

## DECOMPOSITION BASED HEURISTICS FOR A LOT SIZING AND SCHEDULING PROBLEM ON MULTIPLE HETEROGENEOUS PRODUCTION LINES WITH PERISHABLE PRODUCTS

Willy A. de Oliveira Soler<sup>1\*</sup>, Maristela O. Santos<sup>2</sup> and Kerem Akartunalı<sup>3</sup>

Received July 1, 2020 / Accepted March 10, 2021

**ABSTRACT.** In this paper, we propose a novel MIP-based heuristic method to deal with a lot sizing and scheduling problem with multiple heterogeneous production lines in a production setting with perishable items. The problem is inspired by the production processes adopted by some Brazilian food industries and it considers that several production lines share the same scarce production resources. Therefore, only a subset of those lines can simultaneously operate in each production period. Moreover, the production environment is characterized by the existence of sequence-dependent setup times and costs, and by the production of perishable items which can be stocked for a short period only. Firstly, we propose a facility location reformulation for a model previously proposed in the literature. Secondly, we propose a heuristic composed of two phases. The first phase has an elaborated approach to building feasible solutions solving initially an aggregated lot sizing problem to decide which production lines to assemble, followed by the resolution of the various single line lot sizing and scheduling problems. The second phase applies improvement heuristics exploring principles of fix-and-optimize and local branching procedures. Computational results carried out using a data set proposed in the literature are presented in order to study the efficiency of the proposed approach. The results demonstrate that our heuristics provide superior results when benchmarked with a heuristic from the literature specifically developed to solve the problem under consideration, and with a commercial MIP solver.

**Keywords:** lot sizing and scheduling, scarce resources, perishable products, decomposition heuristic.

---

\*Corresponding author

<sup>1</sup>Institute of Mathematics, INMA, Federal University of Mato Grosso do Sul, Av. Costa e Silva, s/n, 79070-900, Campo Grande, MS, Brazil – E-mail: willy.oliveira@ufms.br – <http://orcid.org/0000-0002-7467-2307>

<sup>2</sup>Department of Applied Mathematics and Statistics, Institute of Mathematics and Computer Sciences, University of São Paulo, Av. Trabalhador São-Carlense, 400, 13566-590, São Carlos, SP, Brazil – E-mail: mari@icmc.usp.br – <http://orcid.org/0000-0003-0822-3439>

<sup>3</sup>Department of Management Science, University of Strathclyde, Glasgow, Scotland, United Kingdom – E-mail: kerem.akartunalı@strath.ac.uk – <http://orcid.org/0000-0003-0169-3833>

## 1 INTRODUCTION

In periods of economic crisis, the development of good production plans can be crucial for the survival of a company in the market. However, as highlighted in Almada-Lobo et al. (2015), various industries face problems to plan their operations at minimum cost. In fact, production planning, unless simplified to unrealistic settings, consists of  $\mathcal{NP}$ -hard optimization problems, and many real-world problems are computationally challenging. Such a setting requires specialized solution approaches to provide reasonable feasible solutions at acceptable computational time, as widely observed in the literature (see, e.g., Claassen et al. (2016), Akartunalı et al. (2016), Aouam et al. (2018), Soler et al. (2019), and Soler et al. (2019) for recent examples). For the specific domain of lot sizing and production planning, we refer the interested reader to Brahimi et al. (2017) for an extensive recent survey of single-item problems, to Doostmohammadi & Akartunalı (2018) for a thorough overview of complex multi-item problems, and to Absi & van den Heuvel (2019) for a review of effective relax-and-fix methods in this domain.

For some industries such as food and beverage, the production plan involves the simultaneous determination of the production quantities (lot sizes) and the sequence of production in different production lines with the aim of minimising costs and ensuring to fulfill the customer's demands, as noted by Baldo et al. (2014). In other words, it is necessary to solve the simultaneous lot sizing and scheduling problem (LSP).

In this paper, we consider an LSP on multiple production lines apparent in some Brazilian food industries. This problem was recently introduced by Soler et al. (2019) and it considers a production environment where scarce production resources, like machines, tools and workers, need to be used to assemble the production lines. Due to the scarcity of those resources and the impossibility of acquiring additional resources (as discussed in detail in Soler et al. (2019)), only a subset of the lines can be assembled in each period. Therefore, besides the decisions about lot sizing and scheduling, the managers need to decide which production lines to assemble in each period. Moreover, the considered food industry produces various types of perishable items and it is desirable to avoid that the products be discarded due to expiration of its shelf life. The problem is also characterized by allowing backlogging and by the existence of sequence-dependent setup times and costs. In the literature, this problem is known as capacitated lot-sizing with sequence dependent setup costs, perishable products, scarce resources and multiple production lines (CLSDPRL).

Beyond the mentioned food industry, production planning problems involving the synchronization of scarce resources across multiple production lines also appear in other production environments, such as the semiconductor industry, production and bottling of soft drinks (Almeder & Almada-Lobo (2011)), automotive industry (Güngör et al. (2018)) and block erection in a shipyard (Afzalirad & Rezaeian (2016)).

An MIP model and a relax-and-fix heuristic that efficiently explore the structure of the problem were proposed in Soler et al. (2019) to deal with the CLSDPRL. The computational study presented in the paper evidenced that the addressed problem is challenging from the computational

perspective and that the consideration of scarce resource and perishability of the items makes the problem significantly challenging to solve in comparison to the traditional LSP. Although the proposed relax-and-fix heuristic delivered good feasible solutions and competitive dual bounds (outperforming a commercial solver and an earlier established relax-and-fix procedure from the literature), the results indicated the necessity of new solution approaches to deal with large-sized test instances from real-world scenarios.

Therefore, in this paper we develop specific heuristic approaches to deal with the considered problem that outperform the earlier algorithms proposed in Soler et al. (2019), mainly for large-sized test instances, with respect to the quality of the feasible solutions and the necessary computational time. More specifically, we develop a construction procedure that decomposes the original problem into various small problems that can be easily solved by exact algorithms like branch-and-cut, and then we combine principles of fix-and-optimize heuristic and local branching to obtain improvement procedures applied to the feasible solutions obtained from the construction phase.

The paper is organised as follows: In Section 2, we present a brief literature review, while in Section 3 we present two MIP models for the addressed problem. In Section 4, we present the construction heuristic based on the decomposition of the original problem, while in Section 5 we present the improvement heuristics. The computational study carried out using a data set from the literature is presented in Section 6. Finally, in Section 7 some conclusions and research directions for future studies are presented.

## 2 RELATED RESEARCH AND PROBLEM DESCRIPTION

The lot sizing and scheduling problem (LSP) consists of determining the quantities to be produced of each product and the sequence in which these products are produced in each period, with the aim to ensure the fulfillment of customer demand while minimizing the costs incurred in the production process. This problem naturally appears in various production environments, such as brewery industry (Baldo et al. (2014)), food and drink (Kopanos et al. (2011), Ferreira et al. (2009)) and glass container industry (Almada-Lobo et al. (2008)).

The pioneer mathematical models for the LSP are the capacitated lot-sizing and scheduling problem - CLSP (Karmarkar & Schrage (1985)), the discrete lot-sizing and scheduling problem - DLSP (Fleischmann (1990)), the proportional lot-sizing and scheduling problem - PLSP (Drexl & Haase (1995)), the capacitated lot-sizing with sequence dependent setup costs - CLSD (Haase (1996)), and the general lot-sizing and scheduling problem - GLSP (Fleischmann & Meyr (1997)). The first three models adopt restrictive assumptions limiting to one or two items produced in a period, while the last two models are generalizations of the first ones and can be viewed as general framework models that can be easily extended to consider characteristics observed in real-world problems. Therefore, most LSP models presented in the literature are adaptations of the CLSD and GLSP models (Copil et al. (2017)).

The computational performance of CLSD and GLSP based models was studied in Almeder & Almada-Lobo (2011) and Guimarães et al. (2014) using branch-and-bound algorithms of a commercial MIP solver. In both works, the computational results showed that the CLSD based models perform much better than the GLSP models obtaining best feasible solutions and dual bounds and shorter running times. Furthermore, Guimarães et al. (2014) studied some reformulations of the CLSD model using different constraints to eliminate subtours instead of the MTZ constraints originally used by Haase (1996). More recently, Oliveira & Santos (2017) proposed a simple reformulation for the CLSD model that consists of using explicit binary variables to indicate whether each item is produced in each period. Moreover, the authors proposed a branching rule to guide branch-and-bound algorithms that can significantly speed up the computational performance of the CLSD model.

In this paper, we study an LSP on multiple production lines apparent in some Brazilian meat companies. This problem was recently introduced by Soler et al. (2019) considering industries in which various types of production resources (like machines, tools, and workers) need to be used to assemble the production lines. Due to the scarcity of those resources, only a subset of the lines can simultaneously operate. The assembly of lines requires various operational procedures that consume significant production time. On the other hand, this type of industry usually operates eight hours a day, and a general cleaning of the machines and tools need to be carried out at the end of each production day in order to avoid contamination, requiring the disassembly of the production lines. Therefore, in this problem, the production lines must be assembled only at the start of each production period (a day) and remain working until the end of the period when they must be disassembled for the cleaning process.

