MIP-Heuristics for the Integrated Lot-Sizing and Supplier Selection Problem with Perishability

Caio Paziani Tomazella
Universidade de São Paulo - Instituto de Ciências Matemáticas e de Computação
Av. Trabalhador São-carlense, 400, 13560-970, São Carlos-SP, Brasil
caio.tomazella@usp.br

Maristela Oliveira Santos
Universidade de São Paulo - Instituto de Ciências Matemáticas e de Computação
Av. Trabalhador São-carlense, 400, 13560-970, São Carlos-SP, Brasil
mari@icmc.usp.br

RESUMO

Este artigo aborda o problema de dimensionamento de lotes multi-estágio que integra decisões de produção e compra de matérias-primas. Este problema pode ser encontrado em diferentes indústrias onde os itens produzidos ou matérias-primas são perecíveis, portanto devem ser utilizados na produção ou entregues aos clientes antes de suas validades expirarem. Apesar da relevância deste problema, há uma lacuna na literatura atual referente a métodos heurísticos para este problema. Assim, este artigo propõe MIP-heurísticas do tipo *relax-and-fix* e *fix-and-optimize* que visam encontrar boas soluções para o problema em intervalos de tempo curtos. São apresentadas decomposições que exploram as relações entre produtos e fornecedores, e os resultados mostram que os melhores algoritmos têm um desempenho similar ou, em alguns casos melhor, que a do *solver*. Também é mostrado o impacto da perecibilidade nas soluções em termos de factibilidade e são dadas sugestões de como o método pode ser aprimorado em pesquisas futuras.

PALAVRAS CHAVE. Dimensionamento de lotes. Seleção de fornecedores. Perecibilidade. MIP-Heurística.

ABSTRACT

This article addresses a multi-stage lot-sizing problem that integrates production and raw material procurement decisions. Due to the fact that this problem is applicable to different industries, in several cases the items involved in the supply chain are perishable, and must be either used in production or delivered to the customers before their self-lives are expired. Despite the relevance of this problem, there is a gap in the current literature since no heuristics were yet proposed. Therefore, this article presents relax-and-fix and fix-and-optimize MIP-heuristics with the objective to find good solutions for the problem in a short amount of time. We propose decomposition schemes that explore the relationships between products and suppliers, and show that our algorithms can match and, in some cases, surpass the performance of a commercial solver. We also show how perishability impacts the performance of the heuristics in terms of feasibility and suggest points in which the method can be improved for future research.

KEYWORDS. Lot-sizing. Supplier Selection. Perishability. MIP-Heuristic.

1. Introduction

The Integrated Procurement and Lot-Sizing Problem (IPLSP) has been widely addressed in the literature, due to its relevance in industrial applications [Crama et al., 2004; Mohammadi, 2020; Acevedo-Ojeda and Chen, 2020]. Some of the studies in this field show that the integrated approach provides more cost-efficient solutions than solving both problems separately, since it takes advantage of the inter-dependency of the decisions and reduce costs mostly on the operations related to raw material procurement, which accounts for the majority of the costs incurred in this problem [Cunha et al., 2018; Tomazella et al., 2020].

Supply chain problems that deal with perishable problems are found in several industries, such as food, blood banks and even electronics [Coelho and Laporte, 2014]. Chen et al. [2019] defined four different forms that perishability can be incorporated in a problem: imposing shelf-life constraints; establishing a make-to-order production strategy; using age-dependent holding costs; and measuring the inventory freshness degree. For more in-depth definitions, the reader is referred to these papers.

Perishability is mostly found in the lot-sizing literature in the form of shelf-life constraints, in which items must not be kept in inventory for more than a set number of time periods. Coelho and Laporte [2014] discuss that these constraints are fit for cases in which products are no longer fit for consumption (eg. dairy products and drugs with expiration dates) and products that become obsolete (eg. calendars and electronics).

On the IPLSP literature, Amorim et al. [2016] modeled a food supply chain with shelf-life constraints for all items. In their case, product demand depended on the age of the inventory at hand. Wei et al. [2019] imposed shelf-life constraints for both products and raw materials in a lot-sizing and scheduling problem with a multi-level production structure. Tomazella et al. [2020] considered age-dependent holding costs, in which items got more expensive to be kept in inventory as they aged, enforcing a First In Frist Out (FIFO) consumption policy. Lastly, Acevedo-Ojeda and Chen [2020] modeled raw material perishability in the form of shelf-life constraints, item deterioration and disposal in an advanced composite industry.

