

A hybrid heuristic for capacitated three-level lot-sizing and replenishment problems with a distribution structure

Jesus O. Cunha ^{*} Geraldo R. Mateus [†] Rafael A. Melo ^{‡§}

December 24, 2021

Abstract

In this paper, we consider the capacitated three-level lot-sizing and replenishment problem with a distribution structure (3LSPD-C), recently proposed in the literature. In 3LSPD-C, a single production plant delivers items to the warehouses from where they are distributed to their corresponding retailers. There is a capacity on the total amount the plant can produce in each period, whereas there are no capacities on transportation. The goal of this optimization problem consists in determining an integrated three-echelon production and distribution plan minimizing the total cost, which is composed of fixed costs for production and transportation setups as well as variable inventory holding costs. Additionally, we consider a generalization of the problem which also establishes storage capacities (or inventory bounds) on the warehouses and/or retailers, given the importance of such characteristics in practical industrial and commercial environments. Such extension is denoted generalized capacitated three-level lot-sizing and replenishment problem with a distribution structure (G3LSPD-C). We propose a hybrid mixed integer programming (MIP) heuristic consisting of a relax-and-fix approach to generate initial feasible solutions and a fix-and-optimize improvement procedure grounded on varying-size neighborhoods to obtain high-quality solutions. Computational experiments are performed to analyze the potential cost reductions achieved using the new hybrid heuristic when compared with a state-of-the-art MIP formulation. The results show that the proposed hybrid heuristic can match or improve the solution quality obtained by the MIP formulation for the majority of the 3LSPD-C instances within a specified time limit. Additionally, such superior behavior remains valid when the approaches are applied to the more general G3LSPD-C. Furthermore, we investigate the economic impacts of the storage capacities and how they affect the performance of our newly proposed hybrid heuristic.

Keywords: supply chain management; logistics; three-echelon supply chain; MIP heuristic; fix-and-optimize; storage capacity.

1 Introduction

Supply chain optimization plays a key role in the competitiveness of enterprises nowadays. An important matter that can considerably increase effectiveness and reduce costs is an integrated optimization of the production and transportation plannings, a common need in companies with decentralized markets. In this direction, we consider the capacitated three-level lot-sizing and replenishment problem with a distribution structure (3LSPD-C), introduced in Gruson, Bazrafshan, Cordeau, and Jans (2019). In this three-echelon

^{*}Departamento de Estatística e Matemática Aplicada, Universidade Federal do Ceará, Fortaleza, CE 60440-900, Brazil. (jesus.ossian@dema.ufc.br)

[†]Departamento de Ciência da Computação, Universidade Federal de Minas Gerais, Av. Antônio Carlos 6627, Belo Horizonte, MG 31270-010, Brazil. (mateus@dcc.ufmg.br)

[‡]Institute of Computing, Universidade Federal da Bahia, Salvador, BA 40170-115, Brazil. (rafael.melo@ufba.br)

[§]Part of this work was developed while the author was a visiting scholar at the Departamento de Ciência da Computação, Universidade Federal de Minas Gerais, Brazil.

supply chain optimization problem, a single production plant produces items to replenish multiple warehouses, from where they are sent to fulfill the time-varying deterministic demands of several retailers over a finite planning horizon. Additionally, there is a restriction on the quantity of items the plant can produce in each period. The goal consists in determining an integrated production, inventory, and distribution plan which minimizes the total costs, consisting of fixed production and transportation setups together with per unit inventory holding costs. Figure 1 exemplifies the layout of a three-level lot-sizing and replenishment problem with a distribution structure. Notice that, in practical industrial and commercial settings, it is very natural that the storage facilities have capacity restrictions. In this regard, we also introduce the generalized capacitated three-level lot-sizing and replenishment problem with a distribution structure (G3LSPD-C), in which capacities on storage can occur at the warehouses and/or retailers.

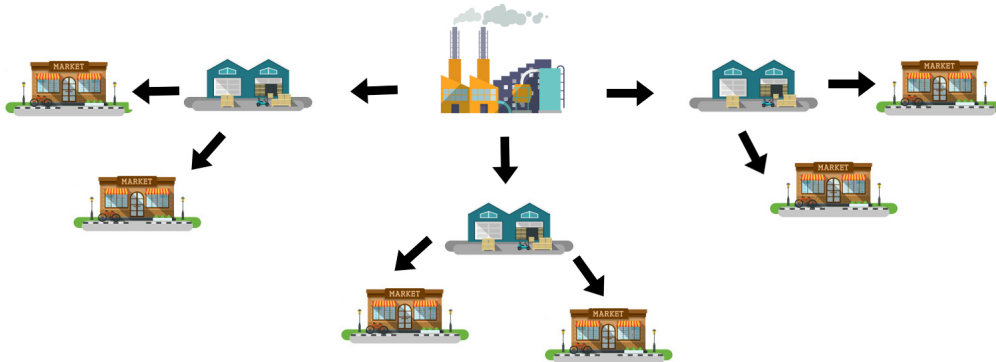


Figure 1: Example of a three-level supply chain with a distribution structure.

Several problems involving integrated production and transportation have been studied in the literature. Among these, we can highlight multi-level lot-sizing problems (Eppen & Martin, 1987; Slama, Ben-Ammar, Dolgui, & Masmoudi, 2020), one-warehouse multi-retailer problems (Solyali & Süral, 2012; Cunha & Melo, 2016) and some of its generalizations (Park, 2005; Melo & Wolsey, 2012; Li & Hai, 2019), problems in the context of production and transportation by third-party logistic companies (Stecke & Zhao, 2007; Melo & Wolsey, 2010), and supply chain optimization problems involving multiple plants in which items can be transferred between the plants (Tanksale & Jha, 2017; Carvalho & Nascimento, 2018; Cunha, Kramer, & Melo, 2021).

Recently, the optimization of three-echelon supply chains has received considerable attention. Cárdenas-Barrón and Treviño-Garza (2014) propose an integer programming formulation for a multi-product three-level supply chain problem. Zhang and Song (2018) present an integer programming approach to tackle a practical three-level problem arising in a case study at Danone Waters China Division. Gruson et al. (2019) introduce the uncapacitated and capacitated three-level lot-sizing and replenishment problem with a distribution structure (3LSPD-U and 3LSPD-C, respectively). The authors propose and compare, both theoretically and computationally, several formulations for both variants. Gruson, Cordeau, and Jans (2020) describe a method based on Benders decomposition for a stochastic three-level lot-sizing and replenishment problem with a distribution structure. Cunha and Melo (2021) tackle 3LSPD-U and propose valid inequalities, pre-processing, and a multi-start randomized bottom-up dynamic programming-based heuristic. The authors managed to solve to optimality all the benchmark instances for the problem by combining the proposed heuristic with a preprocessed multicommodity formulation.

Production planning problems with storage capacities have appeared in several contexts. It is noticeable, however, that most of the works related to storage capacities are very recent when compared to those considering production capacities (Bitran & Yanasse, 1982; Trigeiro, Thomas, & McClain, 1989; Buschkühl, Sahling, Helber, & Tempelmeier, 2010; Vincent, Duhamel, Ren, & Tchernev, 2020; Tavaghof-Gigloo & Min-

ner, 2020). Akbalik, Penz, and Rapine (2015a) study capacitated lot-sizing problems with storage capacities. The authors propose polynomial-time algorithms for the single-item variant of the problem and show NP-hardness results for the multi-item variation. Akbalik, Penz, and Rapine (2015b) establish NP-hardness results for the multi-item uncapacitated lot-sizing with storage capacities under different scenarios of costs and capacities. Brahimi, Absi, Dauzère-Pérès, and Kedad-Sidhoum (2015) propose integer programming formulations and a Lagrangian relaxation-based heuristic for an uncapacitated two-level lot-sizing problem with storage capacities. Melo and Ribeiro (2017) describe mixed integer programming (MIP) formulations and MIP heuristics for the multi-item uncapacitated lot-sizing problem with storage capacities. Jing and Mu (2020) tackle a lot-sizing problem with perishable items, product substitution, and storage capacities. The authors provide a dynamic programming algorithm for the problem and analyze the effects of the different characteristics on the costs of the solutions.

1.1 Main contributions and organization

The main contribution of this paper is an effective hybrid heuristic for the capacitated three-level lot-sizing and replenishment problem with a distribution structure (3LSPD-C), which to the best of our knowledge is the first heuristic for the problem. The proposed hybrid heuristic combines a relax-and-fix approach with a fix-and-optimize improvement procedure which relies on varying-size neighborhoods to improve the local search mechanism and overcome local optima solutions. Computational experiments evidence the effectiveness of the heuristic, as it outperforms a state-of-the-art formulation available for the problem. The heuristic is flexible and can be easily extended to tackle more general variants of the problem, such as the generalized capacitated three-level lot-sizing and replenishment problem with a distribution structure (G3LSPD-C) that we introduce in our work. G3LSPD-C considers storage capacities on the warehouses and/or retailers, which are very relevant characteristics as they often arise in commercial and industrial settings. Last but not least, we analyze the economic impacts (in terms of the increase in the solutions' costs) of storage capacities on the warehouses and retailers.

The remainder of this paper is organized as follows. Section 2 formally defines the considered capacitated three-level lot-sizing and replenishment problems with a distribution structure. Section 3 describes the proposed hybrid heuristic. Section 4 summarizes the performed computational experiments. Section 5 concludes the paper with final remarks.

