CIRRELT-2020-18

Production and Distribution Optimization of Bio-Based Fertilizers

Guilherme O. Chagas
Leandro C. Coelho
Maryam Darvish
Jacques Renaud

June 2020

Document de travail également publié par la Faculté des sciences de l'administration de l'Université Laval, sous le numéro FSA-2020-006.

Bureau de Montréal
Université de Montréal
C.P. 6128, succ. Centre-Ville
Montréal (Québec) H3C 3J7
Tél. : 1-514-343-7575
Télécopie : 1-514-343-7121

Bureau de Québec
Université Laval,
2325, rue de la Terrasse
Pavillon Palais-Prince, local 2415
Québec (Québec) G1V 0A6
Tél. : 1-418-656-2073
Télécopie : 1-418-656-2624

Production and Distribution Optimization of Bio-Based Fertilizers

Guilherme O. Chagas[1], Leandro C. Coelho[1,2], Maryam Darvish[1,*], Jacques Renaud[1,3]

[1] Interuniversity Research Centre on Enterprise Networks, Logistics and Transportation (CIRRELT) and Department of Operations and Decision Systems, Université Laval, 2325 de la Terrasse, Québec, Canada G1V 0A6

[2] Canada Research Chair in Integrated Logistics, Université Laval, Québec, Canada

[3] Centre d'Innovation et Chaînes d'Approvisionnement Durable (CILCAD), Université Laval, Québec, Canada

Abstract. In this paper, we deal with the waste valorization that turns municipal waste into composts and fertilizers to be exploited in agriculture. This has many benefits as it decreases the amount of waste to be disposed of, reduces the sourcing of limited chemical compounds used in fertilizer production, and promotes a circular economy perspective, which is vital in big cities. We design the production and distribution schedule of several types of fertilizers. Through a detailed mathematical description and optimization, we model and simulate the operations of an industrial partner and determine how critical operational parameters affect the performance of the system. Through a series of computational experiments, we demonstrate how the management of operations in our industrial partner biorefinery can be significantly improved. In a broader sense, efficient operation of systems will, in turn, translate into environmental gains, significant contributions to waste valorization, reducing the need for chemical fertilizers, and increasing the awareness towards a circular economy perspective.

Keywords: Waste management, integrated production and distribution, fertilizer, circular economy, scheduling.

Acknowledgements. This project was partly funded by the Natural Sciences and Engineering Council of Canada (NSERC) under grants 2018-03712, 2019-00094, and 2020-0040. We also thank our industrial partner that shared with us all technical specifications as well as their expected production and delivery data. This support is greatly acknowledged.

---

* Corresponding author: maryam.darvish@FSA.ulaval.ca

Dépôt légal – Bibliothèque et Archives nationales du Québec
Bibliothèque et Archives Canada, 2020

1. Introduction

Increasing environmental awareness, especially concerning future resource availability, has highlighted the importance of reusing, reducing, and recycling waste materials. Moreover, with the alarming production rate of solid waste in urban areas, waste management is a challenge in many cities. Therefore, many countries around the world are now promoting the principles of the circular economy. A popular waste valorization method is to turn waste into composts and fertilizers to be exploited for plant growth. Organic fertilizers and biofuel can be generated from the biomass (Awudu and Zhang, 2012) within a typical value chain consisting of several echelons: biomass fields, biorefineries, and consumption markets (Azadeh et al., 2014).

In a circular economy context, waste from a process is fed back to the economy as the raw material for another process (Van Eygen et al., 2018). The closed loop for fertilizers is already promoted in several countries (Trochu et al., 2019; Chojnacka et al., 2020) which results in restoring the land, increasing soil fertility, and thus increased harvests. The three main nutrients used in chemical fertilizer production are nitrogen, phosphorus, and potassium. Scarcity of phosphates aside, as these are non-renewable sources, production, and use of chemical fertilizers contribute to the emission of greenhouse gases (Weigand et al., 2013; Franz, 2008). Therefore, alternative sources of nutrients (sewage, animal, and food wastes) have become increasingly popular as a sustainable replacement for chemical fertilizers. The bio-fertilization requires technical innovation and changes in the legislative framework. However, besides its environmental benefits, it has economic advantages as many countries depend on phosphate imports. In Canada, fertilizer production is one of the most energy and cost intensive industries (Aspire, 2019). Since phosphorus reserves are very limited in Canada, the country depends highly on imports (Gross, 2017). Moreover, as the disposal landfills are becoming full, the country has embraced the idea of the biological treatment of waste (Assamoi and Lawryshyn, 2012). In Quebec, the provincial government has enacted a ban on incineration and disposal of organic waste by 2020 (IISD, 2018), which has become the main driver for waste valorization. Focusing on the organic fertilizer value chain, in this paper we address a fertilizer production and distribution optimization problem observed in Quebec City, Canada.

This paper is motivated by a collaboration with an industrial partner currently building a plant to transform sewage sludge and food residues into fertilizers. Our partner opted for technology of direct trailer loading, which is very fast and flexible but does not allow any temporary storage. Therefore, continuous production is required. Orders from these farms for different products are received months in advance. By using overhead conveyors, products are transferred to loading docks, and full trailers are then sent to local farms, at no cost. Carrying no inventory highlights the importance of a timely and synchronized production and distribution plan as the main operational challenge faced by the company. In this paper, we aim to (i) plan the loading trailers and (ii) plan the delivery schedule to the farms. We call this problem the *fertilizer production and distribution scheduling problem* (FPDSP).

Although the real-case problem we solve in this paper presents similarities to other problems from the literature, to the best of our knowledge, its many specific features have not yet been studied. This paper gives rise to a new integrated production and distribution scheduling problem. Hence, our contributions are manifold. We describe the FPDSP as motivated by real-life, and present a mathematical formulation and several valid inequalities for it. We assess the performance of our model on very large instances based on the future operations of our partner and provide important insights on the operation, planning, and performance of the system under different circumstances.

The remainder of this paper is organized as follows. In Section 2, we review relevant literature. In Section 3, we describe the production and distribution system at hand as well as its cost structure. We present a mathematical formulation for the FPDSP in Section 4. We evaluate the performance of our proposed model on real life data and provide economical and managerial insights in Section 5. Our conclusions follow in Section 6.

2. Literature review

Optimization of the biomass supply chain has recently gained considerable attention (Mantzaras and Voudrias, 2017). Among all the elements of a waste management system, waste collection optimization is the problem broadly studied in the literature (e.g., Das and Bhattacharyya (2015); Shah et al. (2018)) but the work on production and transportation of refined products is still scarce. In this section, we provide an overview of the relevant works in waste management, then look at the problem from an operations research perspective, and briefly discuss similar problems from the literature.

Two studies from the waste management optimization area are relevant to the problem at hand. Quddus et al. (2018) study biofuel production and present a two-stage, chance-constrained model capable of dealing with uncertainty due to feedstock seasonality. Rodias et al. (2019) model the distribution of organic fertilizer (such as liquid manure) by considering several agronomical, legislation, and other specific constraints. They generate several scenarios for the number of available tractors and compare the gains with the base scenario.

A review of relevant operations research literature on solid waste management is presented in Ghiani et al. (2014). From an optimization perspective, this paper deals with an integrated scheduling and distribution problem. However, it is difficult to classify our problem within one specific class of optimization problems as it shares similarities with some classes but also presents its unique features. For instance, it shares some similarities with the truck and trailer routing problem (Derigs et al., 2013) in which full trailers need to be loaded on the truck, but we do not consider any routing to be taken place, since each customer requests a full trailer. Hence the delivery aspect consists of round trips from the production plant to the customers. The problem is similar to the container loading problem (Vélez-Gallego et al., 2020; Kurpel et al., 2020), since we consider trailers with a given capacity to be loaded on docks with different loading rates, but given the contentious production, the material can take any shape to fill the trailer.

