European Journal of Operational Research 303 (2022) 1403–1416

Contents lists available at ScienceDirect

European Journal of Operational Research

journal homepage: www.elsevier.com/locate/ejor

Innovative Applications of O.R.

Vehicle routing for milk collection with gradual blending: A case arising in Chile

Germán Paredes-Belmar[a], Elizabeth Montero[b], Armin Lüer-Villagra[b], Vladimir Marianov[c,*], Claudio Araya-Sassi[d]

[a] School of Industrial Engineering, Pontificia Universidad Católica de Valparaíso, Chile
[b] Department of Engineering Sciences, Universidad Andres Bello, Chile
[c] Department of Electrical Engineering, Pontificia Universidad Católica de Chile and Instituto Sistemas Complejos de Ingeniería (ISCI), Chile
[d] Instituto de Ingeniería Industrial y Sistemas, Universidad Austral de Chile, Chile

ARTICLE INFO

Article history:
Received 11 January 2021
Accepted 28 March 2022
Available online 1 April 2022

Keywords:
Routing
Metaheuristics
Milk collection
Gradual blending
Iterated local search

ABSTRACT

We introduce and solve a new multi-commodity Vehicle Routing Problem, motivated by a case study of milk collection in Chile. Different grades of raw milk are collected from a number of farms scattered over a large area and transported to a single plant, allowing milk blending at the trucks. Previous works allow blending different grades of milk in the trucks, but the resulting blend is classified as its worst grade component, even if there was a single drop of it in the blend. The novelty of our work is the use of a less conservative *gradual blending* rule that associates milk grades to ranges of somatic cell count per milliliter, resulting in a more accurate classification of milk. The volume and somatic cell count per milliliter of the milk produced at each farm are known before the collection, and the farms are paid accordingly. The problem is to route a heterogeneous fleet of vehicles to maximize total profit at the plant, i.e., the revenue from milk minus route cost. All the milk is collected and gradual blending is applied. We propose a mixed integer linear programming formulation and solve the problem using a branch-and-cut method for small instances, and an Iterated Local Search metaheuristic for real-size instances. Both are applied to a large set of standard instances and a real case in Chile. Our results show an increase in profit over previous milk collection strategies, of 20% and 28% for test instances and up to 30% for a real instance.

1. Introduction

Transportation costs of raw milk from farms to processing plants account for a significant percentage of total costs of milk products, ranging from 10 to 30%, depending on the data source (CEPAL, 1998; FAO, 2012; 2019; ODEPA, 2018). Furthermore, transportation costs are at a steady increase. Subsequently, a milk-producing company's essential goal is their reduction (Butler, Herlihy, & Keenan, 2005; Caria, Todde, & Pazzona, 2018). Specifically, milk collection involves the following issues: i) the farms are spread over extensive rural areas with few access roads; ii) the milk production volume can be highly variable between different farms and seasons; and, iii) the milk quality is not the same for all farms, and it is classified in grades, according to the range in which falls its count of somatic cells per milliliter of milk (SC per milliliter). For example, Chilean milk production is carried out by around 6000 farms scattered in about 7600 square kilometers. About 20 big processing plants purchase the produced milk (Consorcio Lechero de Chile, 2015). Milk is produced in daily ranges between 50 and 30,000 liters per farm.

The milk quality is an essential concern for milk farms and processing plants (Paredes-Belmar, Marianov, Bronfman, Obreque, & Lüer-Villagra, 2016; Prasertsri & Kilmer, 2004). The milk quality is defined by a set of characteristics such as its nutritive value, SC per milliliter of milk, concentration of micro-bacteria, proteins, lactose, fat, etc. FAO (2012). In most countries, the quality of milk is regulated, based mainly on the count of SC per milliliter of milk, and classified in grades. Each country has its legislation. The Chilean law defines three grades of milk: i) Grade A (highest), with less than 500,000 SC per milliliter of milk; ii) Grade B, with more than 500,000 SC per milliliter and less than 1,000,000 SC per milliliter; and Grade C (lowest) with more than 1,000,000 SC per milliliter (Ministerio de Agricultura, 1979). The regulation in the United States of America classifies the milk into two grades: Grade A with less than 750,000 SC per milliliter, and Grade B (or non-

* Corresponding author.
E-mail addresses: german.paredes@pucv.cl (G. Paredes-Belmar), elizabeth.montero@unab.cl (E. Montero), armin.luer@unab.cl (A. Lüer-Villagra), marianov@ing.puc.cl (V. Marianov), claudio.araya01@uach.cl (C. Araya-Sassi).

https://doi.org/10.1016/j.ejor.2022.03.050
0377-2217/

grade A milk) with more than 750,000 SC per milliliter (United States Department of Agriculture, 2018). The European Union, Australia, New Zealand, and Canada define a maximum of 400,000 SC per milliliter for grade A (More, Clegg, Lynch, & O'Grady, 2013). The count of SC per milliliter also defines the payment per milk liter to each farm (Sebastino, Uribe, & González, 2020).

Milk collection problems with blending have been studied previously in the literature (Paredes-Belmar, Lüer-Villagra, Marianov, Cortés, & Bronfman, 2017; Paredes-Belmar et al., 2016; Villagrán, Montero, & Paredes-Belmar, 2020). In those papers, the blending rules are very conservative: a single drop of Grade C milk blended with a whole tank of Grade A milk would transform the whole milk in the tank into Grade C, which is unrealistic for practical purposes. Our goal is to model realistically the milk blending process more and solve the resulting challenging models.

As a first contribution, we extend the work of Paredes-Belmar et al. (2016), which we call here the Discrete Milk Blending (DMB), by introducing a new milk collection problem that considers gradual milk blending (GMB), motivated by a real case in southern Chile. A milk processing plant collects milk from farms using a single compartment truck fleet. Each farm produces one of the three qualities of milk, known before its collection. All the produced milk is collected because of the cooperative business structure of the company in our real case. Milk with different qualities can be blended in the same truck if convenient for the plant's profit. The SC per milliliter of milk in each truck is kept count of, which depends on the volume and SC per milliliter count of the milk collected from each one of the farms visited by that truck. This is the main difference with previously published milk collection routing strategies, in which milk is either not blended or blended with *discrete results*, i.e., the resulting grade in each truck is equal to the lowest grade component in the blend (the DMB strategy). By using GMB, we improve the profit by between 20% and 30% as compared to DMB (Paredes-Belmar et al., 2016).

Our second contribution is a mixed-integer linear programming (MILP) model for the GMB problem and a branch-and-cut (B&C) procedure suitable for solving small-size instances optimally.

Our third contribution is an Iterated Local Search metaheuristic (ILS) to solve real-size instances.