2 Capacitated three-level lot-sizing and replenishment problems with a distribution structure

In this section, we define two capacitated three-level lot-sizing and replenishment problems with a distribution structure. Section 2.1 formalizes the capacitated three-level lot-sizing and replenishment problem with a distribution structure (3LSPD-C) studied in Gruson et al. (2019). An available standard mixed integer programming formulation for the problem is also described. Section 2.2 introduces the generalized capacitated three-level lot-sizing and replenishment problem with a distribution structure (G3LSPD-C), which extends 3LSPD-C by considering storage capacities on the warehouses and/or retailers.

2.1 The capacitated three-level lot-sizing and replenishment problem with a distribution structure

3LSPD-C can be formally stated as follows. There is a set $F = P \cup W \cup R$ of facilities composed of a single plant, $P = \{p\}$, a set W of warehouses, and a set R of retailers. Each retailer $r \in R$ has a time-varying deterministic demand d_t^r for a single item for each period $t \in T$. There is a capacity C_t^p on the amount that can be produced at the production plant in period $t \in T$. Each warehouse $w \in W$ attends a set of retailers $\delta(w)$ and each retailer $r \in R$ has a unique associated warehouse $\delta_w(r)$ from which it receives its items. Each facility $i \in F$ has corresponding setup costs sc_t^i and holding costs hc_t^i for each period $t \in T$. The

problem consists in encountering a feasible production/distribution plan which attends to all the demands without backlogging while minimizing the total cost. Denote the demand in period $t \in T$ of the plant as $d_t^p = \sum_{r \in R} d_t^r$ and that of a warehouse $w \in W$ as $d_t^w = \sum_{r \in \delta(w)} d_t^r$. In addition, let the cumulative demand of facility $i \in F$ for periods $1 \leq t \leq k \leq |T|$ be $d_{tk}^i = \sum_{l=t}^k d_l^i$.

Consider the decision variable x_t^i to be the amount produced in the production plant in period $t \in T$ if $i = p$ and to be equal to the amount transported to facility i from its predecessor in period t if $i \in W \cup R$. Besides, let the variable s_t^i represent the amount of inventory in facility $i \in F$ in period $t \in T$. Additionally, define variable y_t^i for $i \in F$ and $t \in T$ to be equal to one if $x_t^i > 0$ and to be equal to zero otherwise. 3LSPD-C can be cast as the following standard MIP formulation (Gruson et al., 2019):

$$(3LSPD-C) \quad \min \sum_{t \in T} \left(\sum_{i \in F} sc_t^i y_t^i + \sum_{i \in F} hc_t^i s_t^i \right) \quad (1)$$

$$s_{t-1}^i + x_t^i = \sum_{j \in \delta(i)} x_t^j + s_t^i, \quad \text{for } i \in P \cup W, t \in T, \quad (2)$$

$$s_{t-1}^r + x_t^r = d_t^r + s_t^r, \quad \text{for } r \in R, t \in T, \quad (3)$$

$$x_t^i \leq d_{t|T}^i y_t^i, \quad \text{for } i \in W \cup R, t \in T, \quad (4)$$

$$x_t^p \leq \min\{C_t^p, d_{t|T}^p\} y_t^p, \quad \text{for } t \in T, \quad (5)$$

$$x_t^i, s_t^i \geq 0, \quad \text{for } i \in F, t \in T, \quad (6)$$

$$y_t^i \in \{0, 1\}, \quad \text{for } i \in F, t \in T. \quad (7)$$

The objective function (1) minimizes the total cost, composed of setup and storage costs. Constraints (2) and (3) define inventory balance constraints for each of the facilities. Constraints (4) and (5) are setup enforcing constraints, with (5) also guaranteeing the plant's production capacities are not violated. Constraints (6) and (7) define, respectively, nonnegativity and integrality requirements on the variables.

2.2 The generalized capacitated three-level lot-sizing and replenishment problem with a distribution structure

G3LSPD-C generalizes 3LSPD-C by considering the fact that the warehouses and/or retailers can have a limit on the amount that can be held in storage in any given period. Define \hat{C}_t^i to be the storage capacity at facility $i \in W \cup R$ in period $t \in T$.

G3LSPD-C can be formulated as the following MIP:

$$(G3LSPD-C) \quad \min \sum_{t \in T} \left(\sum_{i \in F} sc_t^i y_t^i + \sum_{i \in F} hc_t^i s_t^i \right) \quad (8)$$

$$(2) - (7),$$

$$s_t^i \leq \min\{\hat{C}_t^i, d_{t|T}^i\}, \quad \text{for } i \in W \cup R, t \in T. \quad (9)$$

The objective function (8) is the same as (1). Constraints (9) ensure the total inventory at each facility $i \in W \cup R$ at the end of each period does not exceed the storage capacity.

3 The hybrid heuristic

In this section, we detail the hybrid heuristic combining a relax-and-fix approach with a fix-and-optimize improvement procedure, both based on rolling horizon schemes. This hybrid heuristic is motivated by the fact that MIP heuristics using these concepts have been successfully applied to many supply chain optimization problems (Helber & Sahling, 2010; Chen, 2015; Toledo, da Silva Arantes, Hossomi, França, & Akartunali,

2015; Melo & Ribeiro, 2017; Cunha, Kramer, & Melo, 2019). The proposed heuristic aims at providing good quality solutions while allowing flexibility to be easily adapted to variants of the problem. Section 3.1 describes the relax-and-fix approach to obtain an initial feasible solution. Section 3.2 explains the fix-and-optimize improvement procedure. The complete hybrid heuristic is presented in Section 3.3.

In what follows, we say that a variable belongs to an interval $[\alpha, \beta]$ if it corresponds to a period $t \in [\alpha, \beta]$.

3.1 Relax-and-fix approach

The proposed relax-and-fix approach consists of progressively building a solution by solving a series of subproblems in which a subset of the variables is fixed and another subset has its integrality requirements relaxed. It follows a rolling horizon mechanism, which works as follows. Consider integer values α^{RF} and β^{RF} to determine three (possibly empty) intervals, corresponding to how the integer variables are treated when solving the mixed integer programs. The interval $[1, \alpha^{RF} - 1]$ corresponds to variables to be fixed, the interval $[\alpha^{RF}, \beta^{RF}]$ determines the variables to be optimized with the integrality preserved, and the interval $[\beta^{RF} + 1, |T|]$ defines the interval for which the integrality requirements on the variables are relaxed. The approach initiates with $[\alpha^{RF}, \beta^{RF}]$ starting at the beginning of the planning horizon, and it progressively shifts towards the end of the horizon. The relax-and-fix approach is described in Algorithm 1. It takes as inputs a mixed integer programming formulation FORM, an integer k^{RF} defining the size of the horizon of the subproblems, an integer k_f^{RF} representing the size of the interval to be fixed after solving a subproblem, and the maximum allowed time $maxt^{RF}$ for the approach.

Algorithm 1: Relax-and-Fix (FORM, k^{RF} , k_f^{RF} , $maxt^{RF}$)

```

1  $\Phi \leftarrow \emptyset$ ;
2  $\alpha^{RF} \leftarrow 1$ ,  $\beta^{RF} \leftarrow \alpha^{RF} + k^{RF} - 1$ ;
3  $maxt_{sub}^{RF} \leftarrow \frac{maxt^{RF}}{\lceil |T|/k_{fix}^{RF} \rceil}$ ;
4 while complete horizon was not yet treated do
5    $(\hat{s}, \hat{x}, \hat{y}) \leftarrow \text{Solve MIP}_{RF}(\text{FORM}, \alpha^{RF}, \beta^{RF}, \Phi, maxt_{sub}^{RF})$ ;
6   Update  $\Phi$  with the fixings for the variables in the interval  $[\alpha^{RF}, \alpha^{RF} + k^{RF} - 1]$  according to the
   values of  $\hat{y}$ ;
7    $\alpha^{RF} \leftarrow \alpha^{RF} + k^{RF}$ ,  $\beta^{RF} \leftarrow \min\{\alpha^{RF} + k_{fix}^{RF} - 1, |T|\}$ ;
8 return solution  $S = (\hat{s}, \hat{x}, \hat{y})$  determined by the fixings in  $\Phi$ ,  $elapt^{RF}$ ;

```

The set of fixings imposed on the integer y variables, Φ , which defines a partial solution being built, is initialized as empty in line 1. The values of α^{RF} and β^{RF} are initialized in line 2. The time limit for each subproblem, $maxt_{sub}^{RF}$, is determined in line 3. The loop of lines 4-7 is executed while the complete horizon was not yet treated, i.e., a complete integer feasible solution was not yet obtained. Line 5 solves $\text{MIP}_{RF}(\text{FORM}, \alpha^{RF}, \beta^{RF}, \Phi, maxt_{sub}^{RF})$, in which the integer variables related to the periods in the interval $[1, \alpha^{RF} - 1]$ are fixed according to the values determined in the previous rounds, which are defined by Φ , those related to the periods in the interval $[\alpha^{RF}, \beta^{RF}]$ must preserve the integrality, and the integrality constraints are relaxed for those related to the interval $[\beta^{RF} + 1, |T|]$. In line 6, Φ is updated according to the return of the call to MIP_{RF} in line 5. We consider two possible strategies for determining the fixing of the variables. The first strategy (S1) consists of adding to S only the fixings of the variables which assume value 1, which implies that the MIP subproblems to be solved do not necessarily have all the variables in the interval $[1, \alpha^{RF} - 1]$ fixed. The second one (S2) consists of inserting into S the fixings corresponding to all the y variables in the interval. The values of α^{RF} and β^{RF} are updated in line 7. The complete solution S determined according to the fixings in Φ and the elapsed time to run the approach are returned in line 8. Note that S corresponds to the solution obtained in the last call to MIP_{RF} in line 5.

