Operations Research Forum (2023) 4:47
https://doi.org/10.1007/s43069-023-00230-7

ORIGINAL RESEARCH

Relax-and-Fix Heuristics Applied to a Real-World Lot Sizing and Scheduling Problem in the Personal Care Consumer Goods Industry

K. A. G. Araujo[1] · E. G. Birgin[2] · M. S. Kawamura[3] · D. P. Ronconi[3]

Received: 22 July 2021 / Accepted: 20 May 2023 / Published online: 31 May 2023

Abstract

This paper addresses an integrated lot sizing and scheduling problem in the industry of consumer goods for personal care, a very competitive market in which good customer service and cost management are crucial in the competition for clients. In this research, a complex operational environment composed of unrelated parallel machines with limited production capacity and sequence-dependent setup times and costs is studied. There is also a limitation in the total storage capacity for finished goods, a characteristic not found in the literature. Backordering is allowed, but it is extremely undesirable. The problem is described through a mixed integer linear programming formulation. Since the problem is NP-hard, relax-and-fix heuristics with hybrid partitioning strategies are investigated. Computational experiments with randomly generated and real-world instances are presented. The results show the efficacy and efficiency of the proposed approaches. Compared to the current solutions used by the company, the best proposed strategies yield results with substantially lower costs, primarily from the reduction in inventory levels and better allocation of production batches on the machines.

Keywords Lot sizing and scheduling · Mixed integer linear programming models · Relax-and-fix · Real-world instances

1 Introduction

According to the Brazilian Toiletry, Perfumery and Cosmetic Association (ABIHPEC), the Brazilian market for personal care, perfumery, and cosmetics showed growth of 2.2% in 2020 above the −4.5% of overall industry and −4.1% of the GDP (Gross Domestic Product) (both heavily affected by the COVID-19 pandemic) and a CAGR (Compound Annual Growth Rate) of 1.7% over the last 10 years compared to a CAGR of −2.1% for industry overall and −0.3% for GDP over the same period [1]. Among the factors that have contributed to this accelerated growth are

Extended author information available on the last page of the article the growing participation of women in the labor market, the frequent releases of new products, the higher productivity achieved by using cutting-edge technology, and, more recently, the essentiality of the segment's products in combating the COVID-19 pandemic. This segment comprises a large number of small manufacturers and some large companies. In 2020, Brazil held the fourth largest consumer market of this sector in the world, valued at US$ 23.7 billion. Inside this market, the segment studied in this work is the disposable personal care segment, composed of diapers, sanitary pads, toilet paper, towels, and tissues. These products are characterized by (i) frequent consumption by the population, (ii) relatively low cost, and (iii) very similar quality among different brands. Therefore, the processes of production planning, scheduling and control serve important roles for companies to guarantee productivity, cost control, and an adequate customer service level. In most companies, the main responsibility of this activity is to simultaneously analyze several relevant pieces of information, such as the seasonality of the demand, the seasonality of the supply of raw materials, the perishability of the finished products and the raw material, and the manufacturing capacity, to develop a production plan that optimizes the use of productive resources while meeting the demand for manufactured products.

In this work, a case study of a large company in the market of disposable personal care products is performed. The company has factories in southern and southeastern Brazil, and its market is also geographically concentrated in these regions. Its main product categories are diapers, feminine sanitary pads, and toilet paper. These products are manufactured by the company itself, and their demands are predominantly affected by the company's marketing activities and by the competition. Other products, such as tissue paper, which is commonly utilized in the winter due to a higher frequency of respiratory infections, and disposable napkins and paper towels, which are largely sold during holiday times such as Christmas, Easter, and the New Year, among others, have a strong seasonal demand. The production configuration of the company consists of machines of different models that have been acquired over time as demand was increasing. These machines have performance profiles distinct from each other with regard to efficiency, speed, and cost. Moreover, due to technical constraints, not all products can be manufactured by all machines. Between the production of batches of different products on the same line, machine preparation (setup) is required, which may be as simple and rapid as changing the consuming packing, or complex and slow if it involves a change of raw material or reconfiguration of the parameters of the product, among others. Thus, these setups depend on the product to be produced and on the product that was being produced in the previous batch. The manufactured products are shipped to a limited capacity distribution center located close to the factories. Figure 1 schematically represents the process.

Typical production planning decisions, such as lot sizing and scheduling, significantly influence the results of a company by maximizing the fulfillment of sales orders, correctly adjusting the inventory levels, and incurring proper operating costs. The process of lot sizing consists of determining how much of each product to produce in each period to meet a projected demand under the existing conditions and operational capabilities. On the other hand, scheduling means determining when and in which sequence these lots should be produced to maximize the productive resource efficiency and to meet the deadlines. Inefficiencies in these processes can

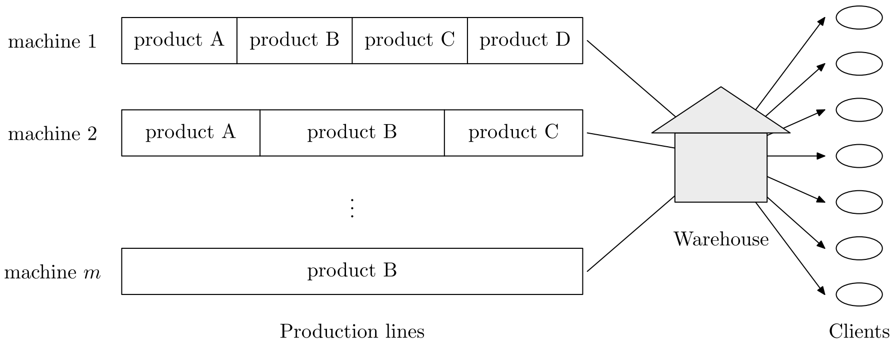

Fig. 1 Schematically representation of the production process

cause overstocking of finished products, unfulfilled sales orders, loss of perishable material, missed due dates, significant reduction in the productive capacity of the production line, stocks of finished products accumulated in advance, and higher preparation machine costs, among others.

The proposed solution method is a customized relax-and-fix heuristic that decomposes a mixed integer linear programming (MILP) model that represents the entire problem into less complex subproblems that are successively solved. This matheuristic approach is mentioned by Copil et al. [2] as the next generation of solution methods for the MILP models of the lot sizing and scheduling problems. It should be noted that the proposed strategy requires minimal technical knowledge to be applied in situ and opens the possibility that subproblems (which are smaller and therefore simpler) may be solved with established off-the-shelf software. The proposed strategy strikes a balance between market-ready tools, where the firm must adapt to the capabilities of the software, and the development of more elaborate ad hoc solution methodologies such as customized metaheuristics that require hiring specialized teams. Eleven different decomposing approaches are presented. Nine of them are problem-dependent strategies that divide the main problem considering several metrics associated with machines, products, and periods, and two of them are problem-independent strategies. The aim of the study is not only to solve a complex real-world problem, but also to assess in which ways the form of partitioning the problem into simpler subproblems and the sequence in which subproblems are solved affect the performance of the relax-and-fix heuristic.

The rest of this paper is organized as follows. A literature review is presented in Section 2. In Section 3, a MILP formulation of the problem is given. In Section 4, the considered relax-and-fix heuristics are described. In Section 5, numerical experiments with randomly generated instances and real-world instances of the company are conducted. The proposed heuristic methods are compared against a commercial solver applied to the MILP model presented in Section 3, while for the real-world instances, a comparison with the solutions adopted by the company is also presented. Conclusions are given in the last section.

2 Literature Review

In the literature, the integrated lot sizing and scheduling problem with dependent setup times and costs on a single machine (or a single line) is called GLSPST (general lot sizing and scheduling problem with sequence-dependent setup time), while the same problem involving parallel machines is called GLSPPL (general lot sizing and scheduling problem with parallel machines). The GLSP with nonzero minimum lot sizes and without setup times has been proven to be NP-complete by Fleischmann and Meyr [3]. Consequently, the GLSP and the GLSPPL with sequence setup times are also NP-hard [4, 5]. The focused problem can be considered as the GLSPPL with sequence-dependent setup times and costs, nonidentical parallel machines, and specific characteristics of the environment addressed, such as limited warehousing capacity for finished products, machine eligibility constraints, backorder costs, and setup state conservation among adjacent periods. To the knowledge of the authors, few studies have addressed the GLSPPL with all the characteristics of the focused real-world problem.

Due to the relevance of production planning for the industry and its complexity, many authors have addressed the lot sizing and scheduling problem with different characteristics. See classical reviews about the theme in [6-10], and for a recent review see [2]. A brief description of the works that address GLSPPL with specific characteristics is presented below. Differences from the problem considered in this work are highlighted.

Kang et al. [11] treated the GLSPPL considering the minimization of setup and inventory costs using column generation techniques. In this work, the cost of sequence-dependent setups is considered, but setup times are ignored. Clark and Clark [12] studied the GLSPPL, whose objective is the minimization of storage and backordering costs, but without considering setup costs. The proposed mathematical model is based on the premise that the maximum number of setups per period is predetermined. A rolling-horizon method and relax-and-fix heuristics are applied. However, the presented computational results show that only small problems can be solved in a reasonable time. Aiming to minimize production, inventory and setup costs without allowing delivery delays, [5] considered the threshold accepting and simulated annealing metaheuristic in small real instances with identical machines. Beraldi et al. [13] developed rolling-horizon and relax-and-fix heuristics to solve the GLSPPL in the textile and fiberglass industry environment. Unlike the mathematical model developed by Meyr [5], the authors introduced a compact formulation for the case of identical machines considering the setup costs but neglected the setup times.

