replace $\tilde{S}_t$ with $\tilde{S}_t^r$ in the relaxed model. Similar to Lemma 1, ordering quantities for a given replenishment schedule in the relaxed MIP model can be explicitly determined, which is given next.

Lemma 2. *Given any $\mathfrak{T}$ and $\overline{\mathfrak{T}}$, the optimal solution for the relaxed MIP model has*

$$\tilde{S}_t^r = \begin{cases} G^{-1}_{d_t+d_{t+1}+\ldots+d_{\bar{t}-1}}(\alpha) & \textit{for} \quad t \in \mathfrak{T} \\ \\ \tilde{S}_{t-1}^r - \tilde{d}_{t-1} & \textit{for} \quad t \in \overline{\mathfrak{T}}, \end{cases} \tag{33}$$

where $\bar{t}$ is given in (14).

Proof. The proof mainly follows from the proof of Lemma 1. Given $\mathfrak{T}$ and $\overline{\mathfrak{T}}$, the relaxed MIP model reduces to the LP given by (16)-(22) without (17), (21) and (22):

$$\min \sum_{t=1}^{N} h(\tilde{S}_t^r - \tilde{d}_t) \tag{34}$$

s.t.

$$\tilde{S}_t^r \leq \tilde{S}_{t-1}^r - \tilde{d}_{t-1} \qquad t \in \overline{\mathfrak{T}} \tag{35}$$

$$\tilde{S}_t^r \geq \tilde{S}_{t-1}^r - \tilde{d}_{t-1} \qquad t \in \overline{\mathfrak{T}} \tag{36}$$

$$\tilde{S}_t^r - \tilde{d}_t \geq G^{-1}_{d_{\underline{t}}+d_{\underline{t}+1}+\ldots+d_t}(\alpha) - \sum_{k=\underline{t}}^{t} \tilde{d}_k \qquad t \in \mathfrak{T} \cup \overline{\mathfrak{T}} \tag{37}$$

where $\underline{t}$ and $\bar{t}$ are defined by (15) and (14), respectively for any $t \in \mathfrak{T} \cup \overline{\mathfrak{T}}$. Note that (22) becomes redundant because $\delta_1 = 1$.

The objective function reduces to minimizing $\sum_{t=1}^{N} \tilde{S}_t^r$ since $h \sum_{t=1}^{N} \tilde{d}_t$ is constant. Following the same line of thought as in the proof of Lemma 1, constraints (35) and (36) can be replaced with

$$\tilde{S}_t^r = \tilde{S}_{t-1}^r - \tilde{d}_{t-1} \qquad t \in \overline{\mathfrak{T}}.$$

The LP formulation reduces to

$$\min \sum_{t=1}^{N} \tilde{S}_t^r$$

s.t.

$$\tilde{S}_t^r = \tilde{S}_{t-1}^r - \tilde{d}_{t-1} \qquad t \in \overline{\mathfrak{T}}$$

$$\tilde{S}_t^r - \tilde{d}_t \geq G^{-1}_{d_{\underline{t}}+d_{\underline{t}+1}+\ldots+d_t}(\alpha) - \sum_{k=\underline{t}}^{t} \tilde{d}_k \qquad t \in \mathfrak{T} \cup \overline{\mathfrak{T}}.$$

Recall from the proof of Lemma 1 how the second constraint is replaced with

$$\tilde{S}_t^r \geq G^{-1}_{d_t+\ldots+d_{\bar{t}-1}}(\alpha) \qquad t \in \mathfrak{T}$$

using the first constraint. Hence, the LP model further simplifies to

$$\min \sum_{t=1}^{N} \tilde{S}_t^r$$

s.t.

$$\tilde{S}_t^r = \tilde{S}_{t-1}^r - \tilde{d}_{t-1} \qquad t \in \overline{\mathfrak{T}}$$

$$\tilde{S}_t^r \geq G^{-1}_{d_t+\ldots+d_{\bar{t}-1}}(\alpha) \qquad t \in \mathfrak{T},$$

having the following optimal solution

$$\tilde{S}_t^r = \begin{cases} G^{-1}_{d_t+\ldots+d_{\bar{t}-1}}(\alpha) & \text{for} \quad t \in \mathfrak{T} \\ \\ \tilde{S}_{t-1}^r - \tilde{d}_{t-1} & \text{for} \quad t \in \overline{\mathfrak{T}}, \end{cases}$$

which completes the proof. □

We can develop an equivalent optimization problem for the relaxed MIP model in a similar way as we do at the end of §2. Lemma 2 simply gives the form of the ordering quantity decisions for the relaxed MIP model, given the replenishment schedule. First $\tilde{S}_1^r$ is computed, then $\tilde{S}_2^r$, and so on. Hence, the relaxed MIP model is equivalent to the following optimization problem:

$$\min_{\delta_1,\ldots,\delta_N} z_r = \sum_{t=1}^{N} \Big(a\delta_t + h(\tilde{S}_t^r - \tilde{d}_t) \Big), \tag{38}$$

where $\tilde{S}_t^r$ for $t = 1, \ldots, N$ are given in (33). The objective is to find the replenishment schedule with the minimum cost such that the inventory levels at the beginning of each period ($\tilde{S}_t^r$ for period t) satisfies the property given in (33).