Observe that strategy S1 ensures a feasible solution can be encountered at the end of the execution of Algorithm 1 whenever the instance is feasible and enough time is available. This is a consequence of the fact that there always exists a feasible solution to each call to MIP_{RF} . On the other hand, note that strategy S2 may lead to infeasible subproblems and, thus, does not ensure obtaining a feasible solution at the end

of Algorithm 1. We remark, though, that such infeasibilities are more likely to arise in situations with very strict capacities and/or very high demands concentrated in a few periods.

3.2 Fix-and-optimize improvement procedure

We now propose a multi-round fix-and-optimize improvement procedure with varying-size neighborhoods. The main intuition behind the varying-size neighborhoods is to improve the search mechanism using the concepts of variable neighborhood search (Hansen, Mladenović, Brimberg, & Pérez, 2019) and to attempt turning the approach less parameter-sensitive. It follows a rolling-horizon approach which can restart from the beginning of the horizon, and its main idea works as follows. Consider values α^{FO} and β^{FO} defining an interval $[\alpha^{FO}, \beta^{FO}]$ determining variables which will be optimized. Additionally, define intervals $[1, \alpha^{FO} - 1]$ and $[\beta^{FO} + 1, |T|]$ corresponding to variables which will be fixed according to an available input solution S . Similar to the relax-and-fix approach, the procedure initiates with $[\alpha^{FO}, \beta^{FO}]$ starting at the beginning of the planning horizon, and it progressively shifts towards the end of the horizon. The difference lies in the fact that $[\alpha^{FO}, \beta^{FO}]$ can be restarted at the beginning of the planning horizon and that the size of the intervals can be dynamically changed during the execution. The procedure is described in Algorithm 2. It receives as inputs a MIP formulation FORM, an initial feasible solution S , the minimum size for the optimization horizon $k_{h,\min}^{FO}$, the minimum size for the fixing interval $k_{f,\min}^{FO}$, the length of the increase in the optimization horizon Δ_h^{FO} , the length of the increase in the fixing horizon Δ_f^{FO} , the minimum number of rounds to be performed rnd_{\min}^{FO} , and a time limit $maxt^{FO}$. The algorithm assumes that $k_{h,\min}^{FO} \geq k_{f,\min}^{FO}$ and $\Delta_h^{FO} \geq \Delta_f^{FO}$.

Algorithm 2: Fix-and-Optimize (FORM, S , $k_{h,\min}^{FO}$, $k_{f,\min}^{FO}$, Δ_h^{FO} , Δ_f^{FO} , rnd_{\min}^{FO} , $maxt^{FO}$)

```

1  $S^*, S' \leftarrow S$ ;
2  $k^{FO} \leftarrow k_{h,\min}^{FO}$ ,  $k_f^{FO} \leftarrow k_{f,\min}^{FO}$ ;
3  $maxt_{sub}^{FO} \leftarrow \frac{maxt^{FO}}{rnd_{\min}^{FO} \times \lceil \frac{|T|}{k_{\min}^{FO}} \rceil}$ ;
4 repeat
5    $\alpha^{FO} \leftarrow 1$ ,  $\beta^{FO} \leftarrow \min\{\alpha^{FO} + k^{FO} - 1, |T|\}$ ;
6   while  $elapt^{FO} < maxt^{FO}$  and complete horizon was not yet treated do
7      $S \leftarrow MIP_{FO}(\text{FORM}, \alpha^{FO}, \beta^{FO}, S^*, maxt_{sub}^{FO})$ ;
8     if  $z(S) < z(S^*)$  then
9        $S^* \leftarrow S$ ;
10     $\alpha^{FO} \leftarrow \min\{\alpha^{FO} + k_f^{FO}, |T|\}$ ,  $\beta^{FO} \leftarrow \min\{\alpha^{FO} + k^{FO} - 1, |T|\}$ ;
11    if  $elapt^{FO} < maxt_{sub}^{FO}$  and  $k^{FO} < |T|$  then
12       $maxt_{sub}^{FO} \leftarrow \min\{maxt_{sub}^{FO}, maxt^{FO} - elapt^{FO}\}$ ;
13  if  $z(S^*) < z(S')$  then
14     $S' \leftarrow S^*$ ;
15  else if  $k^{FO} < |T|$  then
16     $k^{FO} \leftarrow k^{FO} + \Delta_h^{FO}$ ;
17     $k_f^{FO} \leftarrow k_f^{FO} + \Delta_f^{FO}$ ;
18    if  $k^{FO} \geq |T|$  then
19       $maxt_{sub}^{FO} \leftarrow maxt^{FO} - elapt^{FO}$ ;
20  else
21    stop;
22 until stopping criterion is met;
23 return best obtained solution  $S^*$ ;

```

Initially, line 1 defines the input solution S as both the best and previously best-known solutions (S^* and S' , respectively). Line 2 initializes the sizes of the optimization and fixing intervals (k^{FO} and k_f^{FO} ,

correspondingly). Line 3 sets the maximum time for solving each subproblem, which ensures that at least rnd_{\min}^{FO} rounds are performed. Each iteration of the repeat loop (lines 4-22) represents one round of the fix-and-optimize procedure. In each round, the algorithm goes through the whole planning horizon in order to solve the corresponding subproblems. Firstly, the values α^{FO} and β^{FO} , defining the interval related to the subproblem to be optimized, are initialized in line 5. The while loop (lines 6-12) represents the iterations performed in each round. The subproblem corresponding to the interval $[\alpha^{FO}, \beta^{FO}]$ and solution S^* is solved in line 7, and in case the current best solution is improved, it is updated in line 9. After that, the interval $[\alpha^{FO}, \beta^{FO}]$ is updated in line 10. Lines 11-12 update the allowed time for solving the subproblems, whenever appropriate. Next, if the current best solution improves over the previous best, the latter is updated (lines 13-14). Otherwise, if the size of the fixing interval did not achieve the size of the horizon, the sizes of the optimization and fixing intervals are updated. In case the size of the optimization interval reached that of the planning horizon, the remaining allowed time is defined as the time for solving the only subproblem to be tackled (lines 18-19). If none of the two previous conditions are satisfied, the approach has reached a stopping criterion. The best-obtained solution S^* is returned in line 23.

We remark that the choices of the control parameters are crucial for the good behavior of the proposed fix-and-optimize improvement procedure. Note that $k_{h,\min}^{FO}$ should define a reasonable size for the subproblems to be solved to optimality or near optimality in the first rounds. Besides, the size $k_{f,\min}^{FO}$ should allow flexibility in the reoptimization of latter periods in a given round. Furthermore, rnd_{\min}^{FO} should be determined so that it allows enough time for solving the subproblems.

3.3 The hybrid heuristic

The hybrid heuristic consists of generating an initial feasible solution using the relax-and-fix approach, described in Section 3.1, followed by a call to the fix-and-optimize improvement procedure, described in Section 3.2. The heuristic is presented in Algorithm 3 and it takes as inputs the parameters corresponding to the calls to Relax-and-Fix (Algorithm 1) and Fix-and-Optimize (Algorithm 2), as well as a time limit $maxt$.

Algorithm 3: Hybrid-Heuristic (FORM, $k^{RF}, k_f^{RF}, maxt^{RF}, k_{h,\min}^{FO}, k_{f,\min}^{FO}, \Delta_h^{FO}, \Delta_f^{FO}, rnd_{\min}^{FO}, maxt$)

- 1 $S, elapt \leftarrow$ Relax-and-Fix (FORM, $k^{RF}, k_f^{RF}, maxt^{RF}$);
 - 2 $S^* \leftarrow$ Fix-and-Optimize (FORM, $S, k_{h,\min}^{FO}, k_{f,\min}^{FO}, \Delta_h^{FO}, \Delta_f^{FO}, rnd_{\min}^{FO}, maxt - elapt$);
 - 3 **return** S^* ;
-

4 Computational experiments

This section reports the computational experiments conducted to evaluate the performance of the proposed hybrid heuristic. All computational experiments were carried out on a machine running under Ubuntu GNU/Linux, with an Intel(R) Core(TM) i5-3740 CPU @ 3.20GHz processor and 8Gb of RAM. The algorithms were coded in Julia v1.6.2, using JuMP v0.21.8. The formulations were solved using Gurobi 9.0.2 with the standard configurations, except the relative optimality tolerance gap, which was set to 10^{-6} .

4.1 Benchmark instances

3LSPD-C

The benchmark instances for 3LSPD-C were introduced in Gruson et al. (2019). All instances have a single plant. For each instance, the number of retailers $|R|$ lies in $\{50, 100, 200\}$, while that of warehouses belongs to $\{5, 10, 15, 20\}$. The demand for each period is randomly defined using a uniform distribution $U[5, 100]$. Fixed setup costs are defined as follows: sc^p is uniformly chosen from $U[30000, 45000]$, sc^w from $U[1500, 4500]$, and sc^r from $U[5, 100]$. The storage costs are defined as $hc^p = 0.25$, $hc^{\delta_w(r)} = 0.5$, and hc^r

is uniformly chosen from $U[0.5, 1]$. Instances are classified as unbalanced (in which a few warehouses serve most of the retailers) or balanced (in which all the warehouses serve nearly the same amount of retailers). Considering a capacity factor $C \in \{1.50, 1.75, 2.00\}$, the time-invariant plant capacity for every period $t \in T$ is defined as $C_t^p = \frac{C}{|T|} \sum_{i \in R} \sum_{t \in T} d_t^i$. We only consider in our experiments the instances with time-invariant fixed setups and dynamic demands. For each different combination of these parameters, there are five instances, giving a total of 360 unbalanced and 360 balanced instances.