Note that the GMB problem is NP-Hard since, for a single milk grade, the problem is reduced to the capacitated vehicle routing problem (CVRP), which is already NP-Hard (Irnich, Toth, & Vigo, 2014).

The article is organized as follows. Section 2 provides an up-to-date literature review about milk transportation problems. Section 3 describes the milk collection problem with gradual blending. Section 4 presents the MILP model together with the branch-and-cut solving procedure and the ILS heuristic. Section 5 is devoted to the results, and Section 6 presents the conclusions and future research.

2. Literature review

The milk collection problem has been widely studied in specialized literature. Sankaran & Ubgade (1994) propose and solve a milk collection problem in India, where construction and improvement heuristics are implemented in a decision support system (DSS). Butler, Williams, & Yarrow (1997) implement a branch-and-cut algorithm to solve a two-period milk collection problem. Basnet, Foulds, & Wilson (1999) study a milk collection problem in New Zealand, defining a set of routes to be assigned to a set of vehicles using an integer programming model. Also, there is a limited number of pumps that delay the departure of trucks. Prasertsri & Kilmer (2004) develop an insertion heuristic for route construction to solve a milk collection problem in Florida, USA, considering time windows. Hoff & Løkketangen (2007) address a milk collection problem considering trucks and trailers. Palhazi Cuervo & Hendriks (2007) solve a periodic milk collection problem, addressing different collection patterns to minimize the surplus or shortage of milk. Mumtaz, Jalil, & Chatha (2014) present a location and routing model to solve a collection problem, proposing the location of dispatch points and optimizing milk collection routes. Dayarian, Crainic, Gendreau, & Rei (2015a) develop a branch-and-price algorithm for a milk collection problem, considering a heterogeneous fleet and multiple depots and time windows. Dayarian, Crainic, Gendreau, & Rei (2015b) and Dayarian, Crainic, Gendreau, & Rei (2016) study a multi-period vehicle routing problem with seasonal fluctuations using exact and approximated methods, respectively. A real milk collection instance is solved in Quebec, Canada. Masson, Lahrichi, & Rousseau (2015) solve a dairy transportation problem considering weekly variation in milk production using an adaptive large neighborhood search algorithm, which is applied to solve instances of up to 229 producers. O'Callaghan, O'Connor, & Goulding (2018) solve instances up to 50 farms of a milk collection problem in Ireland, considering seasonal variations. Huang, Wu, & Ardiansyah (2019) study a milk collection problem in Indonesia. The authors propose a stochastic route selection model to manage the milk collection from farms and the delivery to several processing plants, solving instances of up to 50 farms. Caria et al. (2018) develop a DSS for a sheep milk collection problem in Sardinia, Italy. The objective is to minimize transportation costs and to minimize carbon emissions. They solve an instance of 187 farms. Chokanat, Pitakaso, & Sethanan (2019) study a homogeneous (one-grade) milk collection problem using trucks with compartments, in which the milk from at most two farms is mixed in the same compartment, to facilitate quality control. An evolutionary algorithm is developed to solve random instances of up to 40 farms. Polat & Topaloğlu (2021) present a milk collection problem using trucks with compartments. In their study, there is one grade of milk and uncertainty in milk production. An Enhanced Iterated Local Search is implemented for solving a real case in Turkey. Montero, Canales, Paredes-Belmar, & Soto (2019) solve a milk collection problem in Chile, considering that a processing plant purchases a certain volume of milk for daily operation. Thus, not all the milk produced by the farms requires being collected. All described papers study the milk collection with a single grade or type of milk.

Only a few papers address the milk collection problem with different grades or types of milk. Dooley, Parker, & Blair (2005) solve a real problem in New Zealand, considering a segregated collection for the two milk types and implementing a genetic algorithm to determine the collection schedule. Caramia & Guerriero (2010) study a milk collection problem in Sardinia, Italy, using trucks with compartments and trailers. A truck travels with a trailer that can be unmounted in a parking site to allow the truck to visit those farms with difficult access for a truck with a trailer. After the visit, the truck returns to the parking site, where the trailer is mounted back. Lahrichi, Crainic, Gendreau, Rei, & Rousseau (2015) study a collection problem in Canada, considering different depots, heterogeneous fleet, collections, and delivery. Their problem has three types of milk that are not allowed to be blended in the same compartment. Sethanan & Pitakaso (2016) solve a milk collection problem considering trucks with compartments. Heuristic procedures are developed to solve the problem. Recently, Paredes-Belmar, Montero, & Leonardini (2021) present a milk collection problem where two milk types are collected, and the mix of them is forbidden. Milk collection centers are located to accumulate milk from distant farms. At this stage, the milk is collected by small trucks. Next, the big trucks coming from a plant collect milk from collection centers and high-volume farms. A three-stage procedure and an ILS metaheuristic are proposed to solve the problem.

Table 1
Summary of literature review. In column *Truck Fleet*: HT = heterogeneous, HM = homogeneous.

Authors	Year	Objective	Compartments	Milk types	Truck fleet	Trailers	Discrete blending	Gradual blending	Solution procedure (E: exact; H: heuristic)
Sankaran & Ubgade	1994	Min cost			HT				H: construction and improvement
Butler et al.	1997	Min cost			HM				E: branch-and-cut
Basnet et al.	1999	Min makespan			HM				E: branch-and-bound
Prasertsri & Kilmer	2004	Min cost			HT				H: construction and improvement
Dooley et al.	2005	Min cost		✓	HM	✓			H:genetic algorithm
Hoff & Løkketangen	2007	Min cost			HT	✓			H: tabu search
Claassen & Hendriks	2007	Min deviations			HT				E: branch-and-bound with special ordered sets
Caramia & Guerriero	2010	Min cost	✓	✓	HT	✓			H: ad-hoc assignment and routing
Mumtaz et al.	2014	Min cost			HT				H: ad-hoc assignment and routing
Dayarian et al.	2015	Min cost			HT				E: branch-and-price
Dayarian et al.	2015	Min cost			HT				E: branch-and-price
Lahrichi et al.	2015	Min cost	✓	✓	HT				H: tabu search
Dayarian et al.	2016	Min cost			HM				H: adaptive large-neighborhood search
Sethanan & Pitakaso	2016	Min cost	✓	✓	HT				H: differential evolution
Paredes-Belmar et al.	2016	Max profit		✓	HT		✓		E: branch-and-cut; H: exact solution in clusters
Masson et al.	2016	Min cost		✓	HT				H: adaptive large neighborhood search
Paredes-Belmar et al.	2017	Max profit		✓	HM		✓		E: branch-and-cut; H: ant colony
Huang et al.	2018	Min time			HT				H: Savings and sweep plus set covering
Caria et al.	2018	Min cost and CO_2			HM				H: ant colony
Chokanat et al.	2019	Min cost	✓	✓	HT				H: differential evolution
O'Callaghan et al.	2018	Min distance			HM				H: large neighborhood search
Montero et al.	2019	Min cost			HT				H: GRASP
Villagrán et al.	2020	Max profit		✓	HT		✓		H: ILS
Paredes-Belmar et al.	2021	Min costs		✓	HM				E: Three-stage approach; H: ILS
Polat & Topaloğlu	2021	Min distance	✓		HT				H: EILS
Our proposal	–	Max profit		✓	HT		✓	✓	E: branch-and-cut; H: ILS

