Lot-sizing for industrial symbiosis

Elodie Suzanne, Nabil Absi, Valeria Borodin, Wilco van den Heuvel

CMP – SITE GEORGES CHARPAK

CENTRE MICROELECTRONIQUE DE PROVENCE

Lot-sizing for industrial symbiosis

Elodie SUZANNE[1], Nabil ABSI[1], Valeria BORODIN[1], Wilco VAN DEN HEUVEL[2]

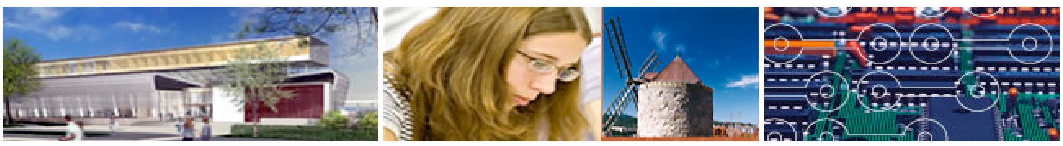
[1]Ecole des Mines de Saint-Etienne, CMP Georges Charpak, CNRS UMR 6158 LIMOS, F-13541 Gardanne, France
{elodie.suzanne, absi, valeria.borodin}@emse.fr

[2]Econometric Institute, Erasmus School of Economics, Erasmus University Rotterdam, P.O. Box 1738, 3000 DR Rotterdam, The Netherlands
{wvandenheuvel}@ese.eur.nl

September 2020

Working Paper EMSE CMP–SFL 2020/1

Waste accumulation in landfills, global warming and the need to preserve finite raw materials push governments and industries to shift towards a circular economy. Industrial symbiosis represents a sustainable way of sharing resources and converting unavoidable production residues into useful and added-value products. In this context, we study a production planning problem arisen between two production units (PU) within an industrial symbiosis. During the production process of a main product, a production residue is generated by the first PU, which is subsequently either used as raw materials by the second PU, or disposed of. The second PU can also purchase raw materials from an external supplier. The resulting combined production planning problem has been formulated as a two-level single-item lot-sizing problem. We prove that this problem is NP-Hard irrespective of the production residue, namely unstorable, or storable with a limited capacity. To efficiently solve this problem, a heuristic based on Lagrangian decomposition is proposed. Extensive numerical experiments highlight the competitiveness of the proposed solution method. The impact of the collaborative framework, in which the production plans of each PU are brought together, has been studied via a comparative analysis of different decentralized and centralized collaboration policies. Valuable insights derived from this analysis are subsequently used to discuss the managerial implications of setting up an industrial symbiosis between a supplier of by-products and its receiver.



Ecole Nationale Supérieure des Mines SAINT-ETIENNE

Lot-sizing for industrial symbiosis

Elodie Suzanne[a], Nabil Absi[a], Valeria Borodin[a], Wilco van den Heuvel[b]

[a]*Department of Manufacturing Sciences and Logistics, Mines Saint-Etienne, University Clermont Auvergne, CNRS, UMR 6158 LIMOS, CMP, Gardanne, France*
[b]*Econometric Institute, Erasmus School of Economics, Erasmus University Rotterdam, P.O. Box 1738, 3000 DR Rotterdam, The Netherlands*

Abstract

Waste accumulation in landfills, global warming and the need to preserve finite raw materials push governments and industries to shift towards a circular economy. Industrial symbiosis represents a sustainable way of sharing resources and converting unavoidable production residues into useful and added-value products. In this context, we study a production planning problem arisen between two production units (PU) within an industrial symbiosis. During the production process of a main product, a production residue is generated by the first PU, which is subsequently either used as raw materials by the second PU, or disposed of. The second PU can also purchase raw materials from an external supplier. The resulting combined production planning problem has been formulated as a two-level single-item lot-sizing problem. We prove that this problem is NP-Hard irrespective of the production residue, namely unstorable, or storable with a limited capacity. To efficiently solve this problem, a heuristic based on Lagrangian decomposition is proposed. Extensive numerical experiments highlight the competitiveness of the proposed solution method. The impact of the collaborative framework, in which the production plans of each PU are brought together, has been studied via a comparative analysis of different decentralized and centralized collaboration policies. Valuable insights derived from this analysis are subsequently used to discuss the managerial implications of setting up an industrial symbiosis between a supplier of by-products and its receiver.

Keywords: Production planning, Two-level lot-sizing, By-product, Industrial symbiosis, Lagrangian decomposition

1. Introduction

Waste accumulation in landfills, the depletion of finite raw materials and global warming push our generation to shift towards a circular economy and to adopt a number of sustainable practices such as recycling end-of-life products, using renewable energy, reducing greenhouse gas emissions. Since few years, the legislation around the world is evolving to promote the reuse of production residues, particularly those that are no longer allowed in landfills[1].

One of the sustainable ways to convert production residues into useful and added-value products is the so-called *industrial symbiosis*. This is a collaborative form between companies based on the *exchange of physical flows*, such as production residues or other secondary resources (e.g. water, and energy), and/or the *sharing of services* like knowledge, logistics,

[1]The directive 1999/31/EC of the Council of 26 April 1999 on the landfill of waste: https://eur-lex.europa.eu/legal-content/EN/TXT/PDF/?uri=CELEX:31999L0031&from=EN Access: 5 February 2020

expertise (Lombardi and Laybourn, 2012). According to the Waste Framework Directive[2], production residues are materials that are not deliberately produced during a production process, and can be divided into two broad classes of products: **(i)** *by-products*, i.e. lawful production residues unavoidably obtained as an integral part of a production process, ready for a certain use without further transformation, and **(ii)** *wastes*, i.e. production residues, which are not by-products.

By its nature, industrial symbiosis offers opportunities for the three dimensions of the sustainable development, namely economic, environmental and social, by: **(i)** avoiding disposal costs and the increasing resource efficiency, **(ii)** reducing the consumption of raw materials, **(iii)** supporting the regional economic development. As a consequence and due to the success of the existing eco-industrial parks, a number of industrial symbiosis emerges all around the world. In 2017, Evans et al. (2017) referenced 281 implemented by-product synergies and about 150 planned or under feasible study, spread over almost all continents.

The particular configuration of an industrial symbiosis system, where the by-products generated by a production unit are used as raw materials by another production unit, is called *by-product synergy*. The by-product exchange can take place either within a single entity, or between two or several different autonomous companies. The resulting network includes necessarily two major actors: **(i)** a *supplier*, which generates by-products and, **(ii)** a *receiver*, which uses them. The intervention of a third party can be required to ensure the connection between the supplier and the receiver of a by-product.

As long as two actors are involved at least in a by-product synergy system, multiple practical questions arise e.g.: During which time slot can the by-products be used? What transportation mode is more convenient to be used? How to distribute the related costs or benefits between the different actors? What collaboration policy should be adopted?

Being industry-dependent, the industrial symbiosis networks can also include specific intermediate facilities required by the by-product handling systems, e.g. for storage or treatment purposes (Suzanne et al., 2020b). These intermediate facilities can belong to production units, or be provided by third parties, complicating thereby the logistics related to the by-product handling. An extra fee is sometimes paid by one of the actors or both of them to compensate the generated extra costs incurred by the owner of these facilities to encourage its participation in the industrial symbiosis (Evans et al., 2017).

The collaboration policies can differ from one industrial symbiosis to another. There exist cases, where the involved actors develop a mutual trust between themselves and share all their information, enabling thus possible a centralized decision-making related to the by-product exchange. The full information sharing is usually encountered when both actors belong to the same parent company or when a third party manages the industrial symbiosis network. In general, the full information sharing between actors may be difficult to be considered for different reasons such as the requirements to keep sensitive information private or not to reveal the risks related to production disruptions or production recipes of products (Vladimirova et al., 2018; Fraccascia and Yazan, 2018). Partial information sharing is commonly addressed via decentralized collaboration policies. Let us distinguish two main types of decentralized collaboration policies with respect to their time frames:

- **Opportunistic (Short-term):** In the decentralized collaboration policy applied to regulate spontaneous exchange of by-products, we assume that both the supplier

[2]The communication from the Commission to the Council and the European Parliament on the Interpretative Communication on waste and by-products, number 52007DC0059: https://eur-lex.europa.eu/legal-content/EN/TXT/?uri=CELEX%3A52007DC0059

and the receiver make their own production plans independently from one another. The production plans are then brought together to put the generated by-products into value. This kind of decentralized collaboration policy is specific to short-term horizontal collaboration in classical supply chains.

- **Symbiotic (Long-term):** A long-term collaboration may be expressed in the framework of sequential decentralized collaboration policies, made either from the supplier to the receiver or from the receiver to the supplier. Commonly encountered in the vertical collaboration in classical supply chains, this kind of decentralized collaboration policies can also take place in an industrial symbiosis framework when a long-term collaboration is committed to promoting the by-product recovery. They are based on the dominance (i.e. on primacy in decision-making) of one actor compared to another one: one actor makes its decisions first, then the second one makes its production plan accordingly.

To cope with the operation management problems posed by the industrial symbiosis, the current paper contributes by:

- Introducing and formalizing a lot-sizing problem in the framework of an industrial symbiosis: The addressed problem extends the work of Suzanne et al. (2020b), by integrating the *receiver* production unit in the management of a by-product generated by the *supplier* production unit. As discussed in Section 2.1, the lot-sizing problem introduced in this paper enriches the joint production planning literature by its novelty and industrial relevance.

- Studying a two-level lot-sizing problem for different characteristics: The studied problem falls within the class of two-level lot-sizing problems, where the first level corresponds to the problem encountered by the supplier, and the second level corresponds to the problem encountered by the receiver. The main differences between the studied problem and the classical state-of-the-art two-level problems are the linking flows of the by-product between levels and the associated inventory capacities at the supplier level.

- Proposing a solution method based on both Lagrangian decomposition and Lagrangian relaxation, whose efficiency and effectiveness is shown via extensive numerical experiments. The performance of several variants of the proposed solution method has been also empirically evaluated.

- Discussing centralized and decentralized collaboration policies, which can be applied in full and none information sharing settings.

The remainder of this paper is organized as follows. Section 2 reviews: **(i)** the literature covering lot-sizing problems, which integrate the joint production, and **(ii)** the literature covering the class of two-level lot-sizing problems. The generic version of the problem under study is described and a complexity analysis is conducted in Section 3. A Lagrangian decomposition approach is proposed in Section 4. Different collaboration policies are introduced and solved in Section 5, in order to be compared in the current paper. The competitiveness of the Lagrangian decomposition algorithm is empirically shown by performing computational experiments on two versions of the problem with: storable by-products with a limited capacity and unstorable by-products in Section 6. Managerial implications of the different collaboration policies are discussed in Section 7. Concluding remarks and perspectives are provided in Section 8.

2. Literature review

As the industrial symbiosis implies the operations management related to the by-product exchange, this section provides a literature review focused on joint production planning problems. In the current paper, the by-products generated by a production unit are used by another production unit. In order to better apprehend the interactions between these two production units, we also present a review on two-level production planning problems, by placing a special focus on the problems with inventory capacities at the supplier level and external intermediate flows between levels. For more details on the production planning problems in general, the reader is referred to the literature reviews of Buschkühl et al. (2010); Díaz-Madroñero et al. (2014); Brahimi et al. (2017); Quadt and Kuhn (2008); Melega et al. (2018).

2.1. Joint production with by-products

The current paper studies a lot-sizing problem in the framework of an environmentally friendly context of an industrial symbiosis. Due to the topicality of the environmental concerns, the literature related to production planning problems under the prism of the circular economy is growing (e.g. Ilgin and Gupta (2010), Govindan and Soleimani (2017), Suzanne et al. (2020a)). However, despite the new issues brought by the flows of end-of-life products and production residues, the material exchange involved in industrial symbiosis networks is not investigated in the production planning literature, to the best of our knowledge. As an industrial symbiosis network is organized on the basis of joint production, let us discuss the production planning literature dealing with co-production and generation of by-products.

According to the European Waste Framework Directive[2], recall that: **(i)** a *by-product* is a production residue whose use is certain and without prior transformation, **(ii)** it cannot trigger the production even if an opportunistic demand arises, and **(iii)** its generation is unavoidable. On the contrary, a co-product has its own demand and can initiate the production process (see e.g. Bitran and Leong (1992); Lu and Qi (2011); Ağralı (2012)). Production planning problems dealing with production residues cannot be thus reduced or generalized to co-production problems. In line with the scope of this paper, let us focus in the following on the literature related to the management of unavoidable products at a tactical level.

To the best of our knowledge, only Sridhar et al. (2014) and Suzanne et al. (2020b) have treated production planning problems dealing with by-products from an academic point of view. Sridhar et al. (2014) studied a generic non-linear production planning problem, where the ratio of undesirable by-products increases monotonically as a convex function of the cumulative production mixture. Suzanne et al. (2020b) proposed a lot-sizing problem, where the by-product is generated in a constant proportion, stored in a limited capacity (constant and time-dependent) and transported to a further destination. They showed that the studied problem is NP-Hard when the capacity is time-dependent, and can be optimally solved in polynomial time when the capacity is constant. Solution approaches based on dynamic programming are proposed to solve these two cases of the lot-sizing problem dealing with by-products.

Given the actual circular economy concerns and the large number of interested industrial sectors (e.g. metal processing (Spengler et al., 1997), glass manufacturing (Taşkın and Ünal, 2009), semiconductor fabrication (Rowshannahad et al., 2018)), the current paper contributes to the joint production literature by investigating a joint production system in the context of a generic industrial symbiosis network. We study a single-item lot-sizing problem subject to different properties related to the storability potential of by-products.

2.2. Two-level lot-sizing problems

The studied problem can be considered as a two-level lot-sizing problem, where the first level corresponds to the problem encountered by the supplier, and the second level corresponds to the problem encountered by the receiver. There are multiple configurations of two-level production planning problems: the production-transportation problem (see e.g. Melo and Wolsey (2010, 2012); Hwang et al. (2016)), the supplier-retailer problem (see e.g. Brahimi et al. (2015); Phouratsamay et al. (2018)) and the One Warehouse Multi-Retailer (OWMR) problem (see e.g. Arkin et al. (1989); Solyalı and Süral (2012)) which are the most studied. As the OWMR problem is different from the problem studied in our paper, we only take interest of the production-transportation and supplier-retailer problems.