In this article, we address the IPLSP with multi-level production and supplier selection, in which both products and raw material have shelf-life constraints. We present a Mixed-Integer Programming (MIP) formulation that allow us to keep track of inventory age, and therefore can be applied to different types of item perishability and inventory aging effects. Moreover, we propose heuristics based in the decomposition of the MIP formulation (MIP-heuristics), that is, relax-and-fix and fix-and-optimize heuristics to solve large-size instances.

The motivation behind this article is that, even though several articles have proposed models for variations of the IPLSP, to the best of our knowledge, no studies on the efficiency of different formulations or on heuristic methods for the IPLSP were published. This research follows the article from Tomazella et al. [2020], who suggested the proposal of a heuristic in order to solve large-size instances of the IPLSP. Here we approach a simplified variation of their problem, in which quantity discounts, procurement budget and some production aspects such as provisioning lead-times and set-up carry-overs are not considered. However, the same formulation used to model perishability is used.

This article is structured as follows: in Section 2 we define the IPLSP and present its MIP formulation; in Section 3 we propose MIP-Heuristics for the IPLSP based on the relax-and-fix and fix-and-optimize procedures; Section 4 consists of the computational experimentation details and a results analysis; Section 5 summarizes the findings of this article and suggests ideas for future research.

2. Problem Description and Mixed-Integer Programming Formulation

In a multi-level production structure, products that have only external demand are called *end products*, while those that also have an internal demand incurred from their use for in-house production are called *intermediate products*. The production processes also consume raw materials, which can be purchased from multiple third party suppliers. Fore a more detailed definition of the big-bucket multi-level lot-sizing problem, the reader is referred to Akartunali and Miller [2009] and Wu et al. [2011], while for the supplier selection problem, to Basnet and Leung [2005] and Cárdenas-Barrón et al. [2021].

In the problem, external demand exists for both end and intermediate products, which must be fulfilled entirely, without backlogging or lost sales. Regarding the lot-sizing problem, we assume sequence-independent setups, zero initial inventory and no provisioning lead-times, therefore an intermediate product can be used in the same period that it is produced. Machines have a limited capacity, which can be extended by overtime hours. As for the procurement problem, raw materials can be purchase from multiple suppliers, and are available to be used in the same periods they are delivered.

Both products and raw materials have limited shelf-lives, which are given in periods. Whenever a product (raw material) is produced (purchased), it is either consumed or stored in inventory with age 1. For each period it is kept in inventory, the age increases until the item reaches its shelf-life, a point in which it must be consumed or it must be discarded. In case of the latter, the item is taken out of inventory and is no longer available for use, with no extra costs incurred. It should be noticed that in this problem that are no conditions that lead to having surplus inventory (i.e., quantity discounts, minimum lot sizes), so an optimal solution will not have items being discarded. We also consider that the inventory age does not affect the production process, and the shelf-life of a product is also not affected by the age of the inputs used in production.

In order to describe the IPLSP in detail, we present a Mixed-Integer Programming (MIP) model for the problem. Tables 1, 2 and 3 contain the notation used through this entire article. This formulation is based on the multi-level lot-sizing formulation from Akartunali and Miller [2009] and the supplier selection formulation from Basnet and Leung [2005]. Perishability is modeled using inventory variables that carry an index representing the inventory age, as introduced by Coelho and Laporte [2014].

Table 1: Sets and indices of the model

J	Set of products, indexed by i, j
F	Set of raw materials, indexed by f
T	Set of periods, indexed by t
M	Set of machines, indexed by m
S	Set of suppliers, indexed by s
g	Index referring to inventory age
$\mathbb{S}(j(f))$	Set of products that use product (raw material) j (f) in their production
$\mathbb{K}(m)$	Set of products assigned for production in machine m
$\mathbb{F}(f)$	Set of suppliers that sell raw material f

Objective Function 1 is the sum of the costs associated with the problem, which is minimized. These costs are, in the order that they appear: production, raw material purchasing costs; product and raw material holding costs; machine setup and overtime costs; supplier ordering costs.