Józefowska and Zimniak [14] developed a decision support system applied to a Polish company that manufactures plastic pipes and whose production environment is composed of unrelated parallel machines. This system is based on a multi-objective model that includes among its criteria the maximization of the machine utilization and the minimization of the deviation between the production schedule and the S &OP (Sales and Operations Plan) and the amount of products below the required level of safety stock. The model is solved using a genetic algorithm after having its solution space reduced by adding restrictions suggested by experienced planners. Mateus et al. [15] approached the lot sizing and scheduling problem by considering a refractory brick factory with different machines in parallel. Unlike the problem addressed in this work, the authors did not consider setup carryover, i.e., when the preparation of the machines is not maintained from one period to the next. The proposed iterative solution method is composed of two modules: the first solves the problem of lot sizing considering aggregate capacity and estimated setup times, while the second searches for a feasible sequencing for pre-sized lots through a GRASP metaheuristic. Meyr and Mann [16] dealt with the GLSPPL composed of heterogeneous parallel machines with the objective of minimizing inventory and sequence-dependent setup and production costs without backlogging. The authors proposed a heuristic that iteratively decomposes the multiline problem into a series of single-line subproblems, which can be easily solved by the heuristic TADR proposed by Meyr [4]. Two strategies for decomposing the problem into subproblems are proposed: (i) a strategy based on priority rules and (ii) another strategy based on aggregating the original problem, solving the aggregate problem, and disaggregating the results to define the demand and initial inventory for each line. Xiao et al. [17] also examined the parallel-machine lot sizing and scheduling problem with sequence-dependent setup times. In addition, the release times of the items and machine eligibility and preference constraints are considered. However, the setup cost is sequence-independent, and the production costs are not considered. The authors initially propose a MILP model; next, the original problem is decomposed into a lot sizing subproblem and a set of single-machine scheduling subproblems by Lagrangian decomposition. A Lagrangian-based heuristic algorithm, which incorporates the simulated annealing algorithm to improve the solution of the scheduling problem, is proposed.

Considering the GLSPPL with secondary resources, [18] approached a problem in the health-related injection molding industry where a number of resources, available in limited quantities (e.g., grinders and driers), are installed on machines to perform certain tasks. The authors tackled the problem through a two-stage decomposition. In the first phase, the machines are grouped according to their criticality (machines that produce products with fewer machine options to be processed are more critical). In the second phase, the subproblems represented by a MILP model are solved using an open-source solver. Almeder and Almada-Lobo [19] presented MILP models to tackle the synchronization of a secondary resource in lot sizing and scheduling problems with parallel unrelated machines. The machines have to be equipped with a special kind of resource (e.g., a tool) with limited capacity, and multiple products can be produced with the same tool; therefore, the tool changeover costs and times are considered instead of product changeovers. A GLSPPL inspired by a real-world production environment in the food industry was addressed by Soler et al. [20]. In the problem at hand, due to the scarcity of resources, only a subset of production lines can operate simultaneously, and these lines need to be assembled at each production period. Due to this feature, no setup carryover between adjacent periods exists; only a subset of products can be produced in a given period, and for each product, only one production line is capable of producing it. The authors define a branching rule to accelerate the performance of the branch-and-bound algorithm of the CPLEX solver, and a relax-and-fix procedure is also implemented.

More recently, [21] tackled a GLSPPL to maximize the profit of assembled products where pieces are produced using auxiliary equipment (molds) to form finished products. Each piece may be processed in a set of molds with different production rates on various machines. An iterative heuristic that decouples the lot sizing and scheduling decisions is proposed. First, the lot size of the products is determined through the solution of a MILP model where a mold can be used in more than one machine at a time. Next, another MILP model is presented that, based on the lot size and mold-machine assignment previously obtained, allows us to determine the mold schedule in the machines in each planning period. de Armas and Laguna [22] addressed the GLSPPL in the context of pipe insulation manufacturing, in which a limited number of resources must be shared by parallel machines and sets of make-to-stock (MTS) and make-to-order (MTO) units must be considered simultaneously. The stock level of each MTS stock keeping unit (SKU) must be within specified limits at the end of each period, while each MTO SKU must be produced within its specified time frame. The objective is to maximize the total amount of production in a given planning horizon considering sequence-dependent setup times; the setup costs are not considered. The authors propose a two-phase procedure. First, a MIP model in which the setup times are considered independently of the sequence is formulated and solved. Next, a one-pass heuristic is applied to search for setup time savings by reordering the products assigned by the MIP model to each machine in each period. An overview of simultaneous lot sizing and scheduling involving secondary resources can be found in [23].

3 Mathematical Model

In the GLSPPL, n types of products are manufactured on a shop floor composed of m different machines over a horizon divided into T time periods. In a period t (the time interval that extends from instant $t-1$ to instant t), a machine can produce more than one type of product, provided its time availability is not exceeded. Machines have different production rates and efficiency levels. Some machines, due to their manufacturer, model, or preservation status, can achieve a higher production speed with lower scrap rates than others. Consequently, the production time $p_{i\ell}$ and the production cost $c^P_{i\ell}$ to produce product i on machine ℓ depend on the product and the machine. Furthermore, due to technical constraints, not all products can be manufactured by all machines. Between production batches of different types of products, it is necessary to prepare the machine and to set the correct product parameters, which generally generate loss of material. These setup times $e_{ij\ell}$ and, consequently, their respective costs $c^S_{ij\ell}$, are sequence dependent. Preemption is not permitted. The demand d_{it} of each product i at the end of period t is dynamic and deterministic, i.e., it is known and varies over time. Backordering is allowed, but each backordered unit of product i is penalized by g_i per period of delay. Finished goods are transferred from the factory to a centralized distribution center with limited warehousing capacity C^W. Each stored unit of product i costs an inventory cost h_i per period. In the considered problem, in addition to defining the quantities of each product to be produced and the production sequence, the solution determines the machine by which each batch is manufactured to minimize the sum of the costs of inventory, setup, production, and backordering.

The MILP formulation presented in the current section is based on the MILP formulation introduced in [5], which does not consider the allowance of backorders, machine eligibility constraints, or the limited capacity of warehousing. In the formulation, many products can be produced by a machine in a certain period of time. The machine availability is consumed by the time necessary to set up the machine and to produce the lot. The setup time and cost depend on the sequence of products that are produced by the machine. The planning horizon is divided into T periods. Within each period t, each machine ℓ has $w_{\ell t}$ variable-length subperiods. A machine can produce a single type of product within each subperiod, and the duration of the subperiod is given by the duration of the production of the lot that is being produced. There may be subperiods with zero length, even though the machine is set up to produce some product. The division into subperiods determines the sequence of the jobs on each machine and defines the associated setup times and costs.

To present the proposed model, we introduce the following notation:

Main constants:

m: number of machines,
n: number of products,
T: time horizon (assumed to start at 0).

Indexes:

i, j: products,
ℓ: machines,
t: periods (a period t corresponds to the time interval between instants $t-1$ and t),
s: subperiods.

Sets:

$\mathcal{I} = \{1, \dots, n\}$:	products,
$\mathcal{L} = \{1, \dots, m\}$:	machines,
$\mathcal{I}_\ell \subseteq \mathcal{I}$:	products that can be produced by machine ℓ ($\ell \in \mathcal{L}$),
$\mathcal{L}_i \subseteq \mathcal{L}$:	machines that can produce product i ($i \in \mathcal{I}$),

$\mathcal{T} = \{1, \dots, T\}$: periods,

$\mathcal{S}_\ell = \{1, \dots, W_\ell\}$: subperiods of machine ℓ ($\ell \in \mathcal{L}$), where $w_{\ell t}$ is the number of subperiods of machine ℓ within period t ($\ell \in \mathcal{L}$, $t \in \mathcal{T}$) and $W_\ell = \sum_{t \in \mathcal{T}} w_{\ell t}$ ($\ell \in \mathcal{L}$),

$\mathcal{S}_{\ell t} = \{s \in S_\ell \mid \bar{w}_{\ell t} + 1 \leq s \leq \bar{w}_{\ell t} + w_{\ell t}\}$: subperiods of machine ℓ within period t ($\ell \in \mathcal{L}$, $t \in \mathcal{T}$), where $\bar{w}_{\ell t} = 0$ for $t = 1$ and $\bar{w}_{\ell t} = \bar{w}_{\ell, t-1} + w_{\ell, t-1}$ for $t = 2, \dots, T$.

Parameters:

C^W: capacity of warehousing,

$C^P_{\ell t}$: amount of time machine ℓ is available (for production and setup) within period t ($\ell \in \mathcal{L}, t \in \mathcal{T}$),

d_{it}: demand for product i at the end of period t, i.e., at instant t ($i \in \mathcal{I}, t \in \mathcal{T}$),

h_i: inventory cost per period of a unit of product i ($i \in \mathcal{I}$),

g_i: backordering cost per period of a unit of product i ($i \in \mathcal{I}$),

$p_{i\ell}$: time required to produce a unit of product i in machine ℓ ($\ell \in \mathcal{L}, i \in \mathcal{I}_\ell$),

$c^P_{i\ell}$: cost of producing a unit of product i in machine ℓ ($\ell \in \mathcal{L}, i \in \mathcal{I}_\ell$),

$q^{\mathrm{lb}}_{i\ell}$: minimum lot of product i that can be produced in machine ℓ ($\ell \in \mathcal{L}, i \in \mathcal{I}_\ell$),

$e_{ij\ell}$: setup time required to produce product j immediately after i in machine ℓ ($\ell \in \mathcal{L}, i, j \in \mathcal{I}_\ell$),

$c^S_{ij\ell}$: cost of the setup required to produce product j immediately after i in machine ℓ ($\ell \in \mathcal{L}, i, j \in \mathcal{I}_\ell$),

I^+_{i0}: inventory (quantity) of product i at instant $t = 0$ ($i \in \mathcal{I}$),

I^-_{i0}: backordering (quantity) of product i at instant $t = 0$ ($i \in \mathcal{I}$),

$x_{i\ell 0}$: 1, if machine ℓ is prepared to produce product i at instant $t = 0$; 0, otherwise ($\ell \in \mathcal{L}, i \in \mathcal{I}_\ell$).

Variables:

$q_{i\ell s}$: quantity of product i produced in machine ℓ within subperiod s ($\ell \in \mathcal{L}$, $s \in S_\ell, i \in \mathcal{I}_\ell$),

$x_{i\ell s}$: 1, if machine ℓ is prepared to produce product i at the beginning of subperiod s (i.e., at instant $s-1$); 0, otherwise ($\ell \in \mathcal{L}, s \in S_\ell, i \in \mathcal{I}_\ell$),

$y_{ij\ell s}$: 1, if the setup required to produce product j immediately after i in machine ℓ occurs within subperiod s, i.e., between instants $s-1$ and s; 0, otherwise ($\ell \in \mathcal{L}, s \in S_\ell, i,j \in \mathcal{I}_\ell$),

I^+_{it}: inventory (quantity) of product i at the end of period t, i.e., at instant t ($i \in \mathcal{I}$, $t \in \mathcal{T}$),

I^-_{it}: backordering (quantity) of product i at the end of period t, i.e., at instant t ($i \in \mathcal{I}, t \in \mathcal{T}$).