G3LSPD-C

For G3LSPD-C, we considered adaptations of the instances for 3LSPD-U and 3LSPD-C (Gruson et al., 2019). Namely, for each instance of 3LSPD-U and 3LSPD-C, we generated instances considering a storage capacity factor $C_s \in \{1.50, 1.75, 2.00\}$ to define time-invariant facility storage capacities $\hat{C}_t^i = \frac{C_s}{|T|} \sum_{t \in T} d_t^i$ for every period $t \in T$. The considered instances either have storage capacities on the warehouses ($i \in W$) or the retailers ($i \in R$), but never on both. Thus, the benchmark set is composed of 2880 unbalanced instances and 2880 balanced ones.

4.2 Considered approaches and settings

The following approaches were compared in our experiments:

- the newly proposed hybrid heuristic (RFFO), presented in Section 3;
- the echelon stock formulation (ES-LS), as described in (Gruson et al., 2019).

The comparison of RFFO with ES-LS is because the latter was shown to be amongst the most effective approaches for 3LSPD-C in Gruson et al. (2019). ES-LS is presented in Appendix A for the sake of completeness.

Preliminary experiments were carried out to determine the parameters for RFFO. They took into consideration 144 randomly selected instances that are not used as part of the final experiments. Firstly, we highlight that RFFO employed the echelon stock formulation ES-LS (Gruson et al., 2019). As a general observation, the fix-and-optimize improvement procedure (FO) was the main ingredient of the heuristic, as it could considerably improve the results of the relax-and-fix approach (RF), even when initial solutions with lower quality were obtained. The preliminary experiments allowed us to notice that the performance of RFFO was not sensitive to the tested parameters for RF (k^{RF} and k_f^{RF}). Besides, it was also not sensitive to the initial parameters of FO ($k_{h,\min}^{FO}$ and $k_{f,\min}^{FO}$) when reasonably small values were used. The results showed that values of $rnd_{\min}^{FO} \in \{2, 3\}$ achieved similar results, which were better than those using other smaller or larger values. At the end, the parameters for the complete experiments were defined as follows: $k^{RF} = 5$, $k_f^{RF} = 3$, $k_{h,\min}^{FO} = 5$, $k_{f,\min}^{FO} = 3$, $\Delta_h^{FO} = 0$, $\Delta_f^{FO} = 1$, and $rnd_{\min}^{FO} = 2$. The time limits for RFFO were set as $maxt = 600$ seconds and $maxt^{RF} = \lceil 0.10 \times maxt \rceil = 60$ seconds.

The formulations were executed with a time limit of 600 seconds to allow a direct comparison with RFFO regarding the quality of the achieved solutions. Besides, for 3LSPD-C they were also executed with a time limit of 3600 seconds to verify whether additional instances could be solved to optimality.

4.3 Results for 3LSPD-C

The results for 3LSPD-C are summarized in Tables 1-4, where Tables 1-2 compare the results of RFFO with those of ES-LS running for 600 seconds, while Tables 3-4 compare our results with those running ES-LS for 3600 seconds. In Tables 1-2, the first three columns identify the instance parameters (param), their corresponding values (value), and the number of instances with the specified characteristics (#inst). The next three columns provide, for ES-LS within the time limit of 600 seconds, the average best solution ($best_{ESLS}$), the average open optimality gap in percent (gap_{ESLS} , calculated for each instance as $100 \times \frac{best_{ESLS} - bbound}{best_{ESLS}}$, where $bbound$ denotes the best dual bound achieved by the solver at the end of the time limit), and the number of instances solved to optimality (#opt). The next eight columns show the average best solution

achieved by RF with a time limit of 60 seconds (best_{RF}), the average best solution obtained at the end of RFFO within 600 seconds (best_{RFFO}), the average open optimality gap in percent obtained by RFFO (gap_{RFFO} , given for each instance by $100 \times \frac{\text{best}_{RFFO} - \text{bbound}}{\text{best}_{RFFO}}$), the average gain in percent achieved by FO over RF (impr_{RF}^{FO} , calculated for each instance as $100 \times \frac{\text{best}_{RF} - \text{best}_{RFFO}}{\text{best}_{RF}}$), the average gain in percent achieved by RFFO over ES-LS ($\text{impr}_{ESLS}^{RFFO}$, calculated for each instance as $100 \times \frac{\text{best}_{ESLS} - \text{best}_{RFFO}}{\text{best}_{ESLS}}$), the number of instances for which RFFO achieved a solution at least as good as that of ES-LS ($\leq \text{best}_{ESLS}$), the number of instances for which RFFO obtained a solution which strictly improves over that encountered by ES-LS ($< \text{best}_{ESLS}$), and the average number of rounds performed by FO (rnd_{FO}). The best average solutions in each line are highlighted in bold. Tables 3-4 present a subset of these just described columns, with the difference that those corresponding to ES-LS report information for the formulation with a time limit of 3600 seconds.

Tables 1-2 show the robustness of RFFO, as it obtained lower average best values than ES-LS running with a time limit of 600 seconds for all the instance configurations. RFFO achieved overall average improvements over ES-LS of 3.07% and 2.95% for the unbalanced and balanced instances, respectively. One can observe that the improvements reached values as high as 7.33% and 7.93% for the instances with a larger number of retailers ($|R| = 200$). It should also be noticed that the gaps of RFFO are remarkably lower than those of ES-LS. Additionally, the number of instances for which RFFO at least matched the best solutions of ES-LS is remarkably high, with an overall value of 303 and 285 out of 360 instances (respectively, 84.17% and 79.17%). Besides, FO allowed considerable improvements over the solutions obtained by RF, especially for the larger instances with $|T| = 30$ and those with $|R| = 200$. Furthermore, it can be noticed that the parameter that influenced the most the average number of rounds performed by FO for both unbalanced and balanced instances was the number of retailers $|R|$. When we compare the results for the unbalanced instances (Table 1) with those for the balanced ones (Table 2), we can see that the behaviors of ES-LS and RFFO do not change much when we analyze the achieved average gaps. In this case, though, it is noteworthy that RFFO encountered a larger number of solutions at least matching those of ES-LS for the unbalanced instances, whereas a larger number of strictly improving solutions was obtained for the balanced ones.

The plot in Figure 2 shows the improvements of RFFO over ES-LS for all the 720 instances (both balanced and unbalanced). It can be seen in the figure that the positive improvements of RFFO over ES-LS can be very significant as they reach close to 30% in certain cases. On the other hand, the improvements of ES-LS over RFFO, determined by the negative values, are negligible.

Tables 3-4 show that even running ES-LS for a much larger time (3600 seconds), RFFO still obtained good comparative numbers even though it was executed for much less time (600 seconds). Note that RFFO achieved lower average best values for all configurations except for the smaller instances with $|T| = 15$ and those with $R = 50$. Besides, remark that the average open optimality gaps achieved by RFFO are reasonably low (1.24% and 1.30% for the unbalanced and balanced instances, respectively), and these values reached a maximum average of 2.39% for $|T| = 30$ in the unbalanced instances and 2.18% for $C = 1.75$ in the balanced ones. We remark that the open optimality gaps for RFFO in these tables are smaller than those reported in Tables 1-2 as the dual bounds obtained by ES-LS within 3600 seconds are better than those obtained with a time limit of 600 seconds. When we compare the results for the unbalanced instances (Table 3) with those for the balanced ones (Table 4), it can be observed that the behavior of ES-LS does not change much for most of the instance configurations when we consider the achieved average gaps, except for the parameter $|W|$. This leads to similar behaviors of RFFO for unbalanced and balanced instances when compared to the solutions encountered by ES-LS, except for the improvements achieved for the different values of $|W|$, as that for $|W| = 5$ shows a much higher average for the balanced instances.

Table 1: Results for 3LSPD-C considering the unbalanced instances and ES-LS running for 600 seconds.

Instances			ES-LS			RFFO							
param	value	#inst	best _{ESLS}	gap _{ESLS}	#opt	best _{RF}	best _{RFFO}	gap _{RFFO}	impr _{RF}	impr _{RFFO/ESLS}	≤best _{ESLS}	<best _{ESLS}	rnd _{FO}
ALL		360	867441.64	4.38	105	888640.79	823577.22	1.41	4.58	3.07	303	204	9.22
T	15	180	555507.71	0.28	105	557932.85	555497.91	0.28	0.41	0.00	139	40	11.03
	30	180	1179375.57	8.48	0	1219348.72	1091656.53	2.53	8.76	6.14	164	164	7.41
W	5	90	752824.44	4.28	32	763302.69	718114.70	1.49	4.01	2.89	89	51	10.91
	10	90	839057.40	4.58	32	850736.95	794950.18	1.46	4.12	3.23	73	49	9.50
	15	90	888552.48	4.43	22	912060.35	841440.29	1.31	4.80	3.21	75	57	8.53
	20	90	989332.25	4.24	19	1028463.18	939803.72	1.36	5.40	2.96	66	47	7.93
R	50	120	717586.93	0.94	50	724468.88	717099.07	0.89	0.79	0.06	96	47	11.55
	100	120	834384.89	2.97	47	846863.10	813536.51	1.18	3.00	1.83	104	65	9.54
	200	120	1050353.10	9.23	8	1094590.39	940096.09	2.15	9.96	7.33	103	92	6.57
C	1.50	120	948724.51	4.49	20	956560.93	891806.31	0.91	4.24	3.66	94	75	8.43
	1.75	120	868440.03	5.04	40	890309.50	825931.35	2.19	4.54	2.98	106	68	9.17
	2.00	120	785160.39	3.61	45	819051.93	752994.00	1.12	4.97	2.57	103	61	10.07

Table 2: Results for 3LSPD-C considering the balanced instances and ES-LS running for 600 seconds.