Moreover, in this type of industry the production lines are specialized and therefore, for each item there is only one line able to produce it. Therefore, the choice of lines to be assembled impacts the set of products that can be produced in each period. The considered industry produces perishable products (meat), which is a very important feature requiring specific attention. More specifically, this type of industry has an extensive product catalogue and each product has its own shelf-life. The shelf-lives can significantly vary, for example, frozen products can be stocked for some months, while seasoned meat products can be stocked only for a few days. Finally, Soler et al. (2019) observed the existence of significant sequence-dependent setup costs and times incurred by changeovers between items produced, and backlogging is allowed, albeit with high penalties.

In Soler et al. (2019) the CLSDPRL model - an extension of the CLSD model - is proposed to deal with the problem considered in this paper. More specifically, the CLSDPRL model uses knapsack constraints to model the production resources utilization and a network based approach to control the age of each product in stock avoiding that the items be damaged by the shelf-life. The reformulation and branching rule suggested in Oliveira & Santos (2017) were also incorporated in the CLSDPRL model. Moreover, Soler et al. (2019) developed a specific relax-and-fix heuristic decomposing the set of the binary variables of the CLSDPRL model into subsets ordered with decreasing significance. Computational results carried out with a set of instances inspired by real-world scenarios showed that the heuristic proposed in Soler et al. (2019) is able to

produce good quality feasible solutions and competitive dual bounds, significantly outperforming a commercial MIP solver and an earlier relax-and-fix heuristic from the literature.

We observe that to the best of our knowledge only Soler et al. (2019) studied the CLSDPRL, but there are some relevant works in the literature addressing similar characteristics that we summarize next, highlighting the differences from the CLSDPRL:

- An LSP with sequence-dependent setup times and backlogging inspired by process industries was studied in Smith-Daniels & Ritzman (1988). This problem does not consider scarce production resources and perishable products;
- The work of Özdamar & Birbil (1998) addressed a simultaneous lot sizing and loading problem on multiple facilities (that we can consider as production lines). The scheduling part of the problem is not considered, and there are no scarce resources. Moreover, the products are not perishable;
- An LSP on multiple production lines that share scarce secondary resources was studied in Almeder & Almada-Lobo (2011). Those secondary resources are used to equip the production lines in order to produce specific items. Therefore, all production lines can simultaneously operate, but only a subset of the products can be produced in each configuration of the lines. They also consider backlogging, but perishability is not considered;
- The work of Afzalirad & Rezaeian (2016) considered a pure scheduling problem with sequence dependent setup times and scarce secondary resources. They also used knapsack constraints to manage the resource utilization. As in Almeder & Almada-Lobo (2011), all production lines can simultaneously operate. The lot sizing aspect of the problem and perishable products are not considered;
- Similarly to Afzalirad & Rezaeian (2016), the study presented in Villa et al. (2018) also considered a scheduling problem with scarce resources without considering lot sizing decisions and perishable products;
- The study presented in Güngör et al. (2018) addresses an LSP on parallel identical production lines that need to be equipped with secondary resources to produce the items. The problem assumes cumulative demands that should be met only at the end of the planning horizon. Besides that, only one product can be produced in each period and the all-or-nothing assumption holds. Assembly of production lines and perishable products are not considered.

Besides the synchronisation of scarce production resource for assembling production lines, another important characteristic of the CLSDPRL is the perishability assumption. In the literature, there exist relevant researches addressing production planning problems involving perishable products. For example, Amorim et al. (2013) studied a simultaneous production and distribution planning problem considering perishable products, sequence dependent setup times and costs,

and delivery time windows. More specifically, they propose two mathematical models in order to compare the batching of orders with lot sizing decisions when perishability is considered. They showed that lot sizing is more promising than batching for make-to-order systems.

In Costa et al. (2014) a vegetable crop supply problem was addressed. The main characteristics of this problem are the perishability of products and the backlogging assumption. The authors developed MIP models able to map the age of the products in stock and propose a column generation algorithm to solve the problem motivated by the very large number of variables used in the proposed model. Moreover, they considered both deterministic and stochastic demands. The lot sizing and scheduling problem considering the characteristics of the food industry was studied in Pires et al. (2015). More specifically, they proposed a base model considering general characteristics of the food companies and then they developed modeling techniques in order to consider perishability of the products, behavior of the customers, discarding costs, value of freshness and age dependent demand. Similarly to Costa et al. (2014), the models developed in Pires et al. (2015) also control the age of the products in stock. Through a comparative analysis, the authors showed that the perishable nature of the food products may be crucial to reduce costs and improve the service level.

Finally, Wei et al. (2019) studied the multi-level lot sizing and scheduling problem considering that the raw materials, intermediates and end products may be perishable. They extend a classical MIP model to consider these perishability assumptions and, through a computational study, they showed the importance of considering the perishability aspect for industrial environments in which this phenomenon occurs. They also observed that the perishability assumption significantly increases the complexity of the multi-level lot sizing and scheduling problem.

### 3 MATHEMATICAL MODELS

In this Section, we present two mathematical models for the addressed problem. Firstly, we just exhibit the CLSDPRL model introduced in Soler et al. (2019). Secondly, motivated by the consideration of perishable products, we develop the facility location reformulation (CLSDPRL<sup>FL</sup>) of the CLSDPRL model.

Table 1 summarizes the necessary parameters and variables (including their domains). The CLSDPRL model consists of minimise the objective function (1) subject to constraints (2) to (20).

**Table 1** – Parameters and variables of the CLSDPRL model.

Parameters	
$T, L$	Set of periods (indexed by $t, p$ ), and of production lines (indexed by $l$ )
$J, K$	Set of items (indexed by $i, j$ ), and of resources (indexed by $k$ )
$d_{jt}, C_{lt}$	Demand of item $j$ , and production capacity of line $l$ , in period $t$
$a_{lj}, m_{lj}$	Per unit production time, and the minimum lotsize of item $j$ on line $l$
$h_j, b_j$	Inventory and backloging costs per unit of item $j$
$sc_{lij}, st_{lij}$	Setup cost and time for changeover from item $i$ to $j$ on line $l$
$r_{kl}$	Amount of resource $k$ necessary to assemble the production line $l$
$R_{kt}$	Capacity of resource $k$ in period $t$
$sl_j, ac_l$	Shelf life of item $j$ , and cost to assemble the line $l$
$\gamma_t$	Maximum number of items that can be produced on line $l$ in period $t$
$P_l$	Set of items that can be produced on line $l$
Variables	
$ve_{jt}^{t'} \geq 0$	Stock of product $j$ with age $t'$ available in period $t$ before meeting the customer demands, where $t' \leq \min\{t, sl_j\}$
$q_{jt}^{t'} \geq 0$	Quantity of item $j$ with age $t'$ effectively used in period $t$ to meet the customer demands
$I_{jt}, B_{jt} \geq 0$	Inventory and backloging of item $j$ at the end of period $t$
$x_{ljt} \geq 0$	Amount of item $j$ produced on line $l$ and period $t$
$V_{ljt} \geq 0$	Auxiliary variables to represent the order of production of item $j$ on line $l$ and period $t$
$y_{ljt} \in \{0, 1\}$	1 if item $j$ is the first item produced on line $l$ and period $t$ , and 0 otherwise
$w_{ljt} \in \{0, 1\}$	1 if item $j$ is produced on line $l$ and period $t$ , and 0 otherwise
$z_{lijt} \in \{0, 1\}$	1 if there is change of production from item $i$ to $j$ on line $l$ and period $t$ , and 0 otherwise
$\delta_{lt} \in \{0, 1\}$	1 if the line $l$ is assembled in period $t$ , and 0 otherwise

