

# **Integrated Poultry Production-Distribution Optimization**

**Samuel Gbéya  
Maryam Darvish  
Jacques Renaud  
Leandro C. Coelho**

**March 2026**

**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 Palasis-Prince, local 2415  
Québec (Québec) G1V 0A6  
Tél : 1-418-656-2073  
Télécopie : 1-418-656-2624

# Integrated Poultry Production-Distribution Optimization

Samuel Gbéya\*, Maryam Darvish, Jacques Renaud, Leandro C. Coelho

Interuniversity Research Centre on Enterprise Networks, Logistics and Transportation (CIRRELT) and  
Laval University, Department of Operations and Decision Systems, Québec (QC), G1V 0A6

**Abstract.** In this paper, we describe, model, and solve an integrated production-distribution planning problem encountered in the poultry supply chain. The production process involves raising chicks on farms before they are sent to slaughterhouses for processing. Production planning decisions include determining the number of chicks to be raised on each farm, the start time, and the duration of their breeding. Distribution planning focuses on scheduling the delivery of mature poultry from farms to slaughterhouses, specifying the timing and quantity of each delivery while accounting for transportation costs and slaughterhouse quota allocations. The objective is to develop cost-efficient production and delivery schedules that respect numerous industry-specific constraints. We formulate the problem as a mixed-integer linear programming model and propose a neighborhood search-based matheuristic to solve large-scale, real-world instances. Computational results on instances derived from data provided by a Québec poultry producer demonstrate the effectiveness of the proposed solution approach.

**Keywords:** Scheduling, livestock production, integrated production-distribution, clustering, matheuristic.

Results and views expressed in this publication are the sole responsibility of the authors and do not necessarily reflect those of CIRRELT.

Les résultats et opinions contenus dans cette publication ne reflètent pas nécessairement la position du CIRRELT et n'engagent pas sa responsabilité.

---

\* Corresponding author: samuel.gbeya.1@ulaval.ca

Dépôt légal – Bibliothèque et Archives nationales du Québec  
Bibliothèque et Archives Canada, 2026

© Gbéya, Darvish, Renaud, Coelho and CIRRELT, 2026

## 1. Introduction

The Canadian agri-food industry plays a vital role in the national economy, with strong interconnections to manufacturing, transportation, and retail. In 2024, agriculture and agri-food exports totaled approximately \$100.3 billion, placing Canada among the world’s top ten exporters (Agriculture and Agri-Food Canada, 2024b). On the domestic front, food-related expenditures reached nearly \$213.6 billion, underscoring the sector’s importance to household spending. Within this broader industry, poultry and egg production constitute a major agricultural segment, generating \$6.8 billion in farm cash receipts, or 7.0% of total agricultural revenues, in 2024 (Agriculture and Agri-Food Canada, 2024a). Chicken production alone contributed over \$4 billion in 2024, with Québec and Ontario accounting for the majority of national output. In Québec, the agri-food sector represents nearly 7% of the province’s GDP, with poultry and egg production serving as a critical pillar of both local consumption and interprovincial trade (Ministère de l’Agriculture, des Pêcheries et de l’Alimentation du Québec (MAPAQ), 2024; Agriculture and Agri-Food Canada, 2024a).

The structure of the poultry supply chain and its level of integration vary between countries and companies (Solano-Blanco et al., 2023a); in Québec, the industry is highly vertically integrated (Ministère de l’Agriculture, des Pêcheries et de l’Alimentation du Québec, 2011). Another layer of complexity arises from the fact that in Canada, chickens, turkeys, eggs, and dairy products are under a supply management system that relies on the allocation of quotas (Mundler and Ubertino, 2022). This system is specifically designed to manage and stabilize the supply of poultry products in the market, prevent overproduction, and ensure fair prices for producers. In addition, slaughterhouse production lines benefit from birds of uniform size and weight, reducing the need for frequent machine adjustments (Solano-Blanco et al., 2023b). Producing poultry of consistent weight can be challenging when sourcing birds from multiple farms. Due to the industry’s emphasis on uniform product quality, poultry farms closely monitor weight gain to ensure consistent chicken size for restaurant supply (Solano-Blanco et al., 2023b). This ensures that the chickens sent to slaughterhouses closely match the target market weight, meeting the industry standards for uniformity.

Although poultry producers aim for high-quality production, managing the supply chain can be difficult. The timing of when each farm should receive its day-old birds from hatcheries is crucial in determining the availability of birds a few weeks later. Another important decision is when each farm should transfer its stock to a slaughterhouse. This decision can have a significant impact on the satisfaction, performance, and profitability of the downstream supply chain.

In this paper, we describe, model, and solve an integrated poultry production-distribution problem (IPDP). The problem is inspired by a collaboration with a Québec-based slaughterhouse and poultry processing company. During the last decade, the company has invested in automating and optimizing production processes to achieve operational excellence. However, they remain concerned about the proportion of birds that fail to meet weight standards at the slaughterhouse, impacting both the quality and commercial value of the poultry. The problem is further complicated by the quota

system that regulates poultry production in slaughterhouses in Québec. Furthermore, the poultry industry faces the challenge of minimizing transportation costs when moving poultry from different farms to slaughterhouses.

The objective is to coordinate production and distribution decisions to minimize total cost while adhering to numerous government-established regulations. We formulate this problem as a mixed-integer linear programming (MILP) model and propose a tailored, scalable solution approach to handle realistically sized instances that accounts for quota deviations, weight deviations, and transportation costs. The contributions of this paper are manifold: (i) a MILP formulation is proposed for the IPPDP; (ii) a neighborhood search-based metaheuristic is designed to solve large-scale, industry-driven instances; (iii) a thorough computational study is conducted to assess the scalability and solution quality of the proposed approach; and finally, (iv) managerial insights derived from the analysis of the structural and operational characteristics of poultry production systems.

The remainder of this paper is as follows. In Section 2, we review the relevant literature. A formal description of the problem is presented in Section 3. The mathematical formulation is described in Section 4 and the solution algorithm is described in Section 5. In Section 6, computational experiments are presented and conclusions are drawn in Section 7.

## 2. Literature review

In this section, we review integrated production-distribution problems, their applications, and specific studies in the chicken industry. The objective is to situate the problem addressed in this paper within the existing literature in terms of modeling approaches, application contexts, and solution methods.

### 2.1. Integrated production-distribution problems

Integrated production-distribution problems have been widely studied in the operations research literature as a means to coordinate decisions across multiple stages of the supply chain. Most integrated models combine at most two core functions, such as production and routing (e.g., the production routing problem, PRP), inventory and routing (e.g., the inventory-routing problem, IRP), or production and inventory (e.g., the lot sizing problem, LSP). Other combinations, including location-routing and location-inventory problems, have also been extensively examined (see, e.g., Cheng et al. (2019); Darvish et al. (2021); Berghman et al. (2023)).

In their comprehensive review, Darvish et al. (2021) classify integrated production-distribution problems into two main research streams: one focusing on facility location decisions combined with distribution planning, and another addressing the integration of lot-sizing with distribution decisions. Early work by Chandra and Fisher (1994) examined the benefits of coordinating production and distribution planning, comparing sequential and integrated decision-making approaches in a multi-product, multi-period setting. Their results demonstrated the potential cost reductions achievable

through integrated planning. Later research further expanded the scope of integration. For instance, Lei et al. (2006) developed a model coordinating production, inventory, and delivery operations to satisfy customer demand while minimizing total costs. Due to the problem's complexity, the authors proposed a two-phase solution approach combining mixed-integer programming and consolidation strategies. Related formulations were later studied by Bard and Nananukul (2009) and Bard and Nananukul (2010), who emphasized the integration of production, inventory, and routing decisions and proposed exact and heuristic solution approaches.

Different naming conventions have been adopted in the literature to describe closely related models. For example, Boudia et al. (2006) and Boudia et al. (2007) refer to these problems as integrated production-distribution problems, whereas Adulyasak et al. (2014) and Absi et al. (2015) use the term production routing problem. Boudia et al. (2007) studied a multi-period, single-product production-distribution problem and proposed a greedy randomized adaptive search procedure for large-scale instances. In contrast, Absi et al. (2015) formulated a capacitated lot-sizing problem coupled with routing decisions and developed a two-phase iterative heuristic alternating between production and distribution decisions.

Several extensions of integrated production-distribution problems have been proposed to capture additional operational features. For example, Hwang et al. (2016) addressed a lot-sizing problem with transportation modeled as a concave minimum cost network flow, while Belo-Filho et al. (2015) studied an operational integrated production-distribution problem for perishable goods and proposed an adaptive large neighborhood search (ALNS) algorithm. Other variants include the integrated production and transportation scheduling (IPTS) problem, which has been studied by Karaođlan and Kesen (2017), Devapriya et al. (2017), Arda et al. (2024), and Jia et al. (2026). Arda et al. (2024) address a IPTS problem in home chemotherapy services, and proposes an ALNS combined with linear programming to optimize production and administration sequences. In contrast, Jia et al. (2026) consider a IPTS rescheduling problem that incorporates type-dependent setup times and multiple shipping modes. They analyze the structural properties of the problem and develop exact algorithms as well as column-generation-based heuristic procedures to solve large-scale instances. When location decisions are incorporated, the problem becomes a production-distribution system design problem, as studied by Elhedhli and Goffin (2005), who proposed decomposition-based exact solution methods.

Despite the extensive body of work on integrated production-distribution problems, most existing formulations focus on simplified network structures or limited combinations of operational decisions. As a result, several features relevant to agri-food and livestock supply chains remain only partially addressed.

## *2.2. Integrated production-distribution applications*

Integrated production-distribution models have been applied to a wide range of industrial contexts. For example, Belo-Filho et al. (2015) proposed an ALNS algorithm to solve a perishable

goods production-distribution problem originally introduced by Amorim et al. (2013). Their model considers a single production facility with parallel production lines and allows split deliveries to satisfy customer demand.

Senoussi et al. (2018) studied a supply chain consisting of a single production facility and multiple geographically dispersed retailers. Their model integrates production setup costs, inventory holding costs at both suppliers and retailers, and distribution costs, and is solved using genetic-algorithm-based heuristics. A multi-plant, multi-period integrated production-distribution problem was introduced by Darvish et al. (2016), who considered delivery due dates and allowed customers to be served by any selected plant.

Other studies have focused on algorithmic developments for large scale instances. Solyalı and Süral (2017) proposed a multiphase heuristic for a single-echelon PRP and demonstrated its effectiveness on benchmark instances from the literature (Boudia et al., 2007; Archetti et al., 2011). Multi-item and multi-echelon extensions were studied by Brahimi and Aouam (2016), Darvish and Coelho (2018), and Neves-Moreira et al. (2019), who highlighted the benefits of integrated planning over sequential approaches in industrial contexts.

More recently, Chagas et al. (2023) studied an integrated production-distribution problem motivated by an industrial application involving direct trailer loading. The absence of storage possibilities in their setting required synchronized production and distribution schedules. The authors proposed several mathematical formulations, valid inequalities, and heuristic procedures, demonstrating significant operational improvements for the industrial partner.

These applications illustrate the practical relevance of integrated production-distribution models and motivate the development of problem-specific formulations and scalable solution approaches.

### *2.3. Integrated production-distribution in the poultry industry*

A growing body of literature has addressed production and distribution planning problems in the poultry industry, often motivated by the biological and operational characteristics of livestock production systems. Early studies focused on allocation and scheduling decisions within poultry production networks. For example, Boonmee et al. (2015) developed a hybrid growing neural gas (HGNG) approach to allocate hens across poultry farms while minimizing transportation and utilization costs. Their work was later extended by Boonmee and Sethanan (2016), who formulated a multilevel lot-sizing and scheduling problem and proposed a growing neural local particle swarm optimization algorithm for large-scale instances.

Broiler production networks have also been studied using integrated modeling approaches. Tahraoui et al. (2020) developed a MILP model for the planning and synchronization of broiler production in a poultry network involving multiple breeders, a slaughterhouse, distributors, and retailers. Their results showed that integrated planning enables synchronized production schedules and cost reductions under varying demand scenarios. More recently, Brevik et al. (2020) introduced the chicken flock sizing, allocation, and scheduling problem (CFSASP), an integrated model encompassing egg

incubation, flock allocation, and slaughter scheduling under an “all-in, all-out” policy. Their formulation penalizes deviations from a target weight and incorporates transportation scheduling to maintain a steady flow of broilers to a single slaughterhouse. To enhance scalability, the authors proposed rolling-horizon-based matheuristics and demonstrated their effectiveness on a real-world broiler operation.

Stochastic extensions have also been proposed to address uncertainty in poultry production. Solano-Blanco et al. (2023a) developed a MILP model for a vertically integrated broiler supply chain, while Solano-Blanco et al. (2023b) introduced a two-stage stochastic programming approach to account for growth uncertainty. Both studies reported significant cost reductions and operational improvements in real-world case studies.

While several studies address integrated poultry production and distribution decisions, most existing models consider single-slaughterhouse settings or simplified network configurations. In addition, solution approaches often rely on commercial solvers or rolling-horizon strategies, with limited attention to scalable neighborhood-based matheuristics for multi-slaughterhouse systems.

#### *2.4. Summary and positioning*

The reviewed literature demonstrates the importance of integrating production and distribution decisions, both in general supply chain settings and in the poultry industry. Existing models capture a wide range of operational features, including lot-sizing, routing, inventory management, and biological constraints specific to livestock production. However, most studies rely on simplified network structures or focus on limited subsets of operational decisions.