Paredes-Belmar et al. (2016) propose a different strategy to address the milk collection problem, allowing, for the first time in the literature, different grades of milk to be mixed in each one-compartment truck. They maximize the profit, considering the revenues from milk and the transportation costs. Mixing may degrade the milk but also reduces transportation costs significantly, resulting in higher profits. The blend of different grades in the same truck is discrete, and it is independent of volume, i.e., the lowest grade of milk defines the grade of the blend, no matter what is the volume of it in the mix. Villagrán et al. (2020) solve the Paredes-Belmar et al. (2016) discrete blending problem using their own ILS. Their ILS cannot be applied to our problem, because it works with feasible and unfeasible solutions. This requires detection and penalization of infeasibilities, which results, in our case, in very long run times or inferior quality solutions. Paredes-Belmar et al. (2017) extend the milk collection problem with blending proposed in Paredes-Belmar et al. (2016), adding milk collection points (MCP) to expedite the collection. The MCP accumulates milk from low-volume and farthest-located producers, reducing the transportation cost and increasing collection profit. To the best of our knowledge, there are no other works that consider milk blending.

Table 1 summarizes the most relevant literature reviewed, according to the discussed criteria. It outlines our contributions regarding previous studies.

An unsolved issue in Paredes-Belmar et al. (2016) and Paredes-Belmar et al. (2017) is that even adding a drop of low-grade milk to a truck, transforms the whole truckload in low-grade. We relax this limitation by considering the volumes and SC per milliliter counts from every visited farm, mixed in a truck. We found that this strategy generates different routes and higher profits than the best result in the literature.

3. Problem description

A number of rural farms, organized in a cooperative, produce milk with different qualities, measured by SC per milliliter. The milk must be collected from all farms by trucks and delivered to a plant with a known location. The number of available trucks is known, and it is enough for collecting all the milk. The milk in the trucks is classified in grades, according to its SC per milliliter. In the Chilean case, milk grade A must have less than 500,000 SC per milliliter, milk grade B has between 500,000 and 1,000,000 SC per milliliter, and milk grade C has more than 1,000,000 SC per milliliter. A better grade and a higher price correspond to lesser SC per milliliter. The SC per milliliter of milk produced in each farm is measured before the collection process and classified in grades so that each farm is paid according to the grade and volume of milk it produces. The problem consists of optimizing the collection of milk from the farms in such a way as to maximize the profit, which is the revenue that can be obtained from the milk, discounted the cost of transportation. The revenue depends on the grade of the milk arriving at the plant, being higher for higher grades. Mixing milk of different grades is allowed in the trucks, when it contributes to a higher profit.

We show through a small example, the problem and proposed procedure, consisting in keeping the count of the SC per milliliter in each truck, by combining the SC per milliliter of all farms from which the truck collects milk. The count is updated after

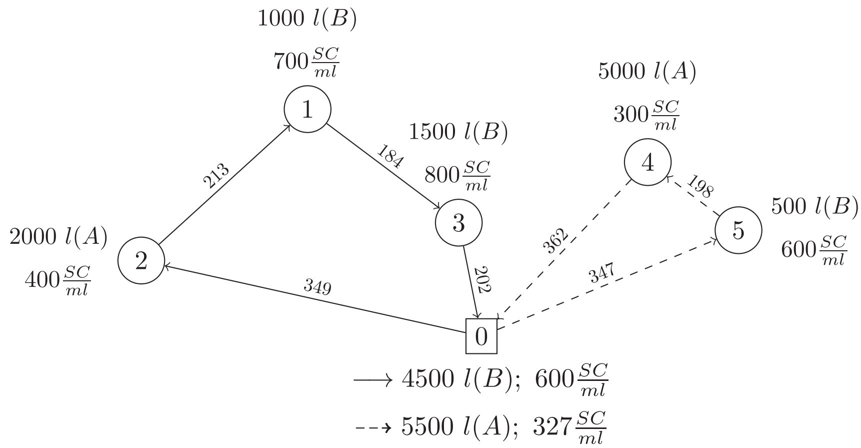

Fig. 1. An illustrative example of the milk collection with gradual blending.

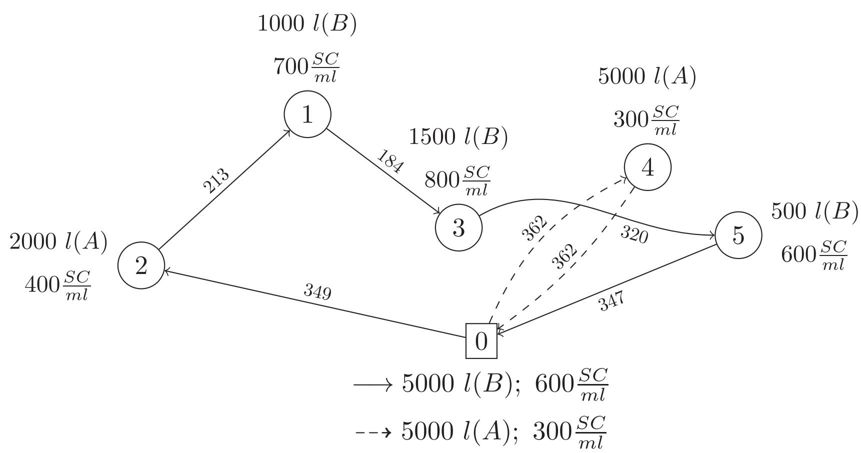

Fig. 2. A small example of discrete milk blending.

each blending and during the whole trip. As we show next, when planned together with the collection process, the blending minimizes the degrading of milk from higher grades to lower grades.

Fig. 1 shows an illustrative scheme for a small collection instance using two trucks. The Plant (0) is denoted with a square. The circles are the farms. The production volume and the number of SC per liter are indicated for each farm. In this example, the concentration tolerance range for milk A is less than $500\frac{SC}{\text{milliliter}}$, while milk with over $500\frac{SC}{\text{milliliter}}$ corresponds to grade B. The milk revenues are 1.0 and 0.7 monetary units [m.u.] per liter of milk A and B, respectively. In this example, there are two vehicles with 6000 liters of capacity.