CLSDPRL:

$$\text{Min } \sum_{t,j} (h_j I_{jt} + b_j B_{jt}) + \sum_{l,t,i,j} sc_{lij} z_{lijt} + \sum_{l,t} ac_l \delta_{lt} \quad (1)$$

$$\text{s.t. } \sum_{t'=0}^{sl_j} q'_{jt} = d_{jt} + B_{j,t-1} - B_{jt}, \forall j, t \quad (2)$$

$$ve'_{j,t+1} = ve'_{jt} - q'_{jt}, \forall j, t, t' \leq sl_j - 1 \quad (3)$$

$$ve'_{jt} = \sum_l x_{ljt}, \forall j, t \quad (4)$$

$$q'_{jt} \leq ve'_{jt}, \forall j, t, t' \leq sl_j \quad (5)$$

$$I_{jt} = \sum_{t'=0}^{sl_j} ve'_{jt} - \sum_{t'=0}^{sl_j} q'_{jt}, \forall j, t \quad (6)$$

$$B_{jT} = 0, \forall j \quad (7)$$

$$\sum_{j \in P_l} a_{lj} x_{ljt} + \sum_{i,j} st_{lij} z_{lijt} \leq C_{lt}, \forall l, t \quad (8)$$

$$m_{lj} w_{ljt} \leq x_{ljt} \leq \frac{C_{lt}}{a_{lj}} w_{ljt}, \forall l, j \in P_l, t \quad (9)$$

$$\sum_{j \in P_l} w_{ljt} \leq \gamma_{lt} \delta_{lt}, \forall l, t \quad (10)$$

$$\sum_l r_{kl} \delta_{lt} \leq R_k, \forall k, t \quad (11)$$

$$w_{ljt} = y_{ljt} + \sum_{i \in P_l} z_{lijt}, \forall l, j \in P_l, t \quad (12)$$

$$\sum_{j \in P_l} y_{ljt} \leq 1, \forall l, t \quad (13)$$

$$y_{ljt} + \sum_{i \in P_l} z_{lijt} \geq \sum_{i \in P_l} z_{ljit}, \forall l, j \in P_l, t \quad (14)$$

$$V_{ljt} \geq V_{lit} + 1 - \gamma_{lt}(1 - z_{lijt}), \forall l, t, i, j \in P_l \quad (15)$$

$$I_{jt} \geq 0, B_{jt} \geq 0, \forall j, t \quad (16)$$

$$\delta_{lt} \in \{0, 1\}, \forall l, t \quad (17)$$

$$x_{ljt} \geq 0, V_{ljt} \geq 0, \forall l, j, t \quad (18)$$

$$y_{ljt} \in \{0, 1\}, w_{ljt} \in \{0, 1\}, \forall l, j, t \quad (19)$$

$$q'_{jt} \geq 0, ve'_{jt} \geq 0, \forall j, t, t' \leq \min\{t, sl_j\} \quad (20)$$

The objective function (1) represents the sum of the inventory holding, backlogging, and setup costs, and costs to assemble the production lines. Constraints (2) are the inventory balance constraints, while the perishability aspect is modeled by constraints (3)-(5). More specifically, constraints (3) iteratively update the age of the products in stock, while constraints (4) compute the



amount of freshly produced stock and constraints (5) ensure that amount used of a particular age does not exceed the available stock of that age. Constraints (6) compute the total inventory of each item for each period, and constraints (7) ensure all demands are satisfied at the end of the planning horizon. Constraints (8) represent the capacity constraints for production lines, while (9) are logical constraints to ensure that a setup occurs if the production of item  $j$  takes place on line  $l$  and period  $t$ . Constraints (10) ensure that the items can only be produced on assembled lines, and constraints (11) are the capacity constraints for production resources. Constraints (12) state that the produced items should be sequenced and constraints (13) dictate that if a production line is assembled, only one item can be the first item produced on that line. Constraints (14) ensure the flow balance for sequencing of lots, and (15) are the subtour elimination constraints. Finally, constraints (16), (17), (18), (19) and (20) define the domain of the decision variables.

We highlight that the approach used to model the perishability aspect consists of controlling the age of the products in stock. This approach was firstly introduced by Costa et al. (2014) and adapted by Soler et al. (2019) for the CLSDPRL. The variable  $ve'_{jt}$  indicates the amount of product  $j$  with age  $t'$  available before meeting the demand of period  $t$ . For example,  $ve^0_{jt}$  denotes the amount of fresh product of type  $j$  produced in period  $t$ . Therefore, we have constraints (4). Moreover, variable  $q'_{jt}$  denotes the amount of item  $j$  with age  $t'$  effectively used in period  $t$  to meet the customer demands. Obviously, the used products can not exceed the amount of available products, hence we have constraints (5). Note that, for each product  $j$ , the amount of items in stock with age  $t' + 1$  in period  $t + 1$  comes from the products with age  $t'$  that were stocked and were not used to meet customer demands in period  $t$ . Therefore, we have constraints (3).

The inventory balance are given by constraints (2). More specifically, these constraints ensure that the number of items effectively used in each period must be equal to the customer demands plus the overdue balance ( $B_{j,t-1} - B_{jt}$ ). Constraints (6) just capture the amount of each item in stock in each period. It is necessary to compute the inventory holding costs in the objective function. These constraints state that the amount of item  $j$  in stock at the end of period  $t$  is given by the sum of amounts in stock with each age  $t' \leq \min\{t, sl_j\}$  minus the amount effectively used in period  $t$  to meet customer demands. Finally, constraints (7) are introduced to ensure that all demands are satisfied until the end of the planning horizon. This assumption is realistic for meat companies, since backlogging may result in loss of customers.

We note that the perishability aspect suggests the reformulation of the CLSDPRL model using the classical facility location approach. Therefore, in this paper, we also study the facility location reformulation CLSDPRL<sup>FL</sup>. As will be discussed later in Section 6, this reformulation can significantly benefit our construction heuristic.

To state the CLSDPRL<sup>FL</sup> model, we introduce the continuous variables  $x_{ljtp}$  to indicate the amount of item  $j$  produced on line  $l$  during period  $t$  to meet the demand of period  $p$ . With these variables we can easily avoid that product  $j$  be stocked for longer than its shelf-life ( $sl_j$ ), as  $x_{ljtp} = 0$  when  $t < p - sl_j$  can be used for preprocessing to eliminate such variables. Therefore, it is not necessary using the variables  $ve'_{jt}, q'_{jt}, I_{jt}$ , and  $B_{jt}$ . Moreover, we define the parameter  $hb_{jtp}$

to represent the inventory holding (if  $t < p$ ) or backlogging (if  $t > p$ ) cost incurred by producing one unit of item  $j$  in period  $t$  to be used in period  $p$ . The parameter  $hb_{jtp}$  is computed according to (21).

$$hb_{jtp} = \begin{cases} (p-t)h_j, & \text{if } t \leq p \\ (t-p)b_j, & \text{if } t > p \end{cases} \quad (21)$$

The CLSDPRL<sup>FL</sup> is obtained adapting the CLSDPRL model according to the redefinition of the variables  $x$ . More specifically, the CLSDPRL<sup>FL</sup> is stated as follows:

CLSDPRL<sup>FL</sup> :

$$\text{Min} \sum_{l,j,t,p} hb_{jtp}x_{ljtp} + \sum_{l,t,i,j} sc_{lij}z_{lijt} + \sum_{l,t} ac_l\delta_{lt} \quad (22)$$

$$\text{s.t.} \quad \sum_{p=\max\{1,t-sl_j\}}^T x_{ljtp} = d_{jt}, \forall l, j \in P_l, t \quad (23)$$

$$\sum_{j \in P_l, p} a_{lj}x_{ljtp} + \sum_{l,j} st_{lij}z_{lijt} \leq C_{lt}, \forall l, t \quad (24)$$

$$m_{lj}w_{ljt} \leq \sum_p x_{ljtp} \leq \frac{C_{lt}}{a_{lj}}w_{ljt}, \forall l, j \in P_l, t \quad (25)$$

$$x_{ljtp} \geq 0, \forall l, j, t, p \quad (26)$$

(10), (11), (12), (13), (14), (15), (17) and (19)

#### 4 DECOMPOSITION BASED CONSTRUCTION HEURISTIC

Decomposition based heuristics usually take into account the specific structure of the addressed problem in order to decompose it into easier-to-solve sub-problems. In Kim et al. (2010) a decomposition based heuristic was proposed to deal with a lot sizing and scheduling problem considering setup carry-over and setup overlapping apparent in process industries that operate 24 hours a day and 7 days a week. Moreover, Cordeau et al. (2015) proposed a decomposition based heuristic for an inventory routing problem that decomposes the original problem into an inventory (planning) and a routing sub-problems. Besides that, the authors developed a reoptimisation phase in which a feedback model is applied in order to improve the obtained solutions. Finally, we note that a similar decomposition of the problem was explored in the relax-and-fix heuristic proposed in Soler et al. (2019).

In this Section, we propose a decomposition based heuristic, simply referred to as D in the remainder of the paper, with the aim of quickly building feasible solutions for the addressed problem. More specifically, D is a MIP based heuristic that decomposes the original problem into  $L + 1$  small sub-problems and it is composed of two main steps. In the first step, we solve an aggregated model (AG) that is easier to solve than the original problem in order to decide which production lines to assemble in each period, while in the second step, we fix the values of  $\delta_{lt}, \forall l, t$  from the value obtained in the first step and solve the resulting sub-problem (SP). We observe that in the sub-problem SP there is no data dependency between the production lines and then

SP can be further decomposed into  $L$  ( $SP^l, l = 1, \dots, L$ ) single machine LSPs that can be quickly solved by a MIP solver.

In the AG model the sequence decisions are not considered and we use the mean setup times ( $ST_{lj}$ ) and costs ( $SC_{lj}$ ) instead of the original sequence-dependent setups ( $st_{lij}$  and  $sc_{lij}$ ) to ensure an aggregated consideration of the costs and of the capacity consumption relating to the setup procedures. Therefore, the AG model is a lot sizing problem considering the perishability and scarce resources aspects.