Let us position the problem studied in this paper within the two-level production planning literature according to its characteristics:

Inventory capacities at the supplier level. The two-level lot-sizing problem with inventory capacities at the retailer level can be polynomially solved using dynamic programming algorithms. Hwang and Jung (2011); Phouratsamay et al. (2018) provided algorithms running in a polynomial time to solve a number of different versions of this problem. Note that the case with inventory capacities at the first level is NP-Hard (Jaruphongsa et al. (2004); Brahimi et al. (2015); Phouratsamay et al. (2018)). Jaruphongsa et al. (2004) proposed a two-level problem with demand time-window constraints and stationary inventory bounds at the first level. By allowing demand splitting, Jaruphongsa et al. (2004) solved the problem using a polynomial algorithm based on dynamic programming. Brahimi et al. (2015) solved the supplier-retailer problem with inventory capacities using a Lagrangian relaxation. Phouratsamay et al. (2018) solved the same problem using a pseudo-polynomial algorithm based on dynamic programming. The problem studied in the current paper is different from Brahimi et al. (2015); Phouratsamay et al. (2018) in the sense that the inventory capacities are not on the product, which has to meet a demand.

External intermediate flows between levels. External intermediate flows between the two levels can occur while considering intermediate demands at the supplier level. Papers dealing with intermediate demands are not numerous. Melo and Wolsey (2010) considered a two-level lot-sizing problem, where an intermediate product is created at the first level, required to meet only the demand in the second level. On the contrary, Zhang et al. (2012) proposed valid inequalities to solve a multi-echelon lot-sizing problem, where the output of each intermediate echelon has its own external demand to fulfill and can also be used as an input to the next echelon. In the same way, Van Vyve et al. (2014) introduced a problem, where a unique intermediate product is used to multiple outputs. Ahmed et al. (2016) and He et al. (2015) dealt with the multi-stage version of the problem with intermediate demands of the final product as a minimum concave cost flow network problem. Against this background, the novelty of the problem under study lies in the consideration of external intermediate flows and the inventory bounds at the supplier level.

3. Problem statement

Consider a lot-sizing problem for an industrial symbiosis (ULS-IS), where two production units (PU1 and PU2) have to plan their production over a planning horizon of T periods, as illustrated in Figure 1. Each production unit produces a product to meet a deterministic demand. Denote by d_t^1 (resp. d_t^2) the demand of PU1 (resp. PU2) at period

$t \in \{1, 2, ..., T\}$. In addition, during the process of producing a quantity of X_t^1 units of main product in PU1, a quantity of X_t^1 units of by-products is generated. In the same way, to produce X_t^2 units of main product in PU2 at period $t \in \{1, 2, ..., T\}$, X_t^2 units of raw materials are required. The by-product generated by PU1 can be assimilated as the raw material needed to produce the main product of PU2.

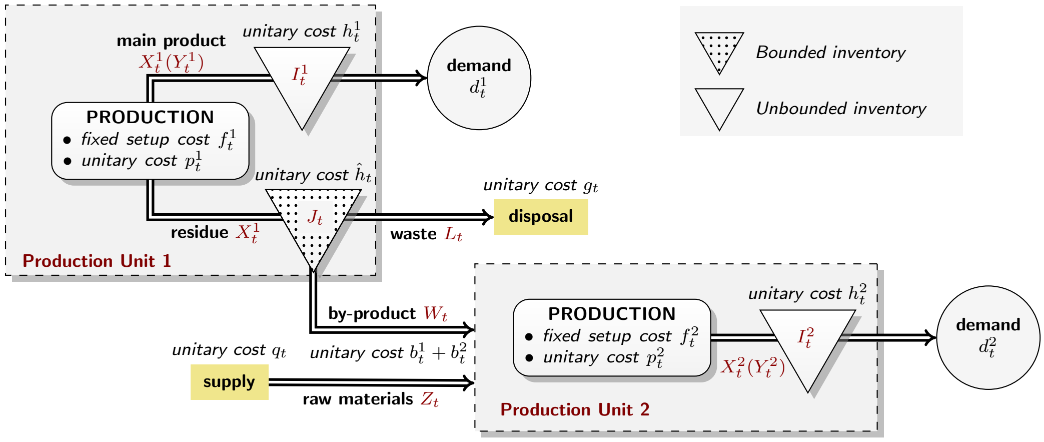


Figure 1: Process flow diagram of the ULS-IS problem

In order to ensure the procurement of raw materials, PU2 can supply its production either with the by-products generated by PU1, or with the raw materials from an external supplier. The quantity of by-products, which is not used by PU2, can be disposed of or stored by PU1, as long as the stored quantity does not exceed a limited capacity B in each period $t \in \{1, 2, ..., T\}$. The quantity of by-products, transported from PU1 to PU2 at each period $t \in \{1, 2, ..., T\}$, is denoted by W_t. The quantity of raw materials bought from the external supplier at period $t \in \{1, 2, ..., T\}$ is denoted Z_t. Let L_t (resp. J_t) be the quantity of by-products disposed of (resp. stored) in period $t \in \{1, 2, ..., T\}$.

The management of the exchange of by-products and the supply of raw materials generates the following unitary costs in each period $t \in \{1, 2, ..., T\}$:

- a unitary disposal cost g_t, paid by PU1,
- a unitary inventory holding cost $\hat{h}_t$ paid by PU1 to store the generated by-products,
- a unitary cost of reusing by-products of PU1 by PU2, decomposed into two unitary costs: b_t^1 (resp. b_t^2) paid by PU1 to prepare by-products for further use (resp. paid by PU2 to transport by-products from PU1 to PU2),
- a unitary purchasing cost q_t of raw materials supplied from an external supplier, paid by PU2.

Moreover, each PU pays the classical lot-sizing costs per period $t \in \{1, 2, ..., T\}$: a unitary production cost p_t^1 (resp. p_t^2), a fixed setup cost f_t^1 (resp. f_t^2), and a unitary holding cost h_t^1 (resp. h_t^2), paid by PU1 (resp. PU2). The binary setup indicators of production for PU1 and PU2 are denoted by Y_t^1 and Y_t^2, respectively. Let I_t^1 be the inventory level of the main product in PU1 at the end of period t, and I_t^2 be the inventory level of the product in PU2. The parameters and variables are summarized in Table 1.

In what follows, a number of assumptions are made, without loss of generality:

(A.1) The by-product inventory level is equal to zero at the end of the horizon, i.e. $J_T = 0$.

Table 1: Summary of the problem parameters

Parameters:	
T	Number of time periods
d_t^1 (d_t^2)	Demand for the main product of PU1 (PU2) in period t
p_t^1 (p_t^2)	Unitary production cost for PU1 (PU2) in period t
f_t^1 (f_t^2)	Fixed setup cost for PU1 (PU2) in period t
h_t^1 (h_t^2)	Unitary holding cost for the main product of PU1 (PU2) in period t
$\hat{h}_t$	Unitary holding cost for the by-product of PU1 in period t
q_t	Unitary purchasing cost of raw materials by PU2 from an external supplier in period t
b_t^1	Unitary treatment or transportation cost imputed to PU1 for the by-product in period t
b_t^2	Unitary treatment or transportation cost imputed to PU2 for the by-product in period t
g_t	Unitary by-product disposal cost of PU1 in period t
B	By-product inventory capacity in PU1 in each period
M_t^1 (M_t^2)	Big number with $M_t^1 = \sum_{i=t}^{T} d_i^1 \left(M_t^2 = \sum_{i=t}^{T} d_i^2 \right)$
Decision variables:	
X_t^1 (X_t^2)	Production quantity in PU1 (PU2) in period t
Y_t^1 (Y_t^2)	Binary setup indicator for PU1 (PU2) for period t
I_t^1 (I_t^2)	Inventory level of the main product of PU1 (PU2) at the end of period t
J_t	Inventory level of the by-product of PU1 at the end of period t
W_t	Quantity of by-products sent from PU1 to PU2 in period t
Z_t	Quantity of raw materials purchased at an external supplier by PU2 in period t
L_t	Disposal quantity of by-products in period t

(A.2) The treatment or transportation cost of the by-products imputed to PU1 is lower than their disposal cost performed by PU1, i.e. $b_t^1 \leq g_t, \forall t \in \{1, 2, \ldots, T\}$.

(A.3) The treatment or transportation cost of the by-product imputed to PU2 is lower than its purchasing cost, i.e. $b_t^2 \leq q_t, \forall t \in \{1, 2, \ldots, T\}$.

(A.4) The needs for raw materials in PU2 cannot trigger the production in PU1, i.e. $q_t \leq p_t^1 + b_t^1 + b_t^2, \forall t \in \{1, 2, \ldots, T\}$.

(A.5) A by-product demand in PU1 cannot trigger the production in PU2, i.e. $g_t \leq p_t^2 + b_t^1 + b_t^2, \forall t \in \{1, 2, \ldots, T\}$.

(A.6) On average, the by-product inventory holding cost is small enough to make possible the storage of by-products instead of their disposing of, i.e. $\sum_{t=1}^{T} \hat{h}_t \leq \sum_{t=1}^{T} (g_t - b_t^1)$. Otherwise, the problem to solve can be reduced to the problem without intermediate storage of the by-product.

Note that, if one of these assumptions is not met, the problem becomes trivial.

3.1. Straightforward formulation (AGG)

Using notations given in Table 1, the ULS-IS problem can be modeled via the following straightforward formulation:

$$\begin{aligned} \min \sum_{t=1}^{T} &(p_t^1 X_t^1 + f_t^1 Y_t^1 + h_t^1 I_t^1 + \hat{h}_t J_t + g_t L_t + b_t^1 W_t) \\ &+ \sum_{t=1}^{T} (p_t^2 X_t^2 + f_t^2 Y_t^2 + h_t^2 I_t^2 + q_t Z_t + b_t^2 W_t) \end{aligned} \tag{1}$$

$$
\begin{aligned}
\text{s.t. } & I^1_{t-1} + X^1_t - I^1_t = d^1_t, & \forall t \in \{1, 2, \ldots, T\} \quad (2)\\
& I^1_0 = 0, & (3)\\
& X^1_t \leq M^1_t Y^1_t, & \forall t \in \{1, 2, \ldots, T\} \quad (4)\\
& I^2_{t-1} + X^2_t - I^2_t = d^2_t, & \forall t \in \{1, 2, \ldots, T\} \quad (5)\\
& I^2_0 = 0, & (6)\\
& X^2_t \leq M^2_t Y^2_t, & \forall t \in \{1, 2, \ldots, T\} \quad (7)\\
& J_{t-1} + X^1_t = W_t + L_t + J_t, & \forall t \in \{1, 2, \ldots, T\} \quad (8)\\
& J_0 = J_T = 0, & (9)\\
& J_t \leq B, & \forall t \in \{1, 2, \ldots, T\} \quad (10)\\
& W_t + Z_t = X^2_t, & \forall t \in \{1, 2, \ldots, T\} \quad (11)\\
& X^1_t, X^2_t, I^1_t, I^2_t, W_t, Z_t, J_t, L_t \geq 0, & \forall t \in \{1, 2, \ldots, T\} \quad (12)\\
& Y^1_t, Y^2_t \in \{0, 1\}, & \forall t \in \{1, 2, \ldots, T\} \quad (13)
\end{aligned}
$$

The objective function (1) minimizes the sum of all costs: production, inventory holding, transportation, disposal and purchasing. Constraints (2) and (5) model the flow conservation of the main products of PU1 and PU2, respectively. Constraints (3) and (6) set to zero the initial inventory levels of the main products of both PU1 and PU2. Constraint (9) fix the initial and ending inventory levels of the by-product to zero. Constraints (4) and (7) are production constraints, which ensure that the production launching at a given period entails a setup operation at the same period. The inventory capacity of the by-product is limited by Constraints (10). Constraints (8) and (11) are conservation constraints of flows of by-products and external raw materials, which ensure that the production residue of PU1 is either disposed of, stored or used, and raw materials required for production of PU2 are bought. Constraints (12) and (13) are the nonnegativity and binary requirement constraints.

3.2. Facility location formulation (FAL)

In the lot-sizing literature, the straightforward formulation is very intuitive and easy to understand, but its linear relaxation usually provides a poor dual bound (Brahimi et al., 2017). As we can fear that the straightforward formulation becomes intractable for large size problems, we propose a disaggregated formulation, called *the facility location model* (Krarup and Bilde, 1977), to link the production variables not only with their production period, but also with their consumption period.

In the ULS-IS problem, there is production of two different main products linked by a flow of storable by-products. Thus, the facility location formulation of the ULS-IS problem requires the introduction of new variables that link production and inventory variables with: **(i)** production periods in PU1, **(ii)** consumption periods of the main product in PU1, **(iii)** periods of using by-products or periods of ordering raw materials (i.e. production periods in PU2) and, **(iv)** consumption periods of the main product in PU2. To do this, we introduce a new set of variables U_{ijkl}, $i \in \{1, ..., T\}$, $j \in \{i, i+1, ..., T\}$, $k \in \{i, i+1, ..., T\}$, $l \in \{k, k+1, ..., T\}$, which represent:

1. the quantity of the main product produced in PU1 in period i,
2. to fulfill a fraction of the demand of the main product in period j in PU1,
3. such that the quantity of by-products generated in i is stored until $k-1$ and sent to PU2 in order to produce the main product of PU2 in period k,

4. to meet the demand of PU2 in period l.

By convention, variables U_{00kl} are used to denote the quantity of raw materials purchased from an external supplier to be used in period k to produce the main product of PU2 for satisfying the demand in period l. We also consider variables U_{ijk0} to represent the quantity of by-products generated by PU1 in period i to satisfy the demand of the main product in period j, which is disposed of after having been stored until period k.