Motivated by these observations, this paper formulates an integrated poultry production-distribution problem that captures industry-specific operational constraints under a direct shipment policy. The proposed model accommodates heterogeneous farms and multiple slaughterhouses, yielding a rich and computationally challenging optimization problem.

### **3. Problem description**

In this section, we formally define the IPPDP. We consider a cooperative that manages several slaughterhouses and farms. Each slaughterhouse has limited capacity and must respect the daily production quota. The production quota of each slaughterhouse is deterministic and determined by government regulations. Any deviation from this quota incurs a cost. In fact, in the case of overproduction, excess production is sent to other slaughterhouses, and in the opposite case, poultry is purchased from other producers to compensate for the deficit supply.

Within a given planning horizon, each farm can dispatch at most one batch. Farms whose stocking decisions would result in dispatches beyond the current planning horizon are not excluded from the system but are deferred to subsequent planning periods within a rolling horizon framework, where they become eligible for consideration in later optimization runs. This restriction is not a structural

limitation, but a deliberate modeling choice reflecting the regulatory and cooperative framework of the system under study. In practice, farms are managed under supply-management rules that aim to ensure equitable treatment across producers. Allowing multiple stocking and dispatch cycles within a single planning horizon would systematically favor farms with advantageous characteristics, such as proximity to slaughterhouses, faster growth rates, or larger capacities, which is inconsistent with operational practice.

The poultry is transported to slaughterhouses from several farms. The distance that poultry need to travel during transportation is a crucial factor that can impact the company's profit as well as the welfare of the animals. To minimize transportation costs, the company aims to ideally assign slaughterhouses to farms located close to them. This approach can be viewed as a strategy to minimize the total transportation cost.

A target weight is defined for the poultry to be sold to a specific market. A range of acceptable deviations from this target weight is also defined. Poultry that does not meet the acceptable weight range must be sold as lower-grade products in alternative markets. Since poultry is raised for a specific market, excess weight beyond the target also results in resource waste, leading to potential losses. Therefore, the company aims to reduce the deviation from the target weight. It should be noted that there is an acceptable weight range even for alternative markets. Poultry that weighs significantly less than the target will be rejected for sale, even in alternative markets.

Breeding takes between three and five weeks and the growth rate of birds on different farms varies depending on the conditions of the farm. Based on this information, the growth rate and the breeding duration are known in advance. Taking into account all this information, the production manager assigns a batch to each farm. Poultry must remain on the farm for a breeding period determined by their growth rate and target weight. During this period, the farm cannot accept any other contracts. We assume that each farm is stocked in a single batch at full capacity. This assumption reflects the "all-in, all-out" production policy commonly enforced in poultry farming and also adopted by a Norwegian broiler production company (Brevik et al., 2020), where partial stocking is not permitted due to biosecurity and sanitation regulations. Once a farm is emptied, a sanitation period is held, which lasts for a few days. During this period, the farm has to be kept empty. Consequently, a stocking decision corresponds to utilizing the farm's full capacity rather than any arbitrary fraction.

At the beginning of each planning horizon, the company's production manager must select a set of farms for breeding. At this time, a farm can be in one of these three possible states: (i) empty without any lot being assigned to it and ready to receive a lot, (ii) empty but within its sanitation period, compulsory for all farms between each two breeding activities, (iii) already occupied by birds from the last planning horizon, i.e., they already have an initial inventory.

The problem assumptions are summarized as follows: (i) farms cannot start breeding on specific days of the week (generally, hatcheries do not deliver on Wednesdays, Saturdays, Sundays, or holidays); (ii) poultry growth rate is deterministic on each farm; (iii) each farm is emptied "all at once", and there is no split delivery; (iv) farms are sanitized (during a cleaning period) after each delivery;

(v) slaughterhouses quotas are deterministic; (vi) slaughterhouses do not work during weekends or holidays.

#### 4. Mathematical formulation

In this section, we present a MILP model for the IPPDP. Let  $T$  be the set of days in the planning horizon. The set  $T$  contains two subsets:  $U$  used for possible days to start breeding in farms and the subset  $V$  for the potential delivery days from farms to slaughterhouses. Therefore, a decision must be made on the timing of sending poultry from a set of farms to each slaughterhouse. To make this decision, one needs to take into account the quota, the weight of the poultry, and the distance between a farm and the slaughterhouse to which it is assigned.

Let  $S$  be the set of slaughterhouses, and  $Q_s$  the quota (in number of birds) assigned to slaughterhouse  $s \in S$  for each day in the planning horizon  $V$ . Let  $p^-$  denote the penalty cost per bird for underproduction, and  $p^+$  the penalty cost per bird for overproduction.

Each farm  $b \in B$  is replenished by a hatchery. Hatcheries deliver day-old birds to farms on day  $u \in U$ , representing the beginning day of breeding for farm  $b$ . The farm breeds the birds until an acceptable weight is reached, depending on the target weight  $W$  in decigrams (dg) and the growth rate  $\eta_b$  in dg per day. The breeding period depends on several factors and ranges from 3 to 5 weeks. During the planning horizon,  $\eta_b$  is assumed to be known and deterministic. An acceptable maximum and minimum deviation in percentage over the expected average weight of a flock is defined as  $\delta^+$  and  $\delta^-$ , respectively. Birds that do not reach a weight within the acceptable range can be sold to an alternative market, but a weight restriction applies.  $\xi^+$  represents the maximum additional non-acceptable deviation in percentage over the expected average weight of a flock, and  $\xi^-$  the minimum, both added to  $\delta^+$  and  $\delta^-$  respectively.

Given all this information, we consider three weight situations illustrated in Figure 1: (i) the poultry is within the acceptable range, which means that the weight obtained during the breeding period is within the range  $[W(1 - \delta^-), W(1 + \delta^+)]$ ; (ii) the poultry weighs less than the acceptable range for the specialized market, but can be sold to an alternative market, therefore the weight is within the range  $[W(1 - \delta^- - \xi^-), W(1 - \delta^-)]$ ; (iii) the weight is more than the acceptable weight for the specialized market but can be sold to an alternative market if it is within the range  $(W(1 + \delta^+), W(1 + \delta^+ + \xi^+)]$ .

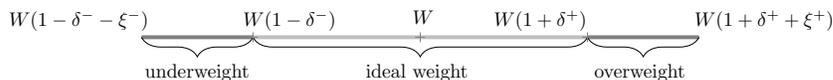


Figure 1: Weight control

We define  $g_1$  and  $g_2$  as the penalty costs per unit deviation (in dg) from the target weight  $W$ , representing penalties for underweight and overweight, respectively, when poultry must be sold in

alternative markets.

We define  $\bar{w}_b^{ut}$  as the deviation from the desired average weight  $W$  for birds entering the farm  $b \in B$  in period  $u \in U$  and leaving for the slaughterhouse in period  $t \in V$ . The weight gained is given by the initial weight  $\varphi_b$  plus the growth rate  $\eta_b$  multiplied by the number of breeding periods  $(t - u)$ . The deviations are calculated as follows:

$$\bar{w}_b^{ut} = |W - (\varphi_b + \eta_b(t - u))| \quad \text{for } t > u. \quad (1)$$

Let  $\bar{H}_b^t$  represent the set of periods  $u$  in which a flock can enter the farm  $b \in B$  if it leaves in period  $t \in V$ . According to Figure 1, we define the following sets.

$$\bar{H}_b^{t1} := \{u \in U : (1 - \delta^- - \xi^-)W \leq (\varphi_b + \eta_b(t - u)) < (1 - \delta^-)W, t > u\}, \quad (2)$$

$$\bar{H}_b^{t0} := \{u \in U : (1 - \delta^-)W \leq (\varphi_b + \eta_b(t - u)) \leq (1 + \delta^+)W, t > u\}, \quad (3)$$

$$\bar{H}_b^{t2} := \{u \in U : (1 + \delta^+)W < (\varphi_b + \eta_b(t - u)) \leq (1 + \delta^+ + \xi^+)W, t > u\}. \quad (4)$$

Obviously,  $\cup_{i=0}^2 \bar{H}_b^{ti} = \bar{H}_b^t$ ,  $\bar{H}_b^{ti} \cap \bar{H}_b^{tj} = \emptyset$  and  $\bar{H}_b^{ti} \neq \emptyset, i \neq j, \forall i, j = 0, 1, 2$ , where

$$\bar{H}_b^t := \{u \in U : (1 - \delta^- - \xi^-)W \leq (\varphi_b + \eta_b(t - u)) \leq (1 + \delta^+ + \xi^+)W, t > u\}. \quad (5)$$

$H_b^u$  is defined as the set of periods  $t$  in which a flock entered farm  $b \in B$  in period  $u \in U$  can leave for the slaughterhouse. A flock can leave the farm during period  $t$  only if it respects the weight tolerance. Similarly to (2), (3), and (4), we define  $H_b^{ui}, i = 0, 1, 2$ ,  $\cup_{i=0}^2 \bar{H}_b^{ui} = \bar{H}_b^u$ ,  $\bar{H}_b^{ui} \cap \bar{H}_b^{uj} = \emptyset$  and  $\bar{H}_b^{ti} \neq \emptyset, i \neq j, \forall i, j = 0, 1, 2$ , where

$$H_b^u := \{t \in V : (1 - \delta^- - \xi^-)W \leq (\varphi_b + \eta_b(t - u)) \leq (1 + \delta^+ + \xi^+)W \text{ and } t > u\}. \quad (6)$$

A sanitation period is required by the law after each breeding on a farm. When a farm is under cleaning at the beginning of the current planning horizon, the number of days left until this period is over,  $\gamma_b$ , is already known. Therefore, after  $\gamma_b$  days from the start day of planning, farm  $b \in B$  would be ready to receive new lots. If a farm is not emptied in the previous planning round, at the beginning of the new planning horizon, it already has an inventory of  $I_b$  birds.

The distance between the farm  $b$  and the slaughterhouse  $s$  is defined by  $d_{bs}$  and we consider  $\bar{c}$  as the cost of transport per kilometer (km). Table 1 summarizes all the parameters defined.

The decision variables are defined as follows.  $r_b^{st}$  represents a binary variable equal to 1 if the poultry from farm  $b \in B$  is sent to slaughterhouse  $s \in S$  during period  $t \in V$ , zero otherwise.  $y_b^{ut}$  is a binary variable equal to 1 if the poultry enter farm  $b \in B$  in period  $u \in U$  and leave in period  $t \in H_b^u$ , zero otherwise.  $q_{st}^+$  represents the number of birds sent to slaughterhouse  $s \in S$  in period  $t \in V$  and exceeding its quota (overproduction), and finally,  $q_{st}^-$  denotes the number of birds sent to slaughterhouse  $s \in S$  in period  $t \in V$  but under its quota (underproduction).

Table 1: Notation table.

Sets	
$T$ :	Set of days in the planning horizon
$V$ :	Set of days in the delivery horizon
$U$ :	Set of days in the breeding horizon
$B$ :	Set of farms
$S$ :	Set of slaughterhouses
Indices	
$b$ :	index of farms
$s$ :	index of slaughterhouses
$u$ :	index of breeding starting days
$t$ :	index of delivery days
Parameters	
$\delta^-, \delta^+$ :	Maximum acceptable deviation (in percentage) under/over the expected average weight of a flock.
$\xi^-, \xi^+$ :	Maximum additional non-acceptable deviation (in percentage) under/over the expected average weight of a flock.
$p^-, p^+$ :	Penalty cost (per poultry) for underproduction/overproduction.
$g_i$ :	Penalty cost per dg for deviation from target weight $W$ , ( $i = 1$ underweight, $i = 2$ overweight).
$d_{bs}$ :	Distance (in km) between farm $b \in B$ and slaughterhouse $s \in S$ .
$\bar{c}$ :	Transportation cost (per kilometer).
$\gamma_b$ :	Number of periods left for the sanitation of farm $b \in B$ at the beginning of the planning horizon.
$\eta_b$ :	Growth rate (in dg per day) for birds in farm $b \in B$ .
$\varphi_b$ :	Initial average weight of the birds in farm $b \in B$ . If no birds are actually breeding in farm $b \in B$ , then their initial weight will be set to their weight when exiting the hatchery.
$Q_s$ :	Quota of slaughterhouse $s \in S$ (in birds) for each day of $t \in V$ .
$C_b$ :	Breeding capacity of farm $b \in B$ .
$I_b$ :	If farm $b \in B$ is already in production, $I_b$ sets its initial inventory. If the farm is not in production, $I_b$ is set to zero.
$W$ :	Average target weight (in dg) for birds to be sold to the specialized market.
$\bar{w}_b^{ut}$ :	Deviation from the average desired weight $W$ for birds entering farm $b \in B$ at period $u \in U$ and leaving for the slaughterhouse at period $t \in V$ .
$\bar{H}_b^t$ :	Set of periods $u$ at which a flock can enter farm $b \in B$ if it leaves at period $t \in V$ to the slaughterhouse.
$H_b^u$ :	Set of periods $t$ at which a flock entered farm $b \in B$ at period $u \in U$ can leave to the slaughterhouse, $H_b^u \subseteq V$ .