Before presenting the AG model, we note that similarly to the CLSDPRL model, the D heuristic also admits a facility location version  $D^{FL}$ . For simplicity, in this section we just present sub-problems related to the  $D^{FL}$  (the sub-problems of the D heuristic can be analogously obtained). The AG model consists of minimising the objective function (27) subject to constraints (28), (10), (11), (23), and (25) where the parameters  $ST_{lj}$  and  $SC_{lj}$  are defined by (29).

$$\text{Min } \sum_{l,j,t,p} hb_{jtp}x_{ljtp} + \sum_{l,t,i,j} SC_{lj}w_{ljt} + \sum_{l,t} ac_l \delta_{lt} \tag{27}$$

$$\text{s.t. } \sum_{j \in P_l,p} a_{lj}x_{ljtp} + \sum_j ST_{lj}w_{ljt} \leq C_{lt}, \forall l,t \tag{28}$$

$$(10), (11), (23), \text{ and } (25)$$

where

$$ST_{lj} = \frac{\sum_{i \in P_l} st_{lij}}{|P_l|}, \forall l, j \quad \text{and} \quad SC_{lj} = \frac{\sum_{i \in P_l} sc_{lij}}{|P_l|}, \forall l, j. \tag{29}$$

Let  $\bar{\delta} = (\bar{\delta}_{lt})$  be a feasible solution of the AG model. Fixing the value of each binary variable  $\delta_{lt}$  from the value  $\bar{\delta}_{lt}$  in the CLSDPRL (or CLSDPRL<sup>FL</sup>) model we obtain a sub-problem  $SP(\bar{\delta})$ . Since  $\bar{\delta}$  satisfy constraints (11), those constraints are not necessary in the  $SP(\bar{\delta})$  model. Moreover, for each product there is only one production line able to produce it, and  $P_l, l = 1, \dots, L$  is a family of disjoint sets. Therefore, in the  $SP(\bar{\delta})$  model there is no data dependency between different production lines, and then  $SP(\bar{\delta})$  can be further decomposed into  $L$  sub-problems ( $SP^l(\bar{\delta}), l = 1, \dots, L$ ), each with a single production line and only the periods where  $\bar{\delta}_{lt} = 0$  does not hold. Moreover, each sub-problem  $SP^l$  considers only  $|P_l|$  products rather than  $J = \sum_l |P_l|$  products considered in the original problem. The  $SP^l(\bar{\delta})$  model is presented in the Appendix and the D heuristic is summarized in Algorithm 1.

In line 3 of the Algorithm 1, the function  $AG(\text{MIP\_solver}, \text{AG\_MAX\_time})$  consists of solving the AG model using an algorithm from a software  $\text{MIP\_solver}$  with the maximum running time limited by  $\text{AG\_MAX\_time}$ . In line 6,  $SP^l(\bar{\delta}, \text{MIP\_solver}, \text{SPL\_MAX\_time})$  consists of solving the model  $SP^l(\bar{\delta})$  with the running time limited in  $\text{SPL\_MAX\_time}$ . Therefore, we can control the maximum computational time consumed by our D heuristic. Moreover, in this study we use

**Algorithm 1** D heuristic

---

```

1: Given parameters AG_MAX_time, SPL_MAX_time;
2: Set FO  $\leftarrow$  0;
3: Solve AG(MIP_solver, AG_MAX_time);
4: Capture the AG-feasible solution  $\bar{\delta}$ ;
5: for  $l = 1, \dots, L$  do
6:   Solve  $SP^l(\bar{\delta}, \text{MIP\_solver}, \text{SPL\_MAX\_time})$ ;
7:   if  $SP^l(\bar{\delta}, \text{MIP\_solver}, \text{SPL\_MAX\_time})$  admits solution then
8:     Capture the objective function value  $f^l$ ;
9:     FO  $\leftarrow$  FO +  $f^l$ ;
10:  else
11:    Stop. D heuristic can not to find a feasible solution;
12:  end if
13:  Capture the solutions  $\bar{x}_{ljp}, \forall j \in P_l, t, p, \bar{w}_{ljt}, \forall j \in P_l, t, \bar{y}_{ljt}, \forall j \in P_l, t$ , and  $\bar{z}_{lit}, \forall i, j \in P_l, t$ ;
14: end for

```

---

the IBM Cplex as the MIP\_solver. In Section 6 we present computational results using different values for parameters AG\_MAX\_tme and SPL\_MAX.time.

We highlight that the proposed D heuristic was able to find feasible solutions for all test instances used in this study as reported in Section 6. However, since the aggregated model AG uses the mean setup times, we cannot theoretically guarantee that D heuristic can provide a solution for all feasible test instances. Therefore, in lines 7 to 11 of the Algorithm 1, we include a condition for the cases in which D heuristic fails to provide feasible solutions. In these cases, the user can set alternative values for parameters  $ST_{lj}$  and  $SC_{lj}$ .

Finally, we note that a similar decomposition of the problem was explored in the relax-and-fix heuristic (RFH) proposed in Soler et al. (2019). However, the computational results presented in Section 6 show that our D heuristic significantly outperforms the RFH with respect to the observed running times and objective function value. This fact occurs because the aggregated model AG proposed in this paper is able to provide better assemble patterns of the production lines than the RFH, primarily for large-sized test instances.

## 5 IMPROVEMENT HEURISTICS

In this Section, we present two MIP-based heuristics in order to improve the feasible solutions built by our D heuristic. More specifically, we propose a stochastic fix-and-optimize heuristic that decomposes the problem by periods and a local search algorithm induced by local branching constraints.

## 5.1 Stochastic fix-and-optimise

Fix-and-optimise (FO) heuristics have been used to solve a range of production planning problems. Some successful examples are described in James & Almada-Lobo (2011) for single and parallel machine LSP, in Toledo et al. (2015) for multi-level lot sizing problems, and in Soler et al. (2019) for an LSP with demand choice flexibility. The FO heuristic was proposed by Pochet & Wolsey (2006) and it starts with a feasible solution on hand and iteratively looks for a new solution by solving a small subproblem (SSP) in which just a few binary variables of the original problem (and all possible continuous variables) are optimised, while the other binary variables have their values fixed in the incumbent value.

In this paper, we propose a stochastic FO heuristic that can be viewed as a particular case of the heuristics proposed in James & Almada-Lobo (2011) and Soler et al. (2019) in which only one neighborhood structure is considered. More specifically, the proposed heuristic decomposes the problem by periods and, in each iteration, we randomly select a set  $S$  composed of the binary variables associated to  $k^{FO}$  adjacent periods to be optimised, while the binary variables in the set  $BV \setminus S$  have their value fixed, where  $BV$  is the set of all binary variables of the addressed problem. Initially, each set  $S$  has the same probability of being selected. However, according to the frequency  $f_S$  in which set  $S$  have been selected, its selection probability is updated according to the function  $P(S) = e^{-\frac{f_S}{\lambda}}$ , where  $\lambda \geq 1$  is a parameter empirically determined. Moreover, the algorithm also avoids the case of the same set being selected in two consecutive iterations.

We note that the SSP subproblems obtained in each FO iteration are simply solved by a commercial solver within a specified maximum time `SSP_MAX_time`. We have established two stop criteria for the FO heuristic: i) three consecutive iterations without improvement in the objective function value; and ii) reaching of the maximum running time.

Finally, Algorithm 2 details the proposed FO heuristic. In line 16 of the Algorithm 2, the function `SSP( $S, \bar{X}$ , MIP_solver, SSP_MAX_time)` consists of solving the subproblem SSP using a MIP solver with the maximum running time fixed by `SSP_MAX_time`. The SSP is obtained from the CLSDPRL model by fixing the value of the binary variables in the set  $BV \setminus S$  in incumbent value  $\bar{X}$ .

In Algorithm 2, the repetition structure defined by lines (7) to (14) is introduced with the aim of selecting a subset  $S$  of the binary variables of problem to be optimized in the current iteration. This structure ensures that a randomly selected set  $S$  is accepted with probability  $P(S) = e^{-\frac{f_S}{\lambda}}$  and the loop is interrupted only when a set  $S$  is accepted.