Table 2: Parameters of the IPLSP used in the model

	Products
p_{jt}	Production cost of product j in period t
pt_{jt}	Production time of product j in period t
s_{jt}	Setup cost of product j in period t
st_{jt}	Setup time of product j in period t
h_{jt}	Holding cost of product j in period t
c_{mt}	Capacity of machine m in period t
otc_{mt}	Overtime cost of machine m in period t
a_{ij}	Units of product i used in the production of an unit of product j
d_{jt}	Demand of product j in period t
	Raw Material
p_{fst}	Cost of purchasing raw material f from supplier s in period t
δ_{fs}	Parameter with value 1 if raw material f is sold by supplier s, 0 otherwise
h_{ft}	Holding cost of raw material f in period t
a_{fj}	Units of raw material f used in the production of an unit of product j
o_{st}	Cost of ordering from supplier s in period t

Table 3: Variables of the MIP formulation for the IPLSP

X_{jt}	Units of product j produced in period t
I^g_{jt}	Inventory of product j with age g at the end of period t
W^g_{jt}	Units of product j with age g used to fulfill external and internal demand in period t
Q_{fst}	Units of raw material f purchased from supplier s in period t
I^g_{ft}	Inventory of raw material f with age g at the end of period t
W^g_{ft}	Units of raw material f with age g used to fulfill internal demand in period t
O_{mt}	Overtime of machine m in period t
Y_{jt}	1 if setup for product j occurs in period t, 0 otherwise
S_{st}	1 if a purchase is made from supplier s in period t, 0 otherwise

$$
\begin{aligned}
\min \sum_j \sum_t p_{jt} \cdot X_{jt} + \sum_f \sum_s \sum_t p_{fst} \cdot Q_{fst} + \\
\sum_j \sum_t \sum_{g=1}^{v_j} h_{jt} \cdot I^g_{jt} + \sum_f \sum_t \sum_{g=1}^{v_f} h_{ft} \cdot I^g_{ft} + \\
\sum_j \sum_t s_{jt} \cdot Y_{jt} + \sum_m \sum_t otc_{mt} \cdot O_{mt} + \sum_s \sum_t o_{st} \cdot S_{st}
\end{aligned}
\tag{1}
$$

Constraints (2)-(5) model the inventory flow of the products while keeping track of inventory age. Constraints (2) maintain the inventory balance throughout the periods. Constraints (3) set that the amount produced in a period that is not immediately used (for production or demand fulfillment) goes to inventory with age 1, while Constraints (4) guarantee that existing inventory that is not used in to be carried to the following period as long as it remains within its shelf-life. Constraints (5) than guarantee that both internal and external demands are fulfilled. A product perishes when it reaches an inventory age of $g = v_j + 1$, therefore it is neither considered in the inventory flow, in

Constraints (2), nor used for demand fulfillment, in Constraints (5).

Constraints (6) enforce the occurrence of machine setup whenever there is production. For these constraints, we use D_{jt} as a BIG-M, which is the accumulated echelon demand of product j from period t onwards, calculated using $D_{jt} = sum_{t'=t}^{|T|}\left(d_{jt'} + \sum_{i \in \mathbb{S}(j)} (a_{ij} \cdot D_{it})\right)$. Constraints (7) limit the time a machine spend on production and setup operations to its capacity, allowing overtime hours to be added at a penalty that it is high enough so it is the least preferable option to be used.

$$\sum_{g=1}^{v_j} I^g_{j(t-1)} + X_{jt} = \sum_{g=0}^{v_j} W^g_{jt} + \sum_{g=1}^{v_j} I^g_{jt} \qquad \forall j,t \qquad (2)$$

$$X_{jt} = W^0_{jt} + I^1_{jt} \qquad \forall j,t \qquad (3)$$

$$I^g_{j(t-1)} = W^g_{jt} + I^{g+1}_{jt} \qquad \forall j,t,1 \leq g \leq v_j \qquad (4)$$

$$\sum_{g=0}^{v_j} W^g_{jt} = \sum_{i \in \mathbb{S}(j)} (a_{ij} \cdot X_{jt}) + d_{jt} \qquad \forall j,t \qquad (5)$$

$$X_{jt} \leq D_{jt} \cdot Y_{jt} \qquad \forall j,t \qquad (6)$$

$$\sum_{j \in \mathbb{K}(m)} (pt_{jt} \cdot X_{jt} + st_{jt} \cdot Y_{jt}) \leq c_{mt} + O_{mt} \qquad \forall m,t \qquad (7)$$