The proposed mathematical formulation is presented below:

$$\text{Minimize}$$

$$\sum_{i\in\mathcal{I}}\sum_{t\in\mathcal{T}} h_i I^+_{it} + \sum_{i\in\mathcal{I}}\sum_{t\in\mathcal{T}} g_i I^-_{it} + \sum_{\ell\in\mathcal{L}}\sum_{i\in\mathcal{I}_\ell}\sum_{j\in\mathcal{I}_\ell}\sum_{s\in S_\ell} c^S_{ij\ell}\, y_{ij\ell s} + \sum_{i\in\mathcal{I}}\sum_{\ell\in\mathcal{L}_i}\sum_{s\in S_\ell} c^P_{i\ell}\, q_{i\ell s} \tag{1}$$

$$\text{subject to}$$

$$I^+_{i,t-1} - I^-_{i,t-1} + \sum_{\ell\in\mathcal{L}_i}\sum_{s\in S_{\ell t}} q_{i\ell s} - I^+_{it} + I^-_{it} = d_{it}, \quad i\in\mathcal{I},\ t\in\mathcal{T}, \tag{2}$$

$$\sum_{i\in\mathcal{I}} I^+_{it} \le C^W, \quad t\in\mathcal{T}, \tag{3}$$

$$\sum_{i\in\mathcal{I}_\ell}\sum_{s\in S_{\ell t}}\left(p_{i\ell}\, q_{i\ell s} + \sum_{j\in\mathcal{I}_\ell} e_{ij\ell}\, y_{ij\ell s}\right) \le C^P_{\ell t}, \quad \ell\in\mathcal{L},\ t\in\mathcal{T}, \tag{4}$$

$$p_{i\ell}\, q_{i\ell s} \le C^P_{\ell t} x_{i\ell s}, \quad \ell\in\mathcal{L},\ t\in\mathcal{T},\ s\in S_{\ell t},\ i\in\mathcal{I}_\ell \tag{5}$$

$$q_{i\ell s} \ge q^{\text{lb}}_{i\ell}\,(x_{i\ell s} - x_{i,\ell,s-1}), \quad \ell\in\mathcal{L},\ t\in\mathcal{T},\ s\in S_{\ell t},\ i\in\mathcal{I}_\ell \tag{6}$$

$$\sum_{i\in\mathcal{I}_\ell} x_{i\ell s} = 1, \quad \ell\in\mathcal{L},\ s\in S_\ell, \tag{7}$$

$$y_{ij\ell s} \ge x_{i\ell,s-1} + x_{j\ell s} - 1, \quad \ell\in\mathcal{L},\ s\in S_\ell,\ i,j\in\mathcal{I}_\ell \tag{8}$$

$$x_{i\ell s} \in \{0,1\}, \quad \ell \in \mathcal{L},\ s \in S_\ell,\ i \in \mathcal{I}_\ell \tag{9}$$

$$q_{i\ell s} \geq 0, \quad \ell \in \mathcal{L},\ s \in S_\ell,\ i \in \mathcal{I}_\ell \tag{10}$$

$$y_{ij\ell s} \geq 0, \quad \ell \in \mathcal{L},\ s \in S_\ell,\ i,j \in \mathcal{I}_\ell \tag{11}$$

$$I^+_{it}, I^-_{it} \geq 0, \quad i \in \mathcal{I},\ t \in \mathcal{T}. \tag{12}$$

Objective function (1) corresponds to the sum of the costs of inventory, backordering, setup, and production. Constraints (2) define inventory and backordering conservation flow, i.e., the relations between inventory, backordering, demand, and produced quantities. Constraints (3) determine that the total amount of inventory cannot exceed the warehousing capacity of the distribution center. In the problem under consideration, the existing storage capacity cannot be increased or violated. (If desired, it would be possible to consider, in the objective function, an additional cost associated with exceeding the current capacity.) Constraints (4) guarantee that the sum of the production time and setup time of each machine within each period does not exceed the corresponding machine's availability. Constraints (5) ensure that machine ℓ produces product i within subperiod s only if the machine has been previously configured for this purpose. Constraints (6) impose a minimum lot size restriction. Constraints (7) determine that, for every machine, within every machine's subperiod, a single product can be produced. Constraints (8) indicate that if the machine ℓ switch from producing product i to producing product j occurs within subperiod s, then the corresponding setup must be accomplished over the course of subperiod s. Constraints (9, 10, 11 and 12) determine the variables' domain. Note that it is not necessary to impose variables $y_{ij\ell s}$ to be binary because they naturally assume values in $\{0, 1\}$ at an optimal solution. This is because constraints (8) restrict them to be larger than or equal to -1, 0, or 1, while constraints (11) inhibit the possibility of the variables being negative. The minimization of the objective function forces them to assume binary values at the optimal solution.

4 Relax-and-Fix Heuristics

The main idea of the relax-and-fix (RF) heuristic [24] for solving a MILP problem is to partition the set X of integer variables into K subsets $X_1, \dots, X_K$ and to solve a sequence of K smaller MILP problems. In the kth subproblem, the variables in X_k are restricted to integers, while the integrality of the variables in $X_{k+1} \cup \cdots \cup X_K$ is relaxed and the variables in $X_1 \cup \cdots \cup X_{k-1}$ are already fixed; see Fig. 2.

Model (1–12), which will be named "Model $\mathcal{M}$" from now on, needs to be modified to be used within the context of an RF-based heuristic. Let $X = \{(i, \ell, s) \mid \ell \in \mathcal{L},\ s \in S_\ell,\ i \in \mathcal{I}_\ell\}$ be the set of all valid indices' 3-uples of variables $x_{i\ell s}$. The RF-based heuristic relies on the partition of the set X into K

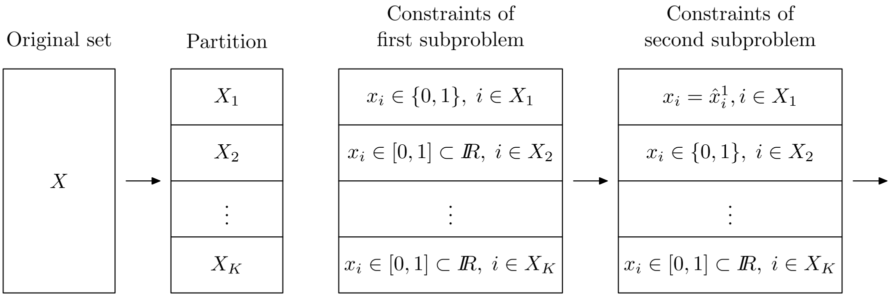
Original set
Partition
Constraints of first subproblem
Constraints of second subproblem
X
X_1
X_2
$\vdots$
X_K
$x_i \in \{0,1\},\ i \in X_1$
$x_i \in [0,1] \subset I\!R,\ i \in X_2$
$\vdots$
$x_i \in [0,1] \subset I\!R,\ i \in X_K$
$x_i = \hat{x}_i^1, i \in X_1$
$x_i \in \{0,1\},\ i \in X_2$
$\vdots$
$x_i \in [0,1] \subset I\!R,\ i \in X_K$

Fig. 2 Relax-and-fix algorithm working structure. In the second subproblem, $\hat{x}_i^1$, $i \in X_1$, correspond to the optimal values found when solving the first subproblem

subsets X_k $(k = 1, \dots, K)$ that verify $\cup_{k=1}^{K} X_k = X$ and $X_{k_1} \cap X_{k_2} = \emptyset$ for all $k_1 \neq k_2$. Let Model $\mathcal{M}_1$ be defined as Model $\mathcal{M}$, in which constraint (9) is substituted with

$$\begin{aligned} &x_{i\ell s} \in \{0,1\},\ (i,\ell,s) \in X_1, \\ &0 \le x_{i\ell s} \le 1,\ (i,\ell,s) \in \cup_{w=2}^{K} X_w. \end{aligned} \tag{13}$$

We can now define Model $\mathcal{M}_k$, for $k = 2, \dots, K$, as Model $\mathcal{M}$ in which constraint (9) is substituted with

$$x_{i\ell s} = \hat{x}_{i\ell s}^{w}, \quad (i,\ell,s) \in X_w \text{ for } w = 1, \dots, k-1, \tag{14}$$

$$x_{i\ell s} \in \{0,1\}, \quad (i,\ell,s) \in X_k, \tag{15}$$

$$0 \le x_{i\ell s} \le 1, \quad (i,\ell,s) \in \cup_{w=k+1}^{K} X_w, \tag{16}$$

where in (14), $\hat{x}_{i\ell s}^{w}$ for $(i,\ell,s) \in X_w$ correspond to the optimal values obtained when solving Models $\mathcal{M}_w$ for $w = 1, \dots, k-1$. This expresses the fact that, in Model $\mathcal{M}_k$, variables $x_{i\ell s}$ with $(i,\ell,s) \in \cup_{w=1}^{k-1} X_w$ are fixed, variables $x_{i\ell s}$ with $(i,\ell,s) \in X_k$ preserve their integrality constraint, and variables $x_{i\ell s}$ with $(i,\ell,s) \in \cup_{w=k+1}^{K} X_w$ are variables whose integrality constraint is relaxed. Note that in Model $\mathcal{M}$, constraint (9), which states that $x_{i\ell s} \in \{0,1\}$ for all $(i,\ell,s) \in X$, implies that all variables $y_{ij\ell s}$, for all $\ell \in \mathcal{L}, \in S_\ell$, $i,j \in \mathcal{I}_\ell$, assume binary values at an optimal solution. On the other hand, a different relation holds in Model $\mathcal{M}_k$ $(k = 1, \dots, K)$. Due to (8), a variable $y_{ij\ell s}$ is guaranteed to assume binary values in an optimal solution of Model $\mathcal{M}_k$ only if $(i,\ell,s-1)$ and $(j,\ell,s) \in X_k$.