3.1. Equivalent Shortest Path Formulation

The optimization problem given in (38) can be transformed into a shortest path problem, which is well-studied in the field of network optimization. Consider an acyclic network $\mathcal{G} = (\mathcal{N}, \mathcal{A})$ with the set of nodes $\mathcal{N} = \{1, 2, \ldots, N+1\}$ denoting periods and arcs (i, j) connecting all pairs of nodes with $i < j$, $(i, j) \in \mathcal{A}$. Node $N+1$ is a dummy period representing the end of the planning horizon. Each arc (i, j) has a cost $c_{i,j}$ that is equal to the cost of ordering in period i to cover demand requirements through period $j-1$ and satisfy the service level α at the end of period $j-1$ by holding the minimum safety stock level. Hence, in period i, the inventory level is $G^{-1}_{d_i+d_{i+1}+\ldots+d_{j-1}}(\alpha)$; as a result

$$c_{ij} \quad = \quad a + h \sum_{k=i}^{j-1} \left(G^{-1}_{d_i+d_{i+1}+\ldots+d_{j-1}}(\alpha) - \sum_{m=i}^{k} \tilde{d}_m \right). \tag{39}$$

Consider a solution for the relaxed model: $\delta_1, \ldots, \delta_N$. Recall that we assume $\delta_1 = 1$ without loss of generality. Let i and j be two consecutive periods with scheduled orders, ie., $i, j \in \{1, \ldots, N\}$, $\delta_i = \delta_j = 1$ and $\delta_{i+1} = \delta_{i+2} = \ldots = \delta_{j-1} = 0$. The contribution of the order placed in period i to the objective function (in other words, the cost of periods $i, \ldots, j-1$) is

$$a + h \sum_{k=i}^{j-1} \left(G^{-1}_{d_i + d_{i+1} + \ldots + d_{j-1}}(\alpha) - \sum_{m=i}^{k} \tilde{d}_m \right),$$

which follows from (38). Note that this cost is equivalent to c_{ij}. Thus, ordering in period i with an order-up-to level $G^{-1}_{d_i + d_{i+1} + \ldots + d_{j-1}}(\alpha)$ and placing no replenishment orders until period j in the optimization problem (38) corresponds to picking arc (i, j) in $\mathcal{G}$. Finding the shortest path between nodes 1 and $N+1$ in network $\mathcal{G}$ is equivalent to solving the optimization problem in (38), which in return is equivalent to the relaxed MIP model.

Recall that $\mathcal{G}$ is an acyclic network, so *the reaching algorithm* solves the shortest path problem from node 1 to $N+1$ in $O(|\mathcal{A}|)$ time, see [5]. Since $|\mathcal{A}| = \frac{N(N+1)}{2}$, we have a very efficient algorithm for solving (38)[3]. Next, we present our main theoretical result that connects the solution for (38) to the MIP model in (1)-(10).

Theorem 3. *Let $\boldsymbol{\delta}^* = (\delta_1^*, \ldots, \delta_N^*)$ and $\mathbf{S}^* = (\tilde{S}_1^{r*}, \ldots, \tilde{S}_N^{r*})$ be an optimal solution to the optimization problem given in (38), and z_r^* be the corresponding optimal objective value. Define z^* as the optimal objective value for the MIP model. If $\mathbf{S}^*$ is feasible for (32), ie., $\{\tilde{S}_t | \delta_1^*, \ldots, \delta_N^*\} = \tilde{S}_t^{r*}$ for all $t = 1, \ldots, N$, then $\boldsymbol{\delta}^*$ and $\mathbf{S}^*$ is also optimal for the MIP model. Otherwise, $z_r^* \leq z^*$.*

Proof. The optimization problem in (38) is equivalent to the relaxed MIP model, so an optimal solution for (38), $\boldsymbol{\delta}^*$ and $\mathbf{S}^*$, is also optimal for the relaxed MIP model. Similarly, we have shown the equivalence between the MIP model and (32), thus, if $(\boldsymbol{\delta}^*, \mathbf{S}^*)$ is feasible for (32), then it is also feasible for the MIP model given in (1)-(10), see [6]. Moreover, the optimal objective value for the relaxed MIP model (z_r^*) is a lower bound on the optimal objective function value of the MIP model (z^*). Therefore, if $(\boldsymbol{\delta}^*, \mathbf{S}^*)$ is feasible for (32) then $z^* = z_r^*$; otherwise, the optimal solution for the MIP model differs from $(\boldsymbol{\delta}^*, \mathbf{S}^*)$, but $z_r^* \leq z^*$ still holds. □

3.2. Subproblem

Consider a version of the optimization problem in (32) where some of the values of the decision variables δ_t are fixed. Define two disjoint sets $\mathfrak{L}$ and $\overline{\mathfrak{L}}$ such that $\mathfrak{L} \cup \overline{\mathfrak{L}} \subset \{1, \ldots, N\}$, $\delta_i = 1$ for all $i \in \mathfrak{L}$ and $\delta_m = 0$ for all $m \in \overline{\mathfrak{L}}$. In other words, $\mathfrak{L}$ denotes the set of periods for which the decision variables (whether to place an order) are set to 1, and similarly $\overline{\mathfrak{L}}$ is the set of periods with decision variables fixed at 0. Using the result of Lemma 1, the MIP formulation with

[3]Tarim and Smith propose a procedure that increases the speed of the solution algorithm for this specific problem setting in [4]. We employ their procedure in our numerical study.