Raw material inventory flow is modeled using Constraints (8)-(11), which are analogous to Constraints (2)-(5). Constraints (12) enforce that an order is placed for a supplier if any raw material is purchased from it, and uses $D_{ft} = \sum_{i \in \mathbb{S}(f)} (a_{if} \cdot D_{it})$. When raw material f is not sold by supplier s, δ_{fs} is equal to zero, which automatically forces $Q_{fst} = 0$.

$$\sum_{g=1}^{v_f} I^g_{f(t-1)} + \sum_{s} Q_{fst} = \sum_{g=0}^{v_f} W^g_{ft} + \sum_{g=1}^{v_f} I^g_{ft} \qquad \forall f,t \qquad (8)$$

$$\sum_{s} Q_{fst} = W^0_{ft} + I^1_{ft} \qquad \forall f,t \qquad (9)$$

$$I^g_{f(t-1)} = W^g_{ft} + I^{g+1}_{ft} \qquad \forall f,t,1 \leq g \leq v_f \qquad (10)$$

$$\sum_{g=0}^{v_f} W^g_{ft} = \sum_{i \in \mathbb{S}(f)} (a_{if} \cdot X_{ft}) \qquad \forall f,t \qquad (11)$$

$$Q_{fst} \leq \delta_{fs} \cdot D_{ft} \cdot S_{st} \qquad \forall f,s,t \qquad (12)$$

Finally, Constraints (13)-(14) define the domain of the variables.

$$X_{jt}, I^g_{jt}, W^g_{jt}, Q_{fst}, I^g_{ft}, W^g_{ft}, O_{mt} \geq 0 \quad \forall j,f,t,m,s,g \qquad (13)$$

$$Y_{jt}, S_{st} \in \{0,1\} \qquad \forall j,t,s \qquad (14)$$

3. MIP-Heuristic Algorithms

3.1. Relax-and-Fix

The relax-and-fix procedure is widely used in the lot-sizing literature to generate solutions for problems, which also serve as a starting point for other algorithms [Stadtler, 2003; Akartunali and Miller, 2009; Baldo et al., 2014]. This heuristic divides the binary variables of the problem into three groups: *fixed*, *optimized* and *relaxed*. Initially, all variables are linearly *relaxed*. Than the domain of a subset of variables is restored to binary so they can be *optimized*; after the model is solved, all or some (in case there is overlapping) variables in *optimized* are *fixed* to the values found. These steps are repeated until all binary variables have been optimized.

$RF_T(w, y)$ heuristic

We first propose a time-based decomposition that constructs a solution starting from the first period and moving forward, and is defined by two parameters, the window size (w) and the overlap (y). Using our nomenclature, w is the amount of periods being optimized, while y is the number of periods that are re-optimized in the following iteration, which means that the variables of the remaining $w - y$ periods are fixed.

In order to illustrate the procedure, we show an example with $(w, y) = (2, 1)$. At first the variables in periods $t = 1, 2$ are optimized; then the model is solved and the variables in period $t = 1$ are fixed; in the second iteration, periods $t = 2, 3$ are optimized and the variables in $t = 2$ are fixed. It should be noticed that in this example overlapping occurs in all variables from period $t = 2$. The pseudocode of the relax-and-fix heuristic, which we call $RF_T(w, y)$, is shown in Algorithm 1.

```
Define model MIP as in Section 2
Relax all Y_jt and S_st variables
Define window (w) and overlap (y)
Define step: s = w − y
Define t1 = 1 and t2 = w
while t2 ≤ |T| do
    Set the domain of variables Y_jt and S_st with t1 ≤ t ≤ t2 to binary
    Solve MIP
    Fix variables Y_jt and S_st with t1 ≤ t ≤ t1 + s to the values found
    t1 = t1 + s
    t2 = t2 + s
    if t2 > |T| then
        t2 = |T|
        t1 = |T| − w
    end
end
```

Algorithm 1: Psudocode of heuristic $RF_T(w, y)$.

RF_{PS} and RF_{SP} heuristics

We also propose two stage-based decomposition schemes to test if this approach yields better results than the traditional time-based heuristics. These heuristics decompose the problem into $|J|$ subsets, each one containing all Y_{jt} variables for a product along the entire planning horizon, and a single subset containing all S_{st} variables. Heuristic RF_{PS} solves first the products subproblems in sequence, followed by the supplier selection subproblem, while RF_{SP} does the opposite, fixing the S_{st} variables first. It should be noticed that after each solve, all optimized variables are fixed and there is no overlap. Contrary to the time-based decomposition, this approach is less integrated in the sense that the Y_{jt} and S_{st} variables are optimized separately.