The solid arrows indicate the route followed by truck 1 and dashed arrows the route of truck 2. Truck 1 arrives at the plant with 4500 liters of milk and $2.7 \cdot 10^9 SC$. So, the truck has a concentration of $\frac{2.7 \cdot 10^9 SC}{4500\,\text{liters}} = 600\frac{SC}{\text{milliliter}}$. This blend is classified as milk B. Truck 2 arrives with 5500 liters and $1.8 \cdot 10^9 SC$. Thus, truck 2 has a concentration of $327.2\frac{SC}{\text{milliliter}}$, and the 5500 liters of the blend are classified as milk A. The total cost is 1855 [m.u.], and the revenue is 8650 [m.u.], obtaining a profit of 6795 [m.u.].

Note that the problem is similar to the milk collection addressed in Paredes-Belmar et al. (2016). However, in their strategy, there is no consideration of the SC per milliliter, but only the grade of the milk. If milk grade A > milk grade B > milk grade C, the milk blending rules are defined as A + B → B, B + C → C, A + C → C, and A + B + C → C. Fig. 2 shows a solution of the previous instance using their strategy. The trucks arrive with 5000 liters of milk A ($300\frac{SC}{\text{milliliter}}$) and 5000 liters of milk B ($600\frac{SC}{\text{milliliter}}$). The cost is 2137 [m.u.], and the revenue is 8500 [m.u.], which results in a profit of 6363 [m.u.], which is 432 [m.u.] less than using the gradual blending we propose here.

The problem becomes the capacitated vehicle routing problem (CVRP) with multiple products if milk blending is not allowed. This problem is separable per milk type, and it is infeasible if only two vehicles of 6000 liters are available. Three trucks are required (two for milk A, one for milk B). The total cost is 2659 [m.u.] and the revenue is 9100 [m.u.], obtaining a profit of 6441 [m.u.].

If trucks with separated compartments were available, there are two possible situations: keeping segregated milk collection (Heßler, 2021; Martins, Ostermeier, Amorim, Hübner, & Almada-Lobo, 2019; Ostermeier, Henke, Hübner, & Wäscher, 2021; Ostermeier & Hübner, 2018) or using gradual blending within compartments. We use the same small example with two trucks, each with 6000 liters of capacity and two 3000-liter compartments each, to compare both situations. Without blending, the profit is 6963 [m.u.]. (Routes 0-5-3-1-2-0 and 0-4-0). If milk blending is allowed within compartments, the profit is 7695 [m.u.] (Routes 0-3-1-2-0 and 0-4-5-0). This result suggests that milk blending also improves the profit in the case of trucks with compartments.

It should be noticed that the milk of all farms is frequently inspected because the payment depends on the grade of produced milk. So, at the beginning of the season, each farm is tested and classified, and the grade is maintained throughout the season. Thus, the payment to each farm is constant, and it does not need to be considered in the problem.

4. Solution methodologies

We present two approaches for solving the GMB problem. The first approach considers a mixed-integer linear programming model to address all conditions of the problem. Then we implement a B&C algorithm for solving small-size instances. The second one is an ILS metaheuristic for solving real-size instances.

4.1. Mixed-integer linear programming model and branch & cut

4.1.1. Mathematical programming formulation

The mathematical programming formulation determines efficient routes for the milk collection using a heterogeneous truck fleet. The vehicles start at and return to the plant. The objective is to maximize the profit, considering revenues per milk at the plant and transportation costs. Recall that the payment to farms is constant.

Let N be the set of nodes (including node plant 0), R the set of milk grades, K the set of trucks, and I the set of ordered pairs that relate node $i \in N \setminus \{0\}$ with the grade of milk $r \in R$ it produces. The transportation cost between nodes i and j is denoted by the parameter c_{ij}. q_i denotes the milk production (liters) of farm $i \in N$ ($q_0 = 0$), Q_k is the capacity of each truck $k \in K$, U_r is the upper bound of SC per milliliter for milk $r \in R \cup \{0\}$, with $U_0 = 0$ the lower bound for the first tolerance range of SC per milliliter. p_r is the revenue per liter of milk type $r \in R$, I_i is the count of SC per milliliter in node $i \in N$ ($I_0 = 0$), and e_i represents the number of SC per milliliter in node i such that $e_i = q_i I_i$.

Decision variables

$$x_{ijk} = \begin{cases} 1 & \text{if node } i \in N \text{ is visited before node } j \in N \\ & \text{by truck } k \in K : i \neq j \\ 0 & \text{otherwise} \end{cases}$$

$$y_{ik} = \begin{cases} 1 & \text{if node } i \in N \text{ is visited by truck } k \in K \\ 0 & \text{otherwise} \end{cases}$$

$$z_{kr} = \begin{cases} 1 & \text{if truck } k \in K \text{arrives with milk } r \in R \text{to the plant} \\ 0 & \text{otherwise} \end{cases}$$

$$w_{kr} = \text{number of SC per milliliter of milk } r \in R \text{ in the plant delivered by truck } k \in K$$

$$v_{kr} = \text{volume of milk } r \in R \text{in the plant delivered by truck } k \in K$$

Objective function

$$\max \quad Z = \sum_{r\in R}\sum_{k\in K} p_r v_{kr} - \sum_{i\in N}\sum_{j\in N: i\neq j}\sum_{k\in K} c_{ij} x_{ijk} \tag{1}$$

Subject to:

$$\sum_{j\in N\setminus\{0\}} x_{0jk} \leq 1 \qquad \forall k \in K \tag{2}$$

$$\sum_{k\in K} y_{ik} = 1 \qquad \forall i \in N \setminus \{0\} \tag{3}$$

$$\sum_{i\in N\setminus\{j\}} x_{ijk} - \sum_{i\in N\setminus\{j\}} x_{jik} = 0 \qquad \forall k \in K, j \in N \tag{4}$$

$$\sum_{j\in N\setminus\{i\}} x_{jik} = y_{ik} \qquad \forall k \in K, i \in N \setminus \{0\} \tag{5}$$

$$\sum_{i\in N} q_i y_{ik} \leq Q_k \qquad \forall k \in K \tag{6}$$

$$\sum_{i\in S}\sum_{j\in S} x_{ijk} \leq |S| - 1 \qquad \forall k \in K, S \subseteq N \setminus \{0\} : |S| \geq 2 \tag{7}$$

$$w_{kr} \leq z_{kr} \sum_{i\in N} e_i \qquad \forall k \in K, r \in R \tag{8}$$

$$w_{kr} \leq v_{kr} U_r \qquad \forall k \in K, r \in R \tag{9}$$

$$w_{kr} \geq v_{kr} U_{r-1} \qquad \forall k \in K, r \in R \setminus \{0\} \tag{10}$$