The key feature of an RF-based heuristic is the determination of the number of subproblems K and the sets $X_1, \dots, X_K$. Problem-dependent and problem-independent strategies will be considered. The problem-dependent strategies divide variables $x_{i\ell s}$ considering the dimensions related to machine, product, and period, and they are based on several metrics associated with machines, products, and periods that we now describe. We define the demand d_i of a product i as

$$d_i = \sum_{t \in \mathcal{T}} d_{it}; \tag{17}$$

and the flexibility f_i of a product i as the number of machines that can produce it, i.e.,

$$f_i = |\mathcal{L}_i| \tag{18}$$

for all $i \in \mathcal{I}$. In addition, inspired by Vogel's approximation method for the transportation problem [25, 26]), we define the discrepancy a_i of a product i as the difference between its two smallest processing times, given by

$$a_i = p_{i\hat{\ell}_2} - p_{i\hat{\ell}_1}, \ \hat{\ell}_1 = \text{argmin}_{\ell \in \mathcal{L}_i}\{p_{i\ell}\}, \ \text{ and } \hat{\ell}_2 = \text{argmin}_{\ell \in \mathcal{L}_i \setminus \{\hat{\ell}_1\}}\{p_{i\ell}\}. \tag{19}$$

Following [18], we consider a machine to be critical if it is able, and thus it potentially will need to process products with low flexibility. Therefore, we define the criticality ω_ℓ of a machine ℓ as

$$\omega_\ell = m - \min_{i \in \mathcal{I}_\ell}\{f_i\} \tag{20}$$

for all $\ell \in \mathcal{L}$. We also associate with a machine ℓ a metric of efficiency ε_ℓ, given by the sum of the machine's average processing times and costs, i.e.,

$$\varepsilon_\ell = \frac{1}{|\mathcal{I}_\ell|}\left(\sum_{i \in \mathcal{I}_\ell} p_{i\ell} + c^P_{i\ell}\right) \tag{21}$$

for all $\ell \in \mathcal{L}$. Finally, for a period $t \in \mathcal{T}$, we define its overall demand δ_t as

$$\delta_t = \sum_{i \in \mathcal{I}} d_{it}. \tag{22}$$

As a whole, nine different problem-dependent and two problem-independent partition strategies are considered. Each strategy consists of a way of sorting the 3-uples of indices $(i, \ell, s) \in X$, where X is the set of valid 3-uples of the variables $x_{i\ell s}$. After sorting, the first $\lfloor |X|/K \rfloor$ variables constitute the set X_1, the next $\lfloor |X|/K \rfloor$ variables constitute the set X_2, and so on. If $|X|$ is not a multiple of K, then we have that $K\lfloor |X|/K \rfloor < |X|$. Therefore, in the first $r = |X| - K\lfloor |X|/K \rfloor$ subsets, we consider $\lceil \cdot \rceil$ instead of $\lfloor \cdot \rfloor$ for the cardinality of $X_1, \dots, X_r$. In most strategies, the suggested order does not imply a total order of the 3-uples. Therefore, tie-breaking criteria play an important role in the strategies. In the suggested strategies, the tie-breaking criterion will always consist of applying a second strategy. Thus, the strategies are intrinsically hybrid. As a second tie-breaking rule, the lexicographic order of the index 3-uples is used. The description of each strategy follows:

- **Time-dimension-based strategies:**

(S1) Chronological time: Given $(i, \ell, s) \in X$, we have that $s \in \mathcal{S}_{\ell t}$ for some $t \in \mathcal{T}$. 3-uples are sorted in increasing lexicographical order of (t, s), where t is the period t to which subperiod s corresponds.

(S2) Periods with larger demand first: The same as (S1), but 3-uples $(i, \ell, s) \in X$ are sorted in decreasing order of δ_t (instead of increasing order of t), where δ_t is the demand of period t given by (22).

- **Product-dimension-based strategies:**

(S3) Most demanded products first: 3-uples $(i, \ell, s) \in X$ are sorted in decreasing order of the item demands d_i given by (17).

(S4) Less demanded products first: The same as (S3) but in increasing order.

(S5) Less flexible products first: 3-uples $(i, \ell, s) \in X$ are sorted in increasing order of the item flexibility f_i given by (18).

(S6) First, the products with the largest discrepancy between the two shortest production times: 3-uples $(i, \ell, s) \in X$ are sorted in decreasing order of the item discrepancy a_i given by (19).

- **Machine-dimension-based strategies:**

(S7) Less efficient machines first: 3-uples $(i, \ell, s) \in X$ are sorted in increasing order of the machine efficiency ε_ℓ given by (21).

(S8) More efficient machines first: The same as (S7) but in decreasing order.

(S9) More critical machines first: 3-uples $(i, \ell, s) \in X$ are sorted in decreasing order of the machine criticality ω_ℓ given by (20).

- **Problem-independent strategies:**

(S10) More fractional variables first: In this strategy, we first solve a linear programming (LP) problem that corresponds to Model $\mathcal{M}$ in which the constraint $x_{i\ell s} \in \{0, 1\}$ is relaxed to $0 \le x_{i\ell s} \le 1$ for *all* $(i, \ell, s) \in X$. Let $\hat{x}^0_{i\ell s}$, for $(i, \ell, s) \in X$, be the optimal solution of the LP problem. For each variable $x_{i\ell s}$ for $(i, \ell, s) \in X$, we compute the "distance to integrality" given by

$$d^0_{i\ell s} = \min\{\hat{x}^0_{i\ell s}, 1 - \hat{x}^0_{i\ell s}\}.$$

3-uples $(i, \ell, s) \in X$ are sorted in decreasing order of their distance to integrality $d^0_{i\ell s}$. In fact, there is no need to sort all 3-uples since, to construct X_1, only the $\lfloor |X|/K \rfloor$ or $\lceil |X|/K \rceil$ 3-uples with the largest $d^0_{i\ell s}$ are required. In general, after having solved the kth subproblem ($k < K$), distances $d^k_{i\ell s} = \min\{\hat{x}^k_{i\ell s}, 1 - \hat{x}^k_{i\ell s}\}$ for $(i, \ell, s) \in X \setminus \cup_{w=1}^k X_w$ are computed, where $\hat{x}^k_{i\ell s}$ is the optimal solution of the kth subproblem, and the $\lfloor |X|/K \rfloor$ or $\lceil |X|/K \rceil$ 3-uples with the largest $d^k_{i\ell s}$ are selected to constitute X_{k+1}.

(S11) More influential variables first: In this strategy, variables with more "influence" in the objective function are considered first. This "influence" could be directly related to the cost associated with the variable in the objective function, and this is the most direct interpretation of this rule. However, in the problem at hand, variables $x_{i\ell s}$ with $(i, \ell, s) \in X$ do not appear in the objective function. However, their values influence the value of variables $q_{i\ell s}$ and $y_{ij\ell s}$. Thus, the influence $\alpha_{i\ell s}$ of variable $x_{i\ell s}$ is defined as

$$\alpha_{i\ell s} = (\sum_{j \in \mathcal{I}_\ell} c^S_{ij\ell}) + c^P_{i\ell}.$$

Note that, in fact, $\alpha_{i\ell s}$ does not depend on s and, therefore, variables $x_{i\ell s_1}$ and $x_{i\ell s_2}$ with $s_1 \neq s_2$ have the same influence measure. Therefore, in the particular problem at hand, this problem-independent strategy depends on the products and the machines. More specifically, products that take a long time to process and/or demand a time-consuming machine preparation after production are thought to be more influential.

Note that strategy (S10) is a dynamic strategy that differs from all other strategies because set X_k is determined after having solved subproblems from 1 to $k-1$, while other strategies determine all subsets $X_1, \dots, X_K$ a priori. Moreover, strategy (S10) requires solving an LP problem first. In the present work, we consider hybrid strategies that consist of applying a problem-dependent strategy between (S1) and (S9), using (S10) or (S11) as the first tie-breaking rule and the lexicographic order in the 3-uples $(i, \ell, s) \in X$ as the second tie-breaking rule. (Hybrid strategies that combined two problem-dependent strategies were also evaluated numerically, but they showed marginal benefits in relation to those presented in this paper.)

5 Numerical Experiments

In this section, we evaluate the performance of the partition strategies described in the previous section. In the first set of experiments, strategies are evaluated with respect to a set of randomly generated instances. In a second experiment, selected strategies are applied to a set of ten real-world instances of the industry of personal care products. The section ends with a basic sensitivity analysis regarding the storage capacity restriction of the warehouses. The experiments were carried out with an Intel Xeon X5690 3.47 GHz machine with 64 GB of RAM. The RF algorithms and the formulations were solved using CPLEX 12.10.0 using default parameters with the concert library and C++ programming language. The code was compiled using the gcc 6.3.0 compiler with Code::Blocks 16.01 IDE. Benchmark instances and code are available at https://github.com/kennedy94/GLSPPL-RF.

5.1 Experiments with Randomly Generated Instances

The benchmark test suite is composed of 50 randomly generated instances inspired by the production environment of the target industry. All instances have a planning horizon composed of 16 time periods divided into 7 subperiods each. As each period corresponds to 1 week, the planning horizon comprises approximately 4 months. (This does not mean that each subperiod corresponds to 1 day. This means that a machine can produce at most seven different products in a period of 1 week. The subperiods correspond to the production of different products on the machine and are of varying duration.)

Five groups of instances were considered (A, B, C, D and E), and ten different random instances were generated within each of the groups, with a total of 50 instances. The random generation used either discrete or continuous uniform distributions, depending on the nature of each parameter. The number of machines m for groups A, B, C, D, and E is 2, 3, 4, 5, and 7, respectively, while the number of products n for groups A, B, C, D, and E is 8, 12, 16, 20, and 28, respectively. It is considered that the machines are not ready for the production of any item at the beginning of the planning horizon, i.e., $x_{i\ell 0} = 0$ for all $i \in \mathcal{I}$ and $\ell \in \mathcal{L}_i$. Table 1 shows details of the random generation of all instance parameters. It should be noted that the random instances were generated from real-world data. In the table, $\bar{e}$ is the average of the setup times $e_{ij\ell}$ for $\ell \in \mathcal{L}$ and $i,j \in \mathcal{I}_\ell$, and d_t (for $t \in \mathcal{T}$) corresponds to an aggregated period demand. For each item $i \in \mathcal{I}$, a proportion $d_i^p \in [0.05, 0.9]$ is randomly generated (independent of t), and d_{it} is defined as $d_{it} = d_t \, d_i^p / (\sum_{i \in \mathcal{I}} d_i^p)$. The production and setup time intervals for each group were obtained from the average times and standard deviations provided by the company. The range for the minimum lot $q_{i\ell}^{\text{lb}}$ of product i on machine ℓ relative to each group was calculated based on the average and the standard deviation, informed by the industrial area of the company, of the minimum feasible quantity to be produced with productivity and material loss within the required standards. In fact, the average and standard deviation were reported for each group in working shifts $\hat{q}^{\text{lb}}$ and transformed into units of products $q_{i\ell}^{\text{lb}}$ considering the processing times $p_{i\ell}$ and the fact that each working shift corresponds to 8 h of labor, i.e., $q_{i\ell}^{\text{lb}} = 8 \, \hat{q}^{\text{lb}} / p_{i\ell}$. Production costs $c_{i\ell}^P$ (for $i \in \mathcal{I}$ and $\ell \in \mathcal{L}_i$) are random with a continuous uniform distribution within the range given in the table. However, they are then redefined as $c_{i\ell}^P - c_{i\xi_i}^P$, where $\xi_i = \text{argmin}_{\ell \in \mathcal{L}_i} \{c_{i\ell}^P\}$ corresponds to the most efficient machine that produces item i (for $i \in \mathcal{I}$). The ranges for the amounts of products in stock and backordered products at the beginning of the time horizon were calculated using averages and standard deviations based on historical data provided by the company. The last line of each group displays average values. Table 2 shows detailed information about the generated instances. In the table, columns IV, CV, and CO correspond to the numbers of integer variables, continuous variables, and constraints of Model $\mathcal{M}$, respectively.