$RF_T(w, y)$ has a natural order to which each subset is solved (starting from the first period onwards). For RF_{PS} and RF_{SP}, we use the product order proposed by Helber and Sahling [2010]: (1) solve the linear relaxation of the MIP; (2) calculate the costs associated with each product (production, inventory, setup and overtime); (3) arrange the products in non-increasing order of the associated costs. For a more detailed explanation of this ordering, the reader is referred to the original article.

3.2. Fix-and-Optimize

The fix-and-optimize heuristic is an improvement MIP-based procedure, which main principal is solving smaller subsets of the MIP in sequence. This procedure is found to be used in several ways, such as a stand-alone method [Helber and Sahling, 2010], applied in conjunction with the relax-and-fix heuristic [Baldo et al., 2014; Toledo et al., 2015] and in conjunction with other heuristic and metaheuristic concepts [James and Almada-Lobo, 2011; Furlan and Santos, 2017].

The variables are assigned into different subsets ($\mathcal{N}_k$) according to a decomposition scheme. Initially all variables are *fixed*, which gives an initial feasible solution. Than each subset is *optimized* in sequence, which gives a solution that is at least as good as the initial one. After all subsets are optimized, the algorithm can either (i) stop in case no improvement on the initial solution is found or (ii) repeat the optimization process otherwise. Algorithm 2 describes the steps of a generalized fix-and-optimize heuristic procedure.

Define an initial solution for model MIP
Define and arrange the K variable subsets $\mathcal{N}_1, \mathcal{N}_2, \ldots, \mathcal{N}_K$
repeat
  for $1 \leq k \leq K$ **do**
    Release variables in $\mathcal{N}_k$ for optimization
    Solve MIP
    Fix variables in $\mathcal{N}_k$ to the values found
  end
until *no improvement in the solution is found*;

Algorithm 2: Fix-and-optimize algorithm

We propose three different decomposition schemes for the IPLSP, highlighting that these aim at optimizing both lot-sizing and supplier selection variables in an integrated manner.

- Period decomposition (FO_T): corresponds to the all binary variables within a range of periods. This decomposition also uses the w and y parameters.
- Product decomposition (F_P): corresponds to the setup variables of a product and the order variables related to the suppliers which sell the raw materials used in its production.
- Process decomposition (FO_{PR}): an extension of the product decomposition, which also considers the setup variables of the intermediate products used in its production.

Since in FO_P and FO_{PR} each subset is tied to a product j, the ordering given to the subsets in both decomposition schemes is the same used in the RF_{PS} and RF_{SP} procedures.

4. Computational Experiments and Results

The models and heuristics presented in this article were implemented in Python 3.6.9, and all models were solved using Gurobi 9.0 with its default parameters. The experiments were done in a computer with an Intel Core i7-2600 processor at 3.40GHz and 16GB RAM.

For the experimentation we used a database of 64 instances, consisting of multi-stage lot-sizing instances of the *New Invented Data* from Tempelmeier and Buschkühl [2009] with newly generated data for the supplier selection parameters. Table 4 summarizes the main parameters of the instances.

The lot-sizing instances are the following (using the nomenclature given by the author): Classes 4 and 6; Assembly and General product structures; demand, capacity and setup profiles 1. Production costs are set to $p_{jt} = 1$ and the original setup costs are multiplied by 100. The supplier selection data was generated as follows: two raw material profiles for each Class, consisting of different numbers of raw materials; the holding costs of each material were set to $h_{ft} = 1$, and the holding costs of the products were updated accordingly; a number of suppliers was set for each Class, and the materials were assigned so that each material was sold by three different suppliers; the raw material purchasing costs (p_{fst}) were generated using uniform distribution in the $[20, 50]$ range, and the supplier order costs (o_{st}) in the $[1000, 2000]$ and $[10000, 12000]$ ranges. These ranges were based on the supplier selection instances used by Cárdenas-Barrón et al. [2021].

Four perishability scenarios are considered, in which v_j (v_f) is common for all products (raw materials): $(v_j, v_f) = (2, 2), (2, 5), (5, 2), (5, 5)$.

Table 4: Characteristics of the instances.