A new parameter a_{ijkl} corresponding to the cost associated with variables U_{ijkl}, $\forall i \in \{0, 1, ..., T\}$, $\forall j \in \{0, i, i+1, ..., T\}$, $\forall k \in \{i, i+1, ..., T\}$, $\forall l \in \{0, k, k+1, ..., T\}$ is added. Parameter a_{ijkl} is defined as follows:

$$a_{ijkl} = \begin{cases} p_i^1 + b_k^1 + b_k^2 + p_k^2 + \sum_{t=i}^{j-1} h_t^1 + \sum_{t=k}^{l-1} h_t^2 + \sum_{t=i}^{k-1} \hat{h}_t, & \text{if } i, j, l \neq 0 \ (i \leq j, i \leq k \leq l) \\ p_i^1 + g_k + \sum_{t=i}^{j-1} h_t^1 + \sum_{t=i}^{k-1} \hat{h}_t, & \text{if } i, j \neq 0, l = 0 \ (i \leq j, i \leq k) \\ p_k^2 + q_k + \sum_{t=k}^{l-1} h_t^2, & \text{if } i, j = 0, l \neq 0 \ (k \leq l) \\ +\infty, & \text{otherwise} \end{cases}$$

The facility location formulation of the ULS-IS problem is given below:

$$\min \sum_{i=1}^{T} \sum_{j=i}^{T} \sum_{k=i}^{T} \left(\sum_{l=k}^{T} a_{ijkl} U_{ijkl} + a_{ijk0} U_{ijk0} \right) + \sum_{k=1}^{T} \sum_{l=k}^{T} a_{00kl} U_{00kl} + \sum_{t=1}^{T} \left(f_t^1 Y_t^1 + f_t^2 Y_t^2 \right) \tag{14}$$

$$\text{s.t.} \sum_{i=1}^{j} \sum_{k=i}^{T} \left(\sum_{l=k}^{T} U_{ijkl} + U_{ijk0} \right) = d_j^1, \qquad \forall j \in \{1, 2, ..., T\} \tag{15}$$

$$U_{ijkl} \leq d_j^1 Y_i^1, \qquad \forall i, j, k, l \in \{1, 2, ..., T\} \ (i \leq j, i \leq k \leq l) \tag{16}$$

$$U_{ijk0} \leq d_j^1 Y_i^1, \qquad \forall i, j, k \in \{1, 2, ..., T\} \ (i \leq j, i \leq k) \tag{17}$$

$$\sum_{k=1}^{l} \left(\sum_{i=1}^{k} \sum_{j=i}^{T} U_{ijkl} + U_{00kl} \right) = d_l^2, \qquad \forall l \in \{1, 2, ..., T\} \tag{18}$$

$$U_{ijkl} \leq d_l^2 Y_k^2, \qquad \forall i, j, k, l \in \{1, 2, ..., T\} \ (i \leq j, i \leq k \leq l) \tag{19}$$

$$U_{00kl} \leq d_l^2 Y_k^2, \qquad \forall k, l \in \{1, 2, ..., T\} \ (k \leq l) \tag{20}$$

$$\sum_{i=1}^{t} \sum_{j=i}^{T} \sum_{k=t+1}^{T} \left(\sum_{l=k}^{T} U_{ijkl} + U_{ijk0} \right) \leq B, \qquad \forall t \in \{1, 2, ..., T\} \tag{21}$$

$$U_{ijkl}, U_{00kl}, U_{ijk0} \geq 0, \qquad \forall i, j, k, l \in \{1, 2, ..., T\} \ (i \leq j, i \leq k \leq l) \tag{22}$$

$$Y_i^1, Y_i^2 \in \{0, 1\}, \qquad \forall i \in \{1, 2, ..., T\} \tag{23}$$

The objective function (14) minimizes the sum of unitary and fixed costs. Constraints (15) and (18) ensure the demand satisfaction of PU1 and PU2, respectively. Constraints (16),(17), (19) and (20) are production constraints, i.e. there is a setup at a given period if there is production at the same period. Constraints (21) limit the level of by-product inventory. The definition domains of decision variables are given in Constraints (22)-(23).

Note that, experimental tests show that adding the following valid inequalities:

$$\sum_{k=i}^{T} \left(\sum_{l=k}^{T} U_{ijkl} + U_{ijk0} \right) \leq d_j^1 Y_i^1, \quad \forall i, j \in \{1.2, ..., T\}, (i \leq j) \tag{24}$$

$$\sum_{i=1}^{k}\sum_{j=i}^{T} U_{ijkl} + U_{00kl} \leq d_l^2 Y_k^2, \quad \forall k,l \in \{1.2,...,T\}, (k \leq l) \tag{25}$$

allows to substantially reduce the computational time needed to optimally solve the facility location formulation. The computational time is even lower when replacing constraints (16)-(17) and (19)-(20) by constraints (24)-(25).

3.3. Complexity analysis

In this section, we study the complexity of the ULS-IS problem. To do this, let us: **(i)** consider a particular case of the ULS-IS problem when the by-product is unstorable, **(ii)** show that this particular case is NP-Hard by performing a reduction from the classical capacitated lot-sizing problem, and **(iii)** derive the complexity of the general case of the ULS-IS problem. The following proposition holds:

Proposition 1. *The ULS-IS problem with an unstorable by-product ($B = 0$), i.e. with no by-product inventory (ULS-IS-NI) is NP-Hard.*

Proof. The proof of NP-Hardness is performed by reduction of ULS-IS-NI from the capacitated lot-sizing (CLS) problem, whose general case is known to be NP-Hard (Florian et al., 1980). The decision version of the CLS problem is defined by:

- a planning horizon of $\tilde{T}$ periods $\{1, 2, ..., \tilde{T}\}$,
- limited production capacities $\tilde{C}_t, \forall t \in \{1, 2, ..., \tilde{T}\}$,
- demands $\tilde{d}_t, \forall t \in \{1, 2, ..., \tilde{T}\}$,
- three cost components: fixed setup costs $\tilde{f}_t$, unit production costs $\tilde{p}_t$ and unit inventory holding costs $\tilde{h}_t, \forall t \in \{1, 2, ..., \tilde{T}\}$.

Let $\tilde{X} = (\tilde{X}_1, \tilde{X}_2, \ldots, \tilde{X}_T)$ be the vector of produced quantities, and $\tilde{I} = (\tilde{I}_1, \tilde{I}_2, \ldots, \tilde{I}_T)$ be the vector of inventory levels during the planning horizon. Denote by $\tilde{Y} = (\tilde{Y}_1, \tilde{Y}_2, \ldots, \tilde{Y}_T)$ the production indicator vector. The question posed by the CLS problem is: Does there exist a production plan $(\tilde{X}, \tilde{I}, \tilde{Y})$ of total cost at most equal to a given value V, which satisfies demands $\tilde{d} = (\tilde{d}_1, \tilde{d}_2, \ldots, \tilde{d}_T)$?

An instance I^{CLS} of the CLS problem can be transformed into an instance I of ULS-IS-NI by making the following substitutions $\forall t \in \{1, 2, \ldots, \tilde{T}\}$:

(S.1) Number of periods: $T = \tilde{T}$;
(S.2) Demands: $d_t^1 = \tilde{C}_t$, $d_t^2 = \tilde{d}_t$;
(S.3) Costs related to the main product of PU1: $f_t^1 = 0$, $p_t^1 = 0$ and $h_t^1 = 1$;
(S.4) Costs related to the product of PU2: $f_t^2 = \tilde{f}_t$, $p_t^2 = \tilde{p}_t$ and $h_t^2 = \tilde{h}_t$;
(S.5) Costs related to by-products of PU1 and raw materials of PU2: $g_t = 0$, $b_t = 0$ and $q_t = V$.

Let us show that instance I^{CLS} has an affirmative answer, if and only if, there exists a feasible solution $(X^1, Y^1, I^1, X^2, Y^2, I^2, W, Z, L)$ for instance I such that:

$$\sum_{t=1}^{T} \left(p_t^1 X_t^1 + f_t^1 Y_t^1 + h_t^1 I_t^1 + g_t L_t + b_t^1 W_t\right) + \sum_{t=1}^{T} \left(p_t^2 X_t^2 + f_t^2 Y_t^2 + h_t^2 I_t^2 + q_t Z_t + b_t^2 W_t\right) \leq V. \tag{26}$$

To do this, we prove the conditional relationship between CLS and ULS-IS-NI problems related to the solution existence.

($\Longrightarrow$). Suppose that instance I^{CLS} has an affirmative answer. Let $\tilde{S} = (\tilde{X}, \tilde{I}, \tilde{V})$ be a production plan, such that $\sum_{t=1}^{\tilde{T}} \left(\tilde{p}_t \tilde{X}_t + \tilde{f}_t \tilde{Y}_t + \tilde{h}_t \tilde{I}_t \right) \leq V$.

A feasible solution $(X^1, Y^1, I^1, X^2, Y^2, I^2, W, Z, L)$ for instance I, such that the total cost is at most equal to V, can be built as follows: **(i)** produce $X^1 = \tilde{C}$ quantities of the main product in PU1, this generates by-product quantities less than $\tilde{C}$ by virtue of substitution (S.2), **(ii)** $I^1 = 0$, hold inventory levels of the main product of PU1 to zero according to (S.3), **(iii)** transport to PU2 the quantity of by-product $W_t = \tilde{X}_t$ in each period and dispose of $L_t = \tilde{C}_t - \tilde{X}_t$ by virtue of substitution (S.5), **(iv)** produce $X^2 = \tilde{X}$ quantities in PU2, and **(v)** $I^2 = \tilde{I}$, hold the levels of the product in PU2 to $\tilde{I}$. Given substitutions (S.1)-(S.5), it follows that equation (26) is hold.

($\Longleftarrow$). Conversely, assume that instance I has a positive answer, i.e. there exists a production plan $(X^1, Y^1, I^1, X^2, Y^2, I^2, W, Z, L)$, which satisfies all demands with a cost at most equal to V. Making use of substitutions (S.1)-(S.5), it can immediately be checked that $\sum_{t=1}^{T} \left(\tilde{p}_t \tilde{X}_t + \tilde{f}_t \tilde{Y}_t + \tilde{h}_t \tilde{I}_t \right) \leq V$, where $\tilde{X} = X^2$, $\tilde{I} = I^2$ and $\tilde{Y} = Y^2$.

□

Remark 1. *As the ULS-IS-NI problem is a particular case of the ULS-IS problem, the ULS-IS problem is also NP-Hard.*

4. Solution method based on Lagrangian decomposition

Lagrangian decomposition approaches have been successfully used to solve a large variety of optimization problems. The main idea is to decompose a complex problem, often NP-Hard, into two or more easy to solve sub-problems. To do this, a set of variables is duplicated and the constraints corresponding to the equivalence of these variables are relaxed and penalized in the objective function by Lagrangian multipliers (see e.g. Fisher (1981)). Lagrangian decomposition provides a lower bound to the initial problem. Upper bounds can be computed using a Lagrangian heuristic, which transforms the obtained infeasible solutions into feasible ones. This procedure is repeated a large number of iterations in order to improve the obtained lower bounds.

The set of variables W_t, which links PU1 and PU2, corresponds to the flows of by-products. Let us duplicate these variables as follows: W_t represents the quantity of by-products to be sent to PU2 in period $t \in \{1, 2, ..., T\}$, and $\bar{W}_t$ represents the quantity of by-products to be bought from PU1 in period $t \in \{1, 2, ..., T\}$. The straightforward formulation of the problem under study becomes:

$$\begin{aligned} \min \sum_{t=1}^{T} (p_t^1 X_t^1 + f_t^1 Y_t^1 + h_t^1 I_t^1 + \hat{h}_t J_t + g_t L_t + b_t^1 W_t) \\ + \sum_{t=1}^{T} (p_t^2 X_t^2 + f_t^2 Y_t^2 + h_t^2 I_t^2 + q_t Z_t + b_t^2 \bar{W}_t) \end{aligned} \tag{27}$$

$$\text{s.t. } (2)-(7), (9)-(10), (12)-(13) \tag{28}$$
$$W_t + L_t + J_t = X_t^1 + J_{t-1}, \quad \forall t \in \{1, 2, ..., T\} \tag{29}$$
$$\bar{W}_t + Z_t = X_t^2, \quad \forall t \in \{1, 2, ..., T\} \tag{30}$$
$$\bar{W}_t = W_t, \quad \forall t \in \{1, 2, ..., T\} \tag{31}$$
$$W_t, \bar{W}_t \geq 0, \quad \forall t \in \{1, 2, ..., T\} \tag{32}$$

By relaxing constraints (31), the objective function becomes:

$$\begin{aligned} C_{LD}(\lambda) = \min \sum_{t=1}^{T} (p_t^1 X_t^1 + f_t^1 Y_t^1 + h_t^1 I_t^1 + \hat{h}_t J_t + g_t L_t + b_t^1 W_t) \\ + \sum_{t=1}^{T} (p_t^2 X_t^2 + f_t^2 Y_t^2 + h_t^2 I_t^2 + q_t Z_t + b_t^2 \bar{W}_t) \\ + \sum_{t=1}^{T} \lambda_t (\bar{W}_t - W_t) \end{aligned}$$

where $\lambda \in \mathbb{R}^T$ is the vector of Lagrangian multipliers.

The problem thus obtained can be separated into two sub-problems SP1 and SP2.

Sub-problem 1 ($SP1(\lambda)$ and $SP1(\lambda, \alpha)$). After applying the Lagrangian decomposition, the sub-problem referring to PU1 can be formulated as follows:

$$\min \sum_{t=1}^{T} \left(p_t^1 X_t^1 + f_t^1 Y_t^1 + h_t^1 I_t^1 + \hat{h}_t J_t + g_t L_t + (b_t^1 - \lambda_t) W_t \right) \tag{33}$$

$$\text{s.t. } (2)-(4), (8)-(10) \tag{34}$$
$$X_t^1, I_t^1, J_t, W_t, L_t \geq 0, \quad \forall t \in \{1, 2, ..., T\} \tag{35}$$
$$Y_t^1 \in \{0, 1\}, \quad \forall t \in \{1, 2, ..., T\} \tag{36}$$

Sub-problem (33)-(36), called for short $SP1(\lambda)$, corresponds to the single-item lot-sizing problem with a by-product, which can be solved using a time consuming polynomial algorithm when the inventory capacity of the byproduct is constant and is proved NP-Hard in the general case (see Suzanne et al. (2020b)). Let us apply a Lagrangian relaxation on capacity constraints (10). Constraints (10) are relaxed and penalized in the objective function by a vector of Lagrangian multipliers denoted $\alpha \in \mathbb{R}_+^T$. The sub-problem, called $SP1(\lambda, \alpha)$ is given as follows:

$$C_{SP1}(\lambda, \alpha) = \min \sum_{t=1}^{T} \left(p_t^1 X_t^1 + f_t^1 Y_t^1 + h_t^1 I_t^1 + (\hat{h}_t + \alpha_t) J_t + g_t L_t + (b_t^1 - \lambda_t) W_t - \alpha_t B \right)$$

$$\text{s.t. } (2)-(4), (8)-(9)$$
$$X_t^1, I_t^1, J_t, W_t, L_t \geq 0, \quad \forall t \in \{1, 2, ..., T\}$$
$$Y_t^1 \in \{0, 1\}, \quad \forall t \in \{1, 2, ..., T\}$$

Being constant, the term $-\alpha_t B$ can be discarded when solving SP1.