$$v_{kr} \leq Q_k z_{kr} \qquad \forall k \in K, r \in R \tag{11}$$

$$\sum_{r\in R} z_{kr} \leq 1 \qquad \forall k \in K \tag{12}$$

$$\sum_{r\in R} w_{kr} = \sum_{i\in N\setminus\{0\}} e_i y_{ik} \qquad \forall k \in K \tag{13}$$

$$\sum_{r\in R} v_{kr} = \sum_{i\in N\setminus\{0\}} q_i y_{ik} \qquad \forall k \in K \tag{14}$$

$$\sum_{r\in R}\sum_{k\in K} v_{kr} = \sum_{i\in N} q_i \tag{15}$$

$$x_{ijk}, y_{ik}, z_{rk} \in \{0, 1\} \qquad \forall k \in K, i \in N, j \in N, r \in R : i \neq j \tag{16}$$

$$w_{kr}, v_{kr} \geq 0 \qquad \forall k \in K, r \in R \tag{17}$$

The objective (1) maximizes the profit of milk collection, considering revenue per volume minus transportation costs. Constraints (2)–(7) assure the route construction for each truck.

Constraints (8) state the relationship between SC per milliliter count variable (w_{kr}) and binary variables (z_{kr}). Constraints (9) and (10) state the maximum and minimum ranges, respectively, for the blend of milk r that arrives at the plant in truck k. The relationship between volume variables v_{kr} and binary variables z_{kr} is enforced by constraints (11). Constraints (12) force that the milk blend on truck k is classified in only one grade of milk r. Constraints (13) and (14) count the number of SC per milliliter and volume of milk that arrive at the plant in each truck. Constraint (15) indicates that the produced milk must be equal to the collected milk. Finally, constraints (16) and (17) state the domain of variables.

In the problem at hand, there are no incompatibilities between the products that will be mixed in a truck. Note, though, that in other applications, such incompatibilities may exist (see Caramia & Guerriero, 2010). The model can accommodate this case by including a set C of incompatibilities, that indicates whether some product $r \in R$ can or cannot be mixed with another product $s \in R$ in the same truck, because of regulations, because the mix would become a dangerous product, or any other reason. In that case, Constraints (18) would preclude these incompatibilities.

$$y_{ik} + y_{jk} \leq 1 \;\; \forall k \in K, i, j \in N\setminus\{0\} : i \neq j, (i, r), (j, s) \in I, (r, s) \in C \tag{18}$$

4.1.2. Branch-and-cut and valid inequalities

The model (1)–(17) represents all conditions of the problem. However, expression (7) generates an intractable number of constraints. We relax this constraint and replace it with valid inequalities in a B&C approach. Within a Branch-and-Bound (B&B) algorithm, every time an integer solution is found, a support graph is built for each truck with $x_{ijk} = 1$. An ad-hoc separation algorithm is run for each support graph to detect violated valid inequalities, and all corresponding cuts are added before solving again.

We propose the following known valid inequalities in the branch-and-cut algorithm.

The inequality:

$$\sum_{h \in S} x_{hlk} \le \sum_{i \in N \setminus S : i \neq l} \sum_{j \in S} x_{ijk} \quad \forall k \in K, l \in N \setminus \{S, 0\}, S \subseteq N \setminus \{0\} \qquad (19)$$

is a valid cut for GMB problem. **Proof.** Paredes-Belmar et al. (2016).

If a truck k uses an arc (h, l), there must be an entering arc from node i to node j because of the flow balance constraints (4).

The inequality:

$$\sum_{i \in N \setminus S} \sum_{j \in S} (Q_k - q_i) x_{ijk} \ge \sum_{i \in S} q_j y_{jk} + \sum_{l \in S} \sum_{m \in N \setminus S} q_m x_{lmk} \quad \forall k \in K, S \subseteq N \setminus \{0\} \qquad (20)$$

is a valid cut for the GMB problem. **Proof.** Yaman (2006).

If a truck k travels to the set S using an arc (i, j), then the same truck must have enough capacity to collect the milk of node i, the milk of nodes within the set S and the demand of the node m in $N \setminus S$.

Separation algorithm We find the integer solutions in the B&B algorithm using the generic callbacks from solver CPLEX 20.1. To identify subtours, for each truck $k \in K$, we start the search with node 0 (depot), which is added to a set S. We use the shrinking technique (Cook, Espinoza, & Goycoolea, 2007), to add nodes to S such that $x_{ijk} = 1$; $i \in S : j \in N \setminus S$. For each truck k, if the cardinality of the set S is not equal to the number of nodes visited by truck k, there is a subtour. In this case, the subtour breaking constraints (19) and (20) are added for the set S. Next, we select a node $i \in N \setminus S$ connected to a node $j \in N \setminus S$ such that $x_{ijk} = 1$. This node j is added to a set S'. We keep adding to the set S', all nodes $t \in N \setminus S$ such that $x_{itk} = 1$, $i \in N \setminus S$. Then we check and add the cuts (19) and (20) to the set S'. The procedure is repeated until there are no more subtours in the support graphs. The complexity of this algorithm is $O(|N|^4)$.

4.2. Metaheuristic: iterated local search with random variable neighborhood descent

4.2.1. Iterated local search

We chose an Iterated Local Search (ILS) algorithm, because it has shown good performance when applied to a variety of vehicle routing problems, while requiring relatively short times compared to other metaheuristics (Brandão, 2020; 2018; Coelho et al., 2016; Cuervo, Goos, Sörensen, & Arráiz, 2014; Lourenço, Martin, & Stützle, 2019; Morais, Mateus, & Noronha, 2014; Paredes-Belmar et al., 2021; Penna, Subramanian, & Ochi, 2013; Polat, Kalayci, Kulak, & Günther, 2015; de Sousa, González, Ochi, & de Lima Martins, 2021; Stützle & Ruiz, 2018).

Villagrán et al. (2020) also use an ILS to solve the DMB, which is similar to the GMB. However, their algorithm cannot be successfully applied to the GMB, as it works with both feasible and unfeasible solutions. Our ILS works only with feasible solutions, because the detection, correction, and penalization of infeasible solutions, in our case, results in very long running times and bad quality solutions. The ILS has the general structure displayed in Algorithm 1,

Algorithm 1 ILS structure (Stützle & Ruiz, 2018).

```
procedure ITERATED LOCAL SEARCH
    s_0 ← GenerateInitialSolution
    s* ← LocalSearch(s_0)
    repeat
        s' ← Perturbation(s*)
        s*' ← LocalSearch(s')
        s* ← AcceptanceCriterion(s*, s*')
    until termination condition met
```

see Stützle & Ruiz (2018) and Lourenço et al. (2019).