$\delta_i = 1$ for all $i \in \mathfrak{L}$ and $\delta_m = 0$ for all $m \in \overline{\mathfrak{L}}$ can equivalently be represented by the following optimization problem:

$$\min_{\delta_1,\ldots,\delta_N} w = \left\{ \sum_{t=1}^{N} \Big(a\delta_t + h(\tilde{S}_t - \tilde{d}_t) \Big) \Big| \delta_i = 1 \ \forall i \in \mathfrak{L},\ \delta_m = 0 \ \forall m \in \overline{\mathfrak{L}} \right\}, \quad (40)$$

where $\tilde{S}_t$ for $t = 1, \ldots, N$ are given in (13). We refer to the problem in (40) as a *subproblem.* In a similar fashion, using Lemma 2, the relaxed MIP formulation of §3 with $\delta_i = 1$ for all $i \in \mathfrak{L}$ and $\delta_m = 0$ for all $m \in \overline{\mathfrak{L}}$ can be represented by

$$\min_{\delta_1,\ldots,\delta_N} w_r = \left\{ \sum_{t=1}^{N} \Big(a\delta_t + h(\tilde{S}_t^r - \tilde{d}_t) \Big) \Big| \delta_i = 1 \ \forall i \in \mathfrak{L},\ \delta_m = 0 \ \forall m \in \overline{\mathfrak{L}} \right\}, \quad (41)$$

where $\tilde{S}_t^r$ for $t = 1, \ldots, N$ are given in (33).[4] Note that (41) is a relaxation of (40), so we refer to it as the *relaxed subproblem.* An equivalent shortest path formulation for (41) can be developed as discussed in §3.1, but with some modifications in the arc costs. For any $i \in \overline{\mathfrak{L}}$, set $c_{ji} = c_{ik} = \infty$ for all $j < i$ and for all $k > i$. This way no arc entering or leaving node i is selected. Similarly, for any $i \in \mathfrak{L}$, set $c_{jk} = \infty$ for all $j < i$ and for all $k > i$, which assures that node i is visited in the shortest path. The reaching algorithm can still be used to solve the resulting shortest path problem.

Next, similar to Theorem 3, we give a new theorem that links the solution of (41) to the MIP formulation with $\delta_i = 1$ for all $i \in \mathfrak{L}$ and $\delta_m = 0$ for all $m \in \overline{\mathfrak{L}}$.

Theorem 4. *Given any $\mathfrak{L}$ and $\overline{\mathfrak{L}}$, let $\boldsymbol{\delta}_s^* = (\delta_1^*, \ldots, \delta_N^*)$ and $\mathbf{S}_s^* = (\tilde{S}_1^{r*}, \ldots, \tilde{S}_N^{r*})$ be an optimal solution to the optimization problem given in (41), and w_r^* be the corresponding optimal objective value. Define w^* as the optimal objective value for the MIP model with $\delta_i = 1$ for all $i \in \mathfrak{L}$ and $\delta_m = 0$ for all $m \in \overline{\mathfrak{L}}$. If $\mathbf{S}_s^*$ is feasible for (40), ie., $\{\tilde{S}_t | \delta_1^*, \ldots, \delta_N^*\} = \tilde{S}_t^{r*}$ for all $t = 1, \ldots, N$, then $\boldsymbol{\delta}_s^*$ and $\mathbf{S}_s^*$ is also optimal for the MIP model with $\delta_i = 1$ for all $i \in \mathfrak{L}$ and $\delta_m = 0$ for all $m \in \overline{\mathfrak{L}}$. Otherwise, $w_r^* \leq w^*$.*

Proof. The proof follows the same line of thought as in the proof of Theorem 3, hence, omitted. □

Theorem 4 extends the result of Theorem 3 to cases where some values of δ_t in the original MIP formulation given in (1)-(10) are fixed. This case is relevant for us because it arises in a branch-and-bound procedure where the branching is done on binary variables δ_t.

At this point, we have all the theoretical results to design a method for the computation of the solution of the original MIP model in (1)-(10). The details of the computational procedure are discussed next.

---

[4]Note that (32) and (38) are equivalent to (40) and (41) respectively when $\mathfrak{L} = \overline{\mathfrak{L}} = \emptyset$.

4. Computational Procedure

The results of Theorems 3 and 4 have significant implications for solving the original MIP model, and leads to the following computational procedure. An optimal solution ($\boldsymbol{\delta}^* = (\delta_1^*, \ldots, \delta_N^*)$ and $\mathbf{S}^* = (\tilde{S}_1^{r*}, \ldots, \tilde{S}_N^{r*})$) for the relaxed MIP model is obtained by solving the shortest path problem in the equivalent network. The feasibility of this solution can be checked by calculating $\tilde{S}_t$ for all t by substituting $\delta_1^*, \ldots, \delta_N^*$ in (13), and comparing these values against $\mathbf{S}^*$. If $\tilde{S}_t = \tilde{S}_t^r$ for all t then we terminate: $(\boldsymbol{\delta}^*, \mathbf{S}^*)$ is an optimal solution for the MIP model and z_r^* (the optimal cost for the relaxed MIP model) is the optimal cost. Otherwise, z_r^* is a lower bound for the optimal cost of the MIP model (z^*). It is worthwhile to mention that solving for the shortest path provides all optimal solutions for the relaxed MIP model in case of multiple optima. All such solutions are candidates for the MIP model, so they can be checked for feasibility.