Sub-problem 2 ($SP2(\lambda)$). Sub-problem $SP2(\lambda)$ corresponds to the single-item lot-sizing problem to be solved by PU2 and can be formulated as follows:

$$C_{SP2}(\lambda) = \min \sum_{t=1}^{T} \left(p_t^2 X_t^2 + f_t^2 Y_t^2 + h_t^2 I_t^2 + q_t Z_t + (b_t^2 + \lambda_t)\bar{W}_t \right)$$

$$\begin{aligned} \text{s.t. } & (5) - (7), (11) & \\ & X_t^2, I_t^2, \bar{W}_t, Z_t \geq 0, & \forall t \in \{1, 2, \ldots, T\} \\ & Y_t^2 \in \{0, 1\}, & \forall t \in \{1, 2, \ldots, T\} \end{aligned}$$

Each of the sub-problems given above can be considered as a single-item lot-sizing problem and solved using a dynamic programming algorithm running in $O(T \log T)$ (Federgruen and Tzur, 1991; Wagelmans et al., 1992; Aggarwal and Park, 1993; van Hoesel et al., 1994). Note that the following properties characterizing the optimal solutions of sub-problems $SP1(\lambda, \alpha)$ and $SP2(\lambda)$ are true:

- $SP1(\lambda, \alpha)$: In an optimal solution, the generated quantities of by-products can either be disposed of or sent to PU2 after storage.
- $SP2(\lambda)$: An optimal solution, where each production period is supplied with raw materials originating from only one supplier, can be found.

Given the values of the Lagrangian multipliers, costs are updated by setting:

- $p'_t := p_t + \min_{t \leq u \leq T} \left(\sum_{v=t}^{u-1} (\hat{h}_v + \alpha_v) + \min\{b_u^1 - \lambda_u, g_u\} \right)$, $\forall t \in \{1, 2, \ldots, T\}$, in SP1,
- $p''_t := p_t + \min\{b_t^2 + \lambda_t, q_t\}$, $\forall t \in \{1, 2, \ldots, T\}$, in SP2.

Thus, the unitary production cost associated with the uncapacitated lot-sizing problems of $SP1(\lambda, \alpha)$ and $SP2(\lambda)$ are respectively p'_t and p''_t.

The pseudo-code of the proposed Lagrangian decomposition algorithm is presented in Algorithm 1. The goal of Algorithm 1 is to successively provide a high number of upper bounds of the ULS-IS problem to move closer to its optimal solution. To do this, at each iteration, sub-problems $SP1(\lambda, \alpha)$ and $SP2(\lambda)$ are independently solved by calling function `Solve(`$SP(\bullet)$`)`, which provides an optimal solution by using the very efficient dynamic programming algorithm of Wagelmans et al. (1992) that runs in $O(T \log T)$. Based on these optimal solutions, a upper bound is computed using the Lagrangian heuristic proposed in Section 4.1. If there is an improvement, the best found lower and upper bounds are updated and a local search procedure is applied (Section 4.2). Lagrangian multipliers are updated using the sub-gradient method described in Section 4.3.

Algorithm 1 Lagrangian decomposition algorithm

$\lambda_t = 0, \alpha_t = 0, \ \forall t \in \{1, 2, ..., T\}$
$\pi := \pi_{init}, \ z_{lb} := 0, \ z_{ub} := +\infty$
while stopping condition is not met **do**
  `Solve`$(SP1(\lambda, \alpha))$ and retrieve X^1, I^1, J, W, L, Y^1 ▷ see Section 4
  `Solve`$(SP2(\lambda))$ and retrieve $X^2, I^2, \bar{W}, Z, Y^2$ ▷ see Section 4
  $z^* \leftarrow C_{SP1}(\lambda, \alpha) + C_{SP2}(\lambda)$
  if $z^* > z_{lb}$ **then**
    $z_{lb} = z^*$
  end if
  $\hat{z}_{ub} \leftarrow$ `LagrangianHeuristic`$(z^*, X^1, I^1, J, W, L, Y^1, X^2, I^2, \bar{W}, Z, Y^2)$ ▷ see Section 4.1
  if $\hat{z}_{ub} < z_{ub}$ **then**
    $z_{ub} = \hat{z}_{ub}$
    $z_{ub} \leftarrow$ `LocalSearch`(z_{ub}, Y^1, Y^2, L, Z) ▷ see Section 4.2
  end if
  `UpdateMultipliers()` ▷ see Section 4.3
end while

4.1. Lagrangian heuristic

Lagrangian decomposition does not generally provide feasible solutions for ULS-SI problem, resulting by merging the obtained solutions of sub-problems $SP1(\lambda, \alpha)$ and $SP2(\lambda)$. The merged solution may violate the relaxed constraints. To recover feasibility, three main phases are executed in the framework of a heuristic formalized in Algorithm 2:

- **Phase** 1 *(Smoothing phase)*: The solutions of $SP1(\lambda, \alpha)$ and $SP2(\lambda)$ are crossed to create a feasible solution of the ULS-IS problem. To do this, quantities of by-products are moved in order to comply with inventory capacity constraints.

- **Phase** 2 *(Inventory balance phase)*: To improve the solution obtained after Phase 1, the by-product is moved in the inventory.

- **Phase** 3 *(Improvement phase)*: To improve the solution obtained in Phase 2, production quantities are moved to reduce the quantities of disposed by-products and purchased raw materials.

Phase 1 of Algorithm 2 creates feasible solutions of the original problem, by: **(i)** checking and satisfying the by-product inventory capacity constraints, and **(ii)** synchronizing the exchange of by-products between PU1 and PU2. Algorithm 2 first disposes of the extra inventory quantities. It ensures subsequently that, if there is a quantity of by-products available in PU1 and production in PU2, then there is an exchange of by-products between PU1 and PU2. Otherwise, no transfer between production units is created. Given Assumptions (A.2)-(A.3), Algorithm 2 exploits the property that, no disposal of by-products and purchasing of raw materials can be simultaneously done.

When the by-product is storable with a limited capacity, Phase 2 is applied to balance the by-product inventory in a less myopic way. The inventory balance phase consists of reducing in each period the disposal of or purchasing quantities by: **(i)** computing the cost of moving backward or forward by-product quantities in the inventory, while satisfying capacity constraints (done via the function called `CostInventory()` in line 3, Algorithm 2), **(ii)** applying the move corresponding to the minimal cost, if it is strictly negative. After Phase 3, Phase 2 is also called to adjust the inventory quantities of by-products.

Algorithm 2 `LagrangianHeuristic`($z_{lb}, X^1, I^1, J, W, L, Y^1, X^2, I^2, \bar{W}, Z, Y^2$)

Phase 1 – Smoothing phase

for $t = 1$ to T **do**
    if $J_t > B$ **then**
        $L_t \leftarrow J_t - B$, $J_t \leftarrow B$
    end if
    $A_t \leftarrow X_t^1 + J_{t-1}$
    if $A_t > 0$ and $X_t^2 > 0$ and $A_t \leq X_t^2$ **then**
        $W_t \leftarrow A_t$, $\bar{W}_t \leftarrow A_t$, $Z_t \leftarrow X_t^2 - A_t$
    else if $A_t > 0$ and $X_t^2 > 0$ and $A_t > X_t^2$ **then**
        $W_t \leftarrow X_t^2$, $\bar{W}_t \leftarrow X_t^2$, $L_t \leftarrow A_t - X_t^2$
    else if $A_t > 0$ **then**
        $L_t \leftarrow A_t$
    else
        $Z_t \leftarrow X_t^2$
    end if
    Update z_{lb}
end for

Phase 2 – Inventory balance phase

for $t = T$ to 1 **do**
    if $L_t > 0$ or $Z_t > 0$ **then**
        $cost \leftarrow$ `CostInventory`(L_t,t) or `CostInventory`(Z_t,t)
        if $cost < 0$ **then**
            Apply the corresponding move
            $z_{ub} \leftarrow z_{ub} + cost$
        end if
    end if
end for

Phase 3 – Improvement phase

for $t = T$ to 1 **do**
    if $L_t > 0$ or $Z_t > 0$ **then**
        $cost1 \leftarrow$ `CostMove`($\min\{L_t, I_t^1\}, t, t+i$) or `CostMove`($\min\{Z_t, I_t^2\}, t, t+i$)
        $cost2 \leftarrow$ `CostMove`($L_t, t, t-i$) or `CostMove`($Z_t, t, t-i$)
        $cost3 \leftarrow$ `CostMove`($L_t, t+i, t$) or `CostMove`($Z_t, t+i, t$)
        $cost4 \leftarrow$ `CostMove`($\min\{L_t, I_{t-1}^2\}, t-i, t$) or `CostMove`($\min\{Z_t, I_{t-1}^1\}, t-i, t$)
        if $\min(cost1, cost2, cost3, cost4) < 0$ **then**
            Apply the move corresponding to the minimal cost
            $z_{ub} \leftarrow z_{ub} + \min(cost1, cost2, cost3, cost4)$
        end if
    end if
end for
if $B \neq 0$ **then**
    Execute *Phase 2*
end if

Phase 3 aims to move the production backward or forward to reduce the quantities of by-products disposed of and raw materials purchased from an external supplier. For each period, if there is disposal of or purchasing operations, there are 4 actions that can contribute to improve the current solution. Each action is applied twice i.e. with: **(i)** next $t-1$ or $t+1$, and **(ii)** the previous or next production period with respect to period

t. Function `CostMove(`Q`,`t`,`t'`)` in Algorithm 2 computes the cost of moving the quantity Q from period t to period t'. More precisely, the goal is to reduce the disposal of and purchasing quantities in each period t, by:

(C.1) Moving a production quantity of PU1 (resp. PU2) in period t to the next period $t+1$ (or the next production period) in order to reduce the disposal of (resp. the purchasing) of byproducts. The cost $cost1$ in Phase 3 of Algorithm 2 corresponds to the cost of moving the minimal quantity between **(i)** L_t and I_t^1 or **(ii)** Z_t and I_t^2, from the current period t to the next period $t+1$ or the next production period $t' > t$.

(C.2) Moving a production quantity of PU1 (resp. PU2) in period t to the previous period $t-1$ (or the last production period before t) in order to reduce the disposal of (resp. the purchasing) of byproducts. The cost of moving quantity L_t or Z_t from the current period t to the previous period $t-1$ or the last production period $t' < t$ is denoted by $cost2$ in Phase 3 of Algorithm 2.

(C.3) Increasing the quantity of by-products transported between PU1 and PU2 by moving a production quantity of PU1 (resp. PU2) from the next period $t+1$ (or the next production period after t) to period t in order to reduce the purchasing of raw materials (resp. the disposable of) in period t. The cost of moving quantity L_t or Z_t from period $t+1$ or the next production period t' after t to the current period t is denoted by $cost3$ in Phase 3 of Algorithm 2.

(C.4) Increasing the quantity of by-products transported between PU1 and PU2 by moving a production quantity of PU1 (resp. PU2) from the previous period $t-1$ (or the last production period before t) to period t in order to reduce the purchasing of raw materials (resp. the disposable of) in period t. The cost of moving the minimal quantity between L_t or Z_t, and the inventory level of the main products in period $t-1$ from period $t-1$ or the last production period t' before t to the current period t is denoted by $cost4$ in Phase 3 of Algorithm 2.

For each possibility (C.1)-(C.4), the satisfaction of capacity constraints is checked. If the inventory capacity prevents from performing a move, the associated cost is set to zero.

In each of the cases (C.1)-(C.4), if the quantity moved equals to the produced quantity, the setup cost is subtracted. Moreover, if a production period is created, a setup cost is added. If the lowest cost thus obtained is negative, the corresponding move is applied and the objective function is updated. Algorithm 2 performs backward and is applied twice to adjust the moves corresponding to periods after the current period.

4.2. Local search procedure

To improve the obtained upper bounds, a local search procedure is executed. It consists of improving a given feasible solution, by applying a number of neighborhood operators.

The operators are applied on the solution obtained by Algorithm 2, and more precisely on the values of binary variables Y_t^1 and Y_t^2 for $t \in \{1, 2, ..., T\}$ corresponding to the setup indicators:

- The *setup removing* operator aims at grouping the production of two consecutive production periods in the earliest period. This consists of switching the value of a single binary variable from 1 to 0: if $Y_t^i = 1$ and $Y_{t'}^i = 1$, then $Y_t^i = 1$ and $Y_{t'}^i = 0$, for $t, t' \in \{1, 2, ..., T\}$, $t < t'$, $i \in \{1, 2\}$.

- The *setup moving* operator aims at exchanging the setup indicators of two consecutive periods: if $Y_t^i = 1$ and $Y_{t'}^i = 0$, then $Y_t^i = 0$ and $Y_{t'}^i = 1$, for $t, t' \in \{2, 3, ..., T\}$, $t < t'$, $i \in \{1, 2\}$.

- The *setup synchronization* operator matches the production periods of PU1 and PU2: if $Y_t^1 + Y_t^2 = 1$, then $Y_t^1 = 1$ and $Y_t^2 = 1$, for $t \in \{2, 3, ..., T\}$.
- The *disposal or purchasing removing* operator fixes a setup in PU2, when there is a disposal of operation in PU1, and a setup in PU1, when there is a purchasing operation in PU2: if $L_t > 0$ then $Y_t^2 = 1$ otherwise if $Z_t > 0$ then $Y_t^1 = 1$, for $t \in \{2, 3, ..., T\}$.

Based on the above introduced neighborhood operators, the local search procedure is formalized in Algorithm 3.

Algorithm 3 `LocalSearch(`z_{ub}, Y^1, Y^2, L, Z`)`

1: $z_{ub} \leftarrow$ `SetupSynchronization(`z_{ub}, Y^1, Y^2`)`
2: $z_{ub} \leftarrow$ `DisposalRemoving(`z_{ub}, Y^1, Y^2, L`)`
3: $z_{ub} \leftarrow$ `PurchasingRemoving(`z_{ub}, Y^1, Y^2, Z`)`
4: $z_{ub} \leftarrow$ `SetupMoving(`z_{ub}, Y^i`, PU_i)`, $i \in \{1, 2\}$
5: $z_{ub} \leftarrow$ `SetupRemoving(`z_{ub}, Y^i`, PU_i)`, $i \in \{1, 2\}$

Function `SetupSynchronization(•)` in Algorithm 3 applies the *setup synchronization* operator to all periods from period 2 to period T. For each period, if the value of the objective function is improved, the associated solution is kept and serves as a basis for the next iteration. Algorithm 3 is composed of the succession of functions corresponding to the neighborhood operators introduced above. The succession, in which the operators are applied, has been empirically determined. To limit the computational time, Algorithm 3 is applied only when the upper bound is improved by Algorithm 2.

4.3. Updating Lagrangian multipliers

The Lagrangian multipliers used in Algorithm 1 are initialized to 0, and then updated using the sub-gradient method. The procedure is named `UpdateMultipliers()` in Algorithm 1. Let δ be the sub-gradient composed by the vectors δ^1 and δ^2 such that:

$$\delta_t^1 = \bar{W}_t - W_t, \quad \delta_t^2 = J_t - B, \quad \forall t \in \{1, 2, ..., T\}.$$

The step size Δ is calculated using the formula:

$$\Delta = \frac{\pi(UB^* - LB)}{\|\delta\|^2}$$

where UB^* is the value of the best known feasible solution and LB is the current lower bound.