ILS starts by constructing a feasible initial solution s_0, to which it applies a local search to find a locally optimal solution s^*. Then, iteratively, perturbs the current solution to find a new one s' in a different neighborhood as a way to escape from the local optimum, followed by local search to find, again, a local optimum $s^{*'}$ in the neighborhood of s'. This perturbation - local search cycle is repeated until a termination condition is met.

The solutions s of the GMB problem are encoded as a set of variable size routes. Each route in a solution corresponds to a truck. The information about the routes, kept in the solution, consists of the ordered sequence of the visited farms, the cost of the trip, the features of the visited farms, and the grade of the milk in the truck. A solution's quality is given by the value of the objective function $f(s)$ (Eq. (1)). The search is performed only within the feasible solution space.

4.2.2. Random variable neighborhood descent

A variety of local search procedures have been used successfully within ILS. We use Variable Neighborhood Descent, analogously to the metaheuristic that Soria-Alcaraz, Ochoa, Sotelo-Figueroa, Carpio, & Puga (2017) call *Iterated Variable Neighborhood Descent*, except that we use the *Random* version of VND (RVND) (de Sousa et al., 2021). All the variants of VND are based on the concept of *neighborhood*.

A neighborhood of a candidate solution s^* is defined as the set of all candidate solutions that can be obtained by applying a specific *operator*, starting from s^*. The neighborhood is searched over by applying the operator repeatedly until no improvement is obtained, and the resulting solution is a local optimum for that particular neighborhood. The search can use first or best improvement policies. Variable Neighborhood Descent (VND) algorithms search over more than one neighborhood to increase the likelihood of finding good local optima. Instead of using a fixed order for the operators, the Random VND (RVND) defines randomly the order in which the different neighborhoods are searched (de Sousa et al., 2021). We use four operators, which generate four neighborhoods.

The RVND is described in Algorithm 2. RVND first applies procedure *application_order()*, which builds a list of the operators (line 2) indexed by $k = 1, \ldots, 4$, ordering them at random, using a uniform distribution. The list indicates the order in which these operators will be used. Within the while-return cycle (lines 5–12), RVND applies the procedure *local_search*(k, s^*) (line 6) for all k, starting from $k = 1$ and following the order of the list. For any k, operator k is applied repeatedly in a first-improvement fashion, starting from solution s^*, until the local optimal solution has been found for the corresponding neighborhood. All routes or pairs of routes (depending on the operator) are considered in the search space. After a local optimum has been found for a neighborhood,

Algorithm 2 RVND (de Sousa et al., 2021).

```
1:  procedure RVND(s')
2:      application_order();          ▷ Random application order of operators
3:      k ← 1;
4:      s* ← s';
5:      while (k <= 4) do                              ▷ All operators
6:          s*' ← local_search(k, s*);    ▷ First improvement local search until local optimum is found
7:          if f(s*') > f(s*) then
8:              s* ← s*';
9:              k ← 1;
10:         else
11:             k ← k + 1
12:     return s*;
```

if it is better than the incumbent, it replaces the incumbent and the RVND starts again the search from operator $k = 1$ (lines 7 to 9).

We use the following four operators, common in VRP solving algorithms, see, e.g., Laporte, Ropke, & Vidal (2014):

- The **cross-exchange** operator, see, e.g., Gendreau & Potvin (2005), applied to a pair of routes, takes one sequence of two or three farm visits from each route, and exchanges them with each other. Both selected sequences have the same number of visits (two or three).
- For two routes, the **exchange** operator, also known as inter-route relocate (Laporte et al., 2014), takes a farm visit from one of the routes and inserts it in some position within the other route.
- The **move** operator, also known as intra-route relocate operator (Laporte et al., 2014), changes the position of a farm visit within its current route.
- The **2-opt** is a classical local search intra-route operator for routing problems (Babin, Deneault, & Laporte, 2007). It reverses a sequence of two or more farm visits within a route.

The full version of our metaheuristic is displayed in Algorithm 3.

Algorithm 3 ILS - RVND metaheuristic.

1: **procedure** ILS(rcl, t_{stuck}, $t_{\max}$)
2: $t_0 \leftarrow$ clock(); ▷ Time at the start of the execution
3: $t_s \leftarrow$ clock(); ▷ Time at which the last improvement happened
4: $s_0 \leftarrow$ construction(rcl);
5: $s^* \leftarrow$ RVND(s_0);
6: **while** (clock() $- t_0 \le t_{\max}$ and clock() $- t_s \le t_{stuck}$) **do** ▷ Execution and stagnation criteria
7: $s' \leftarrow$ perturbation(s^*); ▷ Using random selected operators
8: $s^{*'} \leftarrow$ RVND(s');
9: **if** $f(s^{*'}) > f(s^*)$ **then**; ▷ Acceptance criterion
10: $s^* \leftarrow s^{*'}$;
11: $t_s \leftarrow$ clock();
12: **return** s^*;

Our algorithm stops when the total elapsed time reaches a value $t_{\max}$, or the algorithm stagnates after a time t_{stuck}. After saving the initial time in counters t_0 and t_s (lines 2 and 3), the algorithm uses the procedure *construction*(rcl) to find a feasible initial solution s_0 (line 4). Next, it performs the RVND local search starting at s_0 to find a solution s^* that is locally optimal (line 5). Then, iteratively, it applies the procedure *perturbation*(s^*) to the current solution (line 7) to find a new one s', far enough from s^* as to avoid falling back into the previous local optimum, and performs a RVND local search (line 8) to find again a local optimum $s^{*'}$ in the vicinity of s'. This cycle is repeated and the stagnation timer t_s is reset to the current time whenever there is an improvement in the incumbent solution. The cycle ends when either the elapsed time reaches or exceeds $t_{\max}$ or the solution has not improved in a time t_{stuck}.

Procedure construction (rcl) Farms are listed in ascending order of their value of (SC per milliliter × Volume of milk). One empty route is created, which starts at the plant. At each step, the next visited farm is selected randomly from the top rcl farms in the list. Note that a higher value of rcl implies more randomness in the procedure, while $rcl = 0$ makes the procedure greedy. We use a uniform distribution to make the selection as in GRASP algorithms (Feo & Resende, 1989; Resende & Ribeiro, 2019). When the remaining capacity in the truck is not enough for collecting from an additional farm, a new route is started. This randomized greedy heuristic allows the construction of different quality initial solutions, depending on the setting of its parameters. This construction procedure is intended to group farms that produce milk with a low count of SC per milliliter to ensure a high revenue, while the operators search for the lowest possible transportation cost. *Procedure perturbation (s^*)*

This procedure chooses one of the four RVND local search operators at random. If the operator is intra-route, it chooses a route also at random, and applies the operator once. If the operator is inter-route, it chooses a pair of routes at random, to which it applies the operator. This process is done twice. Note that this procedure is not intended to find local optima. Instead, its function is to escape from the last searched neighborhood and to find a new initial solution to search from. It allows getting out of the current local optimum, but not far enough to transform the whole algorithm in a random search procedure.