Initially, to decide which of the problem-independent strategies would be used as the tie-breaking rule, we ran strategies S10 and S11 while varying $K \in 1, 2, \ldots, 10$. The two rows at the bottom of Table 3 show the results. In the table, the results obtained with $K \geq 2$ are compared with the result obtained with $K = 1$, which simply corresponds to solving Model $\mathcal{M}$ using CPLEX. The results appear separated by group and refer to the 10 instances of the group solved with 9 values of $K \in \{2, \dots, 10\}$. Under the heading $(W, G(\%))$ is displayed for how many, out of the 90 cases, the relax-and-fix strategy found a better result (W stands for "win") than CPLEX and what the average gap was in these cases ($G(\%)$ stands for "average gap in percentage"). Under the heading $(L, G(\%))$ appears for how many of the 90 cases the relax-and-fix strategy performed worse (L stands for "lost") than CPLEX and the average gap in these cases. Since there were no ties, if the number of wins and losses does not add up to 90, it is because for some instances and values of K, relax-and-fix could not find a feasible solution within the running time limit. The last column of the table shows the average gap considering the 50 instances and the 9 tested values of K. The results show that strategy S11 obtained better results than strategy S10 and, for this reason, it will be used as a tie-breaker for all the other strategies that depend on the problem. The performance of strategies S1 to S9, using S11 as the tie-breaker strategy, is shown at the top of Table 3. At this point, it is important to mention that for both CPLEX and the relax-and-fix strategy, a time limit of 1 h was used. (The influence of this time limit on the comparison will be analyzed later in this section.) Furthermore, in the relax-and-fix strategy, the time was divided linearly between the subproblems such that the first subproblem has twice as much time as the last. Preliminary tests when supplying three times as much time to the first subproblem and dividing the time evenly among the subproblems showed similar results. The last column of the table shows that, in aggregate, that is, without evaluating the different values of K and the groups of instances individually, all strategies improved upon the results obtained with CPLEX, with gap values ranging from −22% to −37%. The numbers also show that strategy S1 provided the best results. It is also worth noting that S1 found feasible solutions for all instances and all tested values of K.

Table 4 shows the disaggregated results for strategy S1: that is, for each value of K separately. Figure 3 graphically shows the same results shown in Table 4. The numbers in the table show that, except for the case $K = 2$, the results vary little depending on the value of K chosen, which can be considered as a positive feature of the method. The numbers also show that the relax-and-fix strategies with $K \geq 3$ almost always surpass CPLEX, except for some instances of groups A and B that concentrate the smallest instances.

The comparison with CPLEX depends very much on the 1-h time limit being used, since if the instance is small and CPLEX is able to find an optimal solution (regardless of whether it can prove whether the solution is optimal or not), then there is nothing that relax-and-fix can do. Therefore, we decided to test the influence of the time limit on the comparison. To this end, we reran CPLEX and strategy S1 with time limits of 600, 1, 200, 1, 800, 2, 400, 3, 000 and 3, 600 s. We considered S1 with $K = 6$ because Table 4 shows that this was the best value of K for such a strategy. However, this does not mean that we intend to use these values of K in the next experiments, since as already mentioned before, the method is robust and shows small variations with respect to the value of K. Figure 4 shows the results for strategy S1 (with $K = 6$). Figure 4a shows that when the time limit is reduced, the advantage of the relax-and-fix strategy increases. The average gap versus CPLEX,

Table 1 Detailed information of the random generation of instances' parameters

Parameter	Unit of measure	Group				
		A	B	C	D	E
m	unit	2	3	4	5	7
n	unit	8	12	16	20	28
$\lvert\mathcal{I}_\ell\rvert$	unit	[5, 8]	[3, 9]	[4, 10]	[5, 12]	[2, 12]
d_t	thousand units	[9, 13]	[16, 24]	[19, 78]	[27, 220]	[65, 100]
C^W	thousand units	[10, 14]	[18, 20]	[40, 48]	[180, 220]	[120, 150]
$C^P_{\ell t}$	hours	160	160	160	160	160
h_i	cost per unit	[0.27, 0.54]	[0.2, 0.34]	[0.3, 0.87]	[0.07, 0.21]	[0.087, 0.433]
g_i	cost per unit	$[10, 15] \times h_i$	$[10, 15] \times h_i$	$[10, 15] \times h_i$	$[10, 15] \times h_i$	$[10, 15] \times h_i$
$p_{i\ell}$	hours	[0.012, 0.04]	[0.008, 0.05]	[0.007, 0.017]	[0.003, 0.01]	[0.005, 0.028]
$c^P_{i\ell}$	cost per unit	$[0.8, 1.2] \times p_{i\ell}$	$[0.8, 1.2] \times p_{i\ell}$	$[0.8, 1.2] \times p_{i\ell}$	$[0.8, 1.2] \times p_{i\ell}$	$[0.8, 1.2] \times p_{i\ell}$
$\hat{q}^{lb}$	working shifts	[1, 9]	[3, 6]	[3, 6]	[6, 9]	[3, 6]
$e_{ij\ell}$	hours	[2, 6]	[5, 9]	[2, 6]	[1, 6]	[2, 8]
$c^S_{ij\ell}$	cost per setup	$[80, 100] \times e_{ij\ell}/\bar{e}$	$[100, 200] \times e_{ij\ell}/\bar{e}$	$[100, 200] \times e_{ij\ell}/\bar{e}$	$[230, 1200] \times e_{ij\ell}/\bar{e}$	$[150, 620] \times e_{ij\ell}/\bar{e}$
I^+_{i0}	unit	[0, 4000]	[0, 4000]	[0, 20000]	[0, 50000]	[0, 30000]
I^-_{i0}	unit	[0, 500]	[0, 500]	[0, 2000]	[0, 4000]	[0, 4000]

which was −40.62% for the 1-h time limit, extended to −65.79%. This experiment shows that with less time available, using the relax-and-fix strategy is even more advantageous. However, the question that remains is as follows: With less time, how much do the solutions found by relax-and-fix deteriorate? Fig. 4b compares the solution obtained by relax-and-fix with a reduced budget against the solution found with the 1-h budget. The figures show that the maximum deterioration is, on average, approximately 24.23%. The deterioration of the solutions found by CPLEX with reduced time is much greater, which is why the advantage of the relax-and-fix strategy increases.

5.2 Experiments with Real-world Instances

In this section, we apply the presented methods to eight real instances provided by the company. All instances, as well as the randomly generated ones, include 16 periods divided into 7 subperiods. The number of machines varies between 2 and 7, and the number of products varies between 8 and 26. Table 5 shows some details of the instances and their respective models. Recall that in the table, IV, CV and CO stand for “integer variables”, “continuous variables”, and “constraints”. The table also shows, for each instance, the solution found by the company, which was calculated by an expert with an undisclosed empirical method.

The results with the random instances showed that there is no clear advantage of a certain value of K over others (excluding small values of K) and that small variations in the values of K generate small variations in the results. Because of this and because we do not know how the solutions given by the company were calculated, we solved the eight instances with strategy S1 while varying the time limit and varying K, as we did with the random instances. Figure 5 shows the results. On the one hand, the figure shows that, for fixed K, the longer the time limit, the better the solution found. On the other hand, the figure also shows that strategy S1 finds very similar values for any $K \geq 6$. Because of this, we arbitrarily fixed $K = 8$.

Table 6 shows the results obtained by strategy S1, with $K = 8$ and varying time limit. The table also shows the results obtained by CPLEX while also varying the time limit. For each method and time limit, the table shows the solution obtained and the gap in relation to the solution presented by the company, calculated as

$$100 \times \left((F_{\text{method}} - F_{\text{company}})/F_{\text{company}}\right)\%,$$

where F_{method} and F_{company} correspond to the objective function values of the solutions found by the method and the company, respectively. The numbers in the table show that the proposed strategy always improves the company’s solution by an amount that roughly varies, on average, between 40% and 46%, depending on the time limit given for the strategy. The solutions found by CPLEX are, on average, worse than the solutions reported by the company when the timeout is 600 or 1,200 s, while CPLEX improves the company’s solutions on average between 24% and 35% for longer time limits.