In the FPDSP, trailers need to be assigned to the docks. Depending on the product being loaded, trailers spend different amount of time on each dock. This problem is then similar to the truck-dock assignment (see Gelareh et al. (2016)) observed in cross-dock scheduling, except that here we only have the loading operations on trailers and no unloading docks are considered.

Finally, the FPDSP shares some features with the problem studied by Berghman and Leus (2015). They consider a dock scheduling problem for incoming and outgoing trailers with the objective of minimizing the number of late outgoing trailers as well as their tardiness. They propose several methods to solve large size instances of the problem. In our case, there is no buffer zone for the trailers since they must be loaded on a truck immediately after being charged. Moreover, we consider the loading and transportation of the trailers while in Berghman and Leus (2015) only dock scheduling is studied.

3. Problem description

In this section, we present the main components of the FPDSP. First, we present the characteristics of the production system in Section 3.1, followed by the distribution logistics in Section 3.2. The cost structure and objective are described in Section 3.3.

3.1. Production system

The facility is designed to produce two grades of fertilizer, which come from two different sources. First, sewage sludge is transformed by a *reactor* at an expected rate between 5 to 8 tons per hour. This speed depends on the months of the year since the production is affected by the temperature. The second reactor uses food residues and produces at a slower rate, between 0.5 to 2 tons per hour. Both reactors produce continuously and nonstop, 24/7. At steady state, the plant is expected to produce around 76,000 tons of fertilizer each year.

Using two overhead conveyors, produced fertilizers are transferred from the reactors to a distribution center, which has four loading docks, as depicted in Figure 1. Both conveyors are equipped with scales and sensors to load the trailers and control the weight uniformly. Initially, the first three docks will be assigned to the fertilizer produced from the sewage sludge and fed by conveyor 1. The fourth dock is dedicated to the food residue fertilizer and is fed by the second conveyor. This assignment cannot be modified on a daily basis, as both fertilizers have slightly different chemical properties. However, it can be modified if the long term production rates change. Trailers are already placed on the docks and are filled from above. For the sewage sludge fertilizer, the conveyor can only fill one trailer at a time. When a trailer is full, it automatically moves to the next available dock.

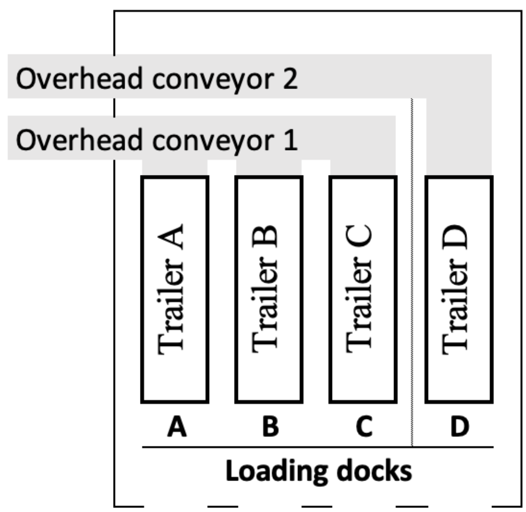

Figure 1: Schematic of the loading docks and their operation

3.2. Distribution system

The facility has long term agreements with agricultural distributors, which identify local farms interested in the fertilizers. Even though the products are requested months in advance, the distribution schedule is made only for a few days at a time, typically one week before the distribution. This enables farms to plan their reception activities. The orders received from farms are converted into full trailer loads. These trailers are then transported to the farms by a fleet of trucks.

Security rules related to maneuvers in the farms' fields suggest that the trailers should be moved during daylight hours. These transportation operations are subcontracted to a common carrier that uses capacitated trailers. The type of trailer used has a capacity of 32 tons, except in the thaw period, when the capacity is reduced to 27 tons. Because larger trailers are more prone to hazardous maneuvers, they cannot be used in this context. Under the contractual agreement, the carrier should always have four trailers at the docks and guarantee that truck drivers are available on call. The agreement also stipulates that the carrier always disposes of two trucks for the trailer delivery.

3.3. Cost structure and formal problem description

The operations of the facility are set up such that some orders *can* be delivered in the current planning horizon, but some may need to be postponed. Hence, the number of orders can be more than the actual delivery capacity of the horizon. Imposed by law, trailer delivery must be conducted during daytime hours, and for every hour that the violation continues, a penalty incurs. The objective of the FPDSP is to select orders to be produced on each dock, and schedule their loading and distribution in order to minimize the overnight delivery penalties.

The delivery time λ_i for each order i includes maneuvers at the facility, travel time to the farm, unloading time, and the time it takes for the truck to return. As mentioned earlier, for security reasons, deliveries should be completed before the sunset, therefore, trucks may depart from time h^- in the morning but all the orders should be delivered before h^+. As a result, it is preferable that any order i leaves the facility before $\left(h^+ - \frac{\lambda_i}{2}\right)$. Violation of this rule incurs a penalty.

To better explain how these penalties are calculated, in Figure 2, we provide an example for an order with $\lambda_i = 4$h, where $h^- = 7{:}00$ and $h^+ = 21{:}00$. Given a time discretization in intervals of one hour, let l be the truck's departure time and δ the hourly penalty fee. Starting from midnight until the last interval, the following four cases may arise for the penalty to be paid:

- Case 1 : $l \in [0{:}00; (h^- - \lambda_i)]$, the penalty is $\lambda_i \delta$
- Case 2 : $l \in [(h^- - \lambda_i + 1); (h^- - 1)]$, the penalty is $(h^- - l)\delta$
- Case 3 : $l \in [h^-; (h^+ - \frac{\lambda_i}{2})]$, the penalty is 0
- Case 4 : $l \in [(h^+ - \frac{\lambda_i}{2} + 1); 23{:}00]$, the penalty is $min(l + \frac{\lambda_i}{2} - h^+; \frac{\lambda_i}{2})\delta$

In Case 1, all trips are entirely performed between midnight and h^-. Therefore, since all the four hours of the trip is performed during the forbidden interval, the company is charged δ per hour. In Case 2 the trip begins before h^- but finishes after h^-. In this case only the part before h^- has to be penalized. For example, if the truck leaves at 5:00, then the company is penalized only for the two hours spent on the road during the forbidden interval. In Case 3, trips start after h^- and at least half of the route is performed before h^+, therefore no penalty is due, as the delivery has already taken place. Obviously, for the time trucks return to the depot with no trailers, no penalty is charged. In Case 4, trips begin after h^+ but are done before midnight, in this situation, as in Case 2, only the part of the trip conducted during the forbidden period is penalized.