## Variables

---

$r_b^{st}$ :	1 if the poultry from farm $b \in B$ is sent to slaughterhouse $s \in S$ during period $t \in V$ , 0, otherwise.
$y_b^{ut}$ :	1 if the poultry enter farm $b \in B$ in period $u$ and leave in period $t \in H_b^u$ , 0, otherwise.
$q_{st}^+$ :	number of birds sent to slaughterhouse $s \in S$ in period $t$ and exceeding its quota (overproduction)
$q_{st}^-$ :	number of birds sent to slaughterhouse $s \in S$ in period $t \in S$ but under its quota (underproduction)

---

The mathematical model of the IPPDP is as follows:

$$\min \sum_{b \in B} \sum_{s \in S} \sum_{t \in V} \bar{c} d_{bs} r_b^{st} + \sum_{b \in B} C_b \sum_{u \in U} \sum_{i=1}^2 \left( g_i \sum_{t \in H_b^{ui}} \bar{w}_b^{ut} y_b^{ut} \right) + \sum_{s \in S} \sum_{t \in V} (p^+ q_{st}^+ + p^- q_{st}^-) \quad (7)$$

Subject to

$$\sum_{b \in B} C_b r_b^{st} = Q_s - q_{st}^- + q_{st}^+ \quad s \in S, t \in V \quad (8)$$

$$\sum_{t \in V} \sum_{s \in S} r_b^{st} \leq 1 \quad b \in B \quad (9)$$

$$\sum_{s \in S} r_b^{st} = \sum_{u \in \bar{H}_b^t} y_b^{ut} \quad b \in B, t \in V \quad (10)$$

$$\sum_{u \in U} \sum_{t \in H_b^u} y_b^{ut} \leq 1 \quad b \in B \quad (11)$$

$$\sum_{t \in H_b^k} y_b^{kt} \leq 0 \quad \begin{matrix} b \in B: \gamma_b \geq 1, \\ k \leq \gamma_b \end{matrix} \quad (12)$$

$$\sum_{t \in H_b^1} y_b^{1t} = 1 \quad b \in B, I_b > 0 \quad (13)$$

$$r_b^{st}, y_b^{ut} \in \{0, 1\} \quad \begin{matrix} b \in B, s \in S, \\ u \in U, t \in V \end{matrix} \quad (14)$$

$$q_{st}^+, q_{st}^- \in \mathbb{Z}_+ \quad s \in S, t \in V \quad (15)$$

The objective function (7) minimizes the total cost including the transportation costs resulting from assigning the production of farms to slaughterhouses over the planning horizon, the penalty costs for not reaching the target slaughter weight at farm  $b$  when production started at  $u$  and finished at feasible periods  $t \in H_b^{ui}$ , and finally the penalty costs for not fulfilling the quota requirement of slaughterhouses at each period  $t$ . Constraints (8) calculate the total supply for each slaughterhouse in each period and set the deviation variables. Birds from a given farm can be sent to only one slaughterhouse during the planning horizon with constraints (9). The constraints (10) state that farm  $b$  can supply slaughterhouse  $s$  in period  $t$  if production has started on farm  $b$  in previous periods  $u \in \bar{H}_b^t$ . Constraints (11) empty each farm at most once during the delivery horizon. Note that if breeding starts at period  $u$ , the exit can occur only on days that belong to the set  $H_b^u$ . In constraints (12), if a farm is currently in the sanitation period (the number of periods left to the sanitation of farm  $b$  at the beginning of the planning horizon is at least one,  $\gamma_b \geq 1$ ), it cannot start production in any period  $k \leq \gamma_b$ . By constraints (13), we ensure that all flocks already on farm  $b$  ( $I_b > 0$ ) will leave for a slaughterhouse at a period  $t \in H_b^1$ . Constraints (14)–(15) define the domain and the nature of the variables.

## 5. Solution algorithm for the IPPDP

This section presents the solution algorithm for the IPPDP. To illustrate its application, Figure 2 shows a small instance involving 16 farms and a slaughterhouse, where  $OF$  denotes the objective value,  $OF_d$  the distance component,  $OF_w$  the weight deviation component, and  $OF_{Q+}$  and  $OF_{Q-}$  the quota deviation components. Twelve farms were selected among the 16.

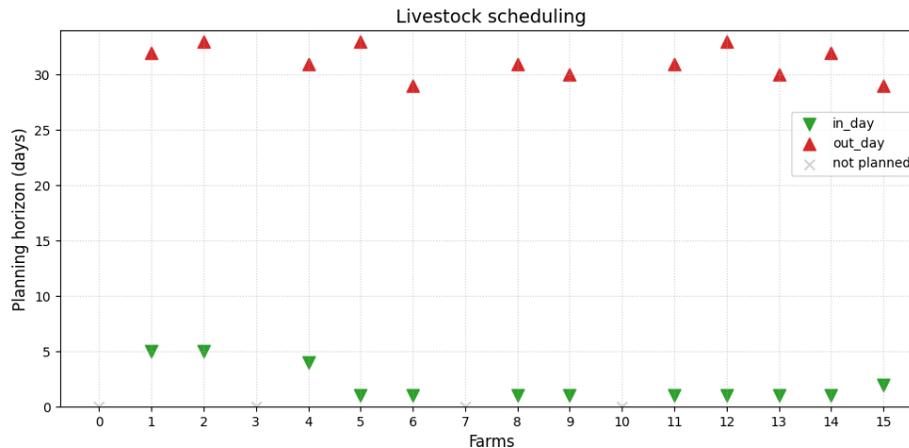


Figure 2: An instance with 16 farms and a slaughterhouse with  $OF = 5025$  ( $OF_d = 648$ ,  $OF_w = 3276$ ,  $OF_{q+} = 353$ ,  $OF_{q-} = 748$ ).

The IPPDP is a joint assignment and scheduling combinatorial problem; as observed in preliminary computational experiments, this makes large instances difficult to solve with commercial solvers. This motivates the development of a two-phase neighborhood search-based matheuristic with two main steps: 1. Initial solution heuristic, 2. Quota deviation penalty costs minimization procedure and solution improvement. In what follows, we present the details of the proposed matheuristic.

### 5.1. Initial solution heuristic

The goal of this phase is to find a feasible initial solution for the IPPDP. Based on quota satisfaction and delivery periods, the algorithm minimizes transportation cost by clustering farms to satisfy the quota for each delivery period  $t \in V$ . The key advantage of this approach is that it implicitly addresses production planning decisions. Farms that cannot be assigned within this phase are temporarily set aside in a separate list for later consideration. The initial solution process unfolds as follows.

#### 5.1.1. Penalty-free breeding durations

First, the set of penalty-free breeding durations for each farm is determined by considering the target slaughter weight, the growth rate of the animals for each farm, and the available working days. These durations are defined as periods during which poultry can reach the required weight range without incurring penalties, thereby qualifying for the specialized market.

For each farm, a penalty-free breeding duration is selected to initiate the clustering process. This duration is chosen randomly among all admissible penalty-free durations. This randomization is intentional and serves two purposes. First, it allows the algorithm to diversify the initial clustering configurations and prevents

systematic biases that may arise from deterministic rules. Second, more elaborate selection rules, such as choosing durations based on the availability of remaining farms, would require evaluating, for each farm and each feasible starting period for breeding, multiple potential delivery periods, leading to a significant computational overhead.

Moreover, such rules are not straightforward to define, as the notion of “best availability” depends on several competing factors, including quota satisfaction, transportation distance, and penalty trade-offs. Instead of embedding these complex decisions at the initialization stage, the proposed approach delegates refinement to the subsequent sequential greedy procedure, which improves the clustering.

### 5.1.2. Clustering

The goal is to create clusters of farms that deliver to the same slaughterhouse on the same day. Knowing that a farm can only deliver to one slaughterhouse in each period, the problem becomes similar to a generalized assignment problem (GAP). Inspired by the greedy algorithm proposed by Martello (1981), we propose the clustering procedure presented in Algorithm 1.

Let  $B = \{1, \dots, n\}$  be the set of  $n$  farms,  $S = \{1, \dots, m\}$  the set of  $m$  slaughterhouses, and  $d_{bs}$  the distance function.  $Q_s$  is the quota that must be met for the slaughterhouse  $s$  on any given day. To control overproduction, we add a quota deviation threshold,  $Q_s^+ = \tau_s Q_s$ , where  $\tau_s \in (0, 1]$ ,  $s = 1, \dots, m$ . Therefore, the new quota for slaughterhouse  $s$  becomes  $Q'_s = Q_s + Q_s^+$ .

Farms are assigned iteratively to slaughterhouses. The main loop iterates as long as the set  $B$  of farms or the set  $L$  is not empty. At each iteration, between lines 4 and 15, for each farm  $b \in B$ , we check the set of candidate slaughterhouses  $\mathcal{F}_b$  where the farm can be assigned based on quota constraints. If no candidate slaughterhouse is available for all farms, then a cluster is constructed, which is a set of farms that can deliver on day  $t \in V$ . Then, we reset the slaughterhouse’s quota, and the cluster is added to the set of clusters  $CL$  if  $V$  is non-empty, otherwise it is added to the set  $R_f$  of farms. We remove  $t$  from  $V$  and update the set  $B$  of farms. The process then returns to line 4 to continue with the remaining farms. Also, if a candidate slaughterhouse is available, the algorithm selects the best slaughterhouse  $s_b^*$  for each farm  $b$  based on the distance minimization function. Then, we determine the next farm  $\hat{b}$  that can deliver on day  $t$  when  $V$  is non-empty. We add it to cluster  $c$  being constructed, remove farm  $\hat{b}$  from set  $B$ , update the quota for the selected slaughterhouse, in lines 20 to 24. For each farm, a penalty-free breeding duration, randomly selected from the set of penalty-free breeding durations, is used to test whether the farm can deliver during period  $t$ . When the delivery days set  $V$  is empty, and it still remains unplanned farms and no assigned farms, we construct clusters based only on distance minimization, in lines 26 to 29. Once all farms are assigned, the algorithm returns the final clustering output  $CL$  and  $R_f$ . Note that  $CL$  denotes the set of clusters and  $R_f$  consists of the set of farms where production has not yet been planned, but the farms have been assigned to specific slaughterhouses for future deliveries. This helps us to improve the initial solution.

The initial solution is obtained by performing complete destruction of  $CL$ , followed by its reconstruction using the sequential greedy insertion method outlined in Section 5.2.2.

### 5.2. Quota deviation minimization procedure and solution improvement

In this phase, we minimize the quota deviation penalty costs using a heuristic based on the large neighborhood search (LNS). LNS has been successfully applied to several integrated supply chain problems (Adulyasak et al., 2014; Eskandarpour et al., 2017). Moreover, the choice of a neighborhood search-based metaheuristic is motivated by the discrete, highly constrained, and structured nature of the IPPDP. The problem involves

**Algorithm 1: Clustering**


---

**Input:**  $V$ : Delivery horizon  
 $B$ : Set of farms  
 $S$ : Set of slaughterhouses  
 $Q_s, Q_s^+ \forall s \in S$ : Quota and threshold

- 1  $CL \leftarrow$  empty set of clusters
- 2  $R_f \leftarrow$  empty set of farms
- 3  $c \leftarrow$  empty cluster of farms
- 4  $L \leftarrow$  empty list of farms
- 5 **while**  $B \neq \emptyset \vee L \neq \emptyset$  **do**
- 6 Choose the first element  $t \in V$
- 7  $\mathcal{F}_b = \{s \in S : C_b \leq Q'_s\}$  for  $b \in B$
- 8 **if**  $\mathcal{F}_b = \emptyset$  for all  $b \in B$  **then**
- 9  $Q'_s = Q_s + Q_s^+$  for  $s = 1, \dots, m$
- 10 **if**  $V \neq \emptyset$  **then**
- 11  $B \leftarrow B \cup L, L \leftarrow \emptyset$
- 12  $CL \leftarrow CL \cup c, c \leftarrow \emptyset$
- 13  $V \leftarrow V \setminus t$
- 14 **else**
- 15  $R_f \leftarrow R_f \cup c, c \leftarrow \emptyset$
- 16 **else**
- 17  $s_b^* = \arg \min_{s \in \mathcal{F}_b} d_{bs}$  for  $b \in B$
- 18  $d_b^* = \min_{i \in \mathcal{F}_b, i \neq s_b^*} d_{bi} - d_{bs_b^*}$  for  $b \in B$
- 19  $\hat{b} = \arg \max_{b \in B} d_b^*$ , i.e.,  $\hat{b}$  is the farm to be assigned next, to the slaughterhouse  $s_{\hat{b}}^*$ :
- 20 **if**  $V \neq \emptyset$  **then**
- 21 **if**  $\hat{b}$  can deliver on period  $t$  **then**
- 22  $c \leftarrow c \cup \{\hat{b}\}$
- 23  $Q'_{s_b^*} = Q'_{s_b^*} - C_{\hat{b}}$
- 24 **else**
- 25  $L \leftarrow L \cup \{\hat{b}\}$
- 26 **else**
- 27  $c \leftarrow c \cup \{\hat{b}\}$
- 28  $Q'_{s_b^*} = Q'_{s_b^*} - C_{\hat{b}}$
- 29  $B \leftarrow B \setminus \{\hat{b}\}$
- 30 **return**  $CL, R_f$

---

assignment and timing decisions subject to strict feasibility constraints on capacity limits, quota compliance, and production policies. Neighborhood search methods, and LNS in particular, are well-suited to such settings, as they allow for systematic exploration of large, problem-specific neighborhoods while maintaining feasibility and exploiting structural information (Pisinger and Ropke, 2018).