If the solution is infeasible, the replenishment schedule $\boldsymbol{\delta}^*$ can still be used to obtain an upper bound. Substituting $\boldsymbol{\delta}^*$ in (13) gives $\tilde{S}_t$ for all t, and these values are substituted in (1) to calculate the objective function value of the MIP model. This cost is an upper bound on z^* and a current best solution for the branch-and-bound (B&B) algorithm. After finding a current best solution, we continue our search by performing a depth-first B&B search. The variable selection branches uniquely on replenishment decisions, δ_t proceeds by selecting those periods t's for which an infeasible (negative) expected order quantity is scheduled. Once we decide on which decision variable δ_t we want to branch, we continue with the value selection: first not to schedule a replenishment in such a period (i.e. $\delta_t = 0$) and then to schedule a replenishment (i.e. $\delta_t = 1$). We remark that every node in the search tree represents a subproblem with some replenishment decisions fixed. Therefore, using the result of Theorem 4, for each subproblem, we can easily obtain either a lower-bound or the exact solution. If the solution of the relaxed subproblem is feasible, we stop exploring the current node, and we store the solution found provided that it improves the current best solution. Otherwise, we use the lower bound to exclude suboptimal part of the search tree. We proceed with this strategy until all the nodes have been explored or pruned.

We demonstrate the effectiveness of the computational procedure in the next section through a numerical study. The procedure outlined above has been implemented in Java, see [7].

5. Numerical Study

A computational experiment is designed to investigate the effectiveness of the method proposed in this paper. The following issues are addressed in the experiment:

- the percentage of the non-stationary instances solved to optimality using solely the relaxed-MIP approach, without resorting to any search effort,

- the effectiveness of the bounds provided by the relaxed-MIP model if the observed solution is infeasible for the original problem,
- the overall solution time performance of the proposed method,
- the scalability of the proposed method.

For this purpose we used four different planning horizon lengths, $N = 30, 40, 50, 60$, and five different mean demand patterns, namely, (i) stationary, P_1, (ii) seasonal, P_2, (iii) decreasing, P_3, (iv) increasing, P_4, and (v) product life-cycle, P_5. The formal definitions of these mean demand patterns are as follows:

$$\begin{aligned}
&P_1 : \mu_t = 50,\\
&P_2 : \mu_t = 50 + 40\sin(2\pi t/N),\\
&P_3 : \mu_t = 10 + 80t/N,\\
&P_4 : \mu_t = 10 + 80(N + 1 - t)/N,\\
&P_5 : \mu_t = \begin{cases} 10 + 80t/(N/3) \quad \text{for} \quad t \leq N/3 \\ 90 \quad \text{for} \quad N/3 < t < 2N/3 \\ 10 + 80(N - t)/(N/3) \quad \text{for} \quad 2N/3 \leq t. \end{cases}
\end{aligned}$$

In all these patterns the average period demand is 50. The mean demand patterns are used in the $r_t\mu_t$ process, where r_t is a $[0.4, 1.6]$-uniform random variable, to generate instance specific demand patterns.

The demand in each period t (d_t) is a normally distributed random variable with a mean $r_t\mu_t$ and a coefficient of variation 0.25 (i.e., the standard deviation is $0.25r_t\mu_t$) for $t = 1, \ldots, N$. The inventory holding cost is fixed at 1 ($h = 1$) and the ordering cost (a) is taken as a uniform random variable in the range [75,2000].

In the first step of the experiment, random instances are generated and solved using the "relaxed-MIP" model. For a given planning horizon length and a mean demand pattern, the instance generation is repeated until 10 infeasible instances are observed or the total instance number is 1,000,000. Table 1 gives the total number of generated random instances with respect to mean demand patterns. In the table, "–" denotes no infeasibility observed in the generated 1,000,000 random instances.

Table 1: Total number of generated random instances.

	P_1	P_2	P_3	P_4	P_5
$N = 30$	–	124,101	238,794	304,239	49,279
$N = 40$	–	104,686	134,702	172,369	54,918
$N = 50$	–	62,780	223,889	151,011	22,251
$N = 60$	–	53,086	246,614	128,831	31,400

From Table 1, we see that, for the given problem parameters, the relaxed model always yields optimal solution under the "stationary" mean demand pattern, even though the stationarity is distorted by a considerable add-on noise. The relatively dynamic patterns, P_2 and P_5, are more inclined to produce infeasible instances compared to more stable ones, P_3 and P_4. It is expected that as N gets bigger less instance is needed to observe an infeasibility. This is observed in P_2 and P_4, whereas P_3 and P_5 present mixed cases. All the instance solution times are at the centisecond level and can be taken as zero for any practical purpose.