The Lagrangian multipliers λ_t are updated using the following formula, $\forall t \in \{1, 2, ..., T\}$:

$$\lambda_t = \lambda_t + \Delta\delta_t^1, \quad \forall t \in \{1, 2, ..., T\}.$$

The Lagrangian multipliers α_t are computed as follows:

$$\alpha_t = \begin{cases} \alpha_t + \Delta\delta_t^2, & \text{if } \alpha_t + \Delta\delta_t^2 > 0 \\ 0, & \text{otherwise.} \end{cases} \quad \forall t \in \{1, 2, ..., T\}$$

Usually, the scalar π is initially equal to 2. This coefficient is divided by 2 whenever the lower bound is not improved in a fixed number of iterations. In our settings, if there is an improvement after each 3 iterations, π is not modified, otherwise π is multiplied by 0.8.

If no improvement is recorded during a fixed number of iterations, a multi-start procedure is called. This procedure consists of taking the values of the last Lagrangian multipliers, multiplying them by random values and continuing the execution of Algorithm 1. For the ULS-IS problem, Lagrangian multipliers are multiplied by values between 0.5 and 2. After 600 iterations without improvement, each Lagrangian multiplier is multiplied by a random value comprised between 0.5 and 2.

5. Collaboration policies

In this section, let us investigate a number of different collaboration policies in an industrial symbiosis system controlled by two decision makers. As previously discussed, a centralized collaboration policy (i.e. the ULS-IB problem) can be only applied in full information sharing environments. Nowadays, the lack of information sharing remains a major barrier in the expansion of industrial symbiosis networks (Fraccascia and Yazan, 2018). At the other extreme, a full decentralized decision-making process is not globally consistent, since each decision maker pursues its own local objectives with its local constraints, which does not necessarily maximize the global benefits.

With respect to the centralized collaboration policy modeled via the ULS-IB problem, we study the following baseline collaboration policies: **(i)** a policy without collaboration, **(ii)** a policy expressing an opportunistic collaboration and, **(iii)** two sequential decentralized collaboration policies expressing a symbiotic partnership, when one production unit dominates another one, i.e. makes first its production plan.

The production plans are obtained as follows:

- **No collaboration:** As no interaction is considered between PU1 and PU2, production plans can be found by solving separately sub-problems $SP1(\lambda)$ and $SP2(\lambda)$ with: **(i)** $W_t = 0,\ \forall t \in \{1, 2, ..., T\}$ in PU1, and **(ii)** $\bar{W}_t = 0,\ \forall t \in \{1, 2, ..., T\}$ in PU2.

- **Opportunistic collaboration:** This policy presupposes the replacement of raw materials with by-products whenever possible, by matching the production plans of PU1 and PU2 calculated separately. The first iteration of Algorithm 1 implements this collaboration policy.

- **Sequential decentralized collaboration policies:** Let us consider the case when one production unit makes its production plan first, then another production unit proceeds to the decision-making accordingly. The management of the by-product flows transported from the supplier PU to the receiver PU is discussed in what follows for both cases, when decisions are made downward, and when decisions are made upward.

Downward sequential decision-making. Let us suppose that PU1 makes its production plan and communicates to PU2 the quantities of by-products available in each period. Subsequently and knowingly, PU2 establishes its informed production plan, by taking advantage of the by-products generated by PU1.

The problem to be solved by PU1 corresponds to the sub-problem $SP1(\lambda)$. It is solved by the dynamic programming algorithm of Wagelmans et al. (1992) that runs in $O(T \log T)$. Let $w_t = W_t$ be the quantity of by-products generated by PU1 and available for PU2, $t \in \{1, 2, \ldots, T\}$. The resulting problem for PU2 is a basic supplier selection problem and can be formulated by a mixed integer linear programming model as follows:

$$\min \sum_{t=1}^{T} \left(p_t^2 X_t^2 + f_t^2 Y_t^2 + h_t^2 I_t^2 + q_t Z_t + b_t^2 W_t \right)$$

$$\begin{aligned} \text{s.t. } & (5) - (7), (11) & \\ & W_t \leq w_t, & \forall t \in \{1, 2, \ldots, T\} \\ & X_t^2, I_t^2, W_t, Z_t \geq 0, & \forall t \in \{1, 2, \ldots, T\} \\ & Y_t^2 \in \{0, 1\}, & \forall t \in \{1, 2, \ldots, T\} \end{aligned}$$

By virtue of Assumption (A.3), this problem can be solved using a dynamic programming algorithm by setting:

$$p_t(Q) = \begin{cases} (p_t^2 + b_t^2)Q, & \text{if } Q \leq w_t \\ p_t^2 Q + b_t^2 w_t + q_t(Q - w_t), & \text{otherwise} \end{cases}$$

Since the resulting production cost is concave, the algorithm proposed by Wagner and Whitin (1958), running in $O(T^2)$, is used to solve the problem.

After obtaining the production plans for both production units, they are crossed in order to update the disposal quantities L_t, as well as, the quantities of by-products W_t transported between PU1 and PU2.

Upward sequential decision-making. PU2 makes first its production plan and provides their needs in terms of by-products (i.e. recovered raw materials) to PU1. This collaboration policy can thus be reduced to a co-product problem with: **(i)** inventory bounds, **(ii)** a disposal of option, and **(iii)** possible lost sales on the by-product, since PU1 is allowed to not meet the demands for by-products.

The problem addressed by PU2 is a classical lot-sizing problem and can be solved using an algorithm running in $O(T \log T)$ (see e.g. Wagelmans et al. (1992)). Let w_t be the quantity of raw materials required by PU2 in period t. The problem to be solved by PU1 can be formulated as follows.

$$\min \sum_{t=1}^{T} \left(p_t^1 X_t^1 + f_t^1 Y_t^1 + h_t^1 I_t^1 + \hat{h}_t J_t + g_t L_t + b_t^1 W_t \right)$$

$$\begin{aligned} \text{s.t. } & (2) - (4), (8) - (10) & \\ & W_t \leq w_t, & \forall t \in \{1, 2, \ldots, T\} \\ & X_t^1, I_t^1, J_t, L_t, W_t \geq 0, & \forall t \in \{1, 2, \ldots, T\} \end{aligned}$$

This model is solved using a commercial solver. Once the production plan is made by PU1, quantities of by-products W_t are recovered by PU2 to adjust the real quantities of raw materials received from PU1 and those purchased from an external supplier.

6. Experimental results

6.1. Instances generation

Computational results have been conducted on heterogeneous instances randomly generated. All the costs are supposed to be stationary. The tested data sets are built for two time horizon lengths $T \in \{24, 96\}$. The horizon length defines the size of the problem. Parameters p^1, p^2, $h^1 \neq 0$, b^1, b^2, g and q are randomly generated between 0 and 10, while respecting Assumptions (A.1)-(A.6). To define setup costs f^1 and f^2, and holding cost h^2, critical parameters are identified, namely: ratio Δ between PUs, Setup cost-Holding cost Ratio (SHR), demands d^1 and d^2, and inventory capacity of by-products B:

- $\Delta = \dfrac{h^2}{h^1}$ aims at linking the holding costs of PU1 and PU2. This ratio can be *low* ($\Delta = 0.75$), *medium* ($\Delta = 1$) or *high* ($\Delta = 1.25$). Note that Δ is insightful to reveal the impact of one PU to the production plan of another PU.

- SHR is a well-known parameter in the lot-sizing literature (see e.g. Trigeiro et al. (1989)). This ratio has an impact on the average number of time periods between two consecutive setups, known as the Time Between Order (TBO). SHR links the setup and holding costs. As far as we consider a problem involving two different PUs, an SHR is generated for each PU, $SHR1$ for PU1 and $SHR2$ for PU2, which take their values in the set $\{3, 4, 5\}$.

- Demands d_t^1 and d_t^2, which have an impact on the size of production units, can be: **(i)** *low*: generated following a normal distribution with an average of 50 and a standard deviation of 10, $\forall t \in \{1, 2, ..., T\}$, **(ii)** *medium*: generated following a normal distribution with an average of 100 and a standard deviation of 20, $\forall t \in \{1, 2, ..., T\}$ or **(iii)** *high*: generated following a normal distribution with an average of 200 and a standard deviation of 40, $\forall t \in \{1, 2, ..., T\}$. We denote by $\bar{d}^1$ and $\bar{d}^2$ the average demands of PU1 and PU2, respectively.

- The by-product inventory capacity B can be: **(i)** *tight*: randomly generated around $1.2\bar{d}^1$, **(ii)** *large*: randomly generated around $3\bar{d}^1$, or **(iii)** *null*: when the by-product is unstorable, i.e. $B = 0$.

Given SHR and holding costs of both PUs, setup costs f^1 and f^2 can be computed via:

$$f^1 = \frac{1}{2} h^1 (SHR1)^2 \bar{d}^1$$
$$f^2 = \frac{1}{2} h^2 (SHR2)^2 \bar{d}^2$$

Data sets for each time horizon length $T \in \{24, 96\}$ are generated by combining all possible values of the critical parameters discussed above. By generating 10 instances for each class, the total number of generated instances per $T \in \{24, 96\}$ is $10 \times 3 \times 3 \times 3 \times 3 \times 3 \times 3 = 7{,}290$.

6.2. Design of experiments

We carried out the comparison between the following approaches:

- `AGG`: Straightforward formulation (1)-(13) of the ULS-IS problem solved via CPLEX.

- `FAL`: Facility location formulation (14)-(23) of the ULS-IS problem solved via CPLEX.

- Several variants of Lagrangian decomposition Algorithm 1 are tested:
  - `LD`: Lagrangian heuristic (i.e. Algorithm 2) without any local search procedure.
  - `LD-LS`: Lagrangian heuristic (i.e. Algorithm 2) embedding the local search procedure given by Algorithm 3.
  - `LD-MS`: Lagrangian heuristic (i.e. Algorithm 2) within a multi-start procedure, called when there is no improvement after a fixed number of iterations. This procedure consists of taking the values of the last Lagrangian multipliers, multiplying them by random values, and continuing Algorithm 1.
  - `LD-MS-LS`: Lagrangian heuristic (i.e. Algorithm 2) embedding the local search procedure given by Algorithm 3 within a multi-start procedure.
  - `LD-LP`: The Lagrangian heuristic given in Algorithm 2 is replaced by the resolution of the linear program obtained by fixing the production periods according to the solutions of sub-problems $SP1(\lambda, \alpha)$ and $SP2(\lambda)$.
  - `LD-LP-MS-LS`: `LD-LP` embedding the local search procedure given by Algorithm 3 within a multi-start framework.

Algorithm 1 is stopped after 1,000 iterations. For large size problems, this stopping criterion is strengthened by a time limit fixed at 10 seconds (s). The gap between the upper bound (UB) and the lower bound (LB) is given by the formula:

$$Gap = 100 \times \frac{UB - LB}{UB}$$

Numerical tests were carried out on a computer with Intel Xeon e5-2620 2.1GHz CPU with 32GB RAM. Models `AGG` and `FAL` and linear programs in `LD-LP` and `LD-LP-MS-LS` were solved using IBM ILOG CPLEX 12.6. The Lagrangian decomposition algorithm was implemented using the C++ programming language on Microsoft Visual Studio 2013.

6.3. Numerical results

The goal of the conducted experiments is manifold: **(i)** to evaluate the impact of critical parameters, **(ii)** to discuss the contribution of the different procedures described above, **(iii)** to show the competitiveness of the proposed solution method with respect to `AGG`, **(iv)** to characterize the generated instances.

On average, `LD` outperforms `LD-LP`, regardless the used improvement procedure. For this reason, only `LD-LP` and `LD-LP-MS-LS` are kept to empirically study the performance of the proposed solution method.

Algorithm 1 and its variants. By summarizing Table A.6, Table 2 provides the distribution of the gaps between UB and LB for the different variants of Algorithm 1. The improvement obtained with the multi-start procedure is too low to be noticeable after rounding of the results. On the contrary, the local search procedure reduces the maximum gaps from 3.17 (3.30) to 2.88 (3.01) in the unstorable (storable) case. In the same way, the CPU time increases when using the local search procedure (multiplied by 2 or even 3), while remaining acceptable (always below 1s). The combination of the multi-start and local search procedures is particularly efficient when the by-product is storable. The average gap decreases from 0.55% to 0.38%.

The performance of the multi-start and local search procedures can be also observed when comparing the results obtained by `LD-LP` and `LD-LP-MS-LS`. The difference of gaps between LD-MS-LS and LD-LP-MS-LS are: **(i)** *relatively small* for $B \neq 0$, LD-LP-MS-LS being twice slower, **(ii)** *rather high* for $B = 0$, LD-LP-MS-LS being three times slower.

To sum up, LD-MS-LS outperforms other variants of Algorithm 1 in terms of both the solution quality and computational time, and is considered by default in what follows.

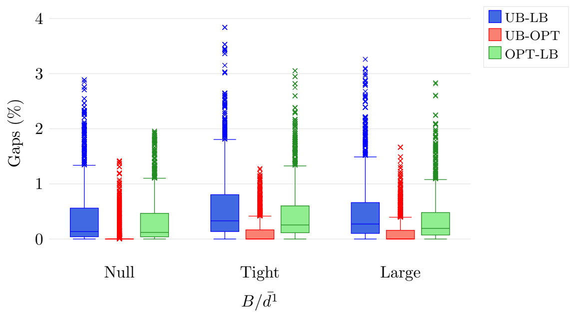
Table 2: Algorithm 1 and its variants: *Gap distribution (in %) between UB and LB for small size instances*

B	Variant	Mean	Standard deviation	Max	Median	CPU time
Null	LD	0.42	0.57	3.17	0.16	0.22
	LD-MS	0.42	0.57	3.17	0.16	0.22
	LD-LS	0.39	0.54	2.88	0.14	0.72
	LD-MS-LS	0.39	0.53	2.88	0.14	1.16
	LD-LP	0.99	1.45	9.46	0.26	3.70
	LD-LP-MS-LS	0.61	0.88	6.24	0.21	3.61
Non-null	LD	0.55	0.66	3.30	0.26	0.40
	LD-MS	0.54	0.65	3.30	0.26	0.40
	LD-LS	0.38	0.54	3.01	0.14	0.83
	LD-MS-LS	0.38	0.54	3.01	0.14	0.85
	LD-LP	0.60	1.10	9.88	0.11	2.21
	LD-LP-MS-LS	0.40	0.72	8.52	0.09	2.29

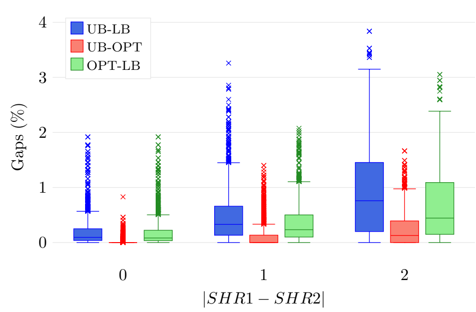
Analysis of the critical parameters. Table A.5 shows that the tightness of the by-product inventory capacity has a high impact on the CPU time spent by CPLEX to solve AGG and FAL. The more the capacity is tight, the higher are computational times needed to solve AGG and FAL.