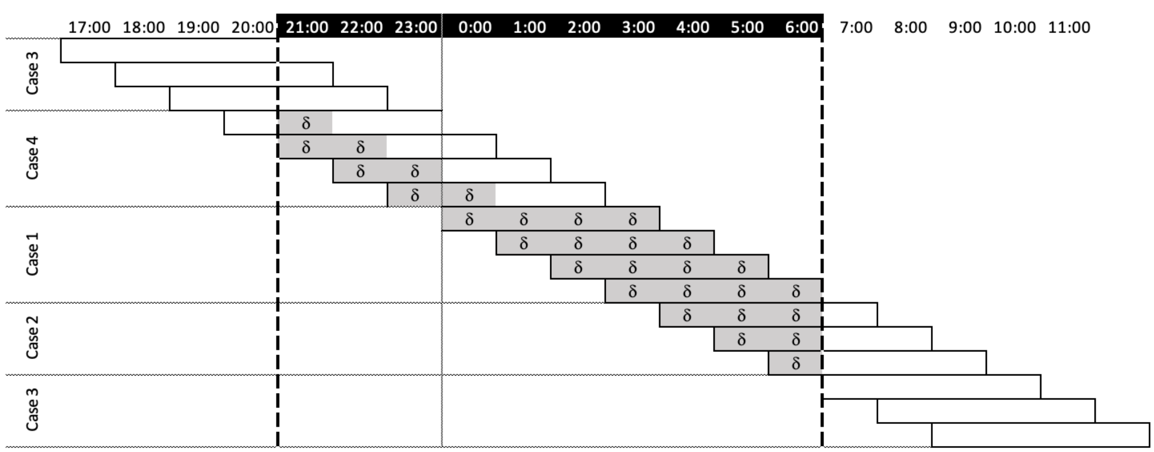

Figure 2: The cost structure

In order to accomplish these operations, several constraints must be considered. First, a trailer should always be loaded from conveyor 1, and another one simultaneously from conveyor 2. For practical reasons, all the orders of a given farm must be loaded consecutively. Once full, a trailer must be placed on a truck and depart for delivery. As soon as a trailer leaves, another one can be placed on the dock. The number of deliveries at any point in time cannot be higher than the number of available trucks.

4. Mathematical formulation

In this section, we first present a mathematical model to solve the FPDSP. We then exploit some problem characteristics to propose several valid inequalities and variable reduction procedures. We provide a numerical example at the end of this section to better illustrate the problem and its formulation.

4.1. A discrete time model

To solve this problem, we propose a discrete time model (DTM) which divides the planning horizon into equal intervals. Since all orders are converted to fully loaded trailers, for which the loading time is constant, and since the loading time is a multiple of the discretization interval, the use of a DTM is valid. Discussions with our industrial partner also confirm that such an approximation, for example intervals of one hour, is very acceptable. Shorter intervals can also be used at the expense of a larger model. Therefore a planning horizon H can be divided into J intervals of equal length.

A number of V trucks are available to transport trailers to the farms. Let us consider a set F of farms. To each farm f is associated an order set S_f, where each order i in this set corresponds to a single trailer-load product. Thus, we have a total of $n = \sum_{f=1}^{F} |S_f|$ orders. The total delivery time λ_i for each order $i \in S_f$ to be delivered to farm f includes is know and all deliveries should be made within time h^- in the morning and before h^+ at night. The total penalty for delivering order i when the truck leaves the facility in time interval l is denoted as c_{il}. This is calculated by using the sum of δ and considering the cost structure presented previously in Section 3.3.

P products are produced on m docks. Each order contains a trailer-load of one product, denoted by p_i, the product of order i. K_p is the set of orders related to product p. Not all docks, however, are capable of producing every product, Q_p is the set of docks that can load product p and also O_k set of orders that can be delivered from dock k. In this sense, Q_{p_i} is the set of docks that can treat product p of order i. Knowing the capacity of the identical trailers used by the company and also given the production rate of each product on each dock, we calculate the required time to fill a trailer with product p, θ_p.

We make use of the following binary decision variables. Variables z_f take value of 1 if and only if the order set from farm, i.e., f, S_f, is selected to be produced. In this case, variables x_{ikj} assign orders to docks and time intervals, taking value of 1 if and only if order i is produced on dock k starting at time j. Once an order is produced, variables y_{ikl} schedule its delivery. Variables y_{ikl} take value of 1 if and only if order i, produced on dock k, departs for delivery during interval l. Note that since trailers must be loaded before departure, variables y_{ikl} do not exists for $l \leq \theta_{p_i}$. Moreover, for all i and $k \in Q_{p_i}$, variable y_{ikl} are only generated for $l \leq J + \max_i(\theta_{p_i})$. Table 1 summarizes the notation used in the DTM.

Table 1: Notation used in the DTM

Parameters	
n	number of orders
m	number of docks
P	number of products
F	number of farms
V	number of trucks
J	number of intervals
c_{il}	total penalty paid for delivering order i when the truck leaves the facility in time interval l
p_i	product of order i
θ_p	time to fill a trailer with product p
λ_i	round trip travel time of order i
h^-, h^+	regular delivery time windows
δ	penalty cost per hour for deliveries during the forbidden hours
Sets	
$\mathcal{P}$	set of products
Q_p	set of docks that can load product p
O_k	set of orders that can be delivered from dock k
K_p	set of orders related to product p
S_f	set of orders to be delivered to farm f ($\sum_{f=1}^{F} \lvert S_f \rvert = n$)
Variables	
z_f	1 if order set S_f, $f = 1, \ldots, F$, is selected to be produced
x_{ikj}	1 if order i is produced on dock k during an interval starting in j
y_{ikl}	1 if order i produced on dock k departs for delivery in an interval starting in l

The DTM formulation is presented next.

$$\text{Minimize} \sum_{i=1}^{n} \sum_{l=\theta_{p_i}}^{J+\max\limits_{p\in\mathcal{P}}(\theta_p)} \sum_{k=1}^{m} c_{il}\, y_{ikl} \tag{1}$$

subject to

$$\sum_{i\in S_f}\sum_{k\in Q_{p_i}}\sum_{j=1}^{J} x_{ikj} = z_f \left|S_f\right| \quad f = 1,\ldots,F \tag{2}$$

$$\sum_{i\in K_p}\sum_{k\in Q_p}\sum_{l=\max(0,j-\theta_p+1)}^{j} x_{ikl} = 1 \quad j = 1,\ldots,J; \quad \forall p\in\mathcal{P} \tag{3}$$

$$\sum_{i\in S_f}\sum_{k\in Q_{p_i}}\sum_{j=1}^{J+\theta_{p_i}} y_{ikj} \le z_f \left|S_f\right| \quad f = 1,\ldots,F \tag{4}$$

$$x_{ikj} + \sum_{l=1}^{j+\theta_{p_i}-1} y_{ikl} \le 1 \quad i = 1,\ldots,n; \quad \forall k\in Q_{p_i}; \quad j = 1,\ldots,J \tag{5}$$

$$x_{ikj} \le \sum_{l=j+\theta_{p_i}}^{J+\theta_{p_i}} y_{ikl} \quad i = 1,\ldots,n; \quad \forall k\in Q_{p_i}; \quad j = 1,\ldots,J; \tag{6}$$

$$\sum_{i=1}^{n}\sum_{k\in Q_p}\sum_{l=\max(1,j-\lambda_i+1)}^{j} y_{ikl} \le V \quad j = 1,\ldots,J+\max_{p\in\mathcal{P}}(\theta_p) \tag{7}$$

$$\sum_{i\in O_k}\sum_{l=\max(1,j-\lambda_i+1)}^{j} y_{ikl} \le 1 \quad j = 1,\ldots,J+\max_{p\in\mathcal{P}}(\theta_p); \quad k = 1,\ldots,m \tag{8}$$

$$x_{\bar{i}kl} \le 1 - \left(\sum_{i\in O_k}\sum_{j=1}^{l-1} x_{ikj} - \sum_{i\in O_k}\sum_{j=1}^{l} y_{ikj}\right) \quad l = 2,\ldots,J; \quad k = 1,\ldots,m; \quad \forall \bar{i}\in O_k \tag{9}$$

$$\begin{gathered}\sum_{i\in S_f\setminus\{i_1\}}\sum_{\bar{k}\in Q_{p_i}} x_{i\bar{k}(l+(i-1)\theta_{p_{i_1}})} \ge (\left|S_f\right| - 1) - M(1 - x_{i_1kl}) \quad f = 1,\ldots,F; \\ i_1\in S_f; \forall k\in Q_{p_{i_1}}; l = 1,\ldots,J-((\left|S_f\right|-1)\theta_{p_{i_1}})\end{gathered} \tag{10}$$

$$\sum_{k\in Q_{p_{i_1}}}\sum_{j=1}^{J-(\left|S_f\right|-1)\theta_{p_{i_1}}} x_{i_1kj} = z_f \quad f = 1,\ldots,F; \quad i_1\in S_f \tag{11}$$

$$z_i; x_{ikl}; y_{ikj} \in \{0,1\} \quad i = 1,\ldots n; k = 1,\ldots,m; l = 1,\ldots,J; j = \theta_{p_i},\ldots,J+\max_{p\in\mathcal{P}}(\theta_p). \tag{12}$$