## 5.2 Local branching based heuristic

In this paper, we also study a simple neighborhood search algorithm induced by local branching constraints. This local branching heuristic (LB) is a deterministic improvement approach that can be used alternatively to the proposed (stochastic) FO heuristic. More specifically, the LB heuristic starts with a feasible solution on hand and in each iteration it explores a neighborhood

**Algorithm 2** FO heuristic

---

```

1: Let  $\bar{X} = (\bar{\delta}, \bar{w}, \bar{y}, \bar{z})$  be a feasible solution;
2: Let  $f_o$  be the incumbent objective function value;
3: Let  $BV$  the set composed of all binary variables of the problem;
4:  $f_S \leftarrow 0, \forall S \subset BV$ ;
5: while Stop criteria are not reached do
6:   accept  $\leftarrow 0$ ;
7:   while accept = 0 do
8:     Randomly choose a set  $S$  composed of all binary variables associated with  $k^{FO}$ 
       contiguous periods;
9:     Randomly choose  $\alpha \in [0, 1]$ ;
10:    if  $\alpha < e^{-\frac{f_S}{\lambda}}$  then
11:      accept  $\leftarrow 1$ ;
12:       $f_S \leftarrow f_S + 1$ ;
13:    end if
14:  end while
15:  Solve SSP( $S, \bar{X}$ , MIP_solver, SSP_MAX_time) obtaining the objective function value  $f$ 
    and the solution  $X^{SSP}$ ;
16:  if  $f < f_o$  then
17:     $f_o \leftarrow f$ ;
18:     $\bar{X} \leftarrow X^{SSP}$ ;
19:  end if
20:  Update stop criteria;
21: end while
22: return  $f_o$ ;

```

---

of the incumbent solution looking for new feasible solutions. The explored neighborhoods are defined according to the local branching constraints that were introduced in Fischetti & Lodi (2003).

Since the choice of the lines to be assembled impacts the set of products that can be produced, the binary variables  $\delta = (\delta_{it})$  play an important role in the solution process deciding the value of some other binary variables. Besides that, as observed in our D heuristic, when we fix the value of the  $\delta$  variables we obtain a subproblem that can be effectively solved by a commercial MIP solver. Therefore, in our LB heuristic we emphasize these variables to define the neighborhoods. We observe that a similar distinction between binary variables in a local branching based heuristic was explored in Fischetti et al. (2004) for a telecommunication network design problem.

Let  $\bar{\delta} = (\bar{\delta}_{it})$  be a feasible solution for the binary variables  $\delta$ . Adapting the definition presented in Fischetti et al. (2004), the  $k^{LB}$ -OPT neighborhood of the solution  $\bar{\delta}$  (indicated by  $\Delta(\bar{\delta}, k^{LB})$ )

is composed of the matrices  $\delta'$  that differ from  $\bar{\delta}$  in the value of at maximum  $k^{LB}$  entries. More specifically,  $\delta' \in \Delta(\bar{\delta}, k^{LB})$  when  $\delta'$  satisfies constraints (30).

$$\sum_{l,t:\bar{\delta}_{lt}=1} (1 - \delta_{lt}) + \sum_{l,t:\bar{\delta}_{lt}=0} \delta_{lt} \leq k^{LB}. \tag{30}$$

Therefore, in each iteration of our LB heuristic, we just include constraints (30) (defined according to the incumbent solution  $\bar{\delta}$ ) in the original CLSDPRL (or CLSDPRL<sup>FL</sup>) model and solve the resulting subproblem  $SLB(\bar{\delta}, k^{LB})$ . Fischetti et al. (2004) observed that adopting small values for parameter  $k$  (e.g.,  $k^{LB} \in \{2, 4, 6\}$ ), the subproblem  $SLB(\bar{\delta}, k^{LB})$  can be much easier to solve than the original problem using a MIP solver. We also observe that according to constraints (13), (10) and (30) we can show that (31) and (32) are valid constraints for subproblem  $SLB(\bar{\delta}, k)$ .

$$\sum_{l,t:\bar{\delta}_{lt}=1} \left( 1 - \sum_{j \in P_l} y_{ljt} \right) + \sum_{l,t:\bar{\delta}_{lt}=0} \left( \sum_{j \in P_l} y_{ljt} \right) \leq k^{LB}, \tag{31}$$

$$\sum_{l,t:\bar{\delta}_{lt}=1} \left( 1 - \sum_{j \in P_l} w_{ljt} \right) + \sum_{l,t:\bar{\delta}_{lt}=0} \left( \sum_{j \in P_l} w_{ljt} \right) \leq k^{LB} \gamma_{lt}. \tag{32}$$

Note that, the incumbent solution  $\bar{\delta}$  belongs to its neighborhood  $\Delta(\bar{\delta}, k^{LB}), \forall k$ , i.e.,  $\bar{\delta}$  is a feasible solution for subproblem  $SLB(\bar{\delta}, k^{LB})$ . Therefore, in order to avoid an algorithmic stagnation, we also add the tabu constraints (33) to each subproblem  $SLB(\bar{\delta}, k^{LB})$ .

$$\sum_{l,t:\bar{\delta}_{lt}=1} (1 - \delta_{lt}) + \sum_{l,t:\bar{\delta}_{lt}=0} \delta_{lt} \geq 1. \tag{33}$$

Finally, we specify two stopping criteria for the LB heuristic: i) the observation of a cycle of solutions, i.e., when the heuristic obtains a feasible solution that was earlier obtained; and ii) reaching the maximum running time. Algorithm 3 presents a pseudo-code for the proposed LB heuristic.

## 6 COMPUTATIONAL RESULTS

In this section, we present computational results in order to study the efficiency of the proposed approaches. We observe that the used test instances that are described in Section 6.1 and the computational implementation of the proposed heuristics (in C programming language) are available at the home page <https://inma.ufms.br/docentes/willy-alves-de-oliveira/willy/>.

### 6.1 Test data

We present a computational study in order to evaluate the effectiveness of the proposed model and heuristics. The computational study is carried out using a data set proposed by Soler et al. (2019) and composed of 100 test instances. According to Soler et al. (2019), the data set covers

---

**Algorithm 3** LB heuristic

---

- 1: Let  $\bar{X} = (\bar{\delta}, \bar{w}, \bar{y}, \bar{z})$  be a feasible solution;
  - 2: Let  $fo$  be the incumbent objective function value;
  - 3: Let  $k^{LB}$  be a integer parameter specified by the user;
  - 4:  $Best\_fo \leftarrow fo$ ;
  - 5: **while** Stop criteria are not reached **do**
  - 6:   Update constraints (30), (31), (32), and (33) according to the incumbent solution  $\bar{\delta}$ ;
  - 7:   Solve SLB( $\bar{\delta}, k^{LB}$ ) obtaining a solution  $X$  with objective function value  $fo$ ;
  - 8:    $\bar{X} \leftarrow X$ ;
  - 9:   **if**  $fo < Best\_fo$  **then**
  - 10:      $Best\_fo \leftarrow fo$ ;
  - 11:   **end if**
  - 12:   Update stop criteria;
  - 13: **end while**
  - 14: **return**  $Best\_fo$ ;
- 

a broad range of scenarios and was generated in line with real-world data observed in some meat industry companies.

Next, we present a brief description of the considered test instances and we note that a detailed discussion about the adopted parameter settings is presented in Soler et al. (2019). We use the notation  $p \in U[a, b]$  to indicate that we generated integer parameters using uniform distributions.

- Number of production periods:  $|T| \in \{10, 12, 14\}$ ; number of production lines:  $|L| \in \{7, 10\}$ ; number of products:  $|J| \in \{45, 80, 90, 110\}$ ; and number of scarce resources types:  $|K| \in \{5, 6, 7\}$ .
- Production capacity:  $C_{it} = 480$ ; unitary processing time:  $a_{ij} = 1$ ; and minimum lot sizes:  $m_{ij} = 2$ .
- Inventory holding costs:  $h_j \in U[1, 10]$ ; and backlogging costs:  $b_j = 10h_j$ .
- Setup times:  $st_{ij} \in U[15, 45]$ ; and setup costs:  $sc_{lij} = 2st_{ij}$ .
- Demand of the products:  $d_{jt} \in U \left[ 0, \frac{C_{it} - \min\{st_{lij}\}\gamma - \varphi^d}{|P_t|} \right]$ , with  $\varphi^d$  as specified in Table 2.
- Amount of resources necessary to assemble the lines:  $r_{kl} \in U[0, 2], k > 0$  and  $r_{0l} \in U[5, 10]$  (resource  $k = 0$  represents the workers).
- Capacity of the scarce production resources:  $R_{kt} = \max \left\{ \max_{l=1, \dots, L} \{r_{kl}\}, \varphi_k^r \sum_{l \in [L]} r_{kl} \right\}$ , with  $\varphi_k^r$  as specified in Table 2.



- Shelf life of the products:  $sl_j \in U[4, T]$ .
- Costs to assemble the production lines:  $ac_l = \sum_k rc_k r_{kl}$ , where  $rc_k \in U[\phi^b, \phi^e]$  with  $\phi^b$  and  $\phi^e$  as specified in Table 2.
- Maximum number of different products that can be produced in each line and period:  $\gamma_{lt} = \gamma$ , with  $\gamma$  as specified in Table 2.
- Set of products that can be produce on each line:  $P_l$  built by randomly allocating each item to a line.

Finally, we remark that the test instances are grouped into five classes, each with 20 test instances representing a specific industry size. The parameter settings adopted in each class are presented in Table 2.