During the search, destroy and repair operators are used; these operators compete to create a new solution. We denote by  $CL_t = \bigcup_{s \in S} B_s^t$ ,  $B_s^t \cap B_{s'}^t = \emptyset, s \neq s'$ , the set of farms that deliver in period  $t$ , where  $B_s^t$  is the set of farms that should deliver to slaughterhouse  $s \in S$ .

An outline of the proposed LNS is provided in Algorithm 2. The steps from lines 2 to 9 are repeated until either the specified number of iterations is reached or the algorithm time limit is exceeded. At each iteration, a new solution  $CL'$  is created by destroying and repairing the previous solution (line 5) using the destroy ( $D$ ) and repair ( $R$ ) operators. From lines 6 to 9, based on the cost of  $CL'$ , it is accepted or rejected.

---

**Algorithm 2: LNS Heuristic**


---

**Input:** Set of destroy operators  
 Set of repair operators  
 $CL$  : current solution

- 1 Solution  $CL_{best} = CL$
- 2 **while** *stopping criterion is not met* **do**
- 3     Generate a random number  $q$  of farms to remove
- 4     Select randomly destroy  $D$  and repair  $R$  operators
- 5     Create solution  $CL'$
- 6     **if**  $Cost(CL') < Cost(CL_{best})$  **then**
- 7          $CL_{best} \leftarrow CL \leftarrow CL'$
- 8     **else**
- 9          $CL \leftarrow CL_{best}$

10 **return**  $CL_{best}$

---

### 5.2.1. Farm Removal

Generally, a destroy operator removes a random number of planned and assigned farms,  $q$ , according to specific criteria. This section describes two removal operators. Both operators take a given solution and an integer number  $q$  as input. The output is a solution where  $q$  farms are removed from their assigned clusters.

- *Random removal:* this operator simply removes  $q$  farms randomly.
- *Related removal:* this operator was first proposed by Shaw (1998) and we slightly adapted it for the IPPDP. The general idea is to remove farms from similar clusters. We define the similarity of two farms  $i$  and  $j$  using a *relatedness measure*  $\mathcal{R}(i, j)$ , where  $\mathcal{R}(i, j) = \chi(|u_i - u_j| + |t_i - t_j|) + \phi \sum_{s=1}^{|S|} |d_{is} - d_{js}| + \mu|C_i - C_j|$ . This measure consists of four terms: the start of the breeding period,  $u$ ; the delivery period,  $t$ ; the distance,  $d_{bs}$  where  $b \in B$  and  $s \in S$ ; and the farm capacity,  $C$ . These terms are weighted using  $\chi$ ,  $\phi$  and  $\mu$ , respectively.

It is assumed that  $u_i, t_i, d_{is}$ , and  $C_i$  are normalized such that  $0 \leq \mathcal{R}(i, j) \leq 2\chi + \phi + \mu$ . This is done by scaling  $u_i, t_i, d_{is}$ , and  $C_i$  such that they only take values from  $[0, 1]$ . Notice that we cannot calculate  $\mathcal{R}(i, j)$  if farms  $i$  or  $j$  are not in the solution. The procedure for removing farms is outlined in Algorithm 3. The algorithm starts by selecting a random farm  $b$  from the current solution  $CL$  and removing it. This farm is then added to a set  $E$ , which will eventually contain  $q$  farms to remove from  $CL$ . The process continues as long as the size of  $E$  is less than  $q$ . In each iteration, a farm from  $E$  is randomly selected as a reference to find similar farms in  $CL$ . A list  $\mathcal{L}$  is created, containing farms not already in  $E$ , and these are sorted based on the similarity function  $\mathcal{R}(b, \mathcal{L}[i])$ , which ranks the farms in ascending order of their proximity to  $b$ . Shaw removal has a parameter  $\sigma$  that determines the degree of randomization. A random number  $h$  between 0 and 1 is then chosen to influence the selection of a farm from  $\mathcal{L}$ , and the index  $h^\sigma |\mathcal{L}|$  determines which farm is selected, with  $\sigma$  controlling the bias toward

farms more similar to  $b$ . Once a new farm is selected, it is added to  $E$ , and the process repeats until  $E$  contains exactly  $q$  farms. Finally, all farms in  $E$  are removed from  $CL$ . The similarity is influenced by the parameter  $\sigma \geq 1$ . This parameter adjusts the balance between randomness and determinism in the selection process. Lower values of  $\sigma$  increase randomness, making the selection process more exploratory, while higher values reduce randomness, leading to a more deterministic selection (Ropke and Pisinger, 2006). In particular, when  $\sigma = 1$ , the Shaw removal becomes random. Note that, each farm in  $E$  is removed from  $CL$  and put into  $R_f$ .

---

**Algorithm 3: Related removal for IPPDP**


---

**Input:**  $CL$  current solution,  $\sigma \in \mathbb{R}_+$ ,  $q \in \mathbb{N}$

- 1 Select a random farm  $b$  from  $CL$
- 2 Remove  $b$  from  $CL$
- 3 Set  $E = \{b\}$
- 4 **while**  $|E| < q$  **do**
  - 5     Select a random farm  $b$  from  $E$
  - 6      $\mathcal{L} = \{b \in CL : b \notin E\}$
  - 7     Sort  $\mathcal{L}$  such that  $i < j \Rightarrow \mathcal{R}(b, \mathcal{L}[i]) < \mathcal{R}(b, \mathcal{L}[j])$
  - 8     Choose a random number  $h$  from  $[0, 1)$
  - 9      $E \leftarrow E \cup \{\mathcal{L}[h^\sigma | \mathcal{L}]\}$
- 10 Remove the farms in  $E$  from  $CL$

---

### 5.2.2. Inserting farms

A repair method is randomly selected to rebuild the solution by inserting farms removed during the destruction phase or remaining in  $R_f$ . For IPPDP, we designed three insertion operators as follows.

- *Parallel greedy insertion:* this construction heuristic evaluates all clusters simultaneously at each iteration, computing the best insertion cost for each farm, and inserts the farm that minimizes the global cost. It performs at most  $|B|$  iterations as it inserts one farm in each iteration.

Let  $\pi_{b,t}$  denote the change in the objective value incurred by inserting farm  $b$  into cluster  $CL_t$  at the corresponding  $B_s^t$  that increases the objective value the least. If we cannot insert farm  $b$  into cluster  $CL_t$ , then, we set  $\pi_{b,t} = \infty$ . We define the best insertion cost of the farm  $b$ , as  $Cost(b) = \min_t \{\pi_{b,t}\}$ . Finally, we choose farm  $b$  that has the minimum cost and insert it into its minimum cost cluster.

- *Sequential greedy insertion:* This operator consists of iteratively inserting a farm into the cheapest possible cluster until all farms are inserted in the solution or no more insertion is feasible. Given a cluster  $CL_t$ , we choose the unplanned farm  $b^*$  that has the cheapest  $\pi_{b,t}$  and insert it into the cluster  $CL_t$ . When no more farm with feasible insertion can be found for the cluster  $CL_t$ , the method is restarted with the next cluster.
- *MILP-based insertion:* This operator selects additional farms to insert into the current solution by solving the MILP formulation introduced in Section 4. The optimization is performed with the decision variables corresponding to the existing partial solution held fixed, ensuring feasibility while exploring promising insertions.

## 6. Computational experiments

In this section, we present and discuss the results of our computational experiments to evaluate the efficiency of the proposed algorithms. We begin by describing the instance generation procedure in Section 6.1, followed by the parameter tuning for the neighborhood search-based matheuristic in Section 6.2, and finally, we discuss the results for the IPPDP in Section 6.3. The algorithms are coded in C++ and all tests are conducted on a computer with an Intel(R) Core(TM) i7-7700 processor at 3.60 GHz and 64 GB of RAM. For the exact method, we have used IBM Concert Technology and 20.1 as the MIP solver. A single thread was used with a time limit of an hour. We always provide the best heuristic solution as a warm start to CPLEX. A runtime of 3600s is imposed to all algorithms.

### 6.1. Instances generation

As the IPPDP has not been studied in the literature, inspired by the data shared by our industrial partner, we generated several test instances. An instance generated for the IPPDP is characterized by the number of farms,  $|B|$ , the number of slaughterhouses,  $|S|$  and the number of periods (in weeks), denoted by  $p$ . The naming convention for an instance follows the format IPPDP- $|B|$ - $|S|$ - $p$ . For each combination of parameters, we generated five random instances and redefine the format to IPPDP- $|B|$ - $|S|$ - $p$ - $i$ , where  $i = 0, \dots, 4$  represents the index of each instance.

The number of periods (in weeks) is a random number selected from the range  $[4, 10]$ . The number of slaughterhouses  $|S|$  is 1, 2, or 3. The number of farms considered depends on the number of slaughterhouses: with one slaughterhouse, the number of farms is selected from the set  $\{40, 50, 60, 70, 80, 90, 100\}$ ; with two slaughterhouses, from  $\{125, 150, 175, 200\}$ ; and with three slaughterhouses, from  $\{225, 250, 275, 300\}$ . The weight deviation acceptance  $\delta^+$  and  $\delta^-$  is set to 0.1 and the additional weight deviations  $\xi^+$  and  $\xi^-$  are set to 0.05. We assume that the farms do not have any inventory at the beginning of the planning horizon, that is,  $I_b = 0$ , but the initial weight of the poultry,  $\varphi$ , is set to 380 dg.

Breeding starts on the first day of the planning horizon, and we only remove Wednesdays, Saturdays, and Sundays from the planning horizon  $T$  to obtain the set  $U$ . We construct the set  $V$  similarly to the set  $U$  by removing Saturdays and Sundays. However, the delivery starts at  $\lceil 7p/2.5 \rceil$  if this period corresponds to a delivery. Otherwise, the start is the delivery period immediately following  $\lceil 7p/2.5 \rceil$ , to ensure that the poultry are ready for delivery.

The coordinates of each farm are integer values randomly placed in a  $270 \times 270$  area, whereas the coordinates of slaughterhouses are integer values randomly placed in a  $50 \times 150$  area. We define  $d_{bs}$  as the distance in kilometers between the farm  $b$  and the slaughterhouse  $s$ . The distances are rounded Euclidean distances and we assume that the transportation cost  $\bar{c}$  is one dollar per km. The breeding capacity of each farm,  $C_b$ , is a random integer value within the range  $[4000, 32000]$ . The target weight of the poultry on each farm is  $W = 22500$  dg. The cost parameters  $g_1 = 0.0007$  and  $g_2 = 0.001$ . The initial cleaning period,  $\gamma_b$  for farm  $b$ , is an integer value randomly selected from the range  $[1, p - 1]$ . The number of farms passing their cleaning period at the beginning of the planning horizon is set to ten percent of the total number of farms available. The growth rate of the animals on each farm is an integer value randomly generated within  $[\alpha, \beta]$ , where  $\alpha = (W(1 - \delta^- - \xi^-) - \varphi) / \min(V)$  and  $\beta = (W(1 + \delta^+ + \xi^+) - \varphi) / \min(V)$ , where  $\min(V)$  is the first period of delivery of each planning horizon. The average number of birds delivered in each period  $t \in V$  is shared between slaughterhouses as their quotas. For single-slaughterhouse instances, the quota must exceed the farm capacities. To achieve this, we first calculate the average available capacity per delivery day, multiply

it by a factor randomly selected from  $(0.1, 0.8)$ . If the result exceeds the maximum farm capacity, it is set as the quota; otherwise, a random number between 0.1 and 0.5 is applied to adjust it. In the case of two slaughterhouses, the share of the quota is roughly 60% and 40% of the adjusted quota for the slaughterhouses. Finally, in the case of three slaughterhouses, the share of the quota is approximately 50%, 30%, and 20% of the adjusted quota for the slaughterhouses. The penalty costs for the deviation of the quota  $p^+$  and  $p^-$ , are one dollar per head of poultry. In addition, it should be noted that the quota deviation penalty costs are greater than those for deviations from target weight, as in practice, the cooperative sends production surpluses to other slaughterhouses and covers production shortfalls by purchasing poultry from other farms. These actions are more costly and operationally complex than minor adjustments to the poultry holding period.

We generated five instances for each combination for a total of 75 test instances.

## 6.2. Parameters tuning

This section describes the parameters used in the solution algorithm, emphasizing their critical role in enhancing performance. We follow the sequential strategy of Ropke and Pisinger (2006) for parameter tuning. The process begins with a reasonable default value for each parameter and is followed by an optimization phase. In the optimization phase, we refine the default value by adjusting one parameter at a time while keeping the others fixed. The matheuristic is applied three times to the test instances for each adjustment. We generated 15 configuration instances, comprising five randomly constructed instances for each scenario involving one, two, and three slaughterhouses selected from our dataset. We select the parameter setting that delivers the best average considering both the objective value and the computation time. The process then moves to the next parameter, using the previously optimized values and the original settings for parameters still to be tuned. This procedure is repeated until all parameters have been fine-tuned.

Key parameters include  $\tau_s, s = 1, \dots, |S|$ , which controls overproduction during clustering, and the maximum number of iterations for the LNS. Other parameters include  $(\chi, \phi, \mu, \sigma)$  that govern the removal operators in LNS.  $r$  defines the percentage range of farms to remove.

For each value of  $\tau_s$ , Figure 3 shows the average over all test instances of the minimum, maximum, and mean objective values computed per instance, along with the average computation time. It demonstrates that  $\tau_s = 0.07$ , for all  $s \in S$  provides the best balance, offering the lowest average objective value with a reasonable computation time. Therefore,  $\tau_s = 0.07$  was selected because it offers the best overall trade-off.