Instances			ES-LS			RFFO							
param	value	#inst	best _{ESLS}	gap _{ESLS}	#opt	best _{RF}	best _{RFFO}	gap _{RFFO}	impr _{RF}	impr _{RFFO/ESLS}	≤best _{ESLS}	<best _{ESLS}	rnd _{FO}
ALL		360	916176.01	4.33	52	936703.77	871189.82	1.47	4.38	2.95	285	237	9.03
T	15	180	589491.91	0.76	52	591987.16	589158.03	0.71	0.43	0.05	128	80	10.57
	30	180	1242860.10	7.90	0	1281420.39	1153221.61	2.23	8.33	5.85	157	157	7.49
W	5	90	809926.18	6.02	17	788994.27	750806.41	1.79	3.32	4.40	81	66	9.81
	10	90	893863.57	4.63	13	904856.00	843910.50	1.59	4.30	3.14	77	66	9.36
	15	90	944040.67	3.97	12	970783.24	900289.37	1.39	4.55	2.64	70	60	8.86
	20	90	1016873.60	2.72	10	1082181.58	989753.00	1.12	5.33	1.63	57	45	8.10
R	50	120	736511.85	0.90	43	743821.56	735904.94	0.84	0.82	0.07	88	47	12.05
	100	120	870414.42	1.96	9	898640.97	861010.05	1.12	3.11	0.86	89	82	9.40
	200	120	1141601.74	10.13	0	1167648.79	1016654.47	2.46	9.19	7.93	108	108	5.64
C	1.50	120	992888.15	4.01	5	1006824.08	939766.40	0.87	4.15	3.21	97	89	8.00
	1.75	120	916194.64	4.99	19	937903.20	873496.42	2.28	4.30	2.83	91	73	9.04
	2.00	120	839445.23	3.99	28	865384.04	800306.64	1.27	4.68	2.82	97	75	10.05

Table 3: Results for 3LSPD-C considering the unbalanced instances and ES-LS running for 3600 seconds.

param	Instances		ES-LS			RFFO				
	value	#inst	best _{ESLS}	gap _{ESLS}	#opt	best _{RFFO}	gap _{RFFO}	impr _{RFFO} _{ESLS}	≤best _{ESLS}	<best _{ESLS}
ALL		360	836685.34	2.11	134	823577.22	1.24	0.90	242	134
T	15	180	555336.71	0.07	134	555497.91	0.09	-0.02	115	7
	30	180	1118033.98	4.16	0	1091656.53	2.39	1.82	127	127
W	5	90	719033.14	1.45	33	718114.70	1.37	0.09	78	37
	10	90	814614.99	2.66	35	794950.18	1.32	1.39	63	37
	15	90	868882.24	2.93	35	841440.29	1.15	1.83	55	33
	20	90	944211.01	1.40	31	939803.72	1.13	0.28	46	27
R	50	120	716942.52	0.79	57	717099.07	0.80	-0.01	76	27
	100	120	814314.07	1.18	54	813536.51	1.12	0.07	87	46
	200	120	978799.44	4.36	23	940096.09	1.80	2.64	79	61
C	1.50	120	910793.00	1.88	34	891806.31	0.73	1.18	69	48
	1.75	120	834645.99	2.64	45	825931.35	2.04	0.62	88	46
	2.00	120	764617.04	1.81	55	752994.00	0.94	0.90	85	40

Table 4: Results for 3LSPD-C considering the balanced instances and ES-LS running for 3600 seconds.

param	Instances		ES-LS			RFFO				
	value	#inst	best _{ESLS}	gap _{ESLS}	#opt	best _{RFFO}	gap _{RFFO}	impr _{RFFO} _{ESLS}	≤best _{ESLS}	<best _{ESLS}
ALL		360	884173.08	2.18	93	871189.82	1.30	0.92	219	155
T	15	180	589045.88	0.42	93	589158.03	0.44	-0.02	101	37
	30	180	1179300.27	3.93	0	1153221.61	2.15	1.85	118	118
W	5	90	798782.99	4.85	27	750806.41	1.57	3.41	76	54
	10	90	845040.99	1.46	24	843910.50	1.39	0.08	60	42
	15	90	902055.34	1.30	23	900289.37	1.19	0.11	46	34
	20	90	990812.99	1.09	19	989753.00	1.03	0.06	37	25
R	50	120	735806.22	0.76	55	735904.94	0.77	-0.01	69	24
	100	120	861601.89	1.06	37	861010.05	1.00	0.05	68	52
	200	120	1055111.12	4.71	1	1016654.47	2.11	2.70	82	79
C	1.50	120	955254.73	1.70	18	939766.40	0.73	0.99	71	56
	1.75	120	887681.20	3.15	36	873496.42	2.18	1.02	78	57
	2.00	120	809583.30	1.68	39	800306.64	0.97	0.74	70	42

4.4 Results for G3LSPD-C

In this section, we assess the performance of our hybrid heuristic when applied to G3LSPD-C and analyze how it behaves under different capacity configurations. We did not perform any specific parameterization for RFFO, instead, we used the same parameters which were applied for 3LSPD-C (see Section 4.2). RFFO is compared against ES-LS, with both approaches executed with a time limit of 600 seconds for each of the instances.

The results are summarized in Tables 5-6. The first three columns identify the capacity configurations of the instances, given by the capacity factors at the plant (C), at the warehouses (C_s^W), and at the retailers (C_s^R), followed by the number of instances with that specific configuration. A value ∞ indicates that there is no limited capacity. The remaining columns are similar to those in Tables 1-2. The results show that RFFO remains very robust for G3LSPD-C. Besides, it clearly outperforms ES-LS when the same time limit of 600 seconds is available for both approaches. RFFO obtained best average values for all the capacity configurations but one for the unbalanced instances ($C = \infty, C_s^W = \infty, C_s^R = 1.50$) and two for the balanced ones ($C = \infty, C_s^W = \infty, C_s^R = 1.50$; and $C = 2.00, C_s^W = \infty, C_s^R = 1.50$). Overall, a solution at least matching the best obtained by ES-LS was achieved by RFFO for around 79% of the instances (2272 and 2270 out of 2880, for the unbalanced and balanced, respectively) and strictly better solutions were achieved for around 48% (1374 out of 2880) of the unbalanced instances and around 56% (1607 out of 2880) of the balanced ones. It can also be noticed that the average optimality gaps of the solutions encountered by RFFO are reasonably low, with the largest values being 2.60% for the unbalanced instances and 2.77% for the balanced ones. It is noteworthy that RFFO achieved good average improvements over ES-LS for the great majority of the capacity configurations, reaching values as high as 10.91% for the unbalanced instances and 12.54% for the balanced ones.

The plot in Figure 3 shows the improvements of RFFO over ES-LS for all the 5760 instances (both balanced and unbalanced). The plot shows that the positive improvements of RFFO over ES-LS are remarkable, as they reach values over 50%. This contrasts with the improvements of ES-LS over RFFO, which are very modest.

Table 5: Results for G3LSPD-C considering the unbalanced instances and ES-LS running for 600 seconds.