Objective function (1) minimizes the penalty incurred for deliveries within the forbidden delivery interval. Constraints (2) impose that if farm f is selected to be served ($z_f = 1$), then all its orders $i \in S_f$ must be assigned to a production/loading interval j and be loaded on compatible docks of Q_{p_i}. Constraints (3) impose continuous production, i.e., a trailer must be loaded on each set of docks all the time. Constraints (4) link the production of order i (from S_f) to both a delivery interval and a dock. The inequality sign here allows some orders to be produced but not delivered within the planning horizon. Constraints (5) impose that if loading of order i on dock k has started in interval j, then it cannot be delivered until the trailer is fully loaded, i.e., on any period $l < j + \theta_{p_i} - 1$. Constraints (6) indicate that if order i has started its loading on dock k in interval j, i.e., $x_{ikj} = 1$, then it should be delivered from the same dock k on any interval $l \geq j + \theta_{p_i}$ (which implies that $\sum_{l=j+\theta_{p_i}}^{J} y_{ikl} = 1$, due to constraints (3)). Similarly, if an order is not loaded on a dock k, i.e., $x_{ikj} = 0$, it cannot be delivered from that dock (which implies $\sum_{l=j+\theta_{p_i}}^{J} y_{ikl} = 0$). For each time interval, the maximum number of deliveries is limited by the number of trucks as indicated in constraints (7). Constraints (8) impose that only one vehicle at a time can depart from each dock. Constraints (9) manage the start of a new loading on dock k in interval l, as one can start loading an order only if all previously produced orders on this dock have already been departed for delivery. Constraints (10) ensure that, for each farm f, all orders in S_f are produced (loaded) consecutively. Here M is a large number and i_1 is the first order of S_f. These should be used in combination with constraints (11) to force the first order of set S_f to be produced before $J - \left(\left(\left|S_f\right| - 1\right) \theta_{p_{i_1}}\right)$, if this farm is selected to be served, as determined by variable z_f. All decision variables are binaries as in constraints (12).

Model (1)–(12) defines the DTM which is valid for solving the FPDSP.

4.2. Valid inequalities and variable reduction

The DTM can become too large to be used. Therefore, in this section we propose efficient valid inequalities and techniques to reduce the number of variables.

4.2.1. Valid inequality

DTM presents several symmetries with respect to the assignment of orders to docks. Suppose that we have three orders to be delivered to farm f as $S_f = \{1, 2, 3\}$ and that these orders can be loaded on three docks, namely a, b, c. Constraints (10) and (11) already impose a loading sequence for these orders and a dock sequence, meaning orders assigned to each dock. Since loading time is the same on all docks, dock assignments of [1, 2, 3], [1, 3, 2], [3, 1, 2], [2, 1, 3], [2, 3, 1] and [3, 2, 1] are all equivalents, as they all lead to the same optimal solution. This symmetry can be avoided by using constraints (13).

$$\sum_{i \in K_p} x_{ik(\theta_p \cdot (l-1)+1)} = \begin{cases} 1, & \text{if } k = ((l-1) \bmod |Q_p|) + k_1 \\ 0, & \text{otherwise} \end{cases} \qquad \forall p \in \mathcal{P};\ l = 1, \dots, \left\lfloor \frac{J}{\theta_p} \right\rfloor;\ \forall k \in Q_p \tag{13}$$

where k_1 is the first dock index that can load product p. Note that these constraints impose that only 1 out of a set of x variables can be selected.

4.2.2. Reduction of x variables

When we take into account the characteristics of the problem, we can eliminate some x_{ikj} variables that are never used in any optimal or even feasible solutions. As demonstrated before by constraints (13), we can order the docks producing the same product.

To make it clearer, we provide an example of a product that can be produced on two docks. We know that the same product cannot be loaded simultaneously on two parallel docks. Thus, in the first interval, only the first dock for Q_{p_i} can be active (and has its $x_{ikj} = 1$) while all other docks dedicated to the same product remain inactive from interval 1 until $(1 + \theta_p - 1)$. Hence, the respective variables can be removed from the model. Then, the loading continues on the second dock, starting at $\theta_p + 1$ and ending at $2\theta_p$. Once the trailer leaves the last dock, the conveyor returns to the first dock at $2\theta_p + 1$.

This reduction can be applied to each set of docks producing product p, since they all have the same loading time θ_p. Figure 3 provides an example on how loading activities alternate between three docks and a production time of 3 units. In this example, the only required variables are $x_{i,1,1}$, $x_{i,2,4}$, $x_{i,3,7}$, $x_{i,1,10}$, $x_{i,2,13}$, and $x_{i,3,16}$.

4.2.3. Reduction of y variables

Following the reduction of loading variables x_{ikj}, a similar procedure can be applied to delivery variables y_{ikl}. However, this is more intricate as we do not know the exact time a trailer leaves the dock since waiting on a dock for a truck to become available is allowed. We know that the delivery time is bounded between the end of the loading time for a trailer and the loading start time for the next one on the same dock. Thus in Figure 3, if order i starts its loading on dock 1 in period 1, it must leave the dock at any interval between 4 to 10. Therefore, all other y_{ikj} variables not respecting this logic can be removed.

This procedure can be formally described as follows. Consider an order of product p and its loading time θ_p, dock set Q_p, and order set K_p. In addition, let $d = 1, \ldots, |Q_p|$ be the indices of each dock. Then, for each dock $k_d \in Q_p$ and $\forall i \in K_p$, the variables y_{ik_dj} do not exist when $j = 1, \ldots, d\theta_p$. Furthermore, for any $j > d\theta_p$ such that $(j - ((d-1)\theta_p) - 1) \mod (|Q_p|\theta_p) = 0$ and $|Q_p| > 1$, the variables y_{ik_dl}, with $l \in [j+1, j+\theta_p - 1]$, are also not defined. Algorithms 1 and 2 present the pseudocode for the generation process of the y variables.

For example, consider variables y_{i1l} and $l \in [j, j+3-1]$, from Figure 3. These variables are not defined when $j = 10$ since $(10 - (0 \times 3) - 1) \mod 9 = 0$. Regarding dock 2, all variables y_{i2l} with $l \in [13, 13+3-1]$ do not exist as $(13 - (1 \times 3) - 1) \mod 9 = 0$. In this example with one product, three docks, 16 periods, and with loading time of three periods, the only y variables generated are y_{i1j} for $j \in [4, 10] \wedge j \in [13, 16]$, y_{i2j} for $j \in [7, 13] \wedge j \in [16, 16]$, and y_{i3j} for $j \in [11, 16]$.