5. Results

Our new GMB is first compared to the previously best-known milk collection strategy, DMB, on small and medium-size instances using B&C (branch-and-cut) (Section 5.1). In Section 5.2 we analyze the effect of the different cuts over the performance of the B&C method. The B&C and the ILS for GMB are compared in Section 5.3, and finally, in Section 5.4 we present the results for the full 500-node instance. All experiments were run on a PC with a processor Intel Xeon CPU E5-2680 v3 @ 2.50 GHz, 64Gb of RAM under Ubuntu x64 16.10 distribution. The MILP model was coded in C++ and solved with solver CPLEX 20.1. In all B&C runs, we set a time limit of four hours (14,400 seconds).

5.1. Comparison of GMB and DMB on test examples

For the comparison between DMB and GMB, we use the respective best known B&C methods, over standard test instances (Augerat et al., 1995; Christofides & Eilon, 1969; Christofides, Mingozzi, & Toth, 1979; Fisher, 1994; Reinelt, 1991; Rochat & Taillard, 1995). Since these test instances do not consider grades of milk, we used two different grade distributions among the farms. *Farms distribution 1* considers a cyclic assignment of grades: farm 1 produces milk A, farm 2 milk B, farm 3 milk C, farm 4 milk A and so on, as in Paredes-Belmar et al. (2016). In *Farms distribution 2*, we assign the milk grades to farms considering that the first 60% of the farms produce milk grade A, the next 30% of the farms produce milk grade B, and the last 10% of the farms produce milk grade C), which are the proportions in the case study. For example, in the instance *eil101* (with 100 farms plus depot), the first 60 farms produce milk A, the next 30 produce milk B, and the last 10, milk C.

We set randomly the values of SC per milliliter of milk of each farm using a uniform distribution within the ranges given by the regulations: $U[0; 5\cdot 10^5]\frac{SC}{\text{milliliter}}$ for milk A; $U[6\cdot 10^5; 10\cdot 10^5]\frac{SC}{\text{milliliter}}$ for milk B; and $U[10\cdot 10^5; 20\cdot 20^5]\frac{SC}{\text{milliliter}}$ for milk C. The revenues per milk type are, in monetary units: 1.0, for milk A; 0.7, for milk B, and 0.3, for milk C. For each instance, there are three available trucks with a single compartment. The data for all instances are available at the repository of Montero (2021).

Table 2 shows the results for the test examples using the two milk collection strategies and the MILP model. Columns Z show the optimal solution if the Integrality Gap (IG %) is zero, or the best incumbent obtained within the time limit. Column T indicates the CPU time (in seconds). Recall that in the DMB strategy, a drop of milk C was enough to classify a truckload as C, even if the rest of the milk was A. This means that in some cases, the milk classification in a truck at the plant does not correspond to its SC count per

Table 2
Comparison of gradual and discrete blending strategies on test instances.