Excluding P_1, for which "relaxed-MIP" yields no infeasible instance, a total of 2,102,950 random instances are generated to produce 10 infeasible instances for each of N and P_i pairs ($N = 30, 40, 50, 60$; $i = 2, 3, 4, 5$) giving a test set of 160 instances. This statistic corresponds to obtaining the optimal solution using solely the relaxed-MIP model with a probability of 99.99%.

In the second step of the experiment, these rare infeasible test instances are solved to optimality using the state-of-the-art mathematical programming solver CPLEX 11.2 with default settings, as well as a Java implementation of our computational method presented in §4. Tests are performed on a 2.0GHz CPU, 32-bit machine. The solution statistics, including the total number of search nodes visited, the solution time, the percentage optimality gap if the search has not been terminated in the allowed time (in our case the limit is 1 hour) and the tightness of the lower and upper bounds at the root node, are listed in Table 2 (for $N = 30, 40$) and Table 3 (for $N = 50, 60$). The formal definitions of the statistics are

$\%\Delta =$	the percentage gap between the best-so-far (BSF) and the corresponding lower bound (LB) for the MIP at the epoch of termination, $100(BSF - LB)/LB$ provided that the solver cannot find the optimal within the time limit of 1 hour;
$\%\Delta_{LB} =$	initial percentage gap between the lower bound ($LB_{rn} = z_r^*$) and the upper bound (UB_{rn}) at the root node before the search starts, $100(UB_{rn} - LB_{rn})/LB_{rn}$;
$\%\Delta_{UB} =$	the percentage gap between the upper bound at the root node and the best-so-far at the epoch of termination for our computational method, $100(UB_{rn} - BSF)/BSF$.

Using our method all instances are solved to optimality without any exception, longest taking 30 secs, most of the time taking less than a second.[5] In the given solution time limit of 1 hour, using MIP, the search procedure terminated with a proven optimal solution in only 84 out of 160 test instances. For $N = 30$ all the instances (between #1 − #40) are solved to optimality. The

[5]Since our method finds the optimal for all the instances considered, $\%\Delta_{UB}$ corresponds to the optimality gap of the upper bound at the root node in Tables 2 and 3.

average solution time is 12.5 secs. For $N = 40$, 39 out of 40 instances (between $\#41 - \#80$) are solved with an average solution time of 529.4 secs. For $N = 50$, only 5 out of 40 (between $\#81 - \#120$) are solved, with an average solution time 1631.0 secs. And finally for $N = 60$ (between $\#121 - \#160$) none of the 40 instances could be solved in the allowed solution time limit of one hour. Mean demand pattern-wise, the solution time performance does not vary much, although P_5 (life-cycle) performs slightly better than the rest. P_5 average for $N = 30$ ($N = 40$) is 7.5 (178.5) secs; P_2, 13.0 (529.7); P_3, 16.8 (763.9); P_4, 12.5 (669.0).

Regarding the search effort, comments similar to the solution time performance can be made. Using MIP, the number of nodes visited during search increases with the number of periods in the planning horizon. For $N = 30$ the average number of search nodes is 20,895, whereas this figure is 636,028 and 1,117,740 for $N = 40$ and $N = 50$, respectively. However, for the same instances, our search procedure enhanced with tight lower and upper bounds requires to visit only a small set of nodes giving an average of 241.

It is important to note that the MIP model does not scale-up well, as all $N = 30$ instances are solved maximum taking 38 secs, but none of the $N = 60$ instances could be solved in 1 hour. Our computational method, however, for these rare infeasible instances presents no shortcoming in scaling-up; average solution times are 0.6, 2.2, 2.4 and 5.4 secs for $N = 30, 40, 50, 60$, respectively.

An investigation of the optimality gap for MIP (the column under $\%\Delta$) shows that for $N = 60$ even after the search is on for 1 hour the average optimality gap is 6.94%. This figure tells us that the relaxation used in the search is not strong enough to prune the search space effectively, hence, even more solution time is dedicated, the optimality could not be proven.

The tightness of the bounds provided in our method can be gauged by examining the columns under the headings $\%\Delta_{LB}$ and $\%\Delta_{UB}$. These figures are calculated at the top of the search tree (i.e., at the root node). The average gaps are $\overline{\Delta}_{LB} = 0.05\%$ and $\overline{\Delta}_{UB} = 0.02\%$, with worst case performances of 0.28% and 0.19%. The conventional LP-relaxation, on the other hand, provides an alternative way of producing lower bounds in B&B. We conducted an ex-post analysis and computed the gap when LP-relaxation provides the LB and the optimal objective function value provides the UB. In this case, among all 160 scenarios considered, the average gap is 66.76%, with a maximum of 71.39% and a minimum of 60.98%. These results clearly demonstrate the effectiveness of the generated lower and upper bounds, which prune the search space aggressively and yield a small search tree, using our method. It is also interesting to check the number of instances in which $\Delta_{UB} = 0.0$. In 114 out of 160 instances the infeasible solution rectified at the root node actually provides the global optimum, but search could be required to prove its optimality (see #6 as an example).