Algorithm 1: y variables generation

input : number of product types P, number of intervals J, docks sets Q_p, orders sets K_p, loading time θ_p.
output: valid y_{ikj} variables.

```
begin
    for p = 1, ..., P do                              // for each product type
        for d = 1, ..., |Q_p| do                      // for each dock related to p
            for j = 1, ..., J + θ_p do
                if should_define_y(p, d, j) then      // Algorithm 2
                    /* create variables y_{ik_d j}, ∀i ∈ K_p and k_d ∈ Q_p     */
                end
            end
        end
    end
    return all variables y_{ikj} created;
end
```

Algorithm 2: should_define_y

input : product p, dock index d, interval j.
output: $true$ if variable must y_{ikj} be defined, $false$ otherwise.

```
begin
    m_j ← (j − ((d − 1)θ_p) − 1) mod (|Q_p|θ_p);
    if (j ≤ dθ_p) or (j > (d − 1)θ_p + |Q_p|θ_p and 1 ≤ m_j ≤ θ_p − 1 and |Q_p| > 1) then
        return false;
    end
    return true;
end
```

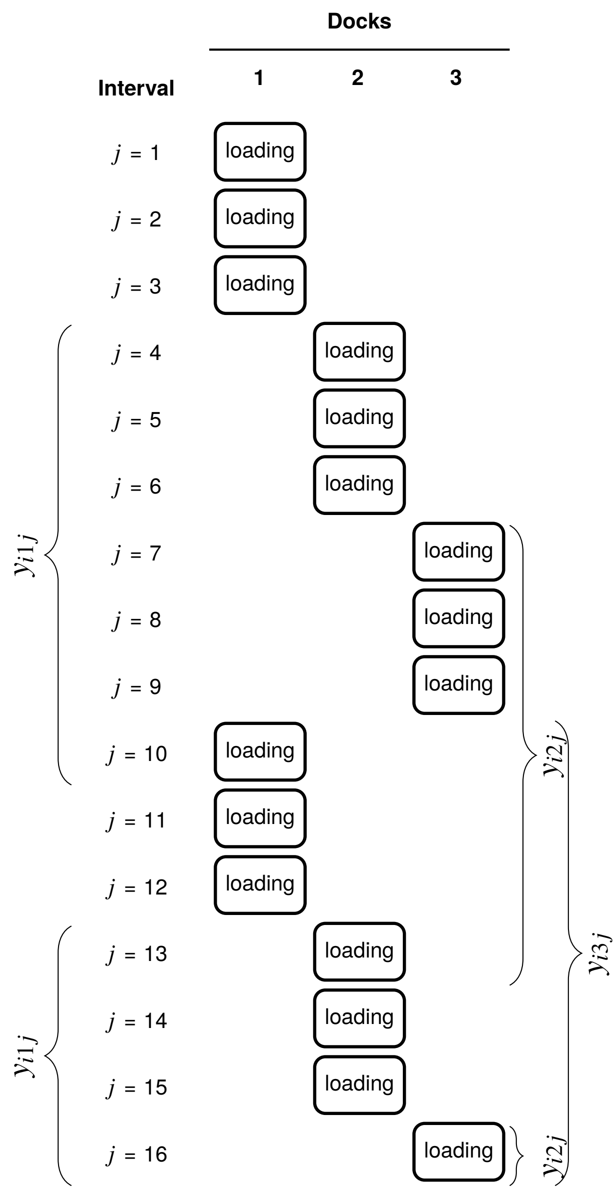

Figure 3: An example of valid y_{ikj} variables. Note that in this example $P = 1$, $|Q_1| = 3$, and $\theta_1 = 3$.

4.3. Detailed example

Consider the following example with two trucks, six trailers, four docks, and two products. Docks 1, 2 and 3 produce product 1, while product 2 is loaded from dock 4. There are 12 orders from eight farms. Regular transportation hours are between 7:00 and 20:59. Table 2 shows a list of *Orders* from *Farms* requesting either of the two *Products*. We also present the *Loading times* required for products and the *Delivery time* of each order, as well as the *Loading docks* to which orders can be assigned to. Note that the loading and delivery time in this table are the number of intervals required to perform the operation. The orders must be treated within a planning horizon of 30 intervals of one hour.

Take order 4, for example, and we define variables x_{4kl} and y_{4kl} for $k = 1, 2, 3$, and for $l = 1, 8, \ldots, 34$. The transportation time of order 4 is four hours, so if an order departs for delivery at 19:00, half of the trip will be performed before 21:00. Therefore, its penalty cost is $c_{4l} = 0$ if $l = 1, \ldots, 13$. Then, $c_{4,14} = \theta, c_{4,15} = c_{4,16} = c_{4,17} = 2\theta$. After midnight, all driving times are penalized until 7:00. Thus $c_{4,18} = c_{4,19} = c_{4,20} = c_{4,21} = 4\theta, c_{4,22} = 3\theta, c_{4,23} = 2\theta, c_{4,24} = \theta$ and $c_{4,25} = 0$ (as $l = 18$ refers to midnight and $l = 25$ corresponds to 7:00).

Table 2: Example for the discrete time model

Order	Farm	Product	Loading time	Delivery time	Loading docks
1	1	1	4	3	1, 2, 3
2	1	1	4	3	1, 2, 3
3	2	1	4	4	1, 2, 3
4	2	1	4	4	1, 2, 3
5	3	1	4	2	1, 2, 3
6	3	1	4	2	1, 2, 3
7	4	2	8	3	4
8	5	2	8	3	4
9	5	2	8	4	4
10	6	2	8	2	4
11	7	1	4	3	1, 2, 3
12	8	1	4	3	1, 2, 3

Figure 4 presents a solution to our numerical example where the letters "L" and "D" indicate loading and delivery, respectively. At 7:00 (period $l = 1$) orders 1 and 7 are on loading docks (thus $x_{111} = x_{741} = 1$). Loading of order 1 ends at 10:59 and its transportation takes place from

11:00 to 13:59 ($y_{1,1,5} = 1$). Immediately at the departure of order 1, order 2 begins its loading process on dock 2. Orders 1 and 2 are loaded consecutively as they are both related to farm 1. At 15:00 (period $l = 9$), orders 2 and 7 finish loading and are both in transportation (D2 and D7). Transportation of order 3 starts at 19:00 and finishes at 22:59 but as half of the travel time takes place before 21:00. The order is delivered before 21:00 and the truck drives back empty (with no trailer loaded) to the facility after 21:00. Since the production on dock 4 cannot be stopped, order 10 is transported at night from 23:00 to 0:59 (D10 is shown in bold characters because of the penalty incurred). Order 10 is served at night as its transportation time is less than that of orders 7 and 8, hence incurring a lower penalty. Order 4 can be processed from 19:00 to 22:59 and it is feasible to postpone its delivery until 7:00 (period $l = 25$) to avoid the penalty. From 23:00 to 6:59, the loading of orders 5 and 6 are completed. At 7:00, order 4 is delivered which allows the loading of order 11 on dock 1. The solution provided for this production-distribution scheduling example requires a penalty for two hours of a trip with a loaded trailer.

5. Computational results

We now present the numerical experiments designed to evaluate the effectiveness of the proposed model and the solution procedure for the FPDSP. We first introduce the problem parameters which are based on the industrial settings of our partner and then we present different-size instances generated as a test bed to assess the quality of solutions obtained by the model, the performance of the valid inequalities, and of the variable reduction procedure.

All implementations are in C++ and compiled with *g*++ compiler version 10.1 using -O3 flag. The model is implemented in Gurobi™ mathematical programming solver, version 9.0.2. All computational tests are executed on a computer with Intel® Core™ i9-9900K CPU 3.60GHz × 16 processor with 16MiB *cache* memory and 126GiB of RAM. The operating system installed on this machine is Ubuntu 18.04.4 64 bits. In addition, the execution time for each instance is limited to 6 hours.