Table 2 Detailed information of benchmark random instances

Group A				Group B				Group C				Group D				Group E			
Inst.	IV	CV	CO	Inst.	IV	CV	CO	Inst.	IV	CV	CO	Inst.	IV	CV	CO	Inst.	IV	CV	CO
A1	1456	11346	12832	B1	2016	14646	16720	C1	3136	27438	30576	D1	4928	52208	56976	E1	5488	46604	52096
A2	1568	13154	14736	B2	2128	16680	18848	C2	2912	23596	26544	D2	4480	42716	47120	E2	5376	45926	51312
A3	1456	11346	12832	B3	2128	18036	20192	C3	3024	25178	28224	D3	4592	43620	48128	E3	5712	52706	58368
A4	1456	11346	12832	B4	2352	20070	22432	C4	3024	25856	28896	D4	5152	55824	60784	E4	5712	52028	57696
A5	1344	9990	11376	B5	2240	18714	20976	C5	3024	24500	27552	D5	5040	52434	57312	E5	5600	53384	58928
A6	1456	11346	12832	B6	2464	21426	23888	C6	2464	17268	19824	D6	4480	42942	47344	E6	5488	53158	58592
A7	1232	8408	9696	B7	1344	7414	8880	C7	3472	31280	34720	D7	4816	48818	53504	E7	4144	32140	36416
A8	1120	7052	8240	B8	2128	16454	18624	C8	2688	20432	23184	D8	4816	48366	53056	E8	6496	68978	75280
A9	1344	9764	11152	B9	2016	15324	17392	C9	3024	25856	28896	D9	4480	43846	48240	E9	5152	45248	50416
A10	1344	9990	11376	B10	2128	16680	18848	C10	3024	24952	28000	D10	5040	52660	57536	E10	4704	43666	48400
avg.	1299.2	9312	10659.2	avg.	2016	15459.6	17526.4	avg.	2934.4	23957.6	26924.8	avg.	4726.4	47326.4	51936	avg.	5196.8	48638	54750.4

Table 3 General performance of hybrid strategies S1 to S9 with S11 as a tie-breaking rule plus standalone strategies S10 and S11 on random instances

Strategy	Group A				Group B				Group C				Group D				Group E				G(%)
	(W,G(%)) / (L,G(%))				(W,G(%)) / (L,G(%))				(W,G(%)) / (L,G(%))				(W,G(%)) / (L,G(%))				(W,G(%)) / (L,G(%))				
S1	78	-16.97	12	13.70	66	-18.92	24	16.33	90	-56.96	–		90	-74.22	–		88	-31.79	2	21.61	-36.88
S2	47	-15.36	43	9.70	65	-17.57	25	11.90	85	-53.86	5	6.39	87	-73.07	2	14.61	88	-30.30	2	15.36	-32.55
S3	59	-11.06	31	9.35	66	-16.75	24	11.18	90	-51.49	–		88	-72.58	1	8.43	82	-30.46	8	17.65	-32.53
S4	62	-14.38	28	10.83	70	-18.24	20	9.85	90	-53.48	–		77	-76.13	13	174.59	71	-30.01	14	60.03	-25.54
S5	52	-10.81	38	13.85	68	-15.37	22	17.33	90	-56.15	–		88	-74.30	2	7.08	87	-33.43	3	716.52	-29.02
S6	64	-14.71	26	7.68	70	-17.54	20	7.24	89	-54.37	1	6.08	81	-73.95	9	56.22	76	-29.32	10	228.18	-27.48
S7	39	-10.24	51	11.02	53	-16.97	37	26.38	89	-52.17	1	21.70	89	-69.55	–		82	-28.38	8	49.66	-27.93
S8	56	-12.11	34	17.70	44	-13.30	46	14.58	90	-53.76	–		90	-72.28	–		86	-30.04	4	9.32	-30.93
S9	61	-11.90	29	16.07	79	-16.61	11	10.94	90	-56.96	–		89	-75.18	1	82.84	78	-32.55	12	151.48	-32.13
S10	36	-11.60	54	19.96	30	-11.91	60	16.89	87	-50.98	3	11.73	74	-72.35	9	30.96	62	-28.28	28	11.77	-22.25
S11	63	-12.34	27	18.19	76	-16.64	14	6.22	90	-56.40	–		88	-75.97	1	47.32	79	-28.57	8	282.68	-29.92

Table 4 Detailed performance of strategy S1 with S11 as a tie-breaking rule varying $K \in \{2, 3, \dots, 10\}$ on random instances

Strategy S1																					
K	Group A				Group B				Group C				Group D				Group E				G(%)
	(W,G(%)) / (L,G(%))				(W,G(%)) / (L,G(%))				(W,G(%)) / (L,G(%))				(W,G(%)) / (L,G(%))				(W,G(%)) / (L,G(%))				
2	10	-12.49	–		8	-12.84	2	13.16	10	-48.10	–		10	-62.72	–		8	-24.48	2	21.60	-29.24
3	10	-18.76	–		8	-16.88	2	6.58	10	-55.22	–		10	-69.46	–		10	-28.44	–		-36.81
4	8	-18.09	2	8.44	8	-17.19	2	10.58	10	-56.44	–		10	-71.99	–		10	-26.51	–		-35.87
5	9	-19.98	1	11.54	8	-19.61	2	23.21	10	-59.94	–		10	-75.69	–		10	-33.41	–		-39.38
6	10	-18.48	–		8	-20.88	2	18.09	10	-60.83	–		10	-76.64	–		10	-34.08	–		-40.62
7	9	-16.48	1	0.66	7	-23.43	3	36.48	10	-59.74	–		10	-78.37	–		10	-34.29	–		-38.53
8	7	-17.77	3	22.84	6	-18.90	4	9.04	10	-55.64	–		10	-77.32	–		10	-34.31	–		-36.12
9	7	-17.07	3	12.13	7	-21.36	3	11.82	10	-59.09	–		10	-77.52	–		10	-36.21	–		-38.51
10	8	-13.76	2	15.21	6	-20.47	4	16.90	10	-57.65	–		10	-78.23	–		10	-34.98	–		-36.87

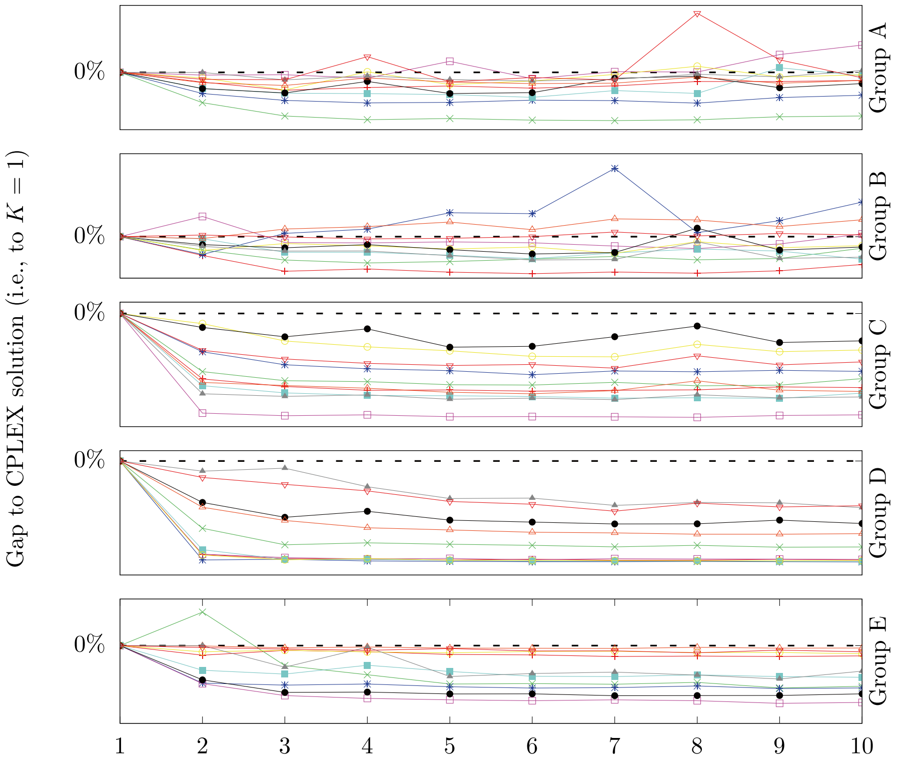

Fig. 3 Gaps to CPLEX solution (i.e., $K = 1$) of strategy S1 on random benchmark instances

Figure 6 shows the data from Table 6 in a different way. In the figure, for each instance, the solution reported by the company corresponds to 100% and the solutions found by the other methods (CPLEX and strategy S1) appear as percentages of the solution reported by the company. In the figure, CPLEX appears in light blue, while strategy S1 appears in violet. Additionally, for each method and each instance, there are 6 thin bars that, from left to right, correspond to the time limits 600, 1,200, … , 3,600. Bars that exceed 100% appear truncated in the figure for presentation purposes, and their true values are indicated with numbers near the top. Note, for example, that in instances P1 and P3, the 6 thin bars of each method have practically the same height. This means that these instances are apparently simple for the methods, which practically find the same solutions regardless of the time limit. For the other instances, the 6 thin bars associated with different time limits seem to have decreasing height from left to right, which shows that, with more time, the methods are able to find better solutions.

Overall, as already determined by examining Table 6, both CPLEX and strategy S1 improve the solutions reported by the company, except for instances P6 and P8, where CPLEX found worse solutions for threshold times below 1,800 s. Regardless, the performances of CPLEX and strategy S1 are similar in instances P1, P2, P3 and P4. In

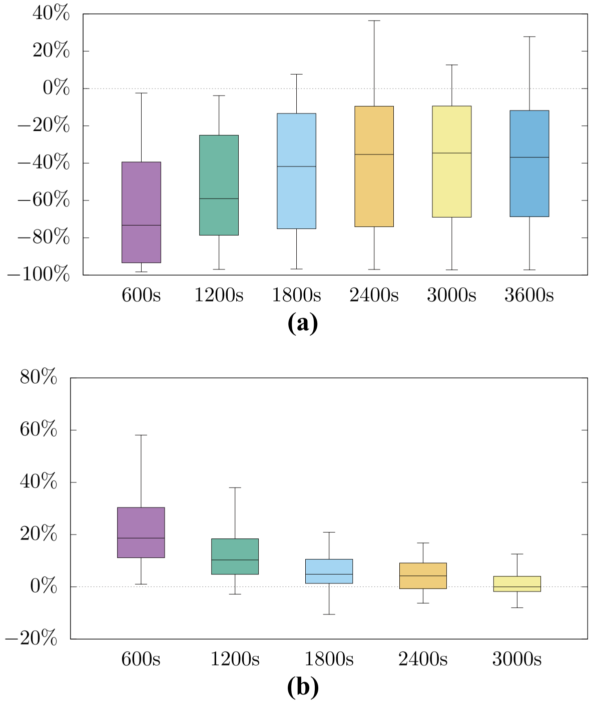

Fig. 4 **a** Boxplot of the gaps between CPLEX and strategy S1 ($K = 6$) applied to the 50 random instances varying the time limits of the methods. (The average gaps are −65.79%, −52.86%,−45.62%, −43.14%, −39.75%, −40.62% for the time limits 600, 1,200, … 3,600, respectively). **b** Boxplot of the gaps between S1 ($K = 6$) applied to the 50 random instances varying its time limit in $\{600, 1{,}200, \ldots, 3{,}000\}$ and S1 ($K = 6$) with a 1-h time limit. (The average gaps are 24.23%, 12.73%, 6.82%, 4.84%, 1.57% for the time limits 600, 1,200, … 3,000, respectively)

these 4 instances, the methods improve the solution presented by the company and return very similar solutions. It is worth noting that these 4 instances are the smallest instances, with dimensions similar to the random instances of groups A and B. In these instances, using CPLEX would be a reasonable alternative. The situation is different in instances P5, P6, P7 and P8, which are the largest within the set considered. In these instances, strategy S1 substantially improves the solutions found by the company in a way that is not matched by CPLEX. Note that even in instances P5, P6 and P8, the solutions found by strategy S1 with the shortest time limit are better than the solutions found by CPLEX with the longest time limit considered.