The above results clearly demonstrate the effectiveness and the computational efficiency of the relaxation method proposed in this paper. On the extensive test instances we used this new method proved that it dominates the MIP solution approach and can handle a real-life size non-stationary (R, S) policy parameter optimization problem in trivial time. The effectiveness of the proposed method hinges on three novelties: (*i*) the proposed relaxation is computationally efficient and yields an optimal solution most of the time (99.99% of the time in our experiments), (*ii*) if the relaxation produces an infeasible solution, this solution can be used as a tight lower bound during search (the average gap is 0.05% in our case), and also (*iii*) this infeasible solution can be modified easily to obtain a feasible solution, which is an upper bound for the optimal solution (the average gap is 0.02% in our case). Due to the tightness of the upper bound, one can even terminate without searching for optimality and still has a close-to-optimal solution.

Note that the efficiency of our computational method heavily depends on the tightness of the upper and lower bounds generated. If this condition is not satisfied (i.e., the initial percentage gap between the lower and upper bound at the root node, $\%\Delta_{LB}$, is high), then the performance of the method is expected to degrade. As an illustration, consider a setting where a high demand period is followed by low demand periods. Such a situation leads to an infeasibility and a loose lower bound. Take the following instance: $N = 3$, $a = 200$, $h = 1$, $\alpha = 95\%$, the coefficient of variation for demand is 0.25, and the mean period demands ($\tilde{d}_t$) are 300, 2, 1 for periods 1, 2 and 3, respectively. This instance produces an infeasible solution with two replenishment orders (the first replenishment being in Period 1 covering the demand for Period 1 only, and the second replenishment in Period 2 covering periods 2 and 3) using the relaxed model. In this solution $S_2 = 4$ is substantially below $I_1 = 123$. The root node relaxation yields $LB_{rn} = 526$ and $UB_{rn} = 764$, whereas the optimal solution is 573. The root node optimality gap $\%\Delta_{LB} = 31.15$ is not strong enough to prune the search space effectively and therefore, even though this is a trivial instance to solve, the standard depth-first search explores 4 nodes before termination, if no other enhancement during search is employed. On the other hand, this relaxation is still better than the LP-relaxation which is the standard relaxation method used during B&B in MIP solvers. For this example the lower bound found using the LP-relaxation is 366.

6. Conclusion

This paper provides an efficient computational approach to solve the MIP model developed by Tarim and Kingsman in [3] for calculating the parameters of an (R, S) policy in a finite horizon of N periods with non-stationary stochastic demand. Our approach is based on a relaxation of the original MIP model and the equivalence of this relaxed model to a shortest path problem, which can be solved with an algorithm having a complexity of $O(N^2)$. We have developed an efficient way to check the feasibility of the resulting solution for the original MIP model. Our extensive numerical experiments show that in 99.99% of more than 2 million randomly generated instances the solution for the relaxed model is found to be feasible, hence, the procedure is terminated by finding an optimal solution at this stage. On the other hand, in case of infeasibility, this solution is used to generate a feasible one, which provides an upper bound. A simple branch-and-bound procedure that implements the relaxation approach at each node of the search tree is used to search for an optimal solution. Numerical evidence shows that the bounds are tight, leading to an efficient and fast search procedure.

In summary, we have developed a computational procedure with a numerically demonstrated better performance compared to a commercially available MIP solver. Our method is scalable and makes it possible to solve practically relevant instances in trivial time. There may still be instances where the bounds generated are not tight enough. For such settings, it is recommended that more sophisticated branch-and-bound methods (as those available in the state-of-the-art MIP solvers) are employed, enhanced with the bounds proposed in this paper. Active research is in progress by the authors to address such cases.

References

[1] E. A. Silver and D. F. Pyke and R. Peterson. *Inventory Management and Production Planning and Scheduling*. John Wiley and Sons, New York, 1998.

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

[3] S. A. Tarim and B. G. Kingsman. The Stochastic Dynamic Production/Inventory Lot–Sizing Problem with Service–Level Constraints. *International Journal of Production Economics*, 88:105–119, 2004.

[4] S. A. Tarim and B. M. Smith. Constraint Programming for Computing Non-Stationary (R,S) Inventory Policies. *European Journal of Operational Research*, 189:1004-1021, 2008.

[5] R. Ahuja and J. Orlin and T. Magnanti. *Network Flows: Theory, Algorithms, and Applications*. Prentice Hall, Upper Saddle River, NJ, 1993.

[6] A. M. Geoffrion. Elements of Large-Scale Mathematical Programming Part I: Concepts. *Management Science*, 16:652-675, 1970.

[7] Java. http://java.sun.com/.