We consider two categories of removal operators: *random* and *related*. Notably, the Shaw removal operator behaves as a random operator when  $\sigma = 1$ . Thus, evaluating Shaw removal with  $\sigma = 1$  effectively corresponds to testing the random operator. To investigate the impact of relatedness on solution quality, we vary  $\sigma$  over the set  $\{1, 2, 4, 6, 8, 10\}$ . Figure 4 illustrates the impact of varying  $\sigma$  on both solution quality and computational time. The results indicate that  $\sigma = 2$  yields the most favorable trade-off between these two objectives.

To determine how many farms to remove at each iteration, we define  $q$  as  $|B| \times r\%$ , where  $r$  is the percentage of farms to remove at each iteration, as suggested in (Masson et al., 2013; Shi et al., 2023). We investigated three different interval settings for  $r$ :  $[5, 10]$ ,  $[10, 20]$ , and  $[20, 30]$ . Figure 5a shows that as the number of farms removed decreases, the average execution time also decreases. Moreover, among the ranges tested, the interval  $[10, 20]$  achieves the best balance between the refinement of the solution and the computational efficiency.

Figure 5b presents the parameter tuning process for the *Related removal* operator. We evaluated several configurations of the tuple  $(\chi, \phi, \mu)$ , specifically  $(3, 3, 3)$ ,  $(3, 1, 1)$ ,  $(1, 3, 1)$ , and  $(1, 1, 3)$ . Among these, the

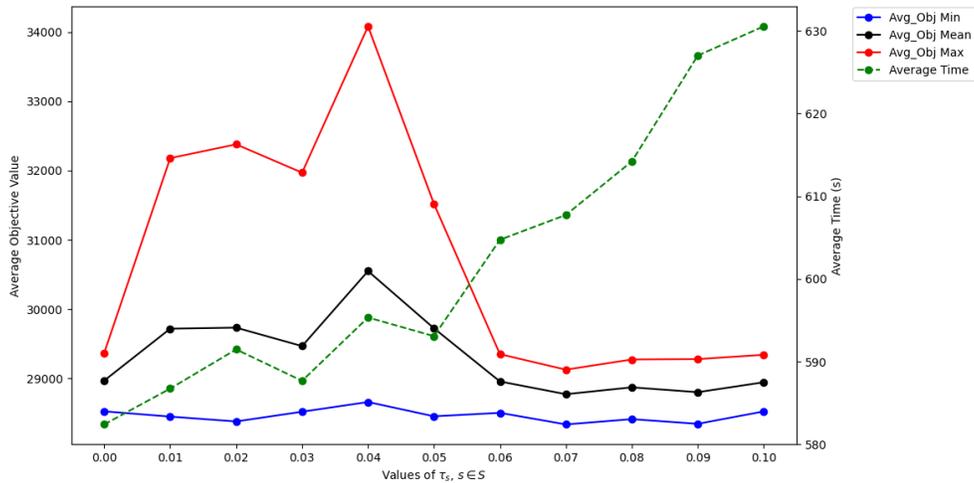


Figure 3: Contribution of  $\tau_s$  in the solution quality.

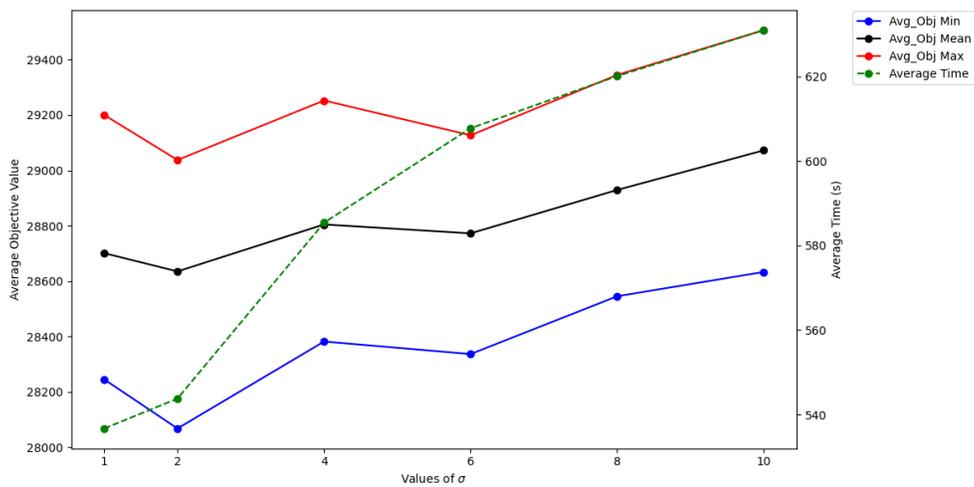
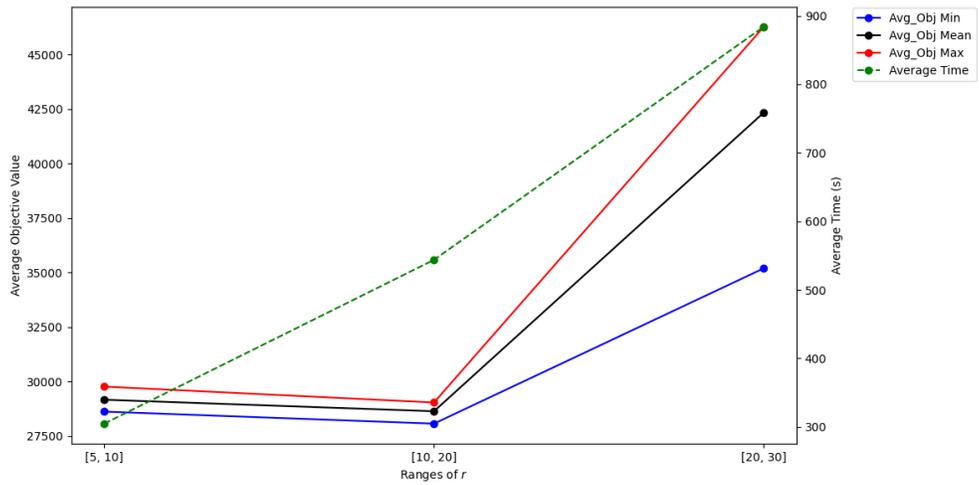


Figure 4: Effect of the values of  $\sigma$  on the performance of the LNS heuristic.

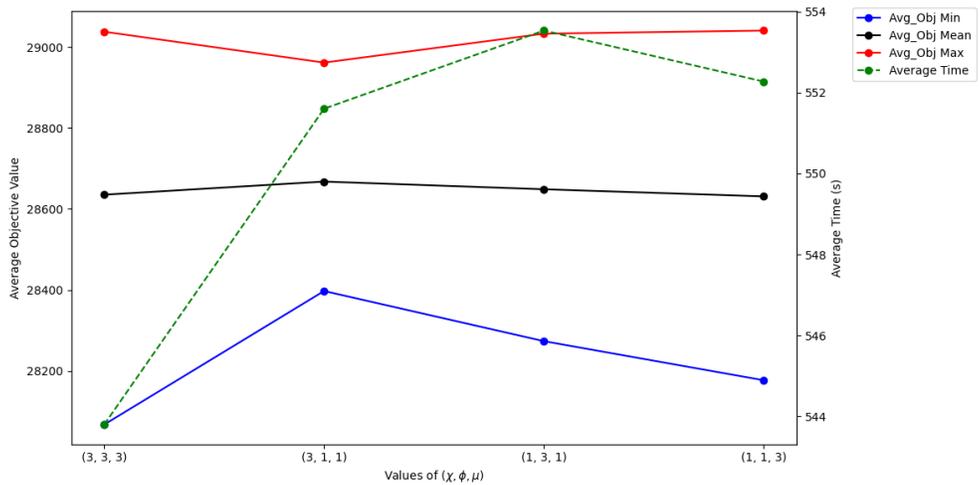
setting (3, 3, 3) was selected based on its performance.

We analyze the comparative performance of the exact insertion algorithm against the combined use of all insertion strategies within our matheuristic algorithm. Figure 6a and Figure 6b depict the evolution of solution quality across number of iterations under both configurations. In the exact insertion algorithm, the MILP formulation is solved at each iteration with the current partial solution held fixed, using a time limit of 0.5 seconds per solve. While this approach consistently outperforms the combined strategy in terms of solution quality, it does so at the cost of increased computational time. To assess the variability in solution quality, we computed relative gaps ( $\text{Gaprel}^{\max}$  and  $\text{Gaprel}^{\min}$ ) between the average objective value and the corresponding minimum and maximum values, aggregated across all test instances with a gap exceeding 1%, where  $\text{Gaprel}^{\max} = \frac{Z^{\max} - \bar{Z}}{\bar{Z}} \times 100\%$ ,  $\text{Gaprel}^{\min} = \frac{\bar{Z} - Z^{\min}}{\bar{Z}} \times 100\%$ ,  $\bar{Z}$  denotes the average objective value over multiple runs, and  $Z^{\min}$  and  $Z^{\max}$  are the minimum and maximum values observed over the same runs.

To enhance the robustness of the algorithm, the valid local branching inequalities of Fischetti and Lodi

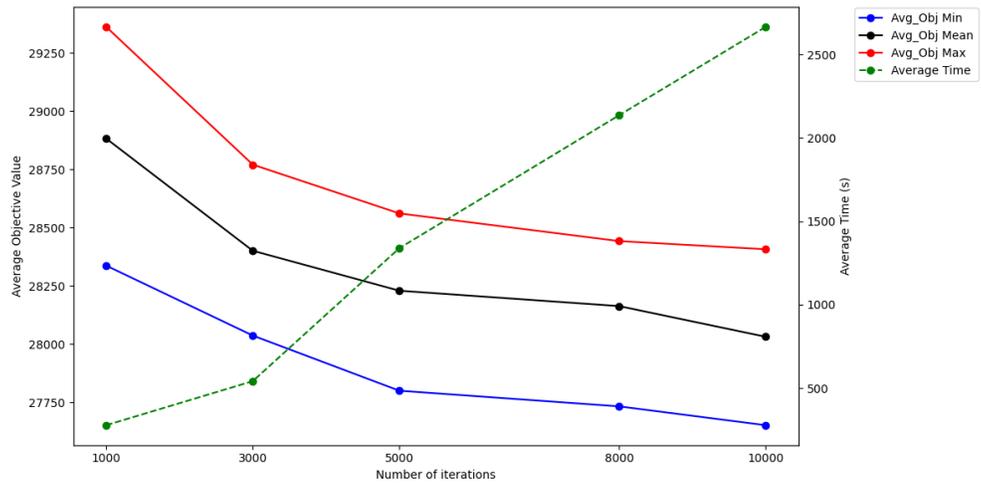


(a) Contribution of  $r$  in the solution quality.

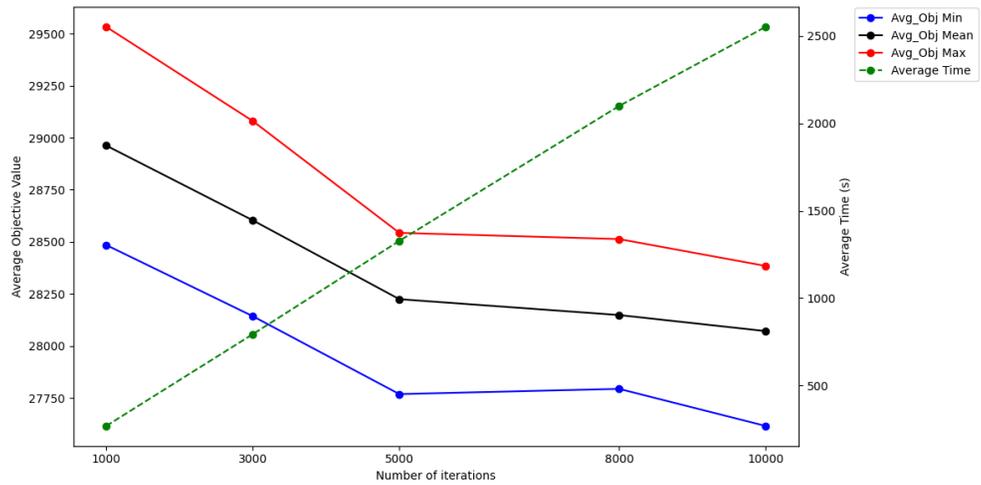


(b) Tuple  $(\chi, \phi, \mu)$  tuning.

Figure 5: Impact of parameter tuning on solution performance.



(a) Only exact insertion algorithm.



(b) All insertion algorithms.

Figure 6: Comparison between exact insertion and combined insertion strategies.

(2003) were incorporated into the model; however, they did not produce any significant improvement. As an alternative, we extended the initial time limit from 0.5 seconds ( Figure 6a) to 1.25 seconds and introduced a dynamic adjustment mechanism that increases the time limit whenever no improvement is observed after 50 iterations. While this approach improves the algorithm’s robustness, it also increases the computational time. Under this setting, the relative gaps are consistently below 1% (see Figure 7). Beyond 3000 iterations, the improvement in solution quality slows down significantly, as indicated by the narrowing gaps between successive iteration levels. Nevertheless, due to computational constraints, the stopping criteria for the LNS algorithm are fixed at 3000 iterations or a maximum runtime of 1800 seconds. Table 2 summarizes the parameter values selected for the final configuration of our algorithm.