Interval	Time	Dock 1	Dock 2	Dock 3	Dock 4
1	7:00	L1			L7
2	8:00	L1			L7
3	9:00	L1			L7
4	10:00	L1			L7
5	11:00	D1	L2		L7
6	12:00	D1	L2		L7
7	13:00	D1	L2		L7
8	14:00		L2		L7
9	15:00		D2	L3	L10+D7
10	16:00		D2	L3	L10+D7
11	17:00		D2	L3	L10+D7
12	18:00			L3	L10
13	19:00	L4		D3	L10
14	20:00	L4		D3	L10
15	21:00	L4		D3	L10
16	22:00	L4		D3	L10
17	23:00	...	L5		L8+**D10**
18	0:00	...	L5		L8+**D10**
19	1:00	...	L5		L8
20	2:00	...	L5		L8
21	3:00	...	...	L6	L8
22	4:00	...	...	L6	L8
23	5:00	...	...	L6	L8
24	6:00	...	...	L6	L8
25	7:00	L11+D4	...	...	L9+D8
26	8:00	L11+D4	...	...	L9+D8
27	9:00	L11+D4	...	...	L9+D8
28	10:00	L11+D4	...	D6	L9
29	11:00	...	L12+D5	D6	L9
30	12:00	...	L12+D5		L9
	...				

Figure 4: Solution of the example
L for loading and D for delivery

5.1. Instance generation

Here, we present two sets of instances. The first set, which we call *industrial*, is generated using the real data from our industrial partner. The second set is generated randomly and its main purpose is to evaluate to which extent we are able to obtain high quality solutions.

We present the parameters used to generate these instance sets in Table 3. Quebec city is divided into six administrative regions, called *zones*. The probability of receiving an order from any of these zones varies, as presented under the *Probability* column. The total delivery time, round trip travel time of each zone, is presented under columns *Industrial set* and *Random set*.

Table 3: Distribution of delivery times for different regions

Zone	Probability	Deliveries times (periods)	
		Industrial set	Random set
1	0.3	2	1
2	0.1	2	2
3	0.2	3	3
4	0.2	4	4
5	0.1	5	5
6	0.1	6	6

We consider two products ($P = 2$) produced on either three or four docks ($m \in \{3, 4\}$). Two or three docks are dedicated to product 1 (sewage sludge) and one dock for product 2 (food residues). In all the instances, we set h^- to 7:00 and h^+ to 20:00, which corresponds to the daytime period in May, and use one-hour time intervals. We add an extra two-hour interval at the end of the planing horizon to allow departures at the end of the production period and avoid the end-of-horizon effect. Therefore, we define the number of intervals as $J \in \{62, 74, 98\}$. Being representative of productions rates observed in May and June, the loading time of products 1 and 2 are respectively equivalents of five and eight intervals.

As products are continuously loaded on docks, the minimum number of orders needed to cover the entire planning horizon for any product i is $\left\lceil \frac{J}{p_i} \right\rceil$ to which we add 25% to allow the model to choose from the orders. Thus, the total number of orders is $n = \left\lceil \frac{1.25J}{p_1} \right\rceil + \left\lceil \frac{1.25J}{p_2} \right\rceil$. To create orders for different zones, we generate a random number between 0 and 1, and compare it against the cumulative probability of the zones, using the "Probability" data from Table 3. Then the order is assigned to the corresponding zone for which the delivery times are known.

We name these industrial instances by the number of periods, orders, and docks where each set contains 5 instances. Thus set I62J26n3m then refers to an industrial set with 62 periods J, 26 orders n, and 3 docks m. Note that all the other parameters are exactly the same in instances with 3 and 4 docks. This enables us to better isolate the effect of dedicating an extra dock to production. The details of these instance sets are shown in Table 4.

Table 4: Parameters of the Industrial Set instances

Set name	Periods (J)	Number of Orders (n)	Docks (m)	Docks for product p_1	p_2
I62J26n3m	62	26	3	0, 1	2
I62J26n4m	62	26	4	0, 1, 2	3
I74J31n3m	74	31	3	0, 1	2
I74J31n4m	74	31	4	0, 1, 2	3
I98J41n3m	98	41	3	0, 1	2
I98J41n4m	98	41	4	0, 1, 2	3

In order to better evaluate the performance of the model, we generated a *Random set* that includes larger instances. Compared to the previous set, the Random Set includes two or three products and as presented in Table 3, the delivery time has more variation. We consider the number of trucks to be equal to the number of products. Each set will be solved with two and three docks dedicated to each product, thus up to 9 docks for the larger instances. The number of orders is calculated as before by including the third product. Details of the random sets are given in Table 5. Instance set R50J50n3P refers to random instance with 50 periods, 50 orders, and three products.

5.2. Computational results on the Industrial set and model calibration

We use the industrial set to assess the performance of five model configurations, as shown in Table 6. These configurations include: the original DTM formulation (1)–(12), DTM with

Table 5: Random instance set parameters

Set name	Periods (J)	Number of Products (P)	Orders (n)	Product loading time p_1	p_2	p_3
R50J29n2P	50	2	29	4	5	–
R74J43n2P	74	2	43	4	5	–
R98J56n2P	98	2	56	4	5	–
R122J70n2P	122	2	70	4	5	–
R146J83n2P	146	2	83	4	5	–
R170J97n2P	170	2	97	4	5	–
R26J28n3P	26	3	28	4	5	3
R50J50n3P	50	3	50	4	5	3
R74J74n3P	74	3	74	4	5	3
R98J98n3P	98	3	98	4	5	3
R122J122n3P	122	3	122	4	5	3

valid inequalities (13), DTM with the variable reduction procedures (Var Red), DTM with both Var Red and valid inequalities (13), and finally, DTM with constraints (13) implemented as a special-ordered-set type 1 constraints (13–SOS). The SOS constraint can provide benefits by allowing the solver to better identify branching decisions.

Table 6 provides an overview on the performance of each configuration by comparing the number of constraints and integer variables used, the computation time, and the value of the objective function obtained. As all configurations achieved the optimal solution within the allotted time of 21,600 seconds, we report the objective function value (Obj. value) only once in Table 6.

In all instances with three docks (two for product one and one for product two), all configurations obtain the optimal solutions. However, the original model reaches the maximum time limit for seven out of 15 instances without proving optimality, with an average computing time of 13,344 seconds. Adding constraints (13) has a huge impact on the computing time which decreases to 2,139 seconds; this is achieved by adding on average only 59 constraints. When using the variable reduction procedure, the size of the model changes considerably (from 16,383 constraints to 2,230, and from 8,960 integer variables to only 3,065) also producing an important reduction in the computing time from 13,344 to 2,347 seconds. Adding either constraints (13) or the variable reduction is almost as good as using both of them simultaneously. However adding constraints (13) as SOS combined with the variable reduction seems to be very efficient in solving larger

Table 6: Computational results for the Industrial Set

	DTM				DTM + (13)			DTM + Var Red			DTM + (13) + Var Red			DTM + (13-SOS) + Var Red		
Set name	Number of constraints	Number of integers	Obj. value	Time (sec)	Number of constraints	Number of integers	Time (s)	Number of constraints	Number of integers	Time (s)	Number of constraints	Number of integers	Time (s)	Number of constraints	Number of integers	Time (s)
Three docks																
I62J26n3m	10148.0	5564.0	73.0	12276.9	10179.0	5564.0	10.0	1474.0	2160.0	5.0	1493.0	2160.0	6.5	1474.0	2160.0	8.1
I74J31n3m	14310.4	7823.8	70.0	6156.3	14347.4	7823.8	81.7	1987.6	3064.8	16.6	2010.6	3064.8	15.8	1987.6	3064.8	20.5
I98J41n3m	24692.4	13494.2	101.0	21600.7	24742.4	13494.2	6328.1	3230.4	3971.2	7019.9	3261.4	3971.2	6618.2	3230.4	3971.2	264.4
Average	16383.6	8960.7	81.3	13344.6	16422.9	8960.7	2139.9	2230.7	3065.3	2347.2	2255.0	3065.3	2213.5	2230.7	3065.3	97.6
Four docks																
I62J26n4m	13902.0	7676.0	51.0	13.2	13945.0	7676.0	3.7	1529.0	3104.0	0.6	1548.0	3104.0	0.7	1529.0	3104.0	0.6
I74J31n4m	19602.4	10787.8	33.0	50.9	19653.4	10787.8	9.5	2054.6	4413.8	5.1	2077.6	4413.8	3.4	2054.6	4413.8	4.6
I98J41n4m	33893.6	18594.2	51.0	21601.1	33962.6	18594.2	71.1	3321.4	6346.2	14.3	3352.4	6346.2	6.2	3416.4	6346.2	15.0
Average	22466.0	12352.7	45.0	7221.7	22520.3	12352.7	28.1	2301.7	4621.3	6.7	2326.0	4621.3	3.4	2333.3	4621.3	6.7

instances with 98 periods. This configuration requires the lowest average computing time of only 97.6 seconds, versus 13,344 seconds for the original DTM. These results demonstrate how introducing the valid inequality and variable reduction procedures are extremely valuable in solving the FPDSP.