5.3 Sensitivity Analysis with Respect to Storage Capacity Constraints

In this section, we perform a basic sensitivity analysis with respect to the storage capacity constraint of the warehouse. This analysis is in fact pertinent because, in the problem considered, storage capacities can be modified rather quickly and at low cost. Leasing or unleasing a warehouse can be done in a few weeks, while, for example, buying and installing a new production machine can take 3 to 5 years. Moreover, the storage capacities considered in real-world instances are intentionally large. On the one hand, the cost of inventory comprises the cost of having resources invested in a commodity stalled in inventory, the cost of the physical space itself, and the cost of handling the inventory. Because many of the items produced are of

Table 5 Detailed information of the real-world instances

Inst.	m	n	IV	CV	CO	Company's solution
P1	4	9	2,016	12,058	14,336	1,069,419
P2	3	12	2,016	14,872	16,944	64,706
P3	4	8	2,128	14,284	16,672	754,967
P4	5	13	2,800	20,556	23,600	888,172
P5	2	20	3,584	64,186	67,120	51,740
P6	5	24	4,480	45,338	49,648	903,501
P7	7	26	5,040	40,660	45,792	636,216
P8	7	26	5,264	56,028	61,248	2,301,544

little value, these three components contribute approximately 20%, 40% and 40% to the cost of inventory, respectively. On the other hand, the backordering cost is relatively high in relation to the inventory cost. Because of this, the solution currently adopted by the company is to produce in advance and store stock to minimize the chance of failing to meet the demand.

To carry out the analysis, we considered, as an example, the real-world instance P6. (The scenario is very similar in the other real-world instances.) Solutions were computed with strategy S1 with $K = 8$ and a CPU time limit of 2 h. For the solution found for this instance, the warehouse capacity constraint is inactive for every instant $t \in \mathcal{T}$; and $\max_{i \in \mathcal{I},\, t \in \mathcal{T}} \{I^+_{it}\} = 457{,}395$. Therefore, we redefined $C^W = 457{,}395 - 1$ and solved the instance again. This procedure was repeated until the instance became infeasible. In the particular case of instance P6, the last solved feasible problem had $C^W = 452{,}939$ and $\max_{i \in \mathcal{I},\, t \in \mathcal{T}} \{I^+_{it}\} = 452{,}834$, and the instance is infeasible if we consider $C^W = 452{,}833$. Figure 7 shows the objective function value of the solutions obtained as a function of the considered value for C^W. The first observation is that the warehouse capacity can be reduced from its original value of 650, 000 to

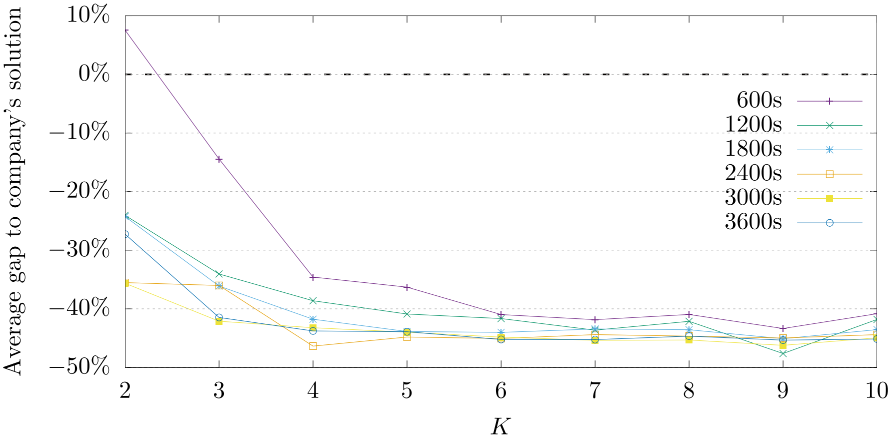

Fig. 5 Average gap to the company's solution of solutions found by strategy S1 varying $K \in \{2, 3, \ldots, 10\}$ and the CPU time limit in $\{600, 1{,}200, \ldots, 3{,}600\}$ s

Table 6 Gap between solutions found by CPLEX and strategy S1 (with $K = 8$) and the company's solutions, with varying CPU time limits in $\{600, 1{,}200, 1{,}800, 2{,}400, 3{,}000, 3{,}600\}$

CPLEX

Inst.	600 s		1200 s		1800 s		2400 s		3000 s		3600 s	
	Objective function value	gap (%)	Objective function value	gap (%)	Objective function value	gap (%)	Objective function value	gap (%)	Objective function value	gap (%)	Objective function value	gap (%)
P1	643,694.00	−39.81	634,612.00	−40.66	634,373.00	−40.68	636,119.00	−40.52	637,932.00	−40.35	635,916.00	−40.54
P2	22,932.20	−64.56	22,495.10	−65.23	20,988.70	−67.56	20,690.20	−68.02	20,775.50	−67.89	20,753.30	−67.93
P3	479,236.00	−36.52	475,105.00	−37.07	474,926.00	−37.09	474,469.00	−37.15	474,895.00	−37.10	473,315.00	−37.31
P4	485,099.00	−45.38	482,778.00	−45.64	480,030.00	−45.95	460,206.00	−48.19	457,134.00	−48.53	450,809.00	−49.24
P5	42,116.70	−18.60	34,643.20	−33.04	34,643.20	−33.04	34,643.20	−33.04	34,643.20	−33.04	34,643.20	−33.04
P6	2,454,100.00	171.62	1,073,460.00	18.81	832,150.00	−7.90	832,150.00	−7.90	832,150.00	−7.90	870,273.00	−3.68
P7	498,090.00	−21.71	413,557.00	−35.00	422,536.00	−33.59	417,686.00	−34.35	414,406.00	−34.86	416,453.00	−34.54
P8	14,885,400.00	546.76	14,885,400.00	546.76	3,935,990.00	71.02	1,975,860.00	−14.15	1,975,860.00	−14.15	1,975,860.00	−14.15
		61.48		38.62		−24.35		−35.42		−35.48		−35.05

Strategy S1 ($K = 8$)

Inst.	600 s		1200 s		1800 s		2400 s		3000 s		3600 s	
	Objective function value	gap (%)	Objective function value	gap (%)	Objective function value	gap (%)	Objective function value	gap (%)	Objective function value	gap (%)	Objective function value	gap (%)
P1	652,170.00	−39.02	648,090.00	−39.40	647,849.00	−39.42	647,849.00	−39.42	644,788.00	−39.71	642,909.00	−39.88
P2	23,447.80	−63.76	23,575.70	−63.56	23,536.40	−63.63	23,588.40	−63.55	23,575.70	−63.56	20,670.90	−68.05
P3	481,900.00	−36.17	478,173.00	−36.66	476,597.00	−36.87	476,597.00	−36.87	476,764.00	−36.85	472,957.00	−37.35
P4	467,198.00	−47.40	456,614.00	−48.59	451,648.00	−49.15	448,914.00	−49.46	453,820.00	−48.90	447,707.00	−49.59
P5	22,689.10	−56.15	19,529.80	−62.25	19,471.70	−62.37	19,000.70	−63.28	18,750.40	−63.76	18,343.40	−64.55
P6	508,391.00	−43.73	536,817.00	−40.58	491,931.00	−45.55	481,842.00	−46.67	480,282.00	−46.84	472,967.00	−47.65

Table 6 (continued)

Strategy S1 ($K = 8$)												
Inst.	600 s		1200 s		1800 s		2400 s		3000 s		3600 s	
	Objective function value	gap (%)	Objective function value	gap (%)	Objective function value	gap (%)	Objective function value	gap (%)	Objective function value	gap (%)	Objective function value	gap (%)
P7	421,610.00	−33.73	421,067.00	−33.82	417,878.00	−34.32	410,308.00	−35.51	417,676.00	−34.35	415,257.00	−34.73
P8	2,126,430.00	−7.61	2,022,190.00	−12.14	1,907,080.00	−17.14	1,782,580.00	−22.55	1,767,330.00	−23.21	1,821,650.00	−20.85
		−40.95		−42.13		−43.56		−44.66		−44.65		−45.33

457, 395 without any impact on the quality of the solution obtained. On the other hand, Fig. 7 shows that reducing C^W from 457, 395 to 452, 834 produces an increase in the objective function value from 451, 067.20 to 494, 141.00. Evidently, this reduction in storage capacity would only be justified if it brought savings of more than 43, 073.80.

Additionally, it should be noted that this analysis does not consider the potentially negative impact of the reduction of warehouse capacity on the inventory cost of products. It would be reasonable to assume that there is a gain of scale when considering large warehouses and that a reduction in the warehouse capacity can increase the 40% share of the physical space that goes into the inventory cost of each product. A deeper parametric analysis, which takes this relationship into consideration, could be performed if the company would effectively be willing to adopt the production scheduling suggested in this study instead of the current policy of production in advance. Furthermore, the analysis could provide a basis for the discussion of contractual backordering costs in the sense that smaller warehouses could be used if the costs for late deliveries were lower or within a certain range.