One can also remark that the closer the values of $SHR1$ and $SHR2$, the faster the problem is to solve. On the contrary, when the average demand of PU1 is close to the average demand of PU2, the optimal solution is found after a higher CPU time. Moreover, when $SHR1$ and $SHR2$ both increase, the CPU times needed to solve AGG and FAL increase (e.g. for AGG, it is around 0.30s for $SHR1 = SHR2 = 3$, and 0.60s for $SHR1 = SHR2 = 5$). The value of Δ intensifies the impact of the parameters on the CPU time, since it operates with the difference between the holding costs, and accordingly between the setup costs. Apart from this fact, the impact of Δ is negligible.

The last row of Table A.5 highlights that the time spent by CPLEX to solve FAL is: **(i)** on average higher than for AGG, **(ii)** inhomogeneous and can be very high for some classes of instances (16.18s in average for $SHR1 = 5$, $SHR2 = 3$ with a tight capacity). We conclude that FAL formulation is less efficient than AGG for most instances, and consequently it is not used to solve large-size instances.



(a) Gaps as a function of $B/\bar{d}^1$



(b) Gaps as function of $|SHR1 - SHR2|$

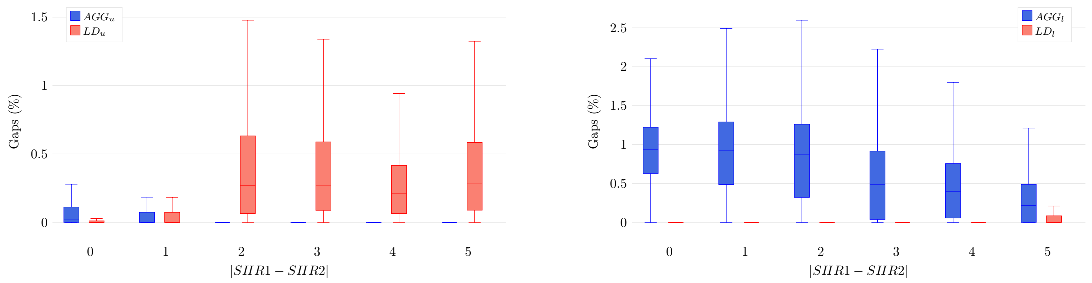
Figure 2: Distribution of gaps (%) for $T = 24$: LD *versus* AGG

Focus on the effectiveness of LD-MS-LS *versus* AGG*.* The solution quality provided by LD-MS-LS is studied on small instances with respect to the value of B, $SHR1$ and $SHR2$. Figure 2 provides the boxplots showing the distributions of gaps between: **(i)** UB and LB for LD-MS-LS, denoted by UB-LB, **(ii)** UB and optimal solution OPT, denoted by UB-OPT, **(iii)** OPT and LB, denoted by OPT-LB.

To evaluate the effectiveness of LD-MS-LS, no time limit has been imposed to CPLEX. The higher gaps are observed for instances with high computational times for AGG. For $B = 0$, the gap UB-LB is always below 3%, and the maximum gap UB-OPT is near 1.7%. When $SHR1 = SHR2$, gaps are very closed to 0 in more than three quarters of instances (see Figure 2b). The higher the difference between $SHR1$ and $SHR2$ is, the significant the gaps are. For 75% of instances with $|SHR1 - SHR2| = 2$, UB-OPT is below 0.5%. Gaps UB-LB are quite large, but rarely exceed 2%. For $B \neq 0$, the gaps UB-OPT increase with the inventory capacity, whereas the gaps OPT-LB tend to decrease (see Figure 2a). When B is large, 75% of gaps UB-OPT and OPT-LB are below 1% for the worst combination of critical parameters. When B is tight, 75% of gaps UB-OPT are below 0.7%, and 75% of gaps OPT-LB are below 1.5% for the same combination of critical parameters.

Focus on the efficiency of LD-MS-LS *versus* AGG*.* Let UB_{LD} and LB_{LD} (resp. UB_{AGG} and LB_{AGG}) be the lower and upper bounds obtained by Algorithm 1 (resp. formulation AGG). The maximum CPU time allowed for CPLEX has been limited to 10s.

To compare the quality of the primal and dual bounds, let us define: **(i)** $\mathtt{LD_u} = UB_{LD} - UB^*$ and $\mathtt{AGG_u} = UB_{AGG} - UB^*$ where $UB^* = \min\{UB_{AGG}, UB_{LD}\}$, **(ii)** $\mathtt{LD_l} = LB^* - LB_{LD}$ and $\mathtt{AGG_l} = LB^* - LB_{AGG}$, where $LB^* = \max\{LB_{AGG}, LB_{LD}\}$. These gaps are provided in Table A.8.



(a) Gaps to UB (b) Gaps to LB

Figure 3: LD versus AGG as function of $|SHR1 - SHR2|$ for $T = 96$

In general, AGG provides better upper bounds than LD-MS-LS. However, when $SHR1 = SHR2$, $\mathtt{LD_u}$ is very close to $\mathtt{AGG_u}$ for a lower running time. In particular, when $B = 0$, LD-MS-LS provides upper bounds almost as good as $\mathtt{AGG_u}$ in only around 1.5s against around 8.8s taken by AGG. Also, the lower bounds obtained by LD-MS-LS are generally better than $\mathtt{AGG_l}$, when $B = 0$. On the contrary, $\mathtt{LD_l}$ is relatively poor compared to $\mathtt{AGG_l}$ for $B \neq 0$. The more SHRs increase, the more the gaps are significant.

Challenging instances. In the light of the findings previously discussed, let us challenge AGG and LD-MS-LS, by increasing the values of SHRs up to 8. The new created instances are discussed in terms of the heterogeneity in SHRs and the maximum value between $SHR1$ and $SHR2$.

- **High heterogeneity in SHR:** As Figure 2b shows, gaps to UB of LD-MS-LS increase when $SHR1$ and $SHR2$ are very different. For instances with $|SHR1 -$

Table 3: Gaps to UB (in %): $T = 96$, $SHR1, SHR2 \in \{6, 7, 8\}$ and $|SHR1 - SHR2| \leq 1$

$SHR1$	$SHR2$	**Null** B		**Tight** B		**Large** B	
		$\texttt{AGG}_u$	$\texttt{LD}_u$	$\texttt{AGG}_u$	$\texttt{LD}_u$	$\texttt{AGG}_u$	$\texttt{LD}_u$
6	6	0.02	0.03	0.04	0.03	0.04	0.06
6	7	0.02	0.07	0.08	0.06	0.07	0.09
7	6	0.02	0.11	0.08	0.08	0.06	0.14
7	7	0.02	0.03	0.11	0.01	0.08	0.03
7	8	0.04	0.03	0.15	0.01	0.12	0.02
8	7	0.02	0.05	0.12	0.02	0.1	0.04
8	8	0.03	0.01	0.14	0	0.13	0.01
Average CPU time		10.0	1.2	10.0	7.2	10.0	7.6

$SHR2| \geq 2$, AGG provides better UBs than LD-MS-LS (see Figure 3a) . However, LB_{AGG} is very poor compared to LB_{LD}. This makes LD-MS-LS globally better than AGG in terms of closeness between bounds. For $|SHR1 - SHR2| = 5$, gaps to UB reach more than 2%, that makes the corresponding instances difficult to solve.

- **High dispersion in SHR:** Focus now on instances with $SHR1, SHR2 \in \{6, 7, 8\}$ and $|SHR1 - SHR2| \leq 1$. They correspond to instances with high setup costs, which may make difficult the synchronization between PUs. The obtained results are provided in Table 3. AGG outperforms LD-MS-LS only when one of the SHRs equals to 6. In all other cases, LD-MS-LS provides better feasible solutions than AGG, especially when B is tight.

7. Managerial implications and research perspectives

In this section, let us discuss the economic opportunities induced by the exchange of by-products between two production units, by examining five baseline collaboration policies, namely:

- No_Co: *No collaboration*, i.e. no symbiotic partnership is considered between production units. The by-products generated by PU1 are disposed of, and raw materials used by PU2 are purchased from an external supplier. Let the costs obtained in the framework of this policy be called *nominal costs*.
- Full_Co: *Full collaboration*, i.e. the exchange of by-products are planned in the framework of a centralized collaboration policy. No other policy can provide better gains. Let the costs obtained in the framework of this policy be called *centralized costs*.
- Opp_Co: *Opportunistic collaboration*, i.e. the exchange of by-products is being done by taking advantage of a fortunate matching between the production plans of the supplier (PU1) and the receiver (PU2).
- Two sequential decentralized collaboration policies:
  - PU1_First: *Downward sequential collaboration*,
  - PU2_First: *Upward sequential collaboration*.

The aforementioned policies were addressed in Section 5, where the used models and the associated solution methods are detailed. For the sake of simplicity and without loss of generality, the quantitative impact of each collaboration policy is evaluated on small size instances, i.e. for $T = 24$ periods and $SHR1, SHR2 \in \{3, 4, 5\}$.

The gains of each production unit i are calculated with respect to its nominal cost c^i obtained outside any symbiotic partnership, as follows:

$$(1 - c_p^i/c^i) \times 100,$$

where c_p^i is the cost of production unit i obtained in the framework of a collaboration policy denoted by p, $p \in \{$`Full_Co`, `Opp_Co`, `PU1_First`, `PU2_First`$\}$, $i \in \{1, 2\}$. The gains obtained for each of the aforementioned policy are provided in Table 4.

Even an opportunistic exchange of by-products creates value-added benefits for both PUs, reaching from slightly over 2.4% on average to a maximum of more than 17%. It is worthwhile to observe that the more the decision-making is informed, the more significant are the benefits of each production unit. Both of the sequential decentralized collaboration policies double the savings procured by exchanging by-products, being far from the centralized costs (`Full_Co`) within a distance of around 1%. In the same line of thought, in the `PU2_First` policy, PU1 knows the by-product needs of PU2. In this case, this knowledge helps PU1 improving its gains by 1% compared to the gains obtained by policy `PU1_First`. In our experiments, it appears that the knowledge of PU2 about the availability of by-products (in the `PU1_First` policy) does improves significantly its gains compared to the gains obtained with the `PU2_First` policy. This means that sometimes the by-product supplier can obtain higher gains when moving at second instance. It is also important to mention that in the `Full_Co` policy, even if the average and the maximal gains are higher than the gains obtained by other policies, PU1 can lose up to 3.6% and PU2 can lose up to 4.2% of their total nominal costs, when the by-product is storable. In this case, to make the proposed solution acceptable by both PUs, compensation mechanisms have to be considered. Some perspectives related to these mechanisms are addressed at the end of this section.

Table 4: Collaboration policies: *Gains of PU1 and PU2 (in %) against* `No_Co`

PU	B	`Opp_Co`			`PU1_First`			`PU2_First`			`Full_Co`		
		min	mean	max	min	mean	max	min	mean	max	min	mean	max
PU1	Null	0	3.1	17.3	0	7	40.1	0	6.9	39.6	-1.4	7.9	40.0
	Tight	0	3.4	21.8	0	7.5	43.5	0	8.3	44.5	-3.6	9.2	48.2
	Large	0	3.2	21.8	0	7.3	43.5	0	8.7	44.5	-1.4	9.4	48.2
PU2	Null	0	2.8	18.2	0	6.6	41.5	0	6.5	39.6	-4.7	7.2	42.1
	Tight	0	2.7	17.6	0	6.3	47.9	0	6.4	48.3	-4.2	7.4	48.4
	Large	0	2.4	17.6	0	5.9	46.3	0	6.7	48.4	-4.2	7.5	48.4

Impact of technology characteristics on collaboration policies. Figures 4-5 highlight the economic impact of the by-product storability with respect to $(SHR1 - SHR2)$ and $(\bar{d}^1 - \bar{d}^2)$.

Figure 4 (resp. Figure 5) reports the average relative gains of each PU obtained by policy `Full_Co` against `No_Co` as function of $(SHR1 - SHR2)$ (resp. $(\bar{d}^1 - \bar{d}^2)$) for storable and unstorable by-products. Once again compared to policy `No_Co`, Figures 6a-6b (resp. Figures 6c-6d) report the relative gains of each PU as function of $(SHR1 - SHR2)$ (resp. $(\bar{d}^1 - \bar{d}^2)$) for the three production policies: `Full_Co`, `PU1_First`, `PU2_First`.