C	Instances			ES-LS			RFFO							
	C_s^W	C_s^R	#inst	best $_{ESLS}$	gap $_{ESLS}$	#opt	best $_{RF}$	best $_{RFFO}$	gap $_{RFFO}$	impr $_{RF}^{FO}$	impr $_{ESLS}^{RFFO}$	\leq best $_{ESLS}$	<best $_{ESLS}$	rnd $_{FO}$
ALL	ALL	ALL	2880	834685.45	6.48	960	816925.90	743460.02	1.34	5.24	5.33	2272	1374	9.83
∞	∞	1.50	120	484160.51	0.53	90	543917.91	484311.72	0.55	5.01	-0.02	101	10	15.51
∞	∞	1.75	120	541489.87	4.87	76	559477.57	480507.27	0.86	6.10	4.22	110	29	14.76
∞	∞	2.00	120	544119.62	5.27	72	559475.95	478024.94	0.98	6.37	4.52	103	29	14.40
∞	1.50	∞	120	607501.86	8.33	49	574476.13	483162.53	1.19	7.00	7.48	102	42	9.33
∞	1.75	∞	120	603603.63	8.18	52	571461.91	482099.47	1.19	6.96	7.31	101	39	9.79
∞	2.00	∞	120	600162.89	7.88	61	572080.80	480906.80	1.12	6.97	7.09	104	40	9.90
1.50	∞	1.50	120	902756.38	0.67	34	943362.97	901812.56	0.60	2.70	0.07	84	55	10.74
1.50	∞	1.75	120	932684.83	2.89	26	948801.66	898322.14	0.76	3.30	2.17	84	63	10.02
1.50	∞	2.00	120	937679.52	3.49	23	951317.95	896930.26	0.89	3.56	2.66	87	71	9.59
1.50	1.50	∞	120	1069909.11	10.15	11	994423.41	899323.09	1.10	6.06	9.24	84	73	6.62
1.50	1.75	∞	120	1053151.71	9.43	11	985842.44	898659.68	1.09	5.63	8.53	91	79	6.73
1.50	2.00	∞	120	1042370.48	8.96	10	985806.15	898482.92	1.14	5.54	8.00	86	74	6.90
1.75	∞	1.50	120	837564.78	1.94	51	879762.05	836332.02	1.84	3.02	0.10	93	49	11.13
1.75	∞	1.75	120	854492.82	3.49	45	887781.96	832540.36	1.99	3.81	1.56	96	56	10.76
1.75	∞	2.00	120	857483.93	3.95	41	887351.69	830461.54	2.06	3.94	1.97	97	64	10.24
1.75	1.50	∞	120	1016222.15	12.28	17	919156.36	834238.27	2.60	5.81	10.14	93	77	6.79
1.75	1.75	∞	120	1006665.90	11.83	25	915495.80	833203.91	2.56	5.68	9.71	94	75	7.13
1.75	2.00	∞	120	988828.15	10.82	24	912322.79	832210.11	2.51	5.47	8.69	97	78	7.09
2.00	∞	1.50	120	764183.61	0.84	57	813729.07	762869.21	0.73	3.68	0.12	96	45	12.02
2.00	∞	1.75	120	777525.17	2.30	49	822126.56	759860.60	0.96	4.46	1.38	97	56	11.71
2.00	∞	2.00	120	778023.04	2.59	46	823774.00	757424.64	1.00	4.77	1.64	95	58	11.09
2.00	1.50	∞	120	952165.01	12.05	26	854816.90	761682.45	1.56	6.74	10.86	95	74	7.65
2.00	1.75	∞	120	952370.36	12.02	32	850168.55	760366.15	1.45	6.58	10.91	88	68	7.82
2.00	2.00	∞	120	927335.35	10.78	32	849291.09	759307.93	1.41	6.54	9.66	94	70	8.08

Table 6: Results for G3LSPD-C considering the balanced instances and ES-LS running for 600 seconds.

C	Instances			ES-LS			RFFO							
	C_s^w	C_s^r	#inst	best _{ESLS}	gap _{ESLS}	#opt	best _{RF}	best _{RFFO}	gap _{RFFO}	impr _{RF} ^{FO}	impr _{ESLS} ^{RFFO}	≤best _{ESLS}	<best _{ESLS}	rnd _{FO}
ALL	ALL	ALL	2880	890596.63	6.71	600	862395.36	785533.74	1.30	5.25	5.56	2270	1607	9.44
∞	∞	1.50	120	508796.98	0.09	96	568798.50	508932.72	0.10	4.57	-0.01	108	3	16.32
∞	∞	1.75	120	508272.09	0.58	90	578266.18	504945.44	0.22	5.51	0.38	108	11	16.14
∞	∞	2.00	120	529881.64	2.49	84	580879.10	502518.49	0.31	5.83	2.22	112	20	15.84
∞	1.50	∞	120	653862.10	8.30	23	608459.06	507482.72	0.71	7.24	7.77	97	64	7.52
∞	1.75	∞	120	648993.73	8.28	26	600896.58	506415.12	0.65	7.04	7.87	98	63	7.80
∞	2.00	∞	120	632063.21	7.75	31	597312.06	503821.95	0.57	6.88	7.33	102	59	8.33
1.50	∞	1.50	120	954138.85	0.75	14	994448.27	953085.25	0.67	2.55	0.08	94	75	9.56
1.50	∞	1.75	120	970392.93	2.12	8	1009293.15	949128.89	0.77	3.63	1.38	85	71	9.03
1.50	∞	2.00	120	974329.76	2.56	4	1008569.90	946729.74	0.83	3.73	1.77	86	79	8.70
1.50	1.50	∞	120	1159592.63	11.40	2	1053885.00	944823.13	1.08	6.69	10.51	98	94	6.21
1.50	1.75	∞	120	1153199.16	11.22	3	1044668.72	943657.85	1.02	6.25	10.37	92	88	6.58
1.50	2.00	∞	120	1134666.22	10.43	3	1042764.87	943061.71	1.00	6.22	9.59	95	91	6.82
1.75	∞	1.50	120	888180.59	2.07	30	927503.39	887447.47	2.02	2.66	0.06	84	56	10.72
1.75	∞	1.75	120	900740.97	3.34	24	940859.12	883700.92	2.18	3.77	1.21	89	65	10.11
1.75	∞	2.00	120	903605.38	3.78	19	943598.46	881140.81	2.28	4.01	1.56	86	71	10.06
1.75	1.50	∞	120	1108148.59	13.85	5	969422.16	878998.97	2.77	6.00	11.55	98	89	6.41
1.75	1.75	∞	120	1101883.31	13.69	5	960839.06	878069.47	2.77	5.50	11.37	95	86	6.71
1.75	2.00	∞	120	1079585.32	12.69	6	960293.46	877398.22	2.67	5.57	10.44	92	85	6.98
2.00	∞	1.50	120	813229.40	0.93	38	861904.13	813603.17	0.95	3.34	-0.02	78	42	11.82
2.00	∞	1.75	120	826247.29	2.28	31	872645.82	809700.31	1.06	4.29	1.26	98	68	11.40
2.00	∞	2.00	120	830062.05	2.85	26	874807.48	807526.02	1.23	4.61	1.67	94	73	11.09
2.00	1.50	∞	120	1053283.81	14.12	6	903104.61	806453.76	1.92	6.90	12.54	95	90	7.08
2.00	1.75	∞	120	1034947.27	13.38	11	900283.39	805274.81	1.82	6.72	11.88	96	85	7.47
2.00	2.00	∞	120	1006215.81	12.05	15	893986.26	806743.57	1.72	6.47	10.69	90	79	7.90

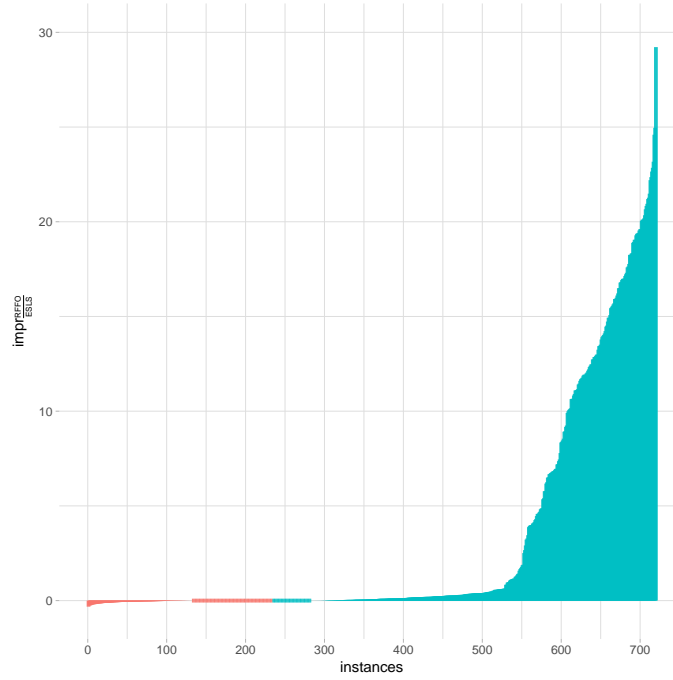


Figure 2: Plot with the improvements of RFFO over ES-LS ($\text{impr}_{\frac{RFFO}{ESLS}}$) for all 3LSPD-C instances.

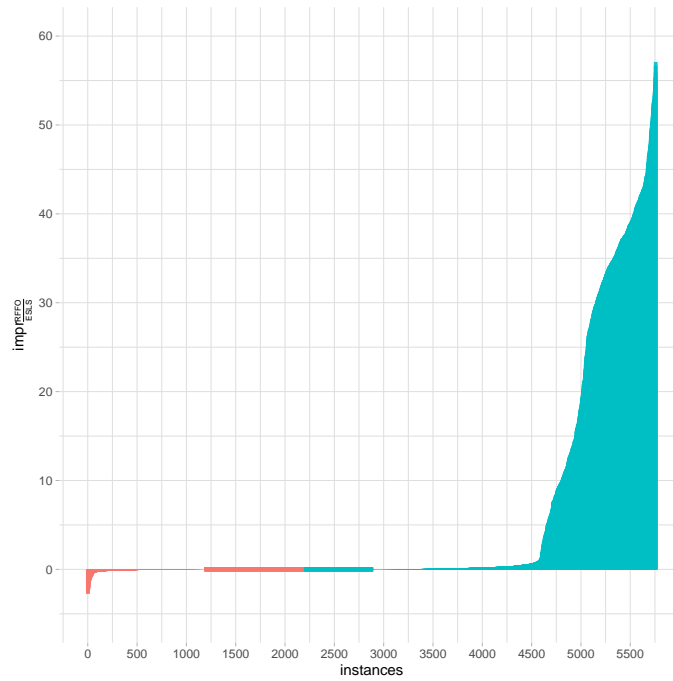


Figure 3: Plot with the improvements of RFFO over ES-LS ($\text{impr}_{\frac{RFFO}{ESLS}}$) for all G3LSPD-C instances.