Class	Periods	Products	Machines	Suppliers	Raw Materials
4a	16	20	6	6	12
4b	16	20	6	6	24
6a	16	40	6	12	24
6b	16	40	6	12	48

4.1. MIP Formulation Results

We first solve the MIP formulation presented in Section 2 in order to obtain upper and lower bounds that serve as a benchmark for the FO heuristic. A time limit of 3600s was given for each instance. Table 5 summarizes the results, showing the number of optimal solutions found, the number of solutions with machine overtime, the optimality gaps and elapsed time. The optimality gap is automatically given by Gurobi, and is calculated using $GAP = (UB - LB)/LB$ (UB being the value of the incumbent solution and LB the value of the lower bound at the end of the execution). Overall, no optimal solutions were found for any of the instances in Classes 4 and 6, with the average gaps being 2.50% and 2.79%, respectively. None of the incumbent solutions have machine overtime, which shows that all instances would be feasible even if overtime was not allowed.

Table 5: Results obtained by solving the MIP formulation.

	Gap			
Class	Average	Minimum	Maximum	Time (s)
4a	1.92%	0.56%	4.43%	3600
4b	3.08%	1.03%	6.14%	3600
6a	1.79%	1.12%	2.45%	3600
6b	3.80%	2.13%	6.28%	3600

4.2. Heuristics Results

In the first of the experimentation, we analyse the results of the relax-and-fix heuristic as a solution method. Let $RF_T(w, y)$ be the relax-and-fix using the time-based decomposition with set

w and y parameters. In each procedure, the execution time for each iteration is limited to $3600/k$ seconds, with $k = \lceil(|T| - w)/(w - y)\rceil + 1$, as proposed by James and Almada-Lobo [2011]. For RF_{PS} and RF_{SP}, we have $k = |J| + 1$, which is the number of iterations.

Table 6 summarizes the results obtained with the relax-and-fix heuristics. We observe that larger optimization windows w and overlaps y provide better solutions, although at the expense of a significant increase in computational time. This trade-off is especially clear in the instances from Set 6, in which the average computational time of $RF_T(3, 2)$ is approximately 20 minutes. We suggest, based on the results obtained with the fix-and-optimized heuristics, that RF_T is better used with shorter windows in order to generate an initial solution rather than as a isolated method.

Both item-based decomposition heuristics, RF_{PS} and RF_{SP}, showed a poor performance both in terms of solution quality, comparable to the worst cases of $RF_T(w, y)$ and, in the case of RF_{SP}, in terms of computational time. We observe that these decomposition schemes do not take advantage of the relationships between variables of different products and suppliers, which leads to non-optimal solutions. As discussed in the introduction, solving the problem in an integrated manner allows us to shape an internal demand profile for raw materials that takes advantage of lower prices and fixed supplier costs. In this case, the shelf-life constraints also become a factor, since they do not allow inventory to be kept for longer periods of time, creating the need of more purchases and production setups, which incurring additional ordering costs and possibly, machine overtime.

Table 6: Results of the relax-and-fix heuristics. Note: Imp: number of instances improved in comparison to the MIP results.

	Instance Set 4			Instance Set 6		
Heuristic	Imp	Gap	Time (s)	Imp	Gap	Time (s)
$RF_T(1, 0)$	0	4.99%	6.32	0	4.73%	21.17
$RF_T(2, 0)$	0	4.58%	10.75	0	4.24%	159.39
$RF_T(2, 1)$	0	4.04%	44.43	0	3.92%	216.29
$RF_T(3, 0)$	0	3.61%	63.39	0	3.54%	731.43
$RF_T(3, 1)$	1	3.26%	85.39	3	3.28%	911.76
$RF_T(3, 2)$	1	3.17%	163.56	3	3.25%	1188.84
RF_{PS}	0	4.42%	8.20	0	4.78%	59.56
RF_{SP}	0	4.24%	571.69	0	4.35%	2406.58

The second experimentation analyses the results of the fix-and-optimize heuristics. As a preliminary step, we have tested the heuristic (Algorithm 2) with the period decomposition using the six combinations of (w, y) parameters from Table 6. We observed a similar trade-off between gap and computational time, and opted to use $(w, y) = (2, 1)$, which are also the same values suggested by James and Almada-Lobo [2011].