Table 2: Solution statistics for $N = 30, 40$

	$N=30$								$N=40$							
		MIP			Our Method					MIP			Our Method			
Demand	#	Nodes	%Δ	secs	Nodes	$\%\Delta_{LB}$	$\%\Delta_{UB}$	secs	#	Nodes	%Δ	secs	Nodes	$\%\Delta_{LB}$	$\%\Delta_{UB}$	secs
$P2$	1	19200	–	12.4	1	0.00	0.00	0.1	41	1026500	–	818.1	79	0.03	0.06	1.4
	2	21900	–	12.2	59	0.12	0.16	0.7	42	646100	–	576.6	1	0.00	0.00	0.2
	3	45800	–	24.5	1	0.00	0.00	0.1	43	147100	–	115.5	79	0.03	0.03	1.2
	4	9000	–	6.7	1	0.00	0.00	0.0	44	249800	–	195.0	79	0.06	0.00	1.2
	5	36100	–	21.8	1	0.00	0.00	0.1	45	702300	–	555.1	103	0.09	0.00	1.2
	6	11500	–	7.6	151	0.08	0.00	1.4	46	561500	–	460.8	459	0.21	0.03	4.3
	7	23400	–	14.4	1	0.00	0.00	0.0	47	440300	–	367.2	199	0.12	0.00	2.3
	8	12100	–	8.6	87	0.16	0.00	0.8	48	851500	–	701.9	1	0.00	0.00	0.1
	9	7400	–	5.3	1	0.00	0.00	0.1	49	1457800	–	1154.6	469	0.09	0.00	5.7
	10	26800	–	16.3	465	0.24	0.04	3.0	50	424400	–	351.9	79	0.16	0.00	1.1
$P3$	11	22700	–	12.1	59	0.04	0.04	0.7	51	922400	–	713.5	79	0.03	0.00	1.4
	12	66500	–	38.4	91	0.08	0.08	0.8	52	5161300	1.04	–	235	0.09	0.00	2.7
	13	28300	–	18.6	107	0.12	0.08	1.0	53	247700	–	236.1	79	0.03	0.00	1.2
	14	30500	–	17.6	59	0.08	0.00	0.7	54	693400	–	583.1	247	0.06	0.00	3.2
	15	17100	–	11.1	59	0.04	0.00	0.7	55	134000	–	117.6	135	0.08	0.00	1.9
	16	22500	–	13.2	1	0.00	0.00	0.1	56	2622200	–	2136.8	79	0.03	0.11	1.3
	17	20400	–	12.4	59	0.04	0.11	0.7	57	937600	–	704.7	147	0.12	0.00	1.9
	18	26900	–	13.0	59	0.08	0.08	0.6	58	409900	–	304.9	79	0.06	0.00	1.2
	19	51700	–	26.7	59	0.04	0.00	0.7	59	866400	–	694.2	1	0.00	0.00	0.1
	20	8400	–	5.4	59	0.04	0.00	0.7	60	1703000	–	1384.0	79	0.06	0.00	1.2
$P4$	21	16800	–	10.7	1	0.00	0.00	0.1	61	991700	–	851.0	153	0.15	0.03	2.2
	22	11600	–	8.4	59	0.04	0.00	0.6	62	197000	–	178.3	79	0.15	0.03	1.4
	23	26600	–	17.0	95	0.04	0.00	0.9	63	911000	–	831.8	995	0.11	0.09	9.0
	24	15400	–	9.6	83	0.28	0.00	1.4	64	732400	–	610.0	109	0.06	0.09	1.5
	25	8000	–	5.6	1	0.00	0.00	0.0	65	152100	–	139.0	79	0.03	0.00	2.0
	26	21700	–	13.2	59	0.04	0.00	0.7	66	366200	–	374.3	97	0.09	0.00	1.6
	27	53300	–	29.6	59	0.04	0.18	0.7	67	1635600	–	1326.3	455	0.14	0.06	5.1
	28	29400	–	17.2	1	0.00	0.00	0.1	68	761800	–	706.3	139	0.03	0.00	1.8
	29	11300	–	8.3	59	0.04	0.07	0.6	69	1323800	–	1159.7	203	0.03	0.03	2.2
	30	7800	–	5.6	87	0.04	0.00	0.9	70	597900	–	513.1	1	0.00	0.00	0.7
$P5$	31	25800	–	13.0	59	0.03	0.07	0.7	71	227600	–	199.8	235	0.05	0.13	2.5
	32	17500	–	9.4	59	0.04	0.00	0.6	72	529600	–	425.2	195	0.08	0.00	2.2
	33	2200	–	1.9	59	0.03	0.07	0.6	73	172300	–	163.3	97	0.05	0.00	1.4
	34	35100	–	19.5	59	0.04	0.00	0.7	74	45600	–	42.9	89	0.08	0.00	1.2
	35	12700	–	8.8	59	0.03	0.00	0.6	75	297700	–	250.9	131	0.05	0.00	1.6
	36	6900	–	4.7	59	0.03	0.00	0.6	76	68600	–	66.3	1061	0.27	0.00	11.4
	37	8800	–	5.8	59	0.07	0.00	0.7	77	147300	–	114.1	151	0.03	0.00	2.0
	38	6400	–	4.4	59	0.03	0.19	0.7	78	318300	–	259.7	239	0.13	0.00	2.4
	39	3000	–	2.6	59	0.07	0.14	0.6	79	29900	–	27.4	79	0.03	0.00	1.2
	40	7300	–	4.9	1	0.00	0.00	0.1	80	254800	–	235.4	1	0.00	0.00	0.1