4.5 Economic impacts of the capacity configurations

In what follows, consider the deviation of a solution for a given capacity configuration from a baseline problem calculated as $100 \frac{\text{best}_{RFFO} - \text{best}_{\text{base}}}{\text{best}_{\text{base}}}$ for a specific instance. In this formula, best_{RFFO} defines the best solution achieved by RFFO for the given capacity configuration while $\text{best}_{\text{base}}$ denotes the optimal solution for the baseline problem 3LSPD-U (determined with the approaches described in Cunha and Melo (2021)) or the best solution achieved by RFFO for the baseline problem 3LSPD-C. Note that the deviation indicates the economical impact (in terms of increase in the objective value) of the given capacity configuration on the obtained solution.

The boxplots in Figure 4 depict the deviations from the baseline problem 3LSPD-U when there are production capacities at the plant. It can be seen that median deviations of 96%, 82%, and 67% were incurred when considering production plant capacity factors of 1.50, 1.75, and 2.00, respectively. It is noteworthy that the economical impacts can reach values as high as 195% for the tighter capacity factor. The large deviations can be explained by the need for additional setups, which have high costs in the considered instances, when the amount that can be produced in a given period is limited.

The boxplots in Figure 5 show the deviations from the baseline problems 3LSPD-U (Subfigure 5(a)) and 3LSPD-C (Subfigures 5(b)-5(d)) when there are limited storage capacities at the warehouses or at the retailers. Notice that all subfigures use the same scale. Overall, the boxplot in Subfigure 5(a) shows that when there is no production capacity at the plant, there is a larger variance on the deviations for the capacity factors at the warehouses, implying reasonably high deviations for several instances. On the other hand, when the baseline problem is 3LSPD-C, the deviations are more modest and stable, with the capacity factors at the retailers implying solutions with higher costs. Notice that some negative values occur in Subfigures 5(b)-5(d), which can be explained by the fact that the hybrid heuristic RFFO can in certain situations tackle some more restricted subproblems of G3LSPD-C better than those of 3LSPD-C, achieving solutions that are closer to optimal.

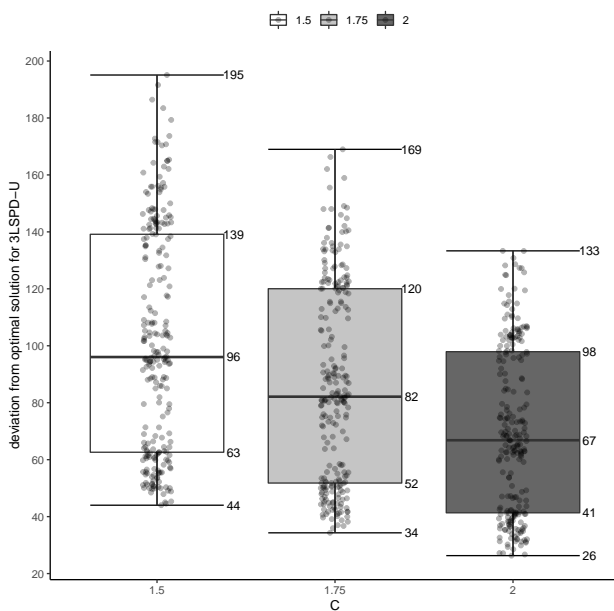
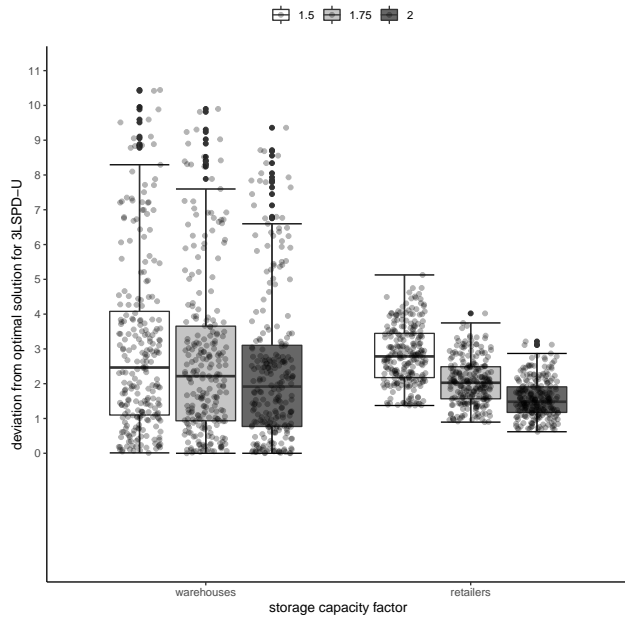
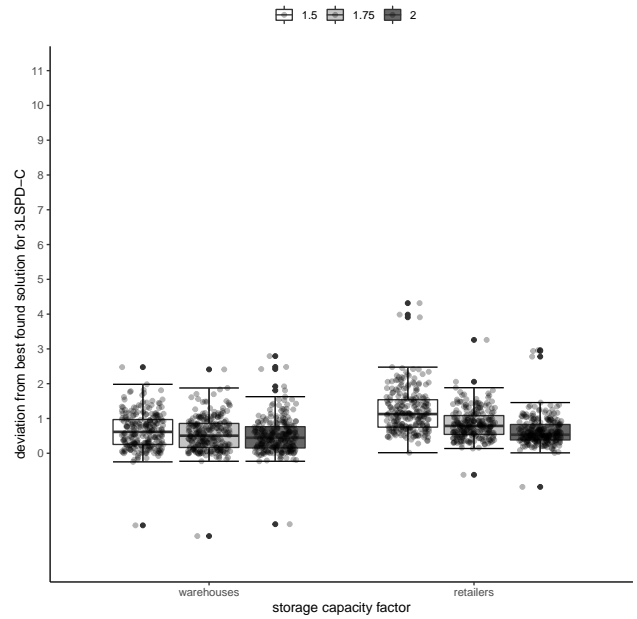


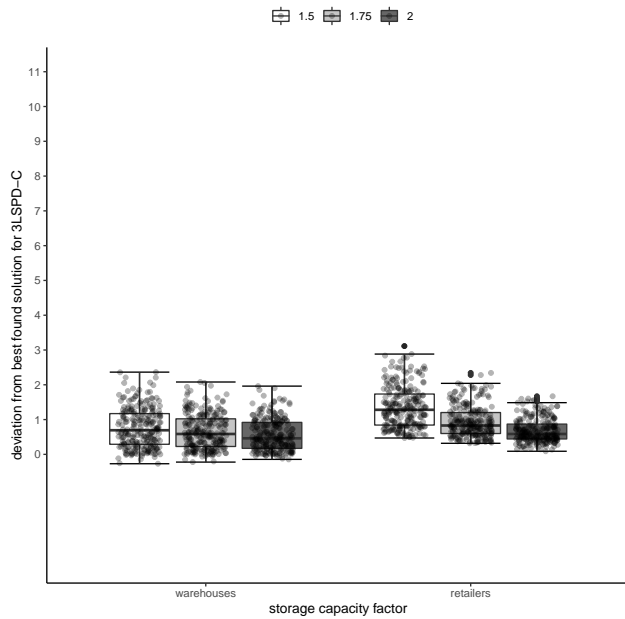
Figure 4: Boxplot summarizing the deviations from the optimal solutions for the 3LSPD-U when capacities occur at the plant.



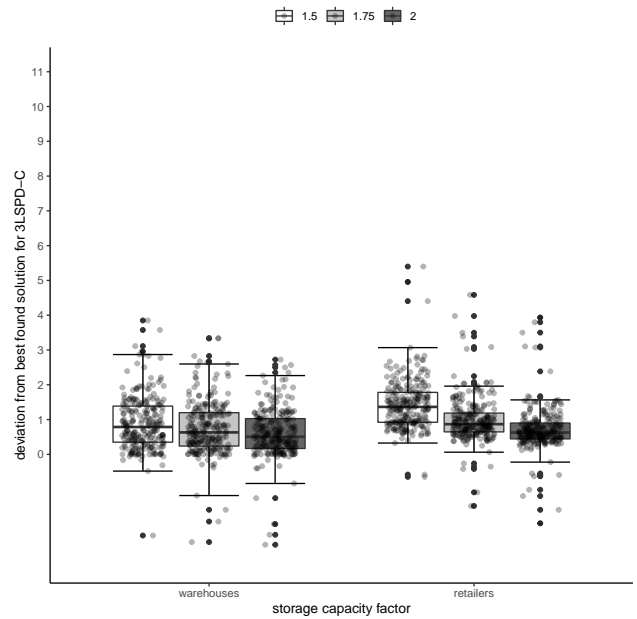
(a) Baseline problem 3LSPD-U.



(b) Baseline problem 3LSPD-C with $C = 1.5$.



(c) Baseline problem 3LSPD-C with $C = 1.75$.



(d) Baseline problem 3LSPD-C with $C = 2.00$.

Figure 5: Boxplot summarizing the deviations from the best known solutions for the baseline problems when capacities occur at the warehouses or at the retailers.

5 Concluding remarks

In this paper, we considered the capacitated three-level lot-sizing and replenishment problem with a distribution structure (3LSPD-C) and an extension denoted the generalized capacitated three-level lot-sizing and replenishment problem with a distribution structure (G3LSPD-C) in which storage capacities restrict the operations on the warehouses and/or retailers. We proposed a very effective hybrid MIP heuristic combining relax-and-fix and fix-and-optimize based on varying-size neighborhoods (RFFO) that can be applied for both problems.