We now test the fix-and-optimize heuristic using the decomposition schemes defined in 3.2. The algorithm starts with $RF_T(2, 1)$ in order to find an initial solution, then applies the procedure described in Algorithm 2. The algorithm stops until no improvement is found or after the time limit of 3600 second is reached. Seven combinations of the decomposition schemes are used:

- FO_1: $F_T(2, 1)$
- FO_2: F_P
- FO_3: F_{PR}
- FO_4: $F_T(2, 1)$, F_P

- FO_5: $F_T(2,1)$, F_{PR}
- FO_6: F_P, F_{PR}
- FO_7: $F_T(2,1)$, F_P, F_{PR}

Table 7 shows the results obtained with these heuristics. We first observe that the benefits of using fix-and-optimize procedures is evident when comparing the gaps and computational times, since FO_1 with $(w, y) = (2, 1)$ results in lower gaps and computational times than using the RF_T heuristic with $w = 3$. The number of fix-and-optimize iterations is also listed, showing that the subsets are often optimized two or more times before a local minimum is found, reassuring the need of multiple procedures.

The fix-and-optimize heuristics with better performance are those that employ the time-based decomposition (FO_1, FO_4, FO_5 and FO_7), and that the addition of schemes F_P and F_{PR} improves the results, although at a significant execution time increase.

When comparing to the MIP results, the best performing heuristics managed to find better solutions in 30% to 40% of the instances. While the average gaps obtained with the heuristics are higher than those obtained while solving the MIP for one hour, the computational times in Table 7 show that they are viable methods for obtaining good solutions for the IPLSP: the $FO(7)$ heuristic, which took the longest times, had an average of less than 2.5 minutes with the instances of Class 4 and less than 12 minutes with those of Class 6.

Overall, no heuristics exceeded the time limit of 3600 seconds, and the worst case was under 2200 seconds. This indicates that the method can be improved by adding different different decomposition schemes (deterministic or stochastic), which can eventually lead the heuristic to consistently find better solutions than the solver within the same time frame.

During the experimentation we have noticed that these subsets take longer to be solved to optimality, often reaching the set time limit. Therefore, we also suggest that this method can be enhanced by, instead of optimizing all product/process variable subsets, we select those that are the most promising in terms of reducing costs and machine overtime.

Table 7: Results of the fix-and-optimize heuristics. Note: Imp: number of instances improved in comparison to the MIP results; Iter: average number of iterations done by the fix-and-optimize heuristics.

	Instance Set 4				Instance Set 6			
Method	Imp	Gap	Time (s)	Iter.	Imp	Gap	Time (s)	Iter.
MIP	-	2.50%	3600.00	-	-	2.79%	3600.00	-
$RF_T(2,1)$	0	4.04%	44.43	-	0	3.92%	216.29	-
FO_1	8	2.76%	78.14	2.16	11	2.95%	460.99	2.97
FO_2	1	3.34%	72.89	1.84	1	3.27%	426.71	1.91
FO_3	4	2.99%	109.82	2.03	2	3.23%	475.04	1.75
FO_4	8	2.69%	98.75	2.16	12	2.92%	520.57	2.09
FO_5	11	2.63%	121.45	1.94	13	2.92%	553.16	2.00
FO_6	4	3.01%	122.87	1.84	1	3.25%	603.29	1.41
FO_7	12	2.61%	142.70	1.91	13	2.92%	679.68	1.94

The last discussion refers to the fact that the proposed methods can result in solutions with machine overtime, which does not happen when solving the MIP formulation. Since overtime hours have an associated cost that is high enough for them to be used as a last option to ensure solution feasibility, the fact that heuristics end with positive overtime indicate a limitation of the methods in dealing with potential feasibility issues. We observe that most cases occur when products have a shorter shelf-life, and that none of the heuristics have a clear tendency in eliminating overtime when compared to the others. For this issue we suggest the proposal of an improvement phase that identifies neighborhoods of variables that cause overtime to occur and optimize them. This procedure will be useful in ensuring feasibility when applying the heuristic to problems in which overtime is not allowed.

5. Conclusion

In this article we have addressed the Integrated Procurement and Lot-Sizing Problem (IPLSP) with perishability in the form of shelf-life constraints. This study was motivated by a gap in the IPLSP literature in the sense that no heuristic methods were proposed to solve this problem. Our objective was to propose decomposition-based MIP-Heuristics that can be used in a number of variations of the problem and aid in the application of the IPLSP in industry cases.

Our results showed that the heuristics that apply time-based decomposition schemes, which optimizes both production and supplier selection decisions across a period interval are more effective, and that the use of other item-based decomposition can improve the solutions, although at the expense of a significant increase in computational time.