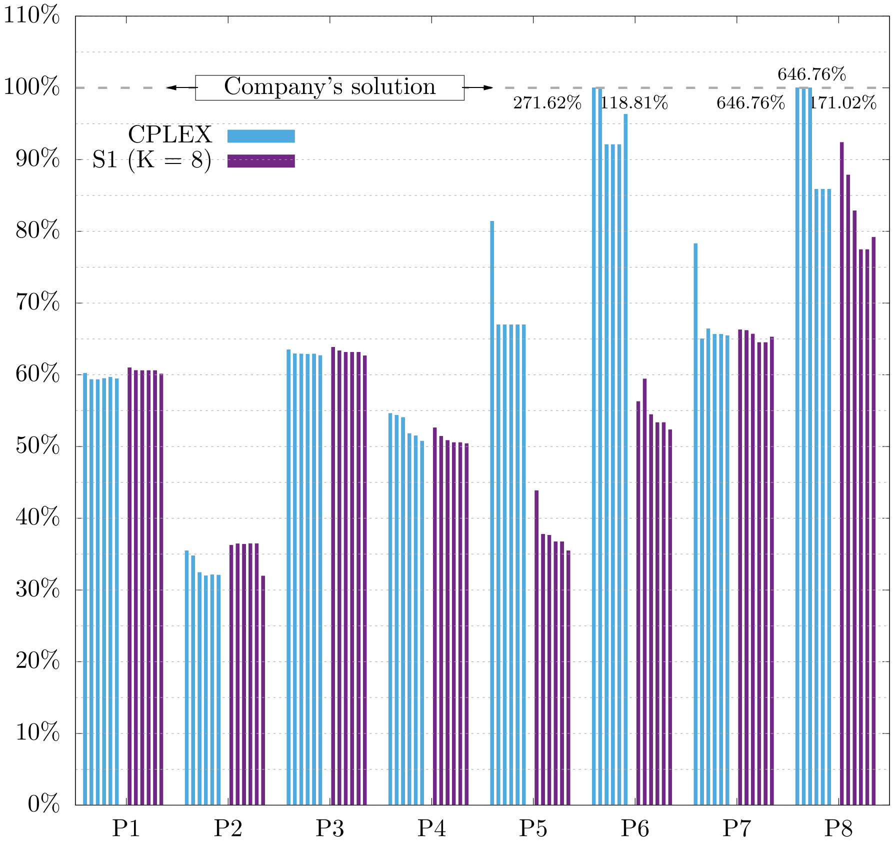

Fig. 6 Solutions obtained with CPLEX and strategy S1 (with $S = 8$) as a percentage of the company's solutions

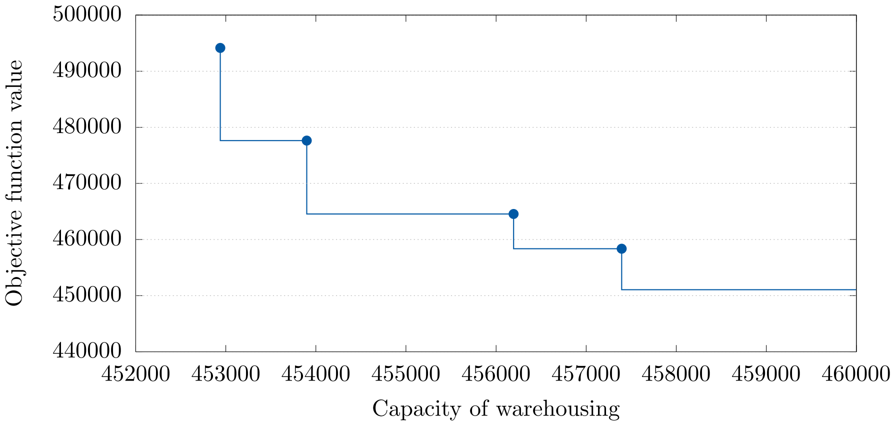

Fig. 7 Objective function value at the solution found as a function of the capacity of warehousing C^W in the real-world instance P6

6 Conclusions

This paper addressed an integrated real-world lot sizing and scheduling problem in a complex operating environment that occurs in a large company in the personal care consumer goods industry. This problem is composed of distinct parallel machines with limited production capacity and sequence-dependent setup times and costs. There is also limited storage capacity for finished goods. To solve this problem, a MILP model was presented, and several problem-dependent and problem-independent strategies based on the relax-and-fix heuristic were developed. The performance of the heuristics was evaluated by solving randomly generated instances and real-world cases. Exhaustive details about the parameters of the real instances are given (see Table 1) in such a way that new random instances can be generated that preserve the characteristics of the real instances. This allows the generation of new test sets to evaluate and compare methods that apply to the problem under consideration.

The relax-and-fix strategies introduced showed the best results overall. Strategy S1, which prioritizes chronological time, showed robustly superior performance compared to the other strategies. In addition, the performance of the relax-and-fix strategies was compared with actual results achieved by the company and the results obtained by solving the MILP model using CPLEX. Compared to the actual solutions used by the company, the strategies produced results with lower costs and a cost reduction between 40% and 46%, mainly from the reduction of inventory levels and better allocation of production lots on the machines. The relax-and-fix heuristics also outperformed CPLEX applied to the MILP model. The proposed strategy confers the advantages of being applicable with minimal technical knowledge and relying on well-established software that solves subproblems. This offers the company confidence in its decision-making. Due to the successful application of the proposed methods, the ideas presented in this work are now being extended to more complex production and logistics environments with multiple distribution centers and factories that are also operated by the target company.

Funding This work has been partially supported by the Brazilian agencies FAPESP (grants 2013/07375-0, 2016/01860-1, and 2018/24293-0) and CNPq (grants 306083/2016-7 and 302682/2019-8).

Data Availability The datasets generated during and/or analyzed during the current study are available in the GitHub repository, https://github.com/kennedy94/GLSPPL-RF.

Declarations

Conflict of Interest The authors declare no competing interests.

References

1. ABIHPEC (2021) Industry Overview of the Brazilian Personal Care, Perfumery and Cosmetics Industry In Portuguese. Available at https://abihpec.org.br/site2019/wp-content/uploads/2021/05/Panorama_do_Setor_Atualizado_Abril-Rev3.pdf. Accessed 21 July 2021
2. Copil K, Wörbelauer M, Meyr H, Tempelmeier H (2017) Simultaneous lotsizing and scheduling problems: a classification and review of models. OR Spectr 39:1–64
3. Fleischmann B, Meyr H (1997) The general lotsizing and scheduling problem. OR Spectr 19:11–21
4. Meyr H (2000) Simultaneous lotsizing and scheduling by combining local search with dual reoptimization. Eur J Oper Res 120:311–326
5. Meyr H (2002) Simultaneous lotsizing and scheduling on parallel machines. Eur J Oper Res 139:277–292
6. Drexl A, Kimms A (1997) Lot sizing and scheduling-survey and extensions. Eur J Oper Res 99:221–235
7. Quadt D, Kuhn H (2008) Capacitated lot-sizing with extensions: A review. 4OR 6:61–83
8. Zhu X, Wilhelm WE (2006) Scheduling and lot sizing with sequence-dependent setup: A literature review. IIE Trans 38:987–1007
9. Robinson P, Narayanan A, Sahin F (2009) Coordinated deterministic dynamic demand lot-sizing problem: A review of models and algorithms. Omega 37:3–15
10. Karimi B, Ghomi SMTF, Wilson JM (2003) The capacitated lot sizing problem: A review of models and algorithms. Omega 31:365–378
11. Kang S, Malik K, Thomas LJ (1999) Lotsizing and scheduling on parallel machines with sequence-dependent setup costs. Manage Sci 45:273–289
12. Clark AR, Clark SJ (2000) Rolling-horizon lot-sizing when set-up times are sequence-dependent. Int J Prod Res 38:2287–2307
13. Beraldi P, Ghiani G, Grieco A, Guerriero E (2008) Rolling-horizon and fix-and-relax heuristics for the parallel machine lot-sizing and scheduling problem with sequence-dependent set-up costs. Comput Oper Res 35:3644–3656
14. Józefowska J, Zimniak A (2008) Optimization tool for short-term production planning and scheduling. Int J Prod Econ 112:109–120
15. Mateus GR, Ravetti MG, de Souza MC, Valeriano TM (2010) Capacitated lot sizing and sequence dependent setup scheduling: an iterative approach for integration. J Sched 13:245–259
16. Meyr H, Mann M (2013) A decomposition approach for the general lotsizing and scheduling problem for parallel production lines. Eur J Oper Res 229:718–731
17. Xiao J, Yang H, Zhang C, Zheng L, Gupta J (2015) A hybrid lagrangian-simulated annealing-based heuristic for the parallel-machine capacitated lot-sizing and scheduling problem with sequence-dependent setup times. Comput Oper Res 63:72–82
18. Dastidar SG, Nagi R (2005) Scheduling injection molding operations with multiple resource constraints and sequence dependent setup times and costs. Comput Oper Res 32:2987–3005
19. Almeder C, Almada-Lobo B (2011) Synchronisation of scarce resources for a parallel machine lot-sizing problem. Int J Prod Res 49:7315–7335
20. Soler WO, Santos MO, Akartunali K (2019) MIP approaches for a lot sizing and scheduling problem on multiple production lines with scarce resources, temporary workstations, and perishable products. J Oper Res Soc, to appear

21. Ríos-Solís YÁ, Ibarra-Rojas OJ, Cabo M, Possani E (2020) A heuristic based on mathematical programming for a lot-sizing and scheduling problem in mold-injection production. Eur J Oper Res 284:861–873
22. de Armas J, Laguna M (2020) Parallel machine, capacitated lot-sizing and scheduling for the pipe-insulation industry. Int J Prod Res 58:800–817
23. Wörbelauer M, Meyr H, Almada-Lobo B (2019) Simultaneous lotsizing and scheduling considering secondary resources: a general model, literature review and classification. OR Spectr 41:1–43
24. Pochet Y, Wolsey LA (2006) Production Planning by Mixed Integer Programming. Series in Operations Research and Financial. Springer Verlag, New York, NY
25. Glover F, Karney D, Klingman D, Napier A (1974) A computation study on start procedures, basis change criteria, and solution algorithms for transportation problems. Manage Sci 20:793–813
26. Reinfeld NV, Vogel WR (1958) Mathematical Programming. Prentice-Hall, Englewood Cliffs, NJ

Publisher's Note Springer Nature remains neutral with regard to jurisdictional claims in published maps and institutional affiliations.

Authors and Affiliations

K. A. G. Araujo[1] · E. G. Birgin[2] · M. S. Kawamura[3] · D. P. Ronconi[3]

✉ E. G. Birgin
egbirgin@ime.usp.br

K. A. G. Araujo
kennedy94@ime.usp.br

M. S. Kawamura
kawamurams@yahoo.com.br

D. P. Ronconi
dronconi@usp.br

[1] Department of Applied Mathematics, Institute of Mathematics and Statistics, University of São Paulo, Rua do Matão, 1010, Cidade Universitária, 05508-090 São Paulo, SP, Brazil

[2] Department of Computer Science, Institute of Mathematics and Statistics, University of São Paulo, Rua do Matão, 1010, Cidade Universitária, 05508-090 São Paulo, SP, Brazil

[3] Department of Production Engineering, Polytechnic School, University of São Paulo, Av. Prof. Luciano Gualberto, 1380, Cidade Universitária, 05508-010 São Paulo, SP, Brazil