Extensive computational experiments showed that RFFO clearly outperforms a state-of-the-art formulation (ES-LS) in terms of solutions quality when both approaches are offered the same time limit. Depending on the instances' characteristics, average improvements can reach approximately 8% for 3LSPD-C. For G3LSPD-C, depending on the capacity configurations, average improvements can come to more than 12%. Such observed improvements can allow considerable economical impacts in a practical setting. The results also show that the solutions encountered by RFFO can be very competitive even when compared to ES-LS running for a much larger time. Overall, RFFO is very robust in that it obtains, for both 3LSPD-C and G3LSPD-C, solutions whose average optimality gaps are below 3% for all the reported instance configurations.

We remark that our approach can be easily extended to other variants and extensions of the problem, such as those involving multiple plants, backlogging, retailers that can receive items from multiple warehouses, among others. Furthermore, we believe that the concepts used in our fix-and-optimize approach may bring enhancements when applied to other problems involving supply chains and/or multi-period logistical optimization problems.

Acknowledgments

Work of Rafael A. Melo was supported by the State of Bahia Research Foundation (FAPESB) and the Brazilian National Council for Scientific and Technological Development (CNPq). Work of Geraldo R. Mateus was supported by Fundação de Amparo à Pesquisa do Estado de Minas Gerais (FAPEMIG) and CNPq.

References

- Akbalik, A., Penz, B., & Rapine, C. (2015a). Capacitated lot sizing problems with inventory bounds. *Annals of Operations Research*, 229(1), 1–18.
- Akbalik, A., Penz, B., & Rapine, C. (2015b). Multi-item uncapacitated lot sizing problem with inventory bounds. *Optimization Letters*, 9(1), 143–154.
- Bitran, G. R., & Yanasse, H. H. (1982). Computational complexity of the capacitated lot size problem. *Management Science*, 28(10), 1174–1186.
- Brahimi, N., Absi, N., Dautère-Pérès, S., & Kedad-Sidhoum, S. (2015). Models and lagrangian heuristics for a two-level lot-sizing problem with bounded inventory. *OR Spectrum*, 37(4), 983–1006.
- Buschkühl, L., Sahling, F., Helber, S., & Tempelmeier, H. (2010). Dynamic capacitated lot-sizing problems: a classification and review of solution approaches. *OR Spectrum*, 32(2), 231–261.
- Carvalho, D. M., & Nascimento, M. C. (2018). A kernel search to the multi-plant capacitated lot sizing problem with setup carry-over. *Computers & Operations Research*, 100, 43 - 53.
- Chen, H. (2015). Fix-and-optimize and variable neighborhood search approaches for multi-level capacitated lot sizing problems. *Omega*, 56, 25–36.
- Cunha, J. O., Kramer, H. H., & Melo, R. A. (2019). Effective matheuristics for the multi-item capacitated lot-sizing problem with remanufacturing. *Computers & Operations Research*, 104, 149 - 158.
- Cunha, J. O., Kramer, H. H., & Melo, R. A. (2021). On the computational complexity of uncapacitated multi-plant lot-sizing problems. *Optimization Letters*, 15(2), 803–812.

- Cunha, J. O., & Melo, R. A. (2016). On reformulations for the one-warehouse multi-retailer problem. *Annals of Operations Research*, 238(1-2), 99–122.
- Cunha, J. O., & Melo, R. A. (2021). Valid inequalities, preprocessing, and an effective heuristic for the uncapacitated three-level lot-sizing and replenishment problem with a distribution structure. *European Journal of Operational Research*, 295(3), 874–892.
- Cárdenas-Barrón, L. E., & Treviño-Garza, G. (2014). An optimal solution to a three echelon supply chain network with multi-product and multi-period. *Applied Mathematical Modelling*, 38(5), 1911 - 1918.
- Eppen, G. D., & Martin, R. K. (1987). Solving multi-item capacitated lot-sizing problems using variable redefinition. *Operations Research*, 35(6), 832–848.
- Gruson, M., Bazrafshan, M., Cordeau, J.-F., & Jans, R. (2019). A comparison of formulations for a three-level lot sizing and replenishment problem with a distribution structure. *Computers & Operations Research*, 111, 297 – 310.
- Gruson, M., Cordeau, J.-F., & Jans, R. (2020). Benders decomposition for a stochastic three-level lot sizing and replenishment problem with a distribution structure. *European Journal of Operational Research*.
- Hansen, P., Mladenović, N., Brimberg, J., & Pérez, J. A. M. (2019). Variable Neighborhood Search. In M. Gendreau & J.-Y. Potvin (Eds.), *Handbook of Metaheuristics* (pp. 57–97). Cham: Springer International Publishing.
- Helber, S., & Sahling, F. (2010). A fix-and-optimize approach for the multi-level capacitated lot sizing problem. *International Journal of Production Economics*, 123(2), 247–256.
- Jing, F., & Mu, Y. (2020). Dynamic lot-sizing model under perishability, substitution, and limited storage capacity. *Computers & Operations Research*, 122, 104978.
- Li, Z., & Hai, J. (2019). Inventory management for one warehouse multi-retailer systems with carbon emission costs. *Computers & Industrial Engineering*, 130, 565–574.
- Melo, R. A., & Ribeiro, C. C. (2017). Formulations and heuristics for the multi-item uncapacitated lot-sizing problem with inventory bounds. *International Journal of Production Research*, 55(2), 576–592.
- Melo, R. A., & Wolsey, L. A. (2010). Optimizing production and transportation in a commit-to-delivery business mode. *European Journal of Operational Research*, 203(3), 614–618.
- Melo, R. A., & Wolsey, L. A. (2012). MIP formulations and heuristics for two-level production-transportation problems. *Computers & Operations Research*, 39(11), 2776 - 2786.
- Park, Y. B. (2005). An integrated approach for production and distribution planning in supply chain management. *International Journal of Production Research*, 43(6), 1205-1224.
- Slama, I., Ben-Ammar, O., Dolgui, A., & Masmoudi, F. (2020). New mixed integer approach to solve a multi-level capacitated disassembly lot-sizing problem with defective items and backlogging. *Journal of Manufacturing Systems*, 56, 50–57.
- Solyali, O., & Süral, H. (2012). The one-warehouse multi-retailer problem: Reformulation, classification, and computational results. *Annals of Operations Research*, 196(1), 517–541.
- Stecke, K. E., & Zhao, X. (2007). Production and transportation integration for a make-to-order manufacturing company with a commit-to-delivery business mode. *Manufacturing & Service Operations Management*, 9(2), 206–224.
- Tanksale, A., & Jha, J. K. (2017). Solving multi-region multi-facility inventory allocation and transportation problem: a case of indian public distribution system. *Computers & Industrial Engineering*, 110, 175–190.
- Tavaghof-Gigloo, D., & Minner, S. (2020). Planning approaches for stochastic capacitated lot-sizing with service level constraints. *International Journal of Production Research*, 1–21.
- Toledo, C. F. M., da Silva Arantes, M., Hossomi, M. Y. B., França, P. M., & Akartunalı, K. (2015). A relax-and-fix with fix-and-optimize heuristic applied to multi-level lot-sizing problems. *Journal of Heuristics*, 21(5), 687–717.
- Trigeiro, W. W., Thomas, L. J., & McClain, J. O. (1989). Capacitated lot sizing with setup times. *Management science*, 35(3), 353–366.
- Vincent, B., Duhamel, C., Ren, L., & Tchernev, N. (2020). A population-based metaheuristic for the capacitated lot-sizing problem with unrelated parallel machines. *International Journal of Production*

Research, 58(21), 6689–6706.

Zhang, S., & Song, H. (2018). Production and distribution planning in Danone Waters China Division. *Interfaces*, 48(6), 578–590.

Appendix A Echelon stock formulation

In this section, we present the echelon stock formulation that is employed in our hybrid heuristic. This formulation was introduced in Gruson et al. (2019), where more details can be obtained. Define the echelon stock I_t^i for facility $i \in F$ in period $t \in T$ as

$$I_t^i = \begin{cases} s_t^i + \sum_{w \in W} s_t^w + \sum_{r \in R} s_t^r, & \text{if } i = p; \\ s_t^i + \sum_{r \in \delta(i)}, & \text{if } i \in W; \\ s_t^i, & \text{if } i \in R. \end{cases}$$

The formulation ES-LS can be cast as

$$z_{ES-LS} = \min \sum_{t \in T} \left(\sum_{i \in F} s c_t^i y_t^i + h c_t^p I_t^p + \sum_{w \in W} (h c_t^w - h c_t^p) I_t^w + \sum_{r \in R} (h c_t^r - h c_t^{\delta_w(r)}) I_t^r \right) \quad (10)$$

(ES-LS) (4)-(5), (7),

$$I_{t-1}^i + x_t^i = d_t^i + I_t^i, \quad \text{for } i \in F, t \in T, \quad (11)$$

$$I_t^i \geq \sum_{j \in \delta(i)} I_t^j, \quad \text{for } i \in P \cup W, t \in T, \quad (12)$$

$$I_{t-1}^i \geq \sum_{j=t}^l d_j^i \left(1 - \sum_{u=t}^j y_u^i \right), \quad \text{for } i \in F, t, l \in T, \text{ with } t \leq l, \quad (13)$$

$$x_t^i, I_t^i \geq 0, \quad \text{for } i \in F, t \in T. \quad (14)$$