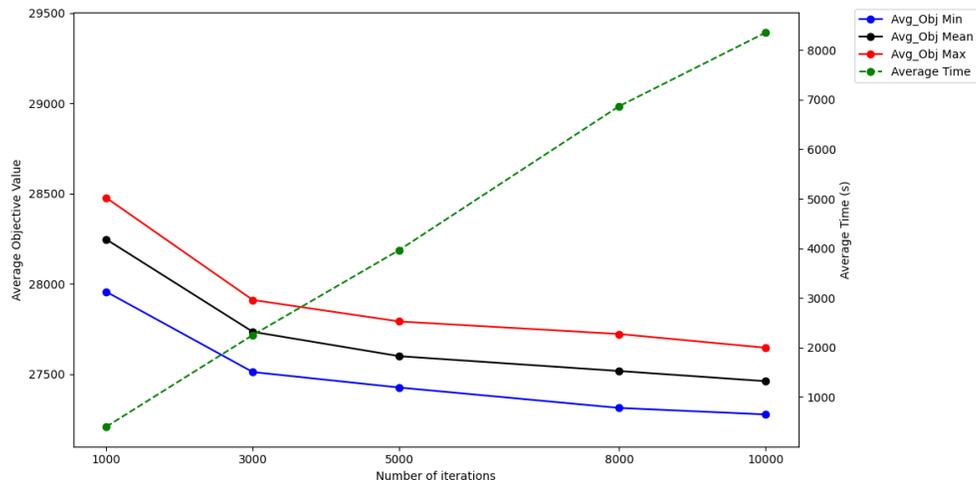


Figure 7: Effect of the number of iterations on the performance of the LNS heuristic.

Table 2: Matheuristic parameters values

Parameters	Description	Values
<b>Clustering</b>		
$\tau_s, s \in S$	Control the overproduction	0.07
<b>LNS</b>		
$(\chi, \phi, \mu, \sigma)$	Removal operator parameters	(3, 3, 3, 2)
$r$	Interval of the percentage of farms to remove	[10, 20]
LNS stopping criteria	Maximum number of iterations or runtime	3000 or 1800 seconds
Exact insertion time limit	maximum runtime of	1.25 seconds

### 6.3. Results for IPPDP on generated test instances

In our preliminary tests, we have solved model (7)–(15) with CPLEX without providing an initial solution. The final solution obtained by CPLEX at the end of its 1 h run was always consistently worse than the solutions of our matheuristic. To assess the quality of our algorithm, we provide our best solution to CPLEX as an initial solution. Our goal is to evaluate whether CPLEX can improve this solution and to compute reasonable optimality gaps. In the following tables, *Instance* indicates the instance details, while *Best sol.* and *Avg sol.* report, respectively, the best and average solutions obtained across the five runs using the matheuristic. The columns *Best time (s)* and *Avg time (s)* indicate the runtime (in seconds) required to

obtain the best solution and the average runtime across runs. Finally,  $BKS$  denotes the best-known solution value, obtained by passing the best solution from the matheuristic to CPLEX and running it for the time limit.  $Gap$  (%) refers to the optimality gap computed as  $100 \times \frac{UB - LB}{UB}$ , where  $LB$  and  $UB$  are the best lower and upper bounds, respectively. We follow Vidal (2022) and evaluate solution quality using the percentage improvement with respect to the BKS, defined as  $Improvement$  (%) =  $100 \times \frac{Z - Z^{BKS}}{Z^{BKS}}$ , where  $Z$  is the objective value obtained by the algorithm (either the best or the average) and  $Z^{BKS}$  denotes the best known solution for the considered instance, either the same as the best from the algorithms or the improved solution from CPLEX.

Table 3 contains the results of instances with a single slaughterhouse, where it is not necessary to select among multiple slaughterhouses to optimize transportation costs. This setup allows us to focus solely on the quality of the solution and computation time without the added complexity of choosing between slaughterhouses. While Table 4 presents the results for instances with two and three slaughterhouses, where the algorithm must make strategic decisions on slaughterhouse selection to minimize transportation costs, introducing additional complexity to the optimization process.

The results presented in Table 3 highlight the effectiveness of the matheuristic in solving instances with a single slaughterhouse. The matheuristic demonstrates consistently strong performance, with minimal variation between its best and average objective values, indicating robustness and stability. Notably, the difference between  $Gap_{best}$  and  $Gap_{avg}$  is often marginal, suggesting that the matheuristic reliably finds consistently good solutions across different runs. In contrast, for almost all instances,  $Gap_{best}$  equals zero, which shows that CPLEX could not improve the matheuristic solutions within the 3600-second time limit. Another significant advantage of the matheuristic is its computational efficiency, as it typically completes each instance within a few minutes, whereas CPLEX requires the full allotted time without necessarily reaching optimality. Overall, the matheuristic approach proves not only more efficient in terms of time but also more effective in maintaining solution quality across various problem sizes.

Table 4 shows the results for instances with two and three slaughterhouses, comparing the matheuristic and CPLEX. Consistent with the earlier analysis for single-slaughterhouse instances, the matheuristic approach consistently achieves similar performance with moderate computation times compared to CPLEX. These results highlight the efficiency of the matheuristic in handling more complex scenarios, offering a reliable and time-effective approach for large-scale instances, where both solution quality and computational speed are important. Furthermore, the length of the planning horizon plays a significant role in performance. As the planning horizons expand, the performance of CPLEX becomes limited by the combinatorial explosion of potential allocations across time, farms, and slaughterhouses. For longer planning horizons, such as in the 10-period configurations seen in these instances, the fixed runtime cap of CPLEX prevents it from exploring the solution space exhaustively, leading to suboptimality (e.g., gaps exceeding 10% in multiple instances). The matheuristic approach handles these extended planning periods with moderate runtime, achieving substantially more practical solutions. This adaptability highlights the matheuristic’s robustness across various planning scenarios.

#### 6.4. Industrial test instance

The primary objective of this test is twofold. First, we evaluate the benefits of our integrated optimization approach by comparing it with the company’s current planning practice. Using real-world data from our partner, we assess whether jointly optimizing production and transportation decisions yields measurable

Table 3: Results for instances with one slaughterhouse for IPPDP.

Instance	Matheuristic (MH)				CPLEX*		Improvement over MH	
	Best sol	Avg sol	Best time (s)	Avg time (s)	BKS sol	Gap (%)	over best (%)	over avg (%)
IPPDP-40-1-4-0	29916	29916	26	27	29916	14.39	0.00	0.00
IPPDP-40-1-4-1	25351	25351	26	26	25351	9.26	0.00	0.00
IPPDP-40-1-4-2	34953	34953	29	29	34953	3.79	0.00	0.00
IPPDP-40-1-4-3	19654	19654	21	21	19654	52.92	0.00	0.00
IPPDP-40-1-4-4	28295	28295	28	27	28295	7.32	0.00	0.00
IPPDP-50-1-5-0	17976	18012	54	41	17976	39.10	0.00	0.20
IPPDP-50-1-5-1	17222	17308	28	30	17222	34.61	0.00	0.50
IPPDP-50-1-5-2	18127	18283	30	32	18127	9.11	0.00	0.86
IPPDP-50-1-5-3	18276	18846	26	34	18276	32.68	0.00	3.12
IPPDP-50-1-5-4	21503	21565	44	43	21503	40.38	0.00	0.29
IPPDP-60-1-6-0	13575	13598	41	42	13553	17.09	0.16	0.33
IPPDP-60-1-6-1	13667	13703	41	40	13660	7.02	0.05	0.32
IPPDP-60-1-6-2	20490	20490	55	55	20490	28.10	0.00	0.00
IPPDP-60-1-6-3	19227	19227	50	51	19227	33.66	0.00	0.00
IPPDP-60-1-6-4	14165	14188	42	42	14165	9.93	0.00	0.17
IPPDP-70-1-7-0	12946	13160	58	58	12946	4.84	0.00	1.65
IPPDP-70-1-7-1	14857	15036	50	59	14857	4.50	0.00	1.21
IPPDP-70-1-7-2	14948	15050	55	57	14948	4.14	0.00	0.69
IPPDP-70-1-7-3	14801	14864	57	63	14801	4.54	0.00	0.43
IPPDP-70-1-7-4	15073	15148	50	55	15073	4.74	0.00	0.50
IPPDP-80-1-8-0	17084	17179	85	87	17084	5.21	0.00	0.55
IPPDP-80-1-8-1	16945	16996	96	87	16934	3.95	0.06	0.36
IPPDP-80-1-8-2	16690	16835	86	83	16690	4.50	0.00	0.87
IPPDP-80-1-8-3	17064	17193	84	100	17064	4.95	0.00	0.75
IPPDP-80-1-8-4	16742	16875	73	168	16742	4.09	0.00	0.79
IPPDP-90-1-9-0	18368	18436	128	128	18368	4.18	0.00	0.37
IPPDP-90-1-9-1	18270	18464	108	117	18270	4.78	0.00	1.06
IPPDP-90-1-9-2	18327	18402	116	316	18327	4.86	0.00	0.41
IPPDP-90-1-9-3	18296	18442	124	119	18296	3.66	0.00	0.80
IPPDP-90-1-9-4	18403	18419	118	369	18403	4.24	0.00	0.09
IPPDP-100-1-10-0	19149	19207	253	240	19149	7.08	0.00	0.30
IPPDP-100-1-10-1	19297	19622	227	269	19297	8.00	0.00	1.69
IPPDP-100-1-10-2	19316	19434	231	229	19316	8.09	0.00	0.61
IPPDP-100-1-10-3	19091	19223	199	246	19091	7.46	0.00	0.69
IPPDP-100-1-10-4	18647	18725	255	227	18647	8.33	0.00	0.42
Average	18763	18860	86	103	18762	12.73	0.01	0.57

\* Warm started with the best matheuristic solution

Table 4: Results for instances with two and three slaughterhouses IPPDP.

Instance	Matheuristic (MH)				CPLEX*		Improvement over MH	
	Best sol	Avg sol	Best time (s)	Avg time (s)	BKS sol	Gap (%)	over best (%)	over avg (%)
IPPDP-125-2-10-0	31418	31686	287	267	31418	31.18	0.00	0.85
IPPDP-125-2-10-1	29140	29274	321	261	29140	22.18	0.00	0.46
IPPDP-125-2-10-2	28296	28484	232	274	28296	28.35	0.00	0.66
IPPDP-125-2-10-3	31527	31765	302	288	31527	13.68	0.00	0.75
IPPDP-125-2-10-4	33363	33464	363	344	33347	10.54	0.05	0.35
IPPDP-150-2-10-0	25278	25438	556	485	25278	17.14	0.00	0.63
IPPDP-150-2-10-1	23612	23807	372	493	23612	22.12	0.00	0.82
IPPDP-150-2-10-2	25876	26104	313	331	25876	18.92	0.00	0.88
IPPDP-150-2-10-3	23498	23720	427	452	23498	20.56	0.00	0.94
IPPDP-150-2-10-4	24681	24879	466	479	24681	21.46	0.00	0.80
IPPDP-175-2-10-0	24029	24119	822	785	24029	13.42	0.00	0.37
IPPDP-175-2-10-1	21766	21888	671	997	21766	11.87	0.00	0.56
IPPDP-175-2-10-2	22491	22741	714	675	22491	16.92	0.00	1.11
IPPDP-175-2-10-3	22876	23022	752	855	22876	14.24	0.00	0.64
IPPDP-175-2-10-4	22815	22945	789	693	22815	14.79	0.00	0.57
IPPDP-200-2-10-0	20919	21173	1198	1194	20919	15.24	0.00	1.21
IPPDP-200-2-10-1	22418	22466	1397	1530	22418	13.26	0.00	0.21
IPPDP-200-2-10-2	25763	25820	1725	1626	25763	8.37	0.00	0.22
IPPDP-200-2-10-3	23316	23492	1686	1745	23307	9.03	0.04	0.79
IPPDP-200-2-10-4	25444	25598	1366	1380	25444	9.71	0.00	0.61
IPPDP-225-3-10-0	30561	30758	1805	1753	30561	16.56	0.00	0.64
IPPDP-225-3-10-1	32854	32971	1329	1485	32854	15.84	0.00	0.36
IPPDP-225-3-10-2	31378	31558	1804	1384	31378	19.15	0.00	0.57
IPPDP-225-3-10-3	29795	30143	1801	1765	29795	19.32	0.00	1.17
IPPDP-225-3-10-4	29273	29434	1712	1676	29273	21.56	0.00	0.55
IPPDP-250-3-10-0	28707	29197	1800	1801	28707	16.87	0.00	1.71
IPPDP-250-3-10-1	31362	31639	1803	1802	31362	15.45	0.00	0.88
IPPDP-250-3-10-2	33282	33457	1801	1801	33282	16.10	0.00	0.52
IPPDP-250-3-10-3	28653	28782	1801	1802	28653	15.75	0.00	0.45
IPPDP-250-3-10-4	28342	28613	1801	1802	28342	13.91	0.00	0.95
IPPDP-275-3-10-0	34621	34754	1803	1803	34621	14.06	0.00	0.38
IPPDP-275-3-10-1	39187	39427	1803	1802	39187	11.68	0.00	0.61
IPPDP-275-3-10-2	34063	34230	1807	1802	34063	11.17	0.00	0.49
IPPDP-275-3-10-3	32996	33105	1805	1804	32996	14.36	0.00	0.33
IPPDP-275-3-10-4	36865	37105	1801	1801	36865	8.69	0.00	0.65
IPPDP-300-3-10-0	36705	36893	1801	1802	36705	8.54	0.00	0.51
IPPDP-300-3-10-1	38548	38680	1805	1803	38548	10.97	0.00	0.34
IPPDP-300-3-10-2	39739	39905	1801	1805	39739	10.87	0.00	0.42
IPPDP-300-3-10-3	40674	40767	1801	1803	40674	7.52	0.00	0.23
IPPDP-300-3-10-4	40163	40536	1806	1804	40163	7.29	0.00	0.93
Average	29657	29846	1256	1256	29657	15.22	0.00	0.65

\* Warm started with the best matheuristic solution

economic gains. Second, we test the performance and scalability of our proposed solution algorithm on real world sized instances, thereby validating its practical applicability to large-scale industrial settings.