**Table 2** – Characteristics of the five classes of test instances.

Class	$ T $	$ L $	$ J $	$ K $	$\varphi^d$	$\varphi_k^r$	$\varphi_0^r$	$\phi^b$	$\phi^e$	$\gamma$
1	10	7	45	5	100	0.8	0.6	0	0	6
2	10	10	80	6	100	0.8	0.6	0	0	8
3	14	10	90	6	90	0.6	0.5	0	0	8
4	12	10	110	7	90	0.6	0.55	100	150	8
5	14	10	110	7	90	0.6	0.55	50	150	8

## 6.2 Computational resources and efficiency measures

The proposed model, heuristics and the benchmark approaches from the literature were implemented in C++ language using Concert Technology of the MIP solver Cplex 12.6. The computational experiments were run on a computer with two processors Intel Xeon E5-2680v2, 2.8GHz, 10 cores, 2 threads/core, 25 MB SmartCache, and 128 GB RAM memory. The models and sub-problems related to the heuristic approaches were solved by the Cplex software using default settings suitably limiting the maximum running time.

In general, for each instance, we have used two efficiency measures to evaluate the computational performance of the studied methods:

- the percentage deviation (gap) of the best obtained feasible solution ( $z_f$ ) from the known dual bound ( $\bar{z}$ ), i.e.,  $gap = 100 \left( \frac{z_f - \bar{z}}{z_f} \right)$ . Obviously, dual bounds can be obtained from Cplex when solving the models CLSDPRL and CLSDPRL<sup>FL</sup>. However, we highlight that the RFH heuristic can also provide dual bounds as proved in Soler et al. (2019). Moreover, the dual bounds obtained from the RFH heuristic significantly outperform the dual bounds obtained from Cplex for some large sized test instances. In this paper, we have used the best of these three obtained dual bounds to compute the gaps; and

- the observed running time (in seconds).

For each class of instances and solution method, we have reported the average (AG), the best (BG), and the worst (WG) gaps, the standard deviation of the gaps (SDG), the average running time (RT), and the number of instances in which, at least, a feasible solution was obtained (FS).

### 6.3 Numerical tests

In computational tests, we first analyze the performance of the constructive heuristics and the proposed models considering different time limits. Then, we analyze the solutions considering the improvement heuristics.

#### 6.3.1 Analysis of mathematical models and constructive heuristics

In Table 3 we analyse the computational performance of the models CLSDPRL and CLSDPRL<sup>FL</sup> (simply referred to as M and M<sup>FL</sup>, respectively) of the D heuristic and its facility location version D<sup>FL</sup>, both proposed in this paper, and of the RFH algorithm proposed by Soler et al. (2019). We have performed tests adopting different maximum running times in order to perform a sensitivity analysis. In Table 3, the adopted maximum running times in seconds are noted as a subscript in the name of each method.

Considering the performance of the CLSDPRL and the CLSDPRL<sup>FL</sup> (columns M and M<sup>FL</sup>, respectively), we can see that both models presented similar results for classes 1 and 2, while CLSDPRL model presented significantly better results than CLSDPRL<sup>FL</sup> for classes 3 and 4. Although CLSDPRL<sup>FL</sup> presented better gaps for class 5, we note that CLSDPRL was able to find feasible solutions for more test instances justifying the observed gap difference. Therefore, we conclude that the facility location reformulation of the original model was not a viable strategy for solving large-sized test instances, likely due to the higher number of variables required to define CLSDPRL<sup>FL</sup>.

We also note that both models faced difficulties to find feasible solutions for some large-sized test instances from classes 4 and 5. More specifically, CLSDPRL provided feasible solutions for 16 and 14 test instances from classes 4 and 5, while CLSDPRL<sup>FL</sup> for 13 and 12 test instances, respectively. Finally, we observe that the obtained mean of the average gaps (AG) ranging around 30% for both models clearly justify the development of specialized heuristic approaches. Note that, in Table 4, we present the average objective function values (OF) and the average dual bounds (DB) obtained for each model, which demonstrates that CLSDPRL<sup>FL</sup> presented slightly better dual bounds than CLSDPRL for all classes.

Table 3 also presents the results for the D heuristics and its facility location version D<sup>FL</sup>, successively fixing the maximum running time ( $T_{lim}$ ) to 900, 1800, and 3600 seconds. For all adopted values for parameter  $T_{lim}$ , D<sup>FL</sup> presented superior results than D with respect to the solution quality. Therefore, different than the behavior observed for the MIP models, the facility location reformulation significantly benefited our D heuristic.

**Table 3** – Computational results for the MIP models and for the constructive procedures.

		Models		Literature		D heuristics					
		M <sub>3600</sub>	M <sub>3600</sub> <sup>FL</sup>	RFH <sub>900</sub>	RFH <sub>3600</sub>	D <sub>900</sub>	D <sub>900</sub> <sup>FL</sup>	D <sub>1800</sub>	D <sub>1800</sub> <sup>FL</sup>	D <sub>3600</sub>	D <sub>3600</sub> <sup>FL</sup>
C1	AG	0.02	0.01	0.09	0.08	0.35	0.35	0.35	0.35	0.35	0.35
	BG	0.00	0.00	0.01	0.00	0.00	0.00	0.00	0.00	0.00	0.00
	WG	0.06	0.06	1.35	1.35	2.38	2.38	2.38	2.38	2.38	2.38
	RT	935	851	48	48	23	19	23	19	23	19
	FS	20	20	20	20	20	20	20	20	20	20
C2	AG	0.09	0.09	0.15	0.12	0.82	0.82	0.82	0.82	0.82	0.82
	BG	0.00	0.00	0.03	0.00	0.04	0.04	0.04	0.04	0.04	0.04
	WG	0.27	0.23	0.51	0.50	3.63	3.63	3.63	3.63	3.63	3.63
	RT	3600	3399	426	426	137	88	137	88	137	88
	FS	20	20	20	20	20	20	20	20	20	20
C3	AG	35.26	59.70	29.71	10.79	20.58	13.42	13.55	9.78	10.64	9.53
	BG	11.14	19.19	4.44	1.93	2.24	1.93	2.24	2.61	2.61	2.61
	WG	72.59	83.90	80.77	36.26	43.35	43.52	37.28	27.16	27.16	27.16
	RT	3600	3600	844	3538	702	708	1605	1615	3424	3416
	FS	20	20	20	20	20	20	20	20	20	20
C4	AG	48.80	56.90	31.31	9.01	13.10	10.43	10.96	9.06	9.80	8.69
	BG	15.40	27.58	6.61	3.05	3.95	4.65	3.95	4.71	5.21	3.74
	WG	74.62	70.93	55.24	16.68	26.11	20.66	19.13	17.51	14.23	17.51
	RT	3600	3600	900	3600	762	766	1668	1671	3473	3481
	FS	16	13	19	20	20	20	20	20	20	20
C5	AG	65.40	57.64	49.60	15.26	15.73	12.59	12.92	9.89	10.16	10.21
	BG	51.50	28.77	30.85	5.64	10.73	5.54	7.85	5.44	5.45	4.72
	WG	86.14	78.86	70.27	26.49	26.72	19.55	19.97	15.37	14.03	16.68
	RT	3600	3600	900	3600	827	835	1716	1741	3527	3544
	FS	14	12	15	20	20	20	20	20	20	20
Mean	AG	29.91	30.91	22.17	7.05	10.11	7.52	7.72	5.98	6.36	5.92
	BG	15.61	15.11	0.00	2.13	3.39	2.43	2.81	2.56	2.66	2.22
	WG	46.74	46.80	80.77	16.26	20.44	17.95	16.48	13.21	12.29	13.47
	RT	3067	3010	624	2242	490	483	1029	1027	2116	2110
	FS	18	17	19	20	20	20	20	20	20	20

In general, for D heuristic, when we increased  $T_{lim}$ , we obtained better solutions. However, for  $D^{FL}$ , we have not observed significant differences in the solutions' quality when we increased parameter  $T_{lim}$  from 1800 to 3600 seconds. This fact encourage the possibility of using  $D^{FL}_{1800}$  to build an initial feasible solution and then apply an improvement heuristic.

The computational performance of the RFH heuristic is reported as benchmark for the algorithms proposed in this paper. We can see that, RFH failed to provide feasible solutions for 6 test instances when we adopted  $T_{lim} = 900$  seconds, while good feasible solutions were obtained for all test instances when  $T_{lim} = 3600$ , significantly overcoming the MIP models as already observed in Soler et al. (2019).

Although RFH provided slight smaller gaps than our D heuristic for instances from classes 1 and 2, we observe that D heuristic consumed significantly less running times for these classes. We also highlight that for both maximum running times adopted for RFH ( $T_{lim} = 900$  and  $T_{lim} = 3600$ ) our  $D^{FL}$  (and also D) provided significantly better feasible solutions and consumed slightly less running times than RFH for large-sized test instances from classes 3, 4, and 5 (and in general). Moreover,  $D^{FL}_{1800}$  significantly overcame  $RFH_{3600}$  providing an average gap of 5.98%,

while  $\text{RFH}_{3600}$  presented an average gap of 7.05%, and significantly less computational time (1027 seconds) in comparison to  $\text{RFH}_{3600}$  (2242 seconds).