Table 3: Solution statistics for $N = 50, 60$

	$N = 50$								$N = 60$							
		MIP			Our Method					MIP			Our Method			
Demand	#	Nodes	%Δ	secs	Nodes	$\%\Delta_{LB}$	$\%\Delta_{UB}$	secs	#	Nodes	%Δ	secs	Nodes	$\%\Delta_{LB}$	$\%\Delta_{UB}$	secs
$P2$	81	2560500	4.22	–	1	0.00	0.00	0.3	121	1787500	7.39	–	1539	0.14	0.02	22.9
	82	2302100	3.00	–	50	0.02	0.00	2.8	122	1574200	6.29	–	1	0.00	0.00	0.3
	83	2624900	1.89	–	50	0.05	0.00	2.0	123	1693100	7.53	–	285	0.08	0.00	5.8
	84	2570800	5.89	–	50	0.05	0.07	2.0	124	1962100	6.26	–	741	0.06	0.00	13.3
	85	2249000	3.60	–	140	0.02	0.09	2.4	125	1408800	5.81	–	1	0.00	0.00	0.3
	86	2541500	4.55	–	1	0.00	0.00	0.2	126	2107000	5.75	–	331	0.04	0.00	7.6
	87	2651300	2.61	–	68	0.12	0.00	2.1	127	1499300	4.97	–	119	0.04	0.00	3.6
	88	2329900	4.22	–	95	0.02	0.00	3.2	128	1605800	5.12	–	259	0.04	0.00	5.6
	89	2323100	4.18	–	1	0.00	0.00	0.2	129	1742700	4.46	–	221	0.04	0.02	6.4
	90	2442800	3.42	–	50	0.09	0.05	2.0	130	1912800	6.40	–	1	0.00	0.00	0.3
$P3$	91	3752300	4.51	–	1	0.00	0.00	0.2	131	2341600	9.92	–	1	0.00	0.00	0.3
	92	2997800	6.99	–	50	0.02	0.00	3.4	132	2424800	10.77	–	197	0.08	0.00	4.6
	93	2747100	4.35	–	50	0.02	0.09	2.4	133	2251200	8.87	–	425	0.04	0.00	9.0
	94	2915200	3.36	–	1	0.00	0.00	0.2	134	2290900	6.85	–	1	0.00	0.00	0.3
	95	2792000	4.56	–	1	0.00	0.00	0.2	135	2119000	8.20	–	1	0.00	0.00	0.3
	96	2829500	6.62	–	1	0.00	0.00	0.1	136	2271400	8.13	–	1	0.00	0.00	0.3
	97	3074500	2.89	–	50	0.07	0.00	2.4	137	2058200	7.56	–	1	0.00	0.00	0.3
	98	2770200	3.59	–	68	0.02	0.00	3.3	138	2294200	7.30	–	1	0.00	0.00	0.3
	99	2845500	2.24	–	1	0.00	0.00	0.2	139	1763200	7.13	–	371	0.06	0.00	7.9
	100	3569400	6.02	–	73	0.09	0.02	3.9	140	1721200	7.50	–	119	0.02	0.04	4.0
$P4$	101	2231800	3.55	–	136	0.05	0.00	3.0	141	2027500	6.60	–	1	0.00	0.00	0.3
	102	2339800	3.69	–	50	0.05	0.02	2.1	142	1950800	6.52	–	1693	0.09	0.00	30.1
	103	1294100	–	2069.0	101	0.11	0.00	4.4	143	2087600	8.02	–	811	0.08	0.00	15.3
	104	2341100	4.28	–	164	0.12	0.00	7.8	144	2173100	7.88	–	669	0.08	0.00	11.9
	105	2403900	0.48	–	93	0.09	0.00	3.1	145	2013800	9.48	–	119	0.04	0.04	4.8
	106	2463500	1.71	–	50	0.04	0.00	2.9	146	1757500	5.86	–	339	0.10	0.04	7.2
	107	2540200	3.45	–	50	0.02	0.02	2.5	147	2174600	7.16	–	119	0.02	0.04	5.1
	108	2554800	3.50	–	82	0.05	0.09	3.4	148	1788200	7.59	–	1	0.00	0.00	0.3
	109	2329100	2.49	–	83	0.07	0.11	3.9	149	2046700	9.20	–	119	0.02	0.02	3.7
	110	2588600	5.08	–	216	0.07	0.00	5.0	150	1850700	7.72	–	119	0.04	0.02	4.2
$P5$	111	2802200	1.79	–	1	0.00	0.00	0.2	151	2083000	6.42	–	119	0.02	0.00	4.0
	112	2493700	0.19	–	95	0.04	0.00	2.9	152	2073300	6.09	–	227	0.02	0.00	5.3
	113	2435300	0.96	–	50	0.04	0.08	2.1	153	2038900	3.98	–	1	0.00	0.00	0.3
	114	3040700	2.37	–	1	0.00	0.00	0.2	154	1919300	7.65	–	125	0.02	0.00	3.2
	115	1692900	–	2310.3	79	0.06	0.06	3.2	155	2098800	5.83	–	207	0.03	0.02	5.0
	116	440700	–	669.6	50	0.04	0.00	1.9	156	1976300	5.84	–	231	0.05	0.03	4.5
	117	3125400	3.74	–	105	0.11	0.00	3.4	157	2294200	4.95	–	303	0.09	0.00	6.4
	118	937000	–	1375.7	96	0.10	0.00	2.5	158	1754900	5.19	–	189	0.02	0.00	4.7
	119	2892500	3.54	–	1	0.00	0.00	0.2	159	2007400	6.31	–	1	0.00	0.00	0.3
	120	1224000	–	1730.3	327	0.12	0.00	7.5	160	1891000	6.90	–	231	0.03	0.02	5.9