The company operates two slaughterhouses with daily processing capacities of 95,000 and 185,000 birds, respectively. Its supply base consists of 601 farms, each characterized by a postal code, production capacity (ranging from 2,800 to 48,500 birds), and growth parameters. Day-old chicks start at 380 dg and must reach a target slaughter weight of 20,875 dg, with daily growth rates ranging from 447 dg to 843 dg.

Current planning practice follows a simple proximity-based rule: each farm is permanently assigned to the slaughterhouse closest in geographic distance, based on distances computed from postal codes. All birds produced at a farm are systematically delivered to the assigned facility. On each slaughter day, direct collection routes are organized independently for each slaughterhouse. Under this policy, farm–slaughterhouse assignments are fixed in advance, and production and transportation decisions are made sequentially rather than jointly optimized.

The average distance between farms and the two slaughterhouses is 129 km and 114 km, respectively. These distances define the transportation cost structure within the integrated model. Unlike the current practice, our formulation allows dynamic farm–slaughterhouse assignments and explicitly captures the interaction between growth processes, capacity constraints, and transportation decisions. The objective is to minimize total system cost while satisfying weight and capacity requirements.

The computational results for the industrial instance are reported in Table 5. The matheuristic exhibits strong stability, as indicated by the small difference between *Gap best* (0.00%) and *Gap avg* (0.64%). This confirms the robustness of the proposed approach on real-world-sized data. To assess solution quality, we also ran CPLEX. Within the imposed time limit, CPLEX terminated with an optimality gap of 5.46%, without improving upon the best solution found by the matheuristic.

When comparing with the company’s current practice, the difference is substantial. The proximity-based policy results in a total cost 13.07% higher than the best solution obtained by the matheuristic. Even under conservative assumptions, given the 5.46% optimality gap, the company’s current practice remains significantly suboptimal. This confirms the structural value of integrating production and distribution decisions within a unified optimization framework.

Table 5: Computational results on the industrial test instance.

Instance	Matheuristic				Improvement over MH			
	Best sol	Avg sol	Best time (s)	Avg time (s)	BKS sol	Gap (%)	Gap best (%)	Gap avg (%)
IPPDP-601-3-10	13626	13713	1802	1802	13626	5.46	0.00	0.64
	Current practice							
	Best sol	Avg sol	Gap best (%)	Gap avg (%)				
IPPDP-601-3-10	15407	15519	13.07	13.89				

### 6.5. Sensitivity and robustness analysis

In the absence of standardized benchmark instances and given the industry-driven nature of the problem, we assess the robustness of the proposed model through a combination of sensitivity and scenario-based analyses. Rather than varying individual parameters in isolation, we examine how solution structure and

performance evolve under changes in the emphasis placed on transportation costs and under alternative modeling assumptions that reflect realistic operational practices.

To conduct the analysis, we generated twenty instances with two slaughterhouses and three farm configurations (15, 20, and 25 farms). Optimal solutions were obtained using both and our matheuristic approach.

### 6.5.1. Effect of transportation cost on production decision

We evaluate the impact of the total transportation cost on production, measured by the number of heads of poultry under- and over produced. To this end, we first minimize the transportation cost by assigning all farms to the nearest slaughterhouse, and then compare the results with those obtained from the integrated approach to highlight the trade-off between transportation efficiency and production balance. The results presented in Table 6, highlight the sensitivity of production outcomes to the structure of the optimization model. When transportation cost is minimized independently, production imbalances (both under- and over-production) remain high, indicating that proximity-based assignments are insufficient to ensure supply-demand balance. In contrast, while the integrated approach slightly increases the transportation cost, it significantly reduces production deviations, demonstrating the advantage of jointly optimizing transportation and production decisions.

Table 6: Comparison between distance minimized and integrated approaches.

B	Distance minimized				Integrated			
	OF	TC	Production (heads)		OF	TC	Production (heads)	
			Under	Over			Under	Over
15	64,835	1,505	60,475	2,854	17,435	2,337	8,228	6,870
20	75,669	2,003	70,292	3,374	11,295	2,974	5,465	2,856
25	71,606	3,324	67,088	1,194	4,837	3,637	597	604
Average	70,703	2,277	65,952	2,474	11,189	2,983	4,763	3,443

OF: Objective Function; TC: Transportation Cost

### 6.5.2. Impact of allowing multiple deliveries per farm

In model (7)–(15), based on company needs, we assumed that farms should deliver at most once during the planning horizon. In this section, we evaluate the effect of the rolling-horizon perspective. The mathematical formulation of this new model is derived from model (7)–(15) by replacing constraints (9) and (11) by (17) and (19) respectively and adding (20). To the parameters of the model (7)–(15), we add  $\varepsilon$  as the duration of the sanitation period. We solved this new formulation using our matheuristic. We also modified the clustering algorithm to allow farms to have multiple batches. So, if all farms are assigned and the set  $V$  is not empty, we reset the set of farms  $B$ . Note that in line 10 of Algorithm (3) all occurrences of each farm in  $E$  are removed from the solution  $CL$ . The new model is shown below.

$$\min \sum_{b \in B} \sum_{s \in S} \sum_{t \in V} \bar{c} d_{bs} r_b^{st} + \sum_{b \in B} C_b \sum_{u \in U} \sum_{i=1}^2 \left( g_i \sum_{t \in H_b^{ui}} \bar{w}_b^{ut} y_b^{ut} \right) + \sum_{s \in S} \sum_{t \in V} (p^+ q_{st}^+ + p^- q_{st}^-) \quad (16)$$

Subject to

$$\sum_{s \in S} r_b^{st} \leq 1 \quad b \in B, t \in V \quad (17)$$

$$(8), (10), (12) - (15) \quad (18)$$

$$\sum_{t \in H_b^u} y_b^{ut} + \sum_{u'=u+1}^{\min H_b^u + \varepsilon} \sum_{t \in H_b^{u'}} y_b^{u't} \leq 1 \quad b \in B, u \in U \quad (19)$$

$$\sum_{t \in H_b^n} y_b^{nt} + \sum_{l \in H_b^k} y_b^{kl} \leq 1 \quad \begin{matrix} b \in B, n, k \in V: \\ (k > n, k \leq t + \varepsilon) \end{matrix} \quad (20)$$

Tables A.7 and A.8 (see appendix) present the results of the matheuristic. We observe that for several instances, some farms delivered twice. This additional flexibility leads to a total cost reduction of approximately \$919 ( $\approx 4.9\%$ ) for single-slaughterhouse instances, and by \$742 ( $\approx 2.5\%$ ) for instances with two or three slaughterhouses. Consequently, if the company's objective is to minimize total cost regardless of the specific farms selected, the new model represents the best option.

## 7. Conclusion

This paper addresses a complex real-world problem in integrated poultry production and distribution for a company that manages multiple slaughterhouses, each operating under strict capacity and daily production quotas. Compliance with these quotas is important, as deviations, whether excess or shortfall, incur high additional costs. Poultry is sourced from multiple farms, where transportation distances play a key role in overall profitability. In addition, poultry must meet specific weight criteria to qualify for premium markets; otherwise, they are sold in secondary markets at a lower value, directly affecting revenue potential. These challenges are further compounded in Québec by the quota system regulating poultry production at the slaughterhouse level.

To address this problem, we formulated a MILP model that integrates production and distribution decisions while explicitly accounting for quota compliance, weight deviations, and transportation costs. While the proposed formulation aggregates transportation costs and deviations penalties costs into a single objective that reflects industrial practice, the problem naturally admits multi-objective extensions that explicitly trade off transportation cost against quota and weight deviation penalties. A neighborhood-search-based matheuristic was developed to solve realistically sized problem instances. Computational experiments on a diverse set of generated instances with varying planning horizons and delivery configurations demonstrate that the proposed approach produces high-quality solutions in reasonable computational time. Benchmarking against a commercial solver further illustrates the scalability of the proposed method for industry-relevant problem sizes.

Several directions for future research emerge from this work. One promising extension is the integration of robust or stochastic optimization techniques to better capture real-world uncertainties, such as variability in growth rates, mortality, or demand conditions. Another avenue is to incorporate explicit transportation routing decisions between farms and slaughterhouses, thereby further enhancing the realism of the distribution

component. Additional extensions include modeling upstream stages, such as hatchery operations, to develop a fully integrated end-to-end poultry supply chain, and considering multiple poultry breeds or categories to reflect more diverse production systems. in line with our industrial partner’s current practice. Alternative production policies could be explored in future research.

## Acknowledgments

This work was partly funded by the Canadian Natural Sciences and Engineering Research Council (NSERC) under grants 2025-04195, 2023-04167, and 2025-03964. The authors thank Compute Canada for providing high-performance parallel computing facilities. This support is gratefully acknowledged. We thank three anonymous referees for their valuable comments on an earlier version of this paper.

## References

- Absi, N., Archetti, C., Dauzère-Pérès, S., and Feillet, D. (2015). A two-phase iterative heuristic approach for the production routing problem. *Transportation Science*, 49(4):784–795.
- Adulyasak, Y., Cordeau, J.-F., and Jans, R. (2014). Optimization-based adaptive large neighborhood search for the production routing problem. *Transportation Science*, 48(1):20–45.
- Agriculture and Agri-Food Canada (2024a). Canada’s poultry and egg industry profile. <https://agriculture.canada.ca/en/sector/animal-industry/poultry-and-egg-market-information/industry-profile>. Accessed: February 9, 2025.
- Agriculture and Agri-Food Canada (2024b). Canadian agriculture at a glance: Sector overview. <https://agriculture.canada.ca/en/sector/overview>. Accessed: July 9, 2025.
- Amorim, P., Belo-Filho, M., Toledo, F. M. B. d., Almeder, C., and Almada-Lobo, B. (2013). Lot sizing versus batching in the production and distribution planning of perishable goods. *International Journal of Production Economics*, 146(1):208–218.
- Archetti, C., Bertazzi, L., Paletta, G., and Speranza, M. G. (2011). Analysis of the maximum level policy in a production-distribution system. *Computers & Operations Research*, 38(12):1731–1746.
- Arda, Y., Cattaruzza, D., François, V., and Ogier, M. (2024). Home chemotherapy delivery: An integrated production scheduling and multi-trip vehicle routing problem. *European Journal of Operational Research*, 317(2):468–486.
- Bard, J. F. and Nananukul, N. (2009). The integrated production–inventory–distribution–routing problem. *Journal of Scheduling*, 12(3):257–280.
- Bard, J. F. and Nananukul, N. (2010). A branch-and-price algorithm for an integrated production and inventory routing problem. *Computers & Operations Research*, 37(12):2202–2217.
- Belo-Filho, M. A. F., Amorim, P., and Almada-Lobo, B. (2015). An adaptive large neighbourhood search for the operational integrated production and distribution problem of perishable products. *International Journal of Production Research*, 53(20):6040–6058.

- Berghman, L., Kergosien, Y., and Billaut, J.-C. (2023). A review on integrated scheduling and outbound vehicle routing problems. *European Journal of Operational Research*, 311(1):1–23.
- Boonmee, A. and Sethanan, K. (2016). A GLNPSO for multi-level capacitated lot-sizing and scheduling problem in the poultry industry. *European Journal of Operational Research*, 250(2):652–665.
- Boonmee, A., Sethanan, K., Arnonkijpanich, B., and Theerakulpisut, S. (2015). Minimizing the total cost of hen allocation to poultry farms using Hybrid Growing Neural Gas approach. *Computers and Electronics in Agriculture*, 110:27–35.
- Boudia, M., Louly, M. A. O., and Prins, C. (2006). A memetic algorithm with population management for a production-distribution problem. *IFAC Proceedings Volumes*, 39(3):541–546.
- Boudia, M., Louly, M. A. O., and Prins, C. (2007). A reactive grasp and path relinking for a combined production–distribution problem. *Computers & Operations Research*, 34(11):3402–3419.
- Brahimi, N. and Aouam, T. (2016). Multi-item production routing problem with backordering: a MILP approach. *International Journal of Production Research*, 54(4):1076–1093.
- Brevik, E., Lauen, A. Ø., Rolke, M. C., Fagerholt, K., and Hansen, J. R. (2020). Optimisation of the broiler production supply chain. *International Journal of Production Research*, 58(17):5218–5237.
- Chagas, G. O., Coelho, L. C., Darvish, M., and Renaud, J. (2023). Modeling and solving the waste valorization production and distribution scheduling problem. *European Journal of Operational Research*, 306(1):400–417.
- Chandra, P. and Fisher, M. L. (1994). Coordination of production and distribution planning. *European Journal of Operational Research*, 72(3):503–517.
- Cheng, B., Leung, J. Y.-T., Li, K., and Yang, S. (2019). Integrated optimization of material supplying, manufacturing, and product distribution: Models and fast algorithms. *European Journal of Operational Research*, 277(1):100–111.
- Darvish, M. and Coelho, L. C. (2018). Sequential versus integrated optimization: Production, location, inventory control, and distribution. *European Journal of Operational Research*, 268(1):203–214.
- Darvish, M., Kidd, M. P., Coelho, L. C., and Renaud, J. (2021). Integrated production-distribution systems: Trends and perspectives. *Pesquisa Operacional*, 41:e246080.
- Darvish, M., Larrain, H., and Coelho, L. C. (2016). A dynamic multi-plant lot-sizing and distribution problem. *International Journal of Production Research*, 54(22):6707–6717.
- Devapriya, P., Ferrell, W., and Geismar, N. (2017). Integrated production and distribution scheduling with a perishable product. *European Journal of Operational Research*, 259(3):906–916.
- Elhedhli, S. and Goffin, J.-L. (2005). Efficient production-distribution system design. *Management Science*, 51(7):1151–1164.