For future research, our results lead to suggestions that can improve the proposed methods: the first it to reduce the iterations of the FO heuristics that use product and process decomposition schemes, by selecting only the variable subsets that are more promising in the sense of improving the solution; the second is to apply an improvement scheme in order to minimize machine overtime, which is an issue that occurs more often in instances with short product shelf-lives.

Acknowledgements

This research was funded by FAPESP (grant 2019/10824-7) and carried out using the computational resources of the Center for Mathematical Sciences Applied to Industry (CeMEAI) funded by FAPESP (grant 2013/07375-0).

References

Acevedo-Ojeda, A. and Chen, M. (2020). Multi-level lot-sizing with raw-material perishability, deterioration, and batch ordering: an application of production planning in advanced composite manufacturing. *Computers & Industrial Engineering*, 145:106484.

Akartunali, K. and Miller, A. J. (2009). A heuristic approach for big bucket multi-level production planning problems. *European Journal of Operational Research*, 193(2):396–411.

Amorim, P., Curcio, E., Almada-Lobo, B., Barbosa-Póvoa, A. P., and Grossmann, I. E. (2016). Supplier selection in the processed food industry under uncertainty. *European Journal of Operational Research*, 252(3):801–814.

Baldo, T. A., Santos, M. O., Almada-Lobo, B., and Morabito, R. (2014). An optimization approach for the lot sizing and scheduling problem in the brewery industry. *Computers & Industrial Engineering*, 72:58–71.

Basnet, C. and Leung, J. M. (2005). Inventory lot-sizing with supplier selection. *Computers & Operations Research*, 32(1):1–14.

Cárdenas-Barrón, L. E., Melo, R. A., and Santos, M. C. (2021). Extended formulation and valid inequalities for the multi-item inventory lot-sizing problem with supplier selection. *Computers & Operations Research*, 130:105234.

Chen, S., Berretta, R., Clark, A., and Moscato, P. (2019). Lot Sizing and Scheduling for Perishable Food Products: A Review. In *Reference Module in Food Science*, p. B9780081005965214443. Elsevier.

Coelho, L. C. and Laporte, G. (2014). Optimal joint replenishment, delivery and inventory management policies for perishable products. *Computers & Operations Research*, 47:42–52.

Crama, Y., Pascual J, R., and Torres, A. (2004). Optimal procurement decisions in the presence of total quantity discounts and alternative product recipes. *European Journal of Operational Research*, 159(2):364–378.

Cunha, A. L., Santos, M. O., Morabito, R., and Barbosa-Póvoa, A. (2018). An integrated approach for production lot sizing and raw material purchasing. *European Journal of Operational Research*, 269(3):923–938.

Furlan, M. M. and Santos, M. O. (2017). BFO: a hybrid bees algorithm for the multi-level capacitated lot-sizing problem. *Journal of Intelligent Manufacturing*, 28(4):929–944.

Helber, S. and Sahling, F. (2010). A fix-and-optimize approach for the multi-level capacitated lot sizing problem. *International Journal of Production Economics*, 123(2):247–256.

James, R. J. and 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.

Mohammadi, M. (2020). Designing an integrated reliable model for stochastic lot-sizing and scheduling problem in hazardous materials supply chain under disruption and demand uncertainty. *Journal of Cleaner Production*, 274:122621.

Stadtler, H. (2003). Multilevel Lot Sizing with Setup Times and Multiple Constrained Resources: Internally Rolling Schedules with Lot-Sizing Windows. *Operations Research*, 51(3):487–502.

Tempelmeier, H. and Buschkühl, L. (2009). A heuristic for the dynamic multi-level capacitated lotsizing problem with linked lotsizes for general product structures. *OR Spectrum*, 31(2):385–404.

Toledo, C. F. M., da Silva Arantes, M., Hossomi, M. Y. B., França, P. M., and 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.

Tomazella, C., Santos, M., and Alem, D. (2020). A Mixed Integer Model for the Integrated Lot-Sizing and Raw Material Procurement Problem with Quantity Discounts and Age-Based Holding Costs. *LII SBPO - Simpósio Brasileiro de Pesquisa Operacional*.

Wei, W., Amorim, P., Guimarães, L., and Almada-Lobo, B. (2019). Tackling perishability in multi-level process industries. *International Journal of Production Research*, 57(17):5604–5623.

Wu, T., Shi, L., Geunes, J., and Akartunali, K. (2011). An optimization framework for solving capacitated multi-level lot-sizing problems with backlogging. *European Journal of Operational Research*, 214(2):428–441.