For instances with four docks (three docks for product 1 and one dock for product 2), we observe that despite the models being larger than the ones with three docks, they are much easier to solve. The first observation is that adding a third dock for product 1 considerably reduces the distribution cost as the extra dock gives more flexibility to hold a full trailer overnight. Computing times have also been drastically reduced. For the four docks instances, the best combinations are using constraints (13) (with or without SOS) and the variable reduction. These versions require less than ten seconds on average in comparison with 7,221 seconds for the initial DTM. From a practical point of view, these results confirm the usefulness of building a facility with three docks for the sewage sludge as it almost halved the distribution costs.

Other parameters of the problem and configurations of the facility are assessed. Due to their significantly better performance, in what follows we will use models DTM + (13) + Var Red and DTM + (13-SOS) + Var Red to solve larger instances from the random set.

5.3. Computational results on Random set

In Table 7, we present the results from the random sets considering two docks per product. The upper part of the table presents the results from two-product sets while the lower part is dedicated to the ones with three products. As before, for each set and model configuration, we present the number of constraints and integer variables, followed by the objective value, the gap reported by Gurobi (calculated as $\frac{\text{objective value}-\text{incumbent solution objective}}{\text{incumbent solution objective}}$) as well as the run time. For the sets with a non null gap, we also identify the number of non-optimal solutions obtained in each set.

For the two-product instances, both configurations always find the same solutions. Out of 25 instances, we find 22 proven optimum, leaving only three instances in set R122J70n2P with no proven optimum. This is achieved by combining the results of both model configurations. As expected, the computing time increases rapidly; among the two models presented, the results indicate that directly considering constraints (13) seems a better choice than using the SOS option. In the lower part of Table 7, we see that adding a third product almost doubles the size of the model, which has an impact on the computing times and the resulting gaps. Detailed results for all instances indicate that 20 out of 25 instances are solved to optimality, yet both configurations fail to validate the optimum of set R122J92n3P with gaps of 75.6% and 77.8%, respectively. Also note that for this set, the model with (13–SOS) configuration does not obtain the same solutions within the time limit. Clearly both configurations reach their limits on solving instances from sets R98J98n3P and R122J92n3P, indicated by the gaps and large computing times.

Finally, for the three-product instances, the value of the objective function almost triples which is explained by the very short loading time of product three. In general, using DTM + (13) + Var Red configuration produces an average gap of 9.6% in 5,362 seconds instead of 11.4% in 6,311 seconds when (13) is generated as SOS.

Table 7: Results for random instances with two docks for each product

Set name	DTM + (13) + Var Red.					DTM + (13–SOS) + Var Red.				
	Number of constraints	Number of integers	Objective value	Gap (%)	Time (sec)	Number of constraints	Number of integers	Objective value	Gap (%)	Time (sec)
R26J16n2P	586.4	631.0	13	0.0	0.2	575.4	631	13	0.0	0.2
R50J29n2P	1664.8	2136.0	20	0.0	14.1	1642.8	2136	20	0.0	17.9
R74J43n2P	3391.6	4679.8	38	0.0	147.8	3359.6	4679.8	38	2.8[1]	4432.7
R98J56n2P	5655.4	8070.2	60	0.0	5188.5	5612.4	8070.2	60	3.6[1]	5393.6
R122J70n2P	7250.0	12534.6	66	21.0[4]	18195.6	7196.0	12534.6	66	25.4[4]	18429.0
Average	3709.6	5610.3	39.4	4.2	4709.3	3677.2	5610.3	39.4	6.4	5656.7
R26J28n3P	1099.6	1185.6	47	0.0	1.3	1080.6	1185.6	47	0.0	0.7
R50J50n3P	2779.6	3959.2	70	0.0	95.8	2741.6	3959.2	70	0.0	57.3
R74J74n3P	6639.4	8612.0	90	0.0	1389.9	6583.4	8612.0	90	4.8[1]	5153.4
R98J98n3P	9518.4	15067.4	119	0.0	7016.8	9443.4	15067.4	119	0.0	8017.1
R122J92n3P	14498.4	23288.6	268	75.6[5]	21600.0	14402.4	23288	280	77.8[5]	21600.3
Average	6906.7	10422.5	118.8	15.1	6020.9	6850.3	10422.5	121.2	16.5	6965.7
Global average	5308.1	8016.4	79.1	9.6	5362.1	5263.7	8016.4	80.3	11.4	6311.2

Results for larger sets with three docks per product are displayed in Table 8. For the two-product sets, again both models always obtain optimal solutions, and generating constraints (13–SOS) is a bit faster. In the second part of Table 8, we see again that adding a third product almost doubles the size of the models. Sixteen out of the 25 instances have been solved to optimality. No model validates an optimum solution for set R122J92n3P with gaps of 61.4% and 58.8% respectively and the model with the (13–SOS) option was not able to find the same solutions within the time allotted. Consistent with the previous results, the value of objective function for the three-product instances increases considerably. Globally, using DTM + (13) + Var Red yields an average gap of 26.9% in 5257 seconds instead of 27.1% in 4544 seconds with (13–SOS).

Table 8: Results for random instances with three docks for each product

	DTM + (13) + Var Red.					DTM + (13–SOS) + Var Red.				
Set name	Number of constraints	Number of integers	Objective value	Gap	Time (sec)	Number of constraints	Number of integers	Objective value	Gap	Time (sec)
R26J16n2P	622.0	969.0	1	0.0	0.1	611.0	969.0	1	0.0	0.1
R50J29n2P	1747.8	3444	1	0.0	5.6	1725.8	3444.0	1	0.0	7.1
R74J43n2P	3522.6	7651.8	1	0.0	89.8	3490.6	7651.8	1	0.0	83.4
R98J56n2P	5834.4	13286.2	8	0.0	200.9	5791.4	13286.2	8	0.0	164.7
R122J70n2P	7477.0	20732	5	0.0	3042.4	7423.0	20732.6	5	0.0	2465.7
Average	3840.7	9216.7	3.2	0.0	667,8	3808,4	9216,7	3.2	0.0	544.2
R26J28n3P	1156.8	1807.6	31	0.0	0.7	1137.8	1807.6	31	0.0	0.7
R50J50n3P	2908.8	6275.2	35	0.0	262.9	2870.8	6275.2	35	0.0	109.5
R74J74n3P	6840.4	13816.0	32	9.4[2]	10790.8	6784.4	13816.0	32	2.8[1]	5240.1
R98J98N3P	9791.6	24313.4	36	29.0[3]	16581.6	9716.6	24313.4	36	25.0[3]	15772.8
R122J92n3P	14841.4	37726.6	119	61.4[5]	21600.7	14747.4	37726.6	121	58.8[5]	21600.0
Average	7107.8	16787.7	50.6	19.9	9847.3	7051.4	16787.7	51	17.3	8544.7
Global average	5474.2	13002.2	26.9	9.9	5257.5	5429.8	13002.2	27.1	8.6	4544.4