- Eskandarpour, M., Dejax, P., and Péton, O. (2017). A large neighborhood search heuristic for supply chain network design. *Computers & Operations Research*, 80:23–37.
- Fischetti, M. and Lodi, A. (2003). Local branching. *Mathematical Programming*, 98:23–47.
- Hwang, H.-C., Ahn, H.-S., and Kaminsky, P. (2016). Algorithms for the two-stage production-capacitated lot-sizing problem. *Journal of Global Optimization*, 65(4):777–799.
- Jia, X., Li, F., Gao, X., and Yang, X. (2026). Integrated production and transportation rescheduling with type-dependent setup times and multiple shipping modes. *European Journal of Operational Research*.
- Karaođlan, İ. and Kesen, S. E. (2017). The coordinated production and transportation scheduling problem with a time-sensitive product: a branch-and-cut algorithm. *International Journal of Production Research*, 55(2):536–557.
- Lei, L., Liu, S., Ruszczyński, A., and Park, S. (2006). On the integrated production, inventory, and distribution routing problem. *IIE Transactions*, 38(11):955–970.
- Martello, S. (1981). An algorithm for the generalized assignment problem. *Operational Research*, pages 589–603.
- Masson, R., Lehuédé, F., and Péton, O. (2013). An adaptive large neighborhood search for the pickup and delivery problem with transfers. *Transportation Science*, 47(3):344–355.
- Ministère de l’Agriculture, des Pêcheries et de l’Alimentation du Québec (2011). Monographie de l’industrie de la volaille au Québec. Technical report, Ministère de l’Agriculture, des Pêcheries et de l’Alimentation du Québec (MAPAQ), Québec, Canada.
- Ministère de l’Agriculture, des Pêcheries et de l’Alimentation du Québec (MAPAQ) (2024). Produit intérieur brut (PIB) de l’industrie bioalimentaire au québec. <https://www.mapaq.gouv.qc.ca/fr/md/statistiques/Pages/pib.aspx>. Accessed: July 9, 2025.
- Mundler, P. and Ubertino, S. (2022). Who gets to farm? agricultural renewal, quotas, and the governance of alternative food networks in quebec. *Canadian Journal of Regional Science*, 45(1):38–47.
- Neves-Moreira, F., Almada-Lobo, B., Cordeau, J.-F., Guimarães, L., and Jans, R. (2019). Solving a large multi-product production-routing problem with delivery time windows. *Omega*, 86:154–172.
- Pisinger, D. and Ropke, S. (2018). Large neighborhood search. In *Handbook of metaheuristics*, pages 99–127. Springer.
- Ropke, S. and Pisinger, D. (2006). An adaptive large neighborhood search heuristic for the pickup and delivery problem with time windows. *Transportation Science*, 40(4):455–472.
- Senoussi, A., Dauzère-Pérès, S., Brahimi, N., Penz, B., and Mouss, N. K. (2018). Heuristics based on genetic algorithms for the capacitated multi vehicle production distribution problem. *Computers & Operations Research*, 96:108–119.

- Shaw, P. (1998). Using constraint programming and local search methods to solve vehicle routing problems. In *International conference on principles and practice of constraint programming*, pages 417–431. Springer.
- Shi, Y., Liu, W., and Zhou, Y. (2023). An adaptive large neighborhood search based approach for the vehicle routing problem with zone-based pricing. *Engineering Applications of Artificial Intelligence*, 124:106506.
- Solano-Blanco, A. L., González, J. E., Gómez-Rueda, L. O., Vargas-Sánchez, J. J., and Medaglia, A. L. (2023a). Integrated planning decisions in the broiler chicken supply chain. *International Transactions in Operational Research*, 30(4):1931–1954.
- Solano-Blanco, A. L., González, J. E., and Medaglia, A. L. (2023b). Production planning decisions in the broiler chicken supply chain with growth uncertainty. *Operations Research Perspectives*, 10:100273.
- Solyalı, O. and Süral, H. (2017). A multi-phase heuristic for the production routing problem. *Computers & Operations Research*, 87:114–124.
- Tahraoui, N., Sari, L. T., and Bennekrouf, M. (2020). Planning and synchronization of broiler production in a poultry network. In *2020 IEEE 13th International Colloquium of Logistics and Supply Chain Management (LOGISTIQUA)*, pages 1–6. IEEE.
- Vidal, T. (2022). Hybrid genetic search for the cvrp: Open-source implementation and swap\* neighborhood. *Computers & Operations Research*, 140:105643.

## Appendix A.

Table A.7: Results for instances with one slaughterhouse for the variant of IPPDP.

Instance	Matheuristic (MH)				CPLEX*		Improvement over MH	
	Best sol	Avg sol	Best time (s)	Avg time (s)	BKS sol	Gap (%)	over best (%)	over avg (%)
IPPDP-40-1-4-0	29916	29916	37	47	29916	10.01	0.00	0.00
IPPDP-40-1-4-1	25351	25351	61	54	25351	9.61	0.00	0.00
IPPDP-40-1-4-2	34953	34953	61	55	34953	4.71	0.00	0.00
IPPDP-40-1-4-3	19654	19654	50	48	19654	52.18	0.00	0.00
IPPDP-40-1-4-4	28295	28295	36	44	28295	10.05	0.00	0.00
IPPDP-50-1-5-0	17988	18485	101	71	17988	34.29	0.00	2.76
IPPDP-50-1-5-1	17094	17271	195	92	17094	29.83	0.00	1.03
IPPDP-50-1-5-2	18135	18212	49	53	18135	8.34	0.00	0.43
IPPDP-50-1-5-3	18407	19008	51	63	18407	28.45	0.00	3.26
IPPDP-50-1-5-4	21503	21550	60	59	21503	6.72	0.00	0.22
IPPDP-60-1-6-0	11388	11544	117	133	11388	9.74	0.00	1.37
IPPDP-60-1-6-1	12186	12434	98	117	12186	5.29	0.00	2.04
IPPDP-60-1-6-2	13291	13528	128	115	13291	6.32	0.00	1.78
IPPDP-60-1-6-3	13832	14181	104	127	13832	6.43	0.00	2.52
IPPDP-60-1-6-4	12215	12323	196	121	12215	4.08	0.00	0.88
IPPDP-70-1-7-0	13024	13220	136	144	13024	5.40	0.00	1.50
IPPDP-70-1-7-1	15001	15035	129	145	15001	5.40	0.00	0.23
IPPDP-70-1-7-2	15019	15078	124	135	15019	4.62	0.00	0.39
IPPDP-70-1-7-3	14662	14945	146	186	14662	3.60	0.00	1.93
IPPDP-70-1-7-4	15145	15205	128	286	15145	5.19	0.00	0.39
IPPDP-80-1-8-0	16513	16636	294	277	16513	6.21	0.00	0.74
IPPDP-80-1-8-1	16204	16326	309	308	16204	4.97	0.00	0.75
IPPDP-80-1-8-2	15648	15924	276	305	15648	5.31	0.00	1.76
IPPDP-80-1-8-3	16184	16241	432	458	16184	5.87	0.00	0.35
IPPDP-80-1-8-4	16086	16205	428	307	16086	6.26	0.00	0.74
IPPDP-90-1-9-0	17587	17643	557	524	17587	6.91	0.00	0.32
IPPDP-90-1-9-1	17157	17501	427	425	17157	5.19	0.00	2.01
IPPDP-90-1-9-2	17431	17547	380	458	17431	6.65	0.00	0.67
IPPDP-90-1-9-3	17363	17549	410	445	17363	5.21	0.00	1.07
IPPDP-90-1-9-4	17350	17525	549	454	17350	5.53	0.00	1.01
IPPDP-100-1-10-0	18064	18138	1205	1005	18064	11.28	0.00	0.41
IPPDP-100-1-10-1	18451	18544	699	1095	18451	15.11	0.00	0.50
IPPDP-100-1-10-2	18428	18460	713	801	18428	11.16	0.00	0.17
IPPDP-100-1-10-3	18155	18270	811	780	18155	12.70	0.00	0.63
IPPDP-100-1-10-4	16877	17072	1694	1244	16877	14.21	0.00	1.16
Average	17844	17993	320	314	17844	10.65	0.00	0.94

\* Warm started with the best matheuristic solution

Table A.8: Results for instances with two and three slaughterhouses for the variant of IPPDP.

Instance	Matheuristic (MH)				CPLEX*		Improvement over MH	
	Best sol	Avg sol	Best time (s)	Avg time (s)	BKS sol	Gap (%)	over best (%)	over avg (%)
IPPDP-125-2-10-0	30457	30679	1088	878	30457	28.25	0.00	0.73
IPPDP-125-2-10-1	28011	28187	677	795	28011	21.48	0.00	0.63
IPPDP-125-2-10-2	26785	27079	1801	1119	26785	29.37	0.00	1.10
IPPDP-125-2-10-3	28614	28946	803	925	28614	13.45	0.00	1.16
IPPDP-125-2-10-4	29858	30096	815	830	29858	13.70	0.00	0.80
IPPDP-150-2-10-0	24777	24897	1801	1763	24777	17.72	0.00	0.49
IPPDP-150-2-10-1	23076	23217	1803	1783	23076	21.71	0.00	0.61
IPPDP-150-2-10-2	25068	25239	1015	1145	25019	18.48	0.20	0.88
IPPDP-150-2-10-3	22947	23113	1801	1648	22947	21.18	0.00	0.73
IPPDP-150-2-10-4	24027	24263	1734	1674	24027	22.04	0.00	0.98
IPPDP-175-2-10-0	23232	23386	1801	1803	23232	14.23	0.00	0.66
IPPDP-175-2-10-1	21219	21406	1802	1802	21219	12.85	0.00	0.88
IPPDP-175-2-10-2	22234	22360	1801	1801	22234	19.70	0.00	0.57
IPPDP-175-2-10-3	22193	22387	1802	1779	22193	15.95	0.00	0.87
IPPDP-175-2-10-4	22346	22616	1800	1800	22346	16.61	0.00	1.21
IPPDP-200-2-10-0	20791	21069	1801	1802	20791	18.85	0.00	1.34
IPPDP-200-2-10-1	21842	22280	1805	1802	21842	15.58	0.00	2.01
IPPDP-200-2-10-2	25037	25229	1801	1802	25037	10.42	0.00	0.77
IPPDP-200-2-10-3	22915	23122	1800	1801	22915	12.30	0.00	0.90
IPPDP-200-2-10-4	25044	25205	1809	1805	25044	12.82	0.00	0.64
IPPDP-225-3-10-0	30178	30307	1802	1802	30178	19.63	0.00	0.43
IPPDP-225-3-10-1	32658	32924	1801	1801	32525	18.15	0.41	1.23
IPPDP-225-3-10-2	31125	31256	1804	1805	31125	20.18	0.00	0.42
IPPDP-225-3-10-3	29658	29830	1801	1804	29658	22.82	0.00	0.58
IPPDP-225-3-10-4	28359	28614	1802	1806	28359	22.80	0.00	0.90
IPPDP-250-3-10-0	28932	29169	1811	1803	28932	21.66	0.00	0.82
IPPDP-250-3-10-1	31499	31622	1806	1804	31499	19.09	0.00	0.39
IPPDP-250-3-10-2	32720	33200	1801	1803	32720	18.01	0.00	1.47
IPPDP-250-3-10-3	28226	28393	1801	1804	28226	19.58	0.00	0.59
IPPDP-250-3-10-4	28276	28412	1804	1803	28276	18.96	0.00	0.48
IPPDP-275-3-10-0	33931	34156	1806	1804	33931	15.82	0.00	0.66
IPPDP-275-3-10-1	39136	39348	1802	1804	39136	15.02	0.00	0.54
IPPDP-275-3-10-2	33036	33275	1802	1803	33036	13.93	0.00	0.72
IPPDP-275-3-10-3	31890	32416	1801	1804	31890	16.20	0.00	1.65
IPPDP-275-3-10-4	35835	36210	1805	1804	35835	11.05	0.00	1.05
IPPDP-300-3-10-0	35966	36282	1803	1804	35966	12.35	0.00	0.88
IPPDP-300-3-10-1	38080	38309	1804	1803	38080	13.86	0.00	0.60
IPPDP-300-3-10-2	38182	38452	1805	1803	38182	12.77	0.00	0.71
IPPDP-300-3-10-3	38986	39281	1807	1804	38986	9.48	0.00	0.76
IPPDP-300-3-10-4	39442	39684	1805	1803	39442	10.94	0.00	0.61
Average	28915	29148	1686	1666	28910	17.22	0.02	0.84

\* Warm started with the best matheuristic solution