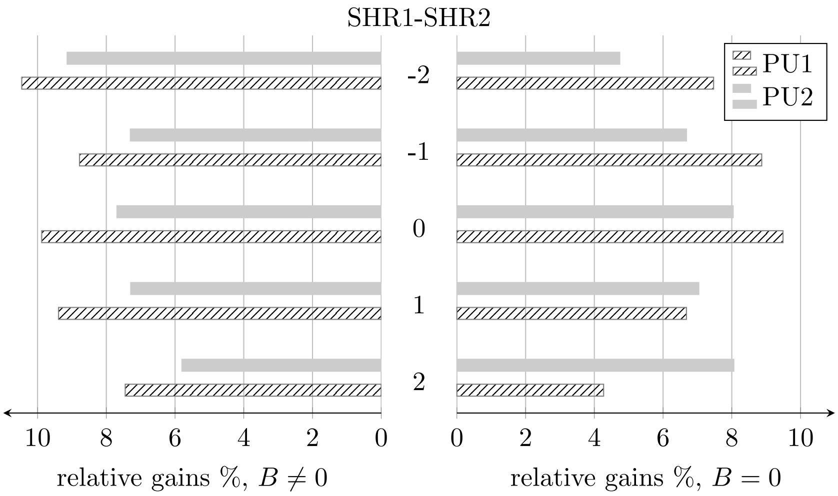


Figure 4: Relative gains (in %) as function of $(SHR1 - SHR2)$: *Policy* `Full_Co`

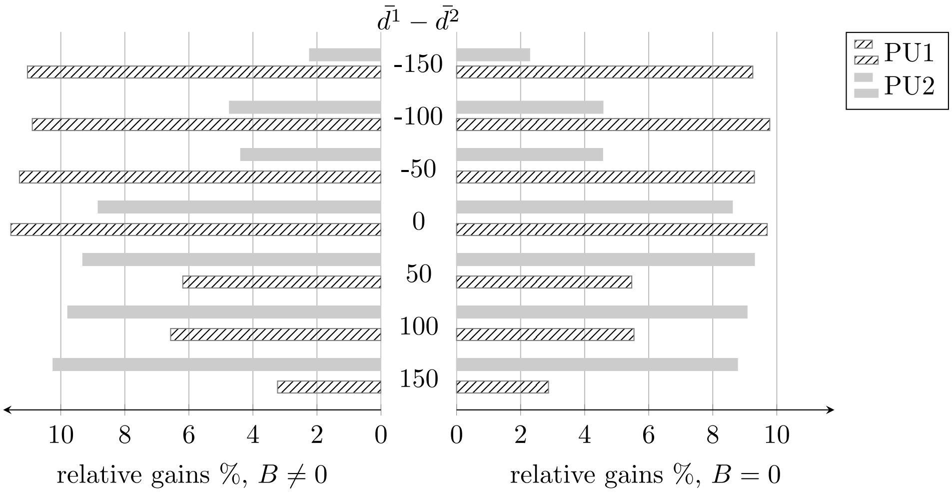
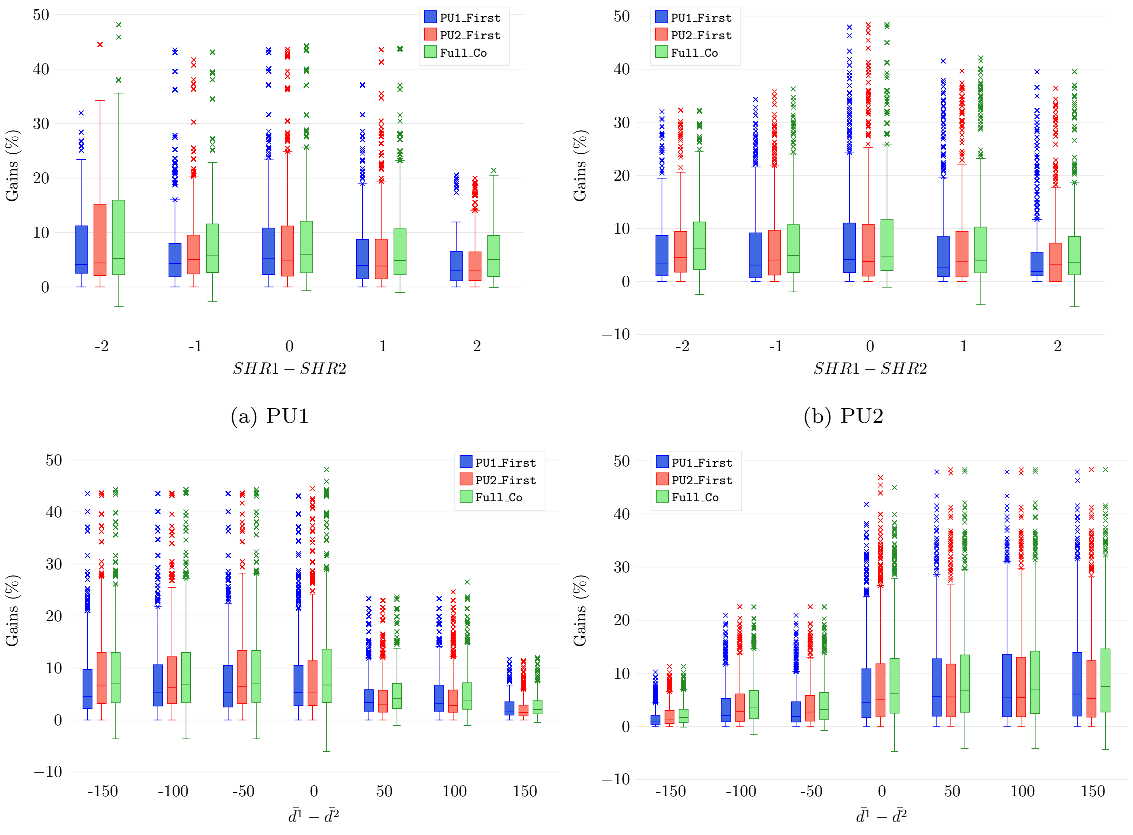


Figure 5: Relative gains (in %) as function of $(\bar{d}^1 - \bar{d}^2)$: *Policy* `Full_Co`

First, let us analyze the impact of $(SHR1 - SHR2)$ on obtained gains regarding the storability or not of by-products. From Figure 4, we can notice that when the by-product is unstorable and $(SHR1 - SHR2)$ is high, PU2 has more gains than PU1. This can be explained by the fact that when the SHR is high for PU1 and low for PU2, PU2 has more flexibility to align its production with the one of PU1. When the situation is inverted, i.e. $(SHR1 - SHR2)$ is very small, the average relative gain of PU1 is higher than the one of PU2. In this case, PU1 has more flexibility to align its production with the one of PU2. When the by-product is storable, we notice that the relative gains of PU1 are always higher than those of PU2 regardless the value of $(SHR1 - SHR2)$. This can be explained by the flexibility of PU1, which has more freedom to manage the by-product inventory level. From Figures 6a-6b, we notice that the higher gains are obtained when the SHR of both PUs are close, i.e. $SHR1 \approx SHR2$. Negative gains are obtained when the difference between the SHR of both PUs is high, i.e. $|SHR1 - SHR2| = 2$.

Focus now on the impact of the balance between average demands of production units on benefits obtained in the framework of different policies. As Figure 5 corroborates in the case of policy `Full_Co`, the higher the difference between demands, the more the gains



(a) PU1 (b) PU2

(c) PU1 (d) PU2

Figure 6: Gains (in %) as function of $(SHR1 - SHR2)$ and $(\bar{d}^1 - \bar{d}^2)$

of PUs are unbalanced. From Figures 6c-6d, it appears to be more beneficial for each PU when the production unit, which has the largest demand, establishes its production plan first. This finding differs from that encountered in classical supply chains, where the primacy in the decision-making ensures the maximum benefits.

- `PU1_first` and $\bar{d}^1 \gg \bar{d}^2$: Having to meet large demands, PU1 generates and makes available large quantities of by-products. This configuration is convenient for PU2 requiring relatively small quantities of by-products.
- `PU2_first` and $\bar{d}^2 \gg \bar{d}^1$: The case when PU2 has to meet greater demands than PU1 is favorable for PU1, since PU2 will tend to deplete the relatively small quantities of by-products generated by PU1.
- $\bar{d}^1 \approx \bar{d}^2$: When demands of production units are balanced, the primacy in decision-making has no drastic effects on costs.

Discussions on industrial symbiosis-based collaboration policies. As previously highlighted, the collaboration schemes applied to coordinate traditional supply chains may not be appropriate to regulate the exchange of by-products between a number of production units.

Apart from the attractiveness in terms of global economic benefits, the centralized collaboration policy suffers from the disadvantage of not being always equitable, as shown in the analysis of Table 4. One of the state-of-the-art remedies to deal with the misalignment of benefits is to explicitly add and operate with financial flows within the network. The centralized policy may be improved by sharing benefits between production units in the form of side payments given by one PU to compensate financial losses incurred by another PU (Daquin et al., 2019).

As far as the decentralized policies are considered, contract schemes serve to align the interests of each production unit, by rigorously framing the by-product transfer and avoiding relationships based on dominance. The global benefits induced by these contracts are situated between the values of solutions obtained by `PU1_First` and `PU2_First`, and deserves to be studied.

One of the most sensitive and crucial points in making successful an industrial symbiosis partnership is the information sharing. Nowadays, a growing number of IT platforms is implementing not only: **(i)** to facilitate the access to information about the by-product location and availability, but also **(ii)** to support the framing of collaboration schemes. Let us mention a couple of platforms dedicated to fostering the industrial symbiosis (Vladimirova et al., 2019): **(i)** SYNERGie 4.0 Platform and Database, promoted by International Synergies[3], **(ii)** MAESTRI Toolkit, EPOS Toolbox, Sharebox or SYNERGie 2.0 Platform, etc, developed in the framework of European projects (respectively MAESTRI[4], EPOS[5] and Sharebox[6] projects), and **(iii)** Industrial Symbiosis Data Repository Platform[7], an open source platform. It is worthwhile to underline the importance of the following questions posed by Fraccascia and Yazan (2018) in achieving the zero-waste goal via the industrial symbiosis: *"What is sensitive information for a company? Which type of information is non-sensitive for a company to implement industrial symbiosis based cooperation? Is the sensitive information really sensitive enough to motivate the limitation for its non-disclosure?"* In line with these issues and as future research, it would be very insightful to evaluate the value of information availability within the system of coordinates defined by the baseline collaboration policies investigated in this paper.

8. Conclusion and perspectives

Inspired by the circular economy paradigm, this paper introduced and investigated a new version of a two-level single-item lot-sizing problem (ULS-IS), posed by an industrial symbiosis network including two production units. We proposed two formulations to model the two-level single-item lot-sizing problem in the framework of a centralized coordination policy. We showed that ULS-IS problem is NP-Hard. A solution method based on Lagrangian decomposition has been proposed. Extensive numerical experiments have been conducted on small and large instances to study the competitiveness and the tractability of the proposed solution method and its variants.

From a managerial point of view, two sequential decentralized collaboration policies have been investigated against two extreme configurations, namely: no collaboration, and full collaboration based on a centralized decision-making. Valuable evidence for policy makers has been discussed, and a number of perspectives has been suggested for further research.

[3]International Synergies https://www.international-synergies.com/
[4]MAESTRI project. https://maestri-spire.eu/
[5]EPOS project. https:/www.spire2030.eu/epos
[6]Sharebox project. http://sharebox-project.eu/partners/
[7]Industrial Symbiosis Data Repository Platform. http://isdata.org

References

Aggarwal, A., Park, J. K., 1993. Improved algorithms for economic lot size problems. Operations research 41 (3), 549–571.

Ağralı, S., 2012. A dynamic uncapacitated lot-sizing problem with co-production. Optimization Letters 6 (6), 1051–1061.

Ahmed, S., He, Q., Li, S., Nemhauser, G. L., 2016. On the computational complexity of minimum-concave-cost flow in a two-dimensional grid. SIAM Journal on Optimization 26 (4), 2059–2079.

Arkin, E., Joneja, D., Roundy, R., 1989. Computational complexity of uncapacitated multi-echelon production planning problems. Operations research letters 8 (2), 61–66.

Bitran, G. R., Leong, T.-Y., 1992. Deterministic approximations to co-production problems with service constraints and random yields. Management Science 38 (5), 724–742.

Brahimi, N., Absi, N., Dauzère-Pérès, S., Kedad-Sidhoum, S., 2015. Models and lagrangian heuristics for a two-level lot-sizing problem with bounded inventory. OR spectrum 37 (4), 983–1006.

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.

Buschkühl, L., Sahling, F., Helber, S., Tempelmeier, H., 2010. Dynamic capacitated lot-sizing problems: a classification and review of solution approaches. Or Spectrum 32 (2), 231–261.

Daquin, C., Allaoui, H., Goncalves, G., Hsu, T., 2019. Collaborative lot-sizing problem for an industrial symbiosis. IFAC-PapersOnLine 52 (13), 1325–1330.

Díaz-Madroñero, M., Mula, J., Peidro, D., 2014. A review of discrete-time optimization models for tactical production planning. International Journal of Production Research 52 (17), 5171–5205.

Evans, S., Benedetti, M., Holgado Granados, M., 2017. Library of industrial symbiosis case studies and linked exchanges.

Federgruen, A., Tzur, M., 1991. A simple forward algorithm to solve general dynamic lot sizing models with n periods in O(n log n) or O(n) time. Management Science 37 (8), 909–925.

Fisher, M. L., 1981. The lagrangian relaxation method for solving integer programming problems. Management science 27 (1), 1–18.

Florian, M., Lenstra, J. K., Rinnooy Kan, A., 1980. Deterministic production planning: Algorithms and complexity. Management science 26 (7), 669–679.

Fraccascia, L., Yazan, D. M., 2018. The role of online information-sharing platforms on the performance of industrial symbiosis networks. Resources, conservation and recycling 136, 473–485.

Govindan, K., Soleimani, H., 2017. A review of reverse logistics and closed-loop supply chains: a journal of cleaner production focus. Journal of Cleaner Production 142, 371–384.

He, Q., Ahmed, S., Nemhauser, G. L., 2015. Minimum concave cost flow over a grid network. Mathematical Programming 150 (1), 79–98.

Hwang, H.-C., Ahn, H.-S., Kaminsky, P., 2016. Algorithms for the two-stage production-capacitated lot-sizing problem. Journal of Global Optimization 65 (4), 777–799.

Hwang, H.-C., Jung, H.-G., 2011. Warehouse capacitated lot-sizing for a two-stage supply chain.

Ilgin, M. A., Gupta, S. M., 2010. Environmentally conscious manufacturing and product recovery (ECMPRO): a review of the state of the art. Journal of environmental management 91 (3), 563–591.

Jaruphongsa, W., Cetinkaya, S., Lee, C.-Y., 2004. Warehouse space capacity and delivery time window considerations in dynamic lot-sizing for a simple supply chain. International Journal of Production Economics 92 (2), 169–180.

Krarup, J., Bilde, O., 1977. Plant location, set covering and economic lot size: An 0 (mn)-algorithm for structured problems. In: Numerische Methoden bei Optimierungsaufgaben Band 3. Springer, pp. 155–180.

Lombardi, D. R., Laybourn, P., 2012. Redefining industrial symbiosis: Crossing academic–practitioner boundaries. Journal of Industrial Ecology 16 (1), 28–37.

Lu, L., Qi, X., 2011. Dynamic lot sizing for multiple products with a new joint replenishment model. European Journal of Operational Research 212 (1), 74–80.

Melega, G. M., de Araujo, S. A., Jans, R., 2018. Classification and literature review of integrated lot-sizing and cutting stock problems. European Journal of Operational Research 271 (1), 1–19.

Melo, R. A., Wolsey, L. A., 2010. Uncapacitated two-level lot-sizing. Operations Research Letters 38 (4), 241–245.

Melo, R. A., Wolsey, L. A., 2012. Mip formulations and heuristics for two-level production-transportation problems. Computers & Operations Research 39 (11), 2776–2786.

Phouratsamay, S.-L., Kedad-Sidhoum, S., Pascual, F., 2018. Two-level lot-sizing with inventory bounds. Discrete Optimization 30, 1–19.