**Table 4** – Primal and dual bounds obtained by the MIP models.

	C1		C2		C3		C4		C5	
	OF	DB	OF	DB	OF	DB	OF	DB	OF	DB
$M_{3600}$	50935	50927	63603	63545	194597	114293	384519	174662	696031	211573
$M_{3600}^{FL}$	50935	50928	63605	63551	357799	115126	442264	174973	600448	212750

**6.3.2 Analysis of improvement heuristics**

Next, we analyse the computational performance of the improvement procedures, as presented in Table 5. Here, we fixed the maximum running time to an hour for all considered methods. This adopted maximum running time was suggested in Soler et al. (2019) in line with real-world conditions observed in meat industry companies.

We note that the proposed improvement heuristics (FO and LB) start from a feasible solution requiring the previous application of a construction procedure. Since  $\text{RFH}_{900}$  is not able to provide feasible solutions for all test instances and  $\text{RFH}_{3600}$  consumes the available computational time (3600 seconds) for most test instances from classes 3, 4, and 5, RFH is not suitable for this purpose. Therefore, we have used the more promising D heuristics ( $D_{900}^{FL}$  and  $D_{1800}^{FL}$ ) to build the necessary initial feasible solutions.

We recall that we need to specify parameters  $k^{FO}$  (number of adjacent periods to be optimized in each iteration) and  $k^{LB}$  (maximum number of  $\delta$ -variables that can have their value changed from the incumbent solution) in order to define our FO and LB heuristics, respectively. According to preliminary computational experiments we have empirically decided to adopt  $k^{FO} = 3$ ,  $k^{LB} = 6$  when using  $D_{900}^{FL}$  and  $k^{LB} = 4$  when using  $D_{1800}^{FL}$  to build initial feasible solutions. In Table 5, these adopted parameters are noted as subscripts in the name of the improvement heuristic, which is preceded by the name of the construction heuristic, e.g.,  $D_{1800}^{FL}$ -FO<sub>3</sub> indicates that the solutions built by  $D_{1800}^{FL}$  were improved by FO heuristic adopting  $k^{FO} = 3$ . Moreover, in Table 5, the reported running times (RT) include the time spent in the construction phase.

Note that, for classes 1 and 2, both heuristics reduced the initial gaps by approx. 50%, providing good quality solutions. For these classes, we highlight that FO heuristic presented significantly smaller computational times than LB heuristic. For classes 3, 4, and 5, the FO and LB heuristics presented only limited improvements to the initial solutions, and for both heuristics the more effective strategy is to spend 1800 seconds to build the initial solutions.

Finally, we observe that for classes 3, 4, and 5, FO heuristic outperformed LB heuristic with respect to the obtained gaps. The better combination of constructive and improvement methods was  $D_{1800}^{FL}$ -FO<sub>3</sub> presenting an average gap of 5.76%. We believe that we have high quality feasible solutions on hand and a significant gap reduction may be obtained improving the quality of the incumbent dual bounds.

**Table 5** – Computational results for the improvement heuristics.

		FO heuristic		LB heuristic	
		$D_{900}^{FL}-FO_3$	$D_{1800}^{FL}-FO_3$	$D_{900}^{FL}-LB_6$	$D_{1800}^{FL}-LB_4$
C1	AG	0.17	0.17	0.17	0.20
	BG	0.00	0.00	0.00	0.00
	WG	1.05	1.05	1.49	1.49
	DPG	0.34	0.33	0.35	0.36
	RT	616	616	1807	1611
C2	AG	0.40	0.40	0.55	0.60
	BG	0.00	0.00	0.02	0.03
	WG	1.43	1.43	2.12	2.12
	DPG	0.42	0.43	0.50	0.53
	RT	3600	3600	3600	3455
C3	AG	11.40	9.46	12.61	9.69
	BG	1.93	2.60	1.93	2.61
	WG	43.52	26.80	39.95	27.16
	DPG	8.97	6.10	9.35	6.16
	RT	3600	3600	3600	3600
C4	AG	9.93	8.88	10.08	9.04
	BG	4.47	4.44	4.52	4.71
	WG	19.79	17.51	19.32	17.51
	DPG	4.24	3.49	4.04	3.68
	RT	3600	3600	3600	3600
C5	AG	11.22	9.83	12.52	9.88
	BG	5.53	5.44	5.54	5.44
	WG	17.24	15.13	19.55	15.16
	DPG	3.31	2.49	3.64	2.51
	RT	3600	3600	3600	3428
Mean	AG	6.62	5.76	7.19	5.88
	BG	2.39	2.50	2.40	2.56
	WG	16.61	12.39	16.48	12.69
	DPG	3.45	2.57	3.58	2.65
	RT	3003	3003	3241	3139

## 7 CONCLUSIONS AND FUTURE RESEARCH

In this paper we studied a lot sizing and scheduling problem apparent in some food industries. In the addressed problem, a large catalog of perishable products need to be produced in various production lines. Moreover, the production lines share the same production resources and due to the scarcity of these resources, only a subset of the lines can operate in each period. We have proposed a MIP model using the facility location strategy to model the perishability aspect, a decomposition based heuristic to build feasible solutions, and two MIP based improvement

heuristics. A computational study carried out using a data set from the literature demonstrated that:

- the facility location reformulation of the standard model is not an effective strategy from the computational perspective due to its size;
- the proposed constructive heuristic can significantly benefit using the facility location strategy to model the perishability aspect;
- the proposed constructive heuristic can obtain superior results in comparison to a recent heuristic from the literature and a commercial MIP solver for large-sized test instances;
- for small-sized test instances, the improvement heuristics can significantly improve the solutions built by the constructive heuristic. However, only limited improvements can be obtained for large-sized test instances within the adopted maximum running times.

Through the study presented in this paper, we have identified some directions for future research. Firstly, we observe the necessity to obtain strong dual bounds in order to effectively evaluate the quality of the solutions on hand. This aspect can be addressed by developing approaches based on Lagrangian relaxation or Dantzig-Wolfe/Benders decomposition. The application of such approaches is also encouraged by the specific structure of the problem with three well-known components: lot sizing, scheduling, and knapsack. Moreover, we believe that improvement heuristic approaches can be further investigated, e.g., variable neighbourhood search, tabu search, and simulated annealing.

## ACKNOWLEDGMENTS

The research of the first author was supported by the Universidade Federal de Mato Grosso do Sul and in part by the Conselho Nacional de Desenvolvimento Científico e Tecnológico (CNPq). Research was carried out using the computational resources of the Center for Mathematical Sciences Applied to Industry (CeMEAI), funded by FAPESP (Grant 2013/07375-0). We also thank three anonymous referees for their valuable comments that helped improve the exposition of the paper.

## References

- [1] ABSI N & VAN DEN HEUVEL W. (2019). Worst case analysis of relax and fix heuristics for lot-sizing problems, *European Journal of Operational Research*, **279**(2): 449-458.
- [2] AFZALIRAD M, & REZAEIAN J. (2016). Resource-constrained unrelated parallel machine scheduling problem with sequence dependent setup times, precedence constraints and machine eligibility restrictions, *Computers & Industrial Engineering*, **98**: 40–52.

- [3] AKARTUNALI K, FRAGKOS I, MILLER AJ & WU T. (2016). Local cuts and two-period convex hull closures for big-bucket lot-sizing problems, *INFORMS Journal on Computing*, **28** (4): 766–780.
- [4] ALMADA-LOBO B, CLARK A, GUIMARÃES L, FIGUEIRA G & AMORIM P. (2015). Industrial insights into lot sizing and scheduling modeling. *Pesquisa Operacional*, **35** (3): 439–464.
- [5] ALMADA-LOBO B, OLIVEIRA JF & CARRAVILLA MA. (2008). Production planning and scheduling in the glass container industry: A VNS approach. *International Journal of Production Economics*, **114** (1): 363–375.
- [6] ALMEDER C & ALMADA-LOBO B. (2011). Synchronisation of scarce resources for a parallel machine lotsizing problem. *International Journal of Production Research*, **49** (24): 7315–7335.
- [7] AMORIM P, BELO-FILHO MAF, TOLEDO FMBD, ALMEDER C & ALMADA-LOBO B. (2013). Lot sizing versus batching in the production and distribution planning of perishable goods. *International Journal of Production Economics*, **146** (1): 208–218.
- [8] AOUAM T, GERYL K, KUMAR K & BRAHIMI N. (2018). Production planning with order acceptance and demand uncertainty. *Computers & Operations Research*, **91**: 145–159.
- [9] BALDO T, SANTOS MO, ALMADA-LOBO B & MORABITO R. (2014). An optimization approach for the lot sizing and scheduling problem in the brewery industry. *Computers & Industrial Engineering*, **72**: 58–71.
- [10] BRAHIMI N, ABSI N, DAUZÈRE-PÉRÈS S & NORDLI A. (2017). Single-item dynamic lot-sizing problems: An updated survey. *European Journal of Operational Research*, **263** (3): 838–863.
- [11] CLAASSEN GDH, GERDESSEN JC, HENDRIX EMT & VAN DER VORST JGAJ. (2016). On production planning and scheduling in food processing industry: Modelling non-triangular setups and product decay. *Computers & Operations Research*, **76**: 147–154.
- [12] COPIL K, WÖRBELAUER M, MEYR H & TEMPELMEIER H. (2017). Simultaneous lot-sizing and scheduling problems: a classification and review of models. *OR Spectrum*, **39** (1): 1–64.
- [13] CORDEAU JF, LAGANÀ D, MUSMANNO R & VOCATURO F. (2015). A decomposition-based heuristic for the multiple-product inventory-routing problem. *Computers & Operations Research*, **55**: 153–166.
- [14] COSTA AM, DOS SANTOS LMR, ALEM DJ & SANTOS RHS. (2014). Sustainable vegetable crop supply problem with perishable stocks. *Annals of Operations Research*, **219** (1): 265–283.