Comparing Table 7 (two docks per product) and Table 8 (three docks per product) shows interesting results. For DTM + (13) + Var Red, the number of constraints goes from 5,308 to 5,474 and the number of integer variables increases from 8,016 to 13,002. Despite this increase in the problem size, the average gap increases just slightly from 9.6% to 9.9% and the average computing time reduces from 5,362 to 5,257. The most important impact is on the value of the objective function which goes from 79.1 to 26.9. Globally, adding a third dock rather simplifies the problem as it gives more versatility to schedule the delivery outside of the hours having an extra cost. Finally, there is no clear conclusion about the utility of using the SOS option for constraints (13). For the results with two docks per product shown in Table 7 use of constraints (13) produce a lowers gap within shorter computing times. However, with three docks per product, as shown in Table 8, it is now constraint (13–SOS) that produces the lowest gap with shorter computing times.

5.4. Managerial insights

Our analyses demonstrate that the proposed model is capable of solving instances large enough to help the plant manager develop better plans. Indeed, in our detailed computational experiments we have solved instances larger than what the facility is expected to face during its steady state production. This is mainly possible by optimizing tactical important decisions of production scheduling and delivery operations in an integrated and efficient way.

The model presented here can also be easily adapted to evaluate the effect of several design and operation settings, for example:

- the model can determine the number of vehicles, if V is set as a decision variable in constraints (7) and its corresponding cost is added to the objective function;
- delivery time windows can also be easily integrated into the model, as timing constraints and variables are already present;
- all seasonal impacts can be modeled by changing parameters such the delivery time, the loading time, and the trailer capacity.

From the strategic point of view, our results show the impact of additional docks and their influence on reducing transportation cost. Considering the trade-offs between adding docks and reducing the transportation costs, this information is very useful to determine the best size of the plant, and to guide development and expansion plans.

6. Conclusions

This paper has been motivated by our collaboration with an industrial partner building a plant to transform sewage sludge and food residues into fertilizer. At full capacity, the plant will produce 76,000 tons of bio-fertilizer per year leading to several managerial and logistics challenges. We have modeled the fertilizer production and distribution scheduling problem as an integer linear program and proposed very powerful valid inequalities and variables reduction procedures to help solve the model. We have generated two sets of instances, one based on the current expectation of the managers (industrial) and a larger one considering any future expansion plans (random). On the industrial sets of instances, the average computing time of the model (with three docks) reduces from 13,344 seconds to 2,139 with the valid inequalities and from 7,221 seconds to only 28 seconds when a fourth dock is added. Similar reductions are achieved with the variable reduction procedures. Further computational results on random sets of instances show that the model can provide optimal four-day distribution schedule. Using both the valid inequalities and the variable reduction procedures allow us to solve instances larger than the expected capacity of the plant. The model is also flexible enough to provide different managerial insights to the operator. As future research, one could develop a fast and flexible metaheuristic to dynamically manage unexpected situations such as a conveyor malfunction (a dock may not be reachable), extreme weather conditions that can impact distribution plans, or the possibility to accept short term orders.

Acknowledgments

This project was partly funded by the Canadian Natural Sciences and Engineering Research Council (NSERC) under grants 2019-00094, 2020-00401, and 2018-03712. We also thank our industrial partner that shared with us all technical specifications as well as their expected production and delivery data. This support is greatly acknowledged.

References

Aspire. Aspire, annual report, 2019. https://fertilizercanada.ca/wp-content/uploads/2019/08/fc_annualreport2019_en_vf-web.pdf.

B. Assamoi and Y. Lawryshyn. The environmental comparison of landfilling vs. incineration of MSW accounting for waste diversion. Waste Management, 32(5):1019–1030, 2012.

I. Awudu and J. Zhang. Uncertainties and sustainability concepts in biofuel supply chain management: A review. Renewable and Sustainable Energy Reviews, 16(2):1359–1368, 2012.

A. Azadeh, H. V. Arani, and H. Dashti. A stochastic programming approach towards optimization of biofuel supply chain. Energy, 76:513–525, 2014.

L. Berghman and R. Leus. Practical solutions for a dock assignment problem with trailer transportation. European Journal of Operational Research, 246(3):787–799, 2015.

K. Chojnacka, K. Moustakas, and A. Witek-Krowiak. Bio-based fertilizers: A practical approach towards circular economy. Bioresource Technology, 295:122223, 2020.

S. Das and B. K. Bhattacharyya. Optimization of municipal solid waste collection and transportation routes. Waste Management, 43:9–18, 2015.

U. Derigs, M. Pullmann, and U. Vogel. Truck and trailer routing-problems, heuristics and computational experience. Computers & Operations Research, 40(2):536–546, 2013.

M. Franz. Phosphate fertilizer from sewage sludge ash (SSA). Waste Management, 28(10):1809–1818, 2008.

S. Gelareh, R. N. Monemi, F. Semet, and G. Goncalves. A branch-and-cut algorithm for the truck dock assignment problem with operational time constraints. European Journal of Operational Research, 249 (3):1144–1152, 2016.

G. Ghiani, D. Laganà, E. Manni, R. Musmanno, and D. Vigo. Operations research in solid waste management: A survey of strategic and tactical issues. Computers & Operations Research, 44:22–32, 2014.

M. Gross. Where is all the phosphorus? Current Biology, 27(3):R1141–R1144, 2017.

IISD. Nutrient recovery and reuse in Canada: Foundations for a national framework, 2018. https://www.iisd.org/sites/default/files/meterial/nutrient-recovery-reuse-canada.pdf.

D. V. Kurpel, C. T. Scarpin, J. E. Pécora Junior, C. M. Schenekemberg, and L. C Coelho. The exact solutions of several types of container loading problems. European Journal of Operational Research, 284(1):87–107, 2020.

G. Mantzaras and Evangelos A. Voudrias. An optimization model for collection, haul, transfer, treatment and disposal of infectious medical waste: Application to a Greek region. Waste Management, 69: 518–534, 2017.

M. A. Quddus, S. Chowdhury, M. Marufuzzaman, F. Yu, and L. Bian. A two-stage chance-constrained stochastic programming model for a bio-fuel supply chain network. International Journal of Production Economics, 195:27–44, 2018.

E. C. Rodias, A. Sopegno, R. Berruto, D. D. Bochtis, E. Cavallo, and P. Busato. A combined simulation and linear programming method for scheduling organic fertiliser application. Biosystems Engineering, 178:233–243, 2019.

P. J. Shah, T. Anagnostopoulos, A. Zaslavsky, and S. Behdad. A stochastic optimization framework for planning of waste collection and value recovery operations in smart and sustainable cities. Waste Management, 78:104–114, 2018.

J. Trochu, A. Chaabane, and M. Ouhimmou. A two-stage stochastic optimization model for reverse logistics network design under dynamic suppliers' locations. Waste Management, 95:569–583, 2019.

E. Van Eygen, D. Laner, and J. Fellner. Circular economy of plastic packaging: Current practice and perspectives in Austria. Waste Management, 72:55–64, 2018.

M. C. Vélez-Gallego, A. Teran-Somohano, and A. E. Smith. Minimizing late deliveries in a truck loading problem. European Journal of Operational Research, 2020.

H. Weigand, M. Bertau, W. Hübner, F. Bohndick, and A. Bruckert. RecoPhos: Full-scale fertilizer production from sewage sludge ash. Waste Management, 33(3):540–544, 2013.