Quadt, D., Kuhn, H., 2008. Capacitated lot-sizing with extensions: a review. 4OR 6 (1), 61–83.

Rowshannahad, M., Absi, N., Dauzère-Pérès, S., Cassini, B., 2018. Multi-item bi-level supply chain planning with multiple remanufacturing of reusable by-products. International Journal of Production Economics 198, 25–37.

Solyalı, O., Süral, H., 2012. The one-warehouse multi-retailer problem: Reformulation, classification, and computational results. Annals of Operations Research 196 (1), 517–541.

Spengler, T., Püchert, H., Penkuhn, T., Rentz, O., 1997. Environmental integrated production and recycling management. European Journal of Operational Research 97 (2), 308–326.

Sridhar, S., Linderoth, J., Luedtke, J., 2014. Models and solution techniques for production planning problems with increasing byproducts. Journal of Global Optimization 59 (2-3), 597–631.

Suzanne, E., Absi, N., Borodin, V., 2020a. Towards circular economy in production planning: Challenges and opportunities. European Journal of Operational Research 287 (1), 168 – 190.

Suzanne, E., Absi, N., Borodin, V., van den Heuvel, W., 2020b. A single-item lot-sizing problem with a by-product and inventory capacities. European Journal of Operational Research 287 (3), 844 – 855.

Taşkın, Z. C., Ünal, A. T., 2009. Tactical level planning in float glass manufacturing with co-production, random yields and substitutable products. European Journal of Operational Research 199 (1), 252–261.

Trigeiro, W. W., Thomas, L. J., McClain, J. O., 1989. Capacitated lot sizing with setup times. Management science 35 (3), 353–366.

van Hoesel, S., Kuik, R., Salomon, M., van Wassenhove, L. N., 1994. The single-item discrete lotsizing and scheduling problem: optimization by linear and dynamic programming. Discrete Applied Mathematics 48 (3), 289–303.

Van Vyve, M., Wolsey, L. A., Yaman, H., 2014. Relaxations for two-level multi-item lot-sizing problems. Mathematical Programming 146 (1-2), 495–523.

Vladimirova, D., Miller, K., Evans, S., 2018. Lessons learnt and best practices for enhancing industrial symbiosis in the process industry. Scaler: Brussels, Belgium.

Vladimirova, D., Miller, K., Evans, S., 2019. Pathways to increase industrial symbiosis including tools and methods for stakeholders. Scaler: Brussels, Belgium.

Wagelmans, A., van Hoesel, S., Kolen, A., 1992. Economic lot sizing: an O(n log n) algorithm that runs in linear time in the wagner-whitin case. Operations Research 40 (1-supplement-1), S145–S156.

Wagner, H. M., Whitin, T. M., 1958. Dynamic version of the economic lot size model. Management science 5 (1), 89–96.

Zhang, M., Küçükyavuz, S., Yaman, H., 2012. A polyhedral study of multiechelon lot sizing with intermediate demands. Operations Research 60 (4), 918–935.

Appendix A. Computational results

Table A.5: AGG and FAL: *Average CPU time for small size instances*

Param values		Null B						Tight B						Large B					
	$SHR1$	3		4		5		3		4		5		3		4		5	
$SHR2$	Δ	AGG	FAL	AGG	FAL	AGG	FAL	AGG	FAL	AGG	FAL	AGG	FAL	AGG	FAL	AGG	FAL	AGG	FAL
3	L	0.29	0.28	0.65	0.84	0.76	0.98	0.32	2.65	0.54	6.27	0.68	8.02	0.25	2.09	0.49	7.72	0.57	5.22
3	M	0.30	0.27	0.63	0.79	0.75	1.04	0.33	2.41	0.53	6.27	0.69	9.18	0.25	2.11	0.47	6.64	0.62	5.60
3	H	0.33	0.26	0.62	0.78	0.76	1.05	0.34	2.47	0.52	5.49	0.67	9.60	0.25	2.13	0.47	6.46	0.60	5.28
4	L	0.38	0.88	0.50	0.23	0.50	0.48	0.49	7.13	0.50	2.70	0.68	3.66	0.39	5.86	0.50	2.53	0.64	4.20
4	M	0.39	0.88	0.47	0.22	0.51	0.50	0.48	7.37	0.52	2.62	0.68	4.09	0.42	5.81	0.48	2.51	0.64	4.27
4	H	0.40	0.97	0.49	0.20	0.51	0.59	0.50	7.25	0.54	2.56	0.66	4.43	0.42	6.13	0.47	2.44	0.61	4.47
5	L	0.47	1.14	0.50	0.59	0.55	0.27	0.55	16.18	0.62	7.17	0.64	2.67	0.48	9.71	0.54	5.83	0.59	2.61
5	M	0.48	1.03	0.46	0.57	0.55	0.27	0.59	13.87	0.60	6.75	0.63	2.70	0.51	8.62	0.57	5.55	0.59	2.49
5	H	0.45	1.12	0.53	0.41	0.52	0.28	0.58	12.41	0.63	6.55	0.63	2.71	0.49	7.33	0.53	5.21	0.63	2.55
	d^1	L		M		H		L		M		H		L		M		H	
d^2	Δ	AGG	FAL	AGG	FAL	AGG	FAL	AGG	FAL	AGG	FAL	AGG	FAL	AGG	FAL	AGG	FAL	AGG	FAL
L	L	0.59	0.84	0.47	0.45	0.45	0.37	0.64	8.34	0.56	4.94	0.44	2.35	0.57	7.94	0.42	2.23	0.42	2.00
L	M	0.58	0.79	0.48	0.47	0.46	0.38	0.65	8.23	0.53	5.62	0.41	2.42	0.60	7.59	0.45	2.48	0.39	2.05
L	H	0.59	0.73	0.53	0.57	0.44	0.43	0.63	7.92	0.53	5.93	0.43	2.45	0.58	7.08	0.45	2.56	0.41	2.06
M	L	0.51	0.68	0.59	0.83	0.50	0.41	0.57	6.20	0.63	8.77	0.54	4.67	0.53	4.59	0.57	9.09	0.41	2.30
M	M	0.47	0.65	0.56	0.81	0.48	0.48	0.56	5.70	0.66	8.48	0.51	5.30	0.53	4.42	0.62	8.21	0.41	2.50
M	H	0.52	0.65	0.55	0.75	0.51	0.58	0.58	5.21	0.68	8.28	0.55	5.66	0.49	4.16	0.60	8.11	0.48	2.57
H	L	0.45	0.60	0.47	0.72	0.59	0.80	0.51	5.63	0.54	7.01	0.62	8.53	0.45	4.25	0.52	4.81	0.57	8.58
H	M	0.45	0.55	0.51	0.63	0.56	0.79	0.51	5.30	0.56	6.31	0.65	7.90	0.48	4.01	0.49	4.36	0.59	7.96
H	H	0.45	0.53	0.50	0.67	0.51	0.72	0.48	4.85	0.52	5.51	0.65	7.66	0.43	3.87	0.48	4.04	0.55	7.54
Avg CPU time (s)		0.51	0.63					0.56	6.12					0.50	4.87				

L: Low, M: Medium, H: High, AGG: Straightforward formulation of ULS-IS, FAL: Facility Location formulation of ULS-IS.

Table A.6: Algorithm 1 and its variants: *Gaps (in %) to optimality for small size instances*

B	Variant	Mean	Standard deviation	Max	Median
Null	LD	0.12	0.27	1.74	0.00
	LD-MS	0.12	0.27	1.74	0.00
	LD-LS	0.09	0.21	1.68	0.00
	LD-MS-LS	0.08	0.21	1.68	0.00
	LD-LP	0.73	1.28	9.29	0.01
	LD-LP-MS-LS	0.31	0.61	5.86	0.00
Non-null	LD	0.37	0.55	3.40	0.12
	LD-MS	0.36	0.55	3.40	0.12
	LD-LS	0.16	0.30	2.05	0.01
	LD-MS-LS	0.16	0.30	2.05	0.01
	LD-LP	0.42	0.90	9.06	0.00
	LD-LP-MS-LS	0.20	0.43	7.20	0.00

Table A.7: LD-MS-LS: *Gaps (in %) for small size instances*

$SHR1$	$SHR2$	**Null B**			**Tight B**			**Large B**		
		UB-LB	UB-OPT	OPT-LB	UB-LB	UB-OPT	OPT-LB	UB-LB	UB-OPT	OPT-LB
3	3	0.16	0	0.15	0.31	0.03	0.28	0.4	0.02	0.37
3	4	0.5	0.06	0.44	0.71	0.1	0.61	0.72	0.11	0.61
3	5	1	0.16	0.84	1.23	0.24	1	1.08	0.3	0.79
4	3	0.43	0.1	0.33	0.81	0.26	0.55	0.96	0.15	0.82
4	4	0.11	0	0.1	0.28	0	0.28	0.55	0.01	0.53
4	5	0.25	0.08	0.17	0.46	0.1	0.37	0.58	0.16	0.42
5	3	0.7	0.22	0.49	0.97	0.35	0.63	0.97	0.3	0.67
5	4	0.2	0.03	0.17	0.28	0.06	0.22	0.46	0.08	0.38
5	5	0.11	0	0.11	0.29	0.01	0.28	0.54	0.04	0.51
d^1	d^2	UB-LB	UB-OPT	OPT-LB	UB-LB	UB-OPT	OPT-LB	UB-LB	UB-OPT	OPT-LB
L	L	0.63	0.08	0.55	0.89	0.16	0.73	0.85	0.19	0.66
L	M	0.35	0.11	0.25	0.46	0.14	0.33	0.38	0.15	0.23
L	H	0.22	0.05	0.17	0.31	0.09	0.22	0.2	0.07	0.13
M	L	0.25	0.05	0.2	0.59	0.1	0.49	1.15	0.07	1.07
M	M	0.64	0.09	0.55	0.92	0.19	0.74	0.86	0.19	0.67
M	H	0.36	0.11	0.25	0.47	0.14	0.33	0.38	0.14	0.23
H	L	0.16	0.03	0.13	0.22	0.07	0.16	0.33	0.07	0.25
H	M	0.24	0.05	0.19	0.6	0.11	0.49	1.24	0.08	1.16
H	H	0.62	0.09	0.53	0.88	0.17	0.72	0.87	0.19	0.68
Average gaps		0.38	0.07	0.31	0.59	0.13	0.46	0.69	0.13	0.56

L: Low, M: Medium, H: High, UB: Upper bound, LB: Lower bound, OPT: Optimality.

Table A.8: AGG and LD-MS-LS: *Gaps (in %) and CPU times for large size instances*

Parameters		**Null** B						**Tight** B						**Large** B					
		Gaps				CPU time		Gaps				CPU time		Gaps				CPU time (s)	
$SHR1$	$SHR2$	AGG_u	LD_u	AGG_l	LD_l	AGG	LD	AGG_u	LD_u	AGG_l	LD_l	AGG	LD	AGG_u	LD_u	AGG_l	LD_l	AGG	LD
3	3	0.00	0.03	0.00	0.18	1.95	1.61	0.00	0.04	0.00	0.22	0.65	8.49	0.00	0.03	0.00	0.22	0.68	8.91
3	4	0.00	0.31	0.05	0.10	8.94	1.57	0.00	0.26	0.00	0.41	8.32	8.03	0.00	0.18	0.00	0.3	6.12	8.50
3	5	0.00	0.73	0.18	0.07	9.07	1.50	0.00	0.56	0.02	0.34	9.82	8.11	0.00	0.36	0.02	0.26	7.95	8.36
4	3	0.00	0.38	0.09	0.06	9.09	1.60	0.00	0.32	0.00	0.57	6.68	7.83	0.00	0.33	0.00	0.43	5.53	8.50
4	4	0.01	0.06	0.31	0.03	9.88	1.44	0.00	0.09	0.00	0.33	4.09	7.58	0.00	0.13	0.00	0.23	5.30	8.45
4	5	0.00	0.30	0.65	0.01	10.02	1.40	0.00	0.23	0.04	0.16	9.75	7.77	0.00	0.22	0.04	0.10	8.28	7.95
5	3	0.00	0.61	0.22	0.05	10.01	1.49	0.00	0.47	0.01	0.60	8.05	7.56	0.00	0.36	0.01	0.47	7.16	7.67
5	4	0.00	0.37	0.70	0.00	10.02	1.42	0.00	0.28	0.06	0.27	8.91	7.34	0.00	0.28	0.05	0.21	8.63	7.66
5	5	0.00	0.05	0.64	0.00	10.01	1.29	0.00	0.04	0.02	0.25	8.09	7.07	0.00	0.08	0.02	0.19	7.59	7.86
d^1	d^2	AGG_u	LD_u	AGG_l	LD_l	AGG	LD	AGG_u	LD_u	AGG_l	LD_l	AGG	LD	AGG_u	LD_u	AGG_l	LD_l	AGG	LD
L	L	0.00	0.42	0.34	0.11	9.06	1.47	0.00	0.35	0.01	0.67	7.91	7.70	0.00	0.31	0.01	0.51	8.17	8.44
L	M	0.00	0.20	0.32	0.03	8.73	1.49	0.00	0.18	0.04	0.17	7.68	7.30	0.00	0.16	0.03	0.15	7.49	7.41
L	H	0.00	0.11	0.21	0.06	8.41	1.49	0.00	0.10	0.04	0.14	7.05	7.27	0.00	0.08	0.03	0.12	6.73	7.23
M	L	0.00	0.38	0.36	0.03	8.72	1.48	0.00	0.28	0.01	0.25	6.71	8.55	0.00	0.16	0.01	0.19	3.73	9.10
M	M	0.00	0.40	0.33	0.10	8.95	1.46	0.00	0.36	0.00	0.69	7.95	7.58	0.00	0.35	0.01	0.50	8.56	8.37
M	H	0.00	0.21	0.33	0.03	8.73	1.50	0.00	0.21	0.04	0.17	7.87	7.36	0.00	0.17	0.03	0.15	7.47	7.46
H	L	0.00	0.27	0.22	0.04	8.54	1.49	0.00	0.15	0.00	0.12	4.31	7.99	0.00	0.16	0.00	0.10	3.11	8.30
H	M	0.00	0.41	0.38	0.02	8.78	1.48	0.00	0.29	0.01	0.26	6.91	8.47	0.00	0.18	0.01	0.21	3.63	9.11
H	H	0.00	0.44	0.33	0.10	9.06	1.46	0.00	0.38	0.01	0.69	7.97	7.56	0.00	0.38	0.01	0.50	8.32	8.45

L: Low, M: Medium, H: High, AGG_l (AGG_u): Lower (upper) bound of the straightforward formulation of ULS-IS, LD_l (LD_u): Lower (upper) bound of LD-MS-LS.