- [15] DOOSTMOHAMMADI M & AKARTUNALIK, (2018). Valid inequalities for two-period relaxations of bigbucket lot-sizing problems: Zero setup case. *European Journal of Operational Research*, **267**:86–95.
- [16] DREXL A & HAASE K. (1995). Proportional lotsizing and scheduling. *International Journal of Production Economics*, **40** (1): 73–87.
- [17] FERREIRA D, MORABITO R & RANGEL S. (2009). Solution approaches for the soft drink integrated production lot sizing and scheduling problem. *European Journal of Operational Research*, **196** (2): 697–706.
- [18] FISCHETTI M & LODI A. (2003). Local branching. *Mathematical programming*, **98**: 23–47.
- [19] FISCHETTI M, POLO C & SCANTAMBURLO M. (2004). A local branching heuristic for mixed-integer programs with 2-level variables, with an application to a telecommunication network design problem. *Networks: An International Journal*, **44**(2): 61–72.
- [20] FLEISCHMANN B. (1990). The discrete lot-sizing and scheduling problem. *European Journal of Operational Research*, **44** (3): 337–348.
- [21] FLEISCHMANN B & MEYR H. (1997). The general lotsizing and scheduling problem. *OR Spectrum*, **19** (1): 11–21.
- [22] GUIMARÃES L, KLABJAN D & ALMADA-LOBO B. (2014). Modeling lotsizing and scheduling problems with sequence dependent setups. *European Journal of Operational Research*, **239** (3): 644–662.
- [23] GÜNGÖR M, ÜNAL AT & TAŞKIN ZC. (2018). A parallel machine lot-sizing and scheduling problem with a secondary resource and cumulative demand. *International Journal of Production Research*, **56** (9): 3344–3357.
- [24] HAASE K. (1996). Capacitated lot-sizing with sequence dependent setup costs. *OR Spectrum*, **18** (1): 51–59.
- [25] JAMES RJ & ALMADA-LOBO B. (2011). Single and parallel machine capacitated lotsizing and scheduling: New iterative MIP-based neighborhood search heuristics. *Computers & Operations Research*, **38** (12): 1816–1825.
- [26] KARMARKAR US & SCHRAGE L. (1985). The deterministic dynamic product cycling problem. *Operations Research*, **33** (2): 326–345.
- [27] KIM SI, HAN J, LEE Y & PARK E. (2010). Decomposition based heuristic algorithm for lot-sizing and scheduling problem treating time horizon as a continuum. *Computers & Operations Research*, **37** (2): 302–314.



- [28] KOPANOS GM, PUIGJANER L & GEORGIADIS M C. (2011). Resource-constrained production planning in semicontinuous food industries. *Computers & Chemical engineering*, **35** (12): 2929–2944.
- [29] OLIVEIRA WAD & SANTOS MO. (2017). A New Branching Rule to Solve the Capacitated Lot Sizing and Scheduling Problem with Sequence Dependent Setups. *TEMA (São Carlos)*, **18** (3): 515–529.
- [30] ÖZDAMAR L & BIRBİL Şİ. (1998). Hybrid heuristics for the capacitated lot sizing and loading problem with setup times and overtime decisions. *European Journal of Operational Research*, **110** (3): 525–547.
- [31] PIRES MJ, AMORIM P, MARTINS S & ALMADA-LOBO B. (2015). Production planning of perishable food products by mixed-integer programming. In: *Operational Research*. Cham: Springer, p. 331-352.
- [32] POCHET Y & WOLSEY LA. (2006). *Production planning by mixed integer programming*. Springer Science & Business Media.
- [33] SMITH-DANIELS VL & RITZMAN L P. (1988). A model for lot sizing and sequencing in process industries. *International Journal of Production Research*, **26** (4): 647–674.
- [34] SOLER WAO, 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, *Journal of the Operational Research Society*. doi: 10.1080/01605682.2019.1640588
- [35] SOLER WAO, POLDI KC & SANTOS MO. (2019). Capacitated lot sizing and scheduling with order acceptance and delivery time windows: mathematical model and a MIP-based heuristic. *Pesquisa Operacional*, **39** (3): 471–496.
- [36] TOLEDO CFM, DA SILVA ARANTES M, HOSSOMI MYB, FRANÇA PM & AKARTUNALI 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.
- [37] VILLA F, VALLADA E & FANJUL-PEYRO L. (2018). Heuristic algorithms for the unrelated parallel machine scheduling problem with one scarce additional resource. *Expert Systems with Applications*, **93**: 28–38.
- [38] WEI W, AMORIM P, GUIMARÃES L & ALMADA-LOBO B. (2019). Tackling perishability in multi-level process industries. *International Journal of Production Research*, **93**(17): 5604–5623.

### How to cite

SOLER WAO, SANTOS MO & AKARTUNALI K. 2021. Decomposition based heuristics for a lot sizing and scheduling problem on multiple heterogeneous production lines with perishable products. *Pesquisa Operacional*, **41** (spe): e240377. doi: 10.1590/0101-7438.2021.041s1.00240377.

**APPENDIX**

In the second iteration of D heuristic proposed in Section 4, we need to solve the sub-problems  $SP^l(\bar{\delta}), l = 1, \dots, L$  obtained by fixing the values of the binary variables  $\delta$  from the values obtained in the first iteration ( $\bar{\delta}$ ). Using the same parameters described in Table 1, for each  $l = 1, \dots, L$ , the sub-problem  $SP^l(\bar{\delta})$  is defined as follows.

$$SP^l(\bar{\delta}):$$

$$\text{Min } \sum_{t,j \in P_l} (h_j I_{jt} + b_j B_{jt}) + \sum_{i,j \in P_l,t} sc_{lij} z_{ijt} + \sum_t ac_l \bar{\delta}_{lt} \tag{34}$$

$$\text{s.t. } \sum_{t'=0}^{sl_j} q'_{jt} = d_{jt} + B_{j,t-1} - B_{jt}, \forall t, j \in P_l \tag{35}$$

$$ve'_{j,t+1} = ve'_{jt} - q'_{jt}, \forall j \in P_l, t, t' \leq sl_j - 1 \tag{36}$$

$$ve^0_{jt} = x_{jt}, \forall j \in P_l, t \tag{37}$$

$$q'_{jt} \leq ve'_{jt}, \forall j \in P_l, t, t' \leq sl_j \tag{38}$$

$$I_{jt} = \sum_{t'=0}^{sl_j} ve'_{jt} - \sum_{t'=0}^{sl_j} q'_{jt}, \forall j \in P_l, t \tag{39}$$

$$B_{jT} = 0, \forall j \in P_l \tag{40}$$

$$\sum_{j \in P_l} a_{lj} x_{jt} + \sum_{i,j \in P_l} st_{lij} z_{ijt} \leq C_{lt}, \forall t \tag{41}$$

$$m_{lj} w_{jt} \leq x_{jt} \leq \frac{C_{lt}}{a_{lj}} w_{jt}, \forall j \in P_l, t \tag{42}$$

$$\sum_{j \in P_l} w_{jt} \leq \gamma_t \bar{\delta}_{lt}, \forall t \tag{43}$$

$$w_{jt} = y_{jt} + \sum_{i \in P_l} z_{ijt}, \forall j \in P_l, t \tag{44}$$

$$\sum_{j \in P_l} y_{jt} \leq 1, \forall t \tag{45}$$

$$y_{jt} + \sum_{i \in P_l} z_{ijt} \geq \sum_{i \in P_l} z_{jit}, \forall j \in P_l, t \tag{46}$$

$$V_{jt} \geq V_{it} + 1 - \gamma_t (1 - z_{ijt}), \forall t, i, j \in P_l \tag{47}$$

As mentioned earlier, the sub-problem  $SP^l(\bar{\delta})$  also admits a facility location version  $SP^l_{FL}(\bar{\delta})$  that can be easily adapted from the presented sub-problem  $SP^l(\bar{\delta})$ .