Instance	Farms distribution 1								Farms distribution 2							
	Gradual blending			Discrete blending				Δ (%)	Gradual blending			Discrete blending				Δ (%)
	Z	IG (%)	T	Z	IG (%)	T	Z*		Z	IG (%)	T	Z	IG (%)	T	Z*	
a32	30,898.0	0.00	122	26,960.0	0.00	34	26,960.0	14.61	40,366.0	0.00	192	31,478.0	0.00	96	**33,878.0**	19.15
a33	36,287.0	0.00	9132	29,417.0	0.00	89	29,417.0	23.35	41,527.0	0.16	14,400	36,033.0	0.00	38	36,033.0	15.25
a34	35,466.0	0.07	14,400	30,536.0	0.00	66	30,536.0	16.14	44,646.0	0.00	2203	38,046.0	0.00	33	38,046.0	17.35
a36	33,476.0	0.69	14,400	29,513.0	0.00	138	29,513.0	13.43	40,246.0	0.10	14,400	32,455.0	0.00	56	32,455.0	24.01
a37	30,032.0	0.00	1720	24,837.0	0.00	38	24,837.0	20.92	38,346.0	0.00	2248	31,558.0	0.00	125	31,558.0	21.51
a38	35,998.0	0.16	14,400	28,596.0	0.00	704	28,596.0	25.88	44,610.0	0.00	2912	35,613.0	0.00	217	35,613.0	25.26
a39	35,720.0	0.25	14,400	31,067.0	0.00	73	31,067.0	14.98	45,220.0	0.17	14,400	35,773.0	0.00	460	35,773.0	26.41
a44	46,630.0	0.13	14,400	39,072.0	0.00	144	39,072.0	19.34	51,734.0	0.22	14,400	41,705.0	0.00	935	41,705.0	24.05
a45	47,144.0	0.37	14,400	40,922.0	0.00	188	40,922.0	15.20	54,246.0	0.13	14,400	42,437.0	0.00	908	**48,157.0**	12.64
a46	49,272.0	0.32	14,400	40,696.0	0.00	175	40,696.0	21.07	55,919.0	0.31	14,400	47,705.0	0.00	690	47,705.0	17.22
a48	45,360.0	0.44	14,400	39,947.0	0.00	2082	39,947.0	13.55	59,545.0	0.15	14,400	48,881.0	0.00	1642	48,881.0	21.82
a53	56,686.0	0.60	14,400	46,662.0	0.00	160	46,662.0	21.48	67,071.0	0.22	14,400	53,573.0	0.12	14,400	**58,181.0**	15.28
a54	27,030.0	0.93	14,400	22,414.0	0.00	217	22,414.0	20.59	31,666.0	0.42	14,400	24,421.0	0.00	607	24,421.0	29.67
a55	27,560.0	0.18	14,400	24,694.0	0.00	339	24,694.0	11.61	38,712.0	0.27	14,400	31,690.0	0.00	473	31,690.0	22.16
a60	30,490.0	0.41	14,400	24,934.0	0.00	438	24,934.0	22.28	38,268.0	0.40	14,400	32,101.0	0.00	366	32,101.0	19.21
a61	71,317.0	0.60	14,400	60,644.0	0.00	304	60,644.0	17.60	82,213.0	0.14	14,400	64,268.0	0.04	14,400	64,268.0	27.92
a62	27,216.0	1.87	14,400	22,949.0	0.00	340	22,949.0	18.59	31,869.0	0.69	14,400	26,507.0	0.19	14,400	26,507.0	20.23
a63	31,059.0	1.19	14,400	25,826.0	0.00	409	25,826.0	20.26	41,352.0	0.41	14,400	33,808.0	0.00	158	33,808.0	22.31
a64	32,388.0	0.75	14,400	26,117.0	0.00	2236	26,117.0	24.01	39,257.0	0.54	14,400	32,603.0	0.01	14,400	32,603.0	20.41
a65	29,900.0	0.46	14,400	28,046.0	0.00	467	28,046.0	6.61	38,784.0	0.60	14,400	28,687.0	0.66	14,400	**34,227.0**	13.31
a69	30,636.0	0.60	14,400	25,872.0	0.00	709	25,872.0	18.41	39,049.0	0.85	14,400	31,910.0	0.00	9361	31,910.0	22.37
a72	90,978.6	0.08	14,400	74,695.1	0.00	601	74,695.1	21.80	111,024.0	0.13	14,400	93,336.7	0.00	13,151	93,336.7	18.95
a80	37,346.0	1.28	14,400	29,997.0	0.00	4135	29,997.0	24.50	43,581.0	1.28	14,400	34,944.0	0.40	14,400	34,944.0	24.72
att48	44,475.0	1.82	14,400	20,402.0	0.00	2300	20,402.0	117.99	65,000.0	9.75	14,400	38,330.0	3.78	14,400	38,330.0	69.58
c50	59,065.0	0.34	14,400	50,128.0	0.00	145	50,128.0	17.83	73,331.0	0.14	14,400	59,241.0	0.04	14,400	59,241.0	23.78
c75	105,589.0	0.35	14,400	88,828.0	0.00	3631	88,828.0	18.87	124,185.0	0.59	14,400	101,765.0	0.00	1931	**121,125.0**	2.53
eil22	19,696.0	0.00	201	15,947.0	0.00	33	15,947.0	23.51	20,086.0	0.00	15	18,075.0	0.00	11	18,075.0	11.13
eil23	8,908.3	0.00	229	7,207.3	0.00	92	7,207.3	23.60	9,669.0	0.00	23	8,641.0	0.00	12	8,641.0	11.90
eil30	9,143.0	1.09	14,400	7,117.0	0.00	128	7,117.0	28.47	11,671.0	0.00	422	9,542.0	0.00	80	9,542.0	22.31
eil31	73,720.0	0.09	14,400	60,080.0	0.00	88	60,080.0	22.70	64,515.0	0.02	8733	59,222.0	0.00	82	59,222.0	8.94
eil33	23,916.0	0.00	4138	20,409.0	0.00	51	20,409.0	17.18	24,585.0	0.00	960	21,584.0	0.00	236	21,584.0	13.90
eil51	58,365.0	0.06	14,400	50,128.0	0.00	392	50,128.0	16.43	73,282.0	0.21	14,400	59,221.0	0.07	14,400	59,221.0	23.74
eil76	107,161.0	0.36	14,400	91,461.0	0.00	1180	91,461.0	17.17	124,154.0	0.38	14,400	101,767.0	0.00	2650	**121,127.0**	2.50
eil101	106,342.0	15.10	14,400	97,657.0	0.00	10,814	97,657.0	8.89	135,369.0	0.40	14,400	106,598.0	0.39	14,400	**118,918.0**	13.83
f45	28,535.5	0.01	14,400	23,705.0	0.00	141	23,705.0	20.38	33,783.5	0.12	14,400	29,668.0	0.00	12,584	29,668.0	13.87
f71	90,528.5	0.18	14,400	72,863.7	0.00	6775	72,863.7	24.24	109,965.0	0.19	14,400	89,137.2	0.00	9978	89,137.2	23.37
f135	8,006.4	45.58	14,400	7074.6	17.00	14,400	7,074.6	13.17	9,704.8	35.78	14,400	9422.0	16.13	14,400	9,422.0	3.00
tai75A	48,311.5	0.77	14,400	41,349.5	0.00	4031	41,349.5	16.84	62,471.0	0.00	12,584	53,222.0	0.27	14,400	53,222.0	17.38
tai75B	56,270.5	1.92	14,400	49,461.0	0.00	1111	49,461.0	13.77	68,920.0	0.74	14,400	56,637.5	0.00	2764	**66,359.5**	3.86
tai75C	31,446.5	0.45	14,400	26,123.5	0.00	2559	26,123.5	20.38	43,775.0	4.42	14,400	37,749.5	0.49	14,400	37,749.5	15.96
tai75D	52,084.5	0.06	14,400	42,484.0	0.00	2559	42,484.0	22.60	67,851.0	0.83	14,400	56,540.0	0.50	14,400	56,540.0	20.01
tai100A	8,432.1	46.61	14,400	8,000.8	0.88	14,400	8,000.8	5.39	10,716.7	27.04	14,400	9,148.2	34.52	14,400	9,148.2	17.15
tai100B	11,879.1	21.87	14,400	10,115.8	1.94	14,400	10,115.8	17.43	14,912.8	12.80	14,400	12,991.0	22.93	14,400	12,603.2	18.33
tai100C	15,055.9	9.80	14,400	13,377.4	0.44	14,400	13,377.4	12.55	17,417.5	10.54	14,400	15,162.2	4.22	14,400	15,162.2	14.87
tai100D	8,170.0	34.75	14,400	6,958.7	0.65	14,400	6,958.7	17.41	10,625.6	14.35	14,400	8,173.2	33.42	14,400	8,173.2	30.01
mn101	742.0	43.55	14,400	208.0	1.54	14,400	208.0	256.73	1,067.0	65.76	14,400	878.0	17.64	14,400	878.0	21.53
mn121	285.5	111.28	14,400	−922.5	146.25	14,400	−922.5	130.95	434.4	75.57	14,400	294.7	115.15	14,400	294.7	47.40
Average	–	7.39	12,892	–	3.59	3211	–	27.89	–	5.68	11,981	–	5.34	7397	–	19.83

milliliter. To make the comparison between the two routing strategies as fair as possible, we computed, for the DMB, the *actual* grade of the milk that arrives at the plant in each truck, i.e., assigning to the milk in a truck the grade that corresponds to its count of SC per milliliter. Column Z^* of the Table 2 shows the objective values (incumbents) corrected by the actual grade of the milk. Whenever there is a re-classification of the milk, the new objective value is shown in bold font. Finally, the $\Delta(\%)$ columns display the percentage differences between the GMB Z and the corrected DMB Z^* values.

Most of the instances have low IGs within four hours of computation. However, in instances with more than 100 nodes, the integer gaps increase significantly. The DMB tends to provide lower IG than the GMB. Note that the GMB generalizes the DMB (which is NP-Hard), and solving it requires more time, especially as the size of the instances increases.

For the first farm distribution, the average difference is around 28% in favor of GMB. For the second farm distribution, the average difference is close to 20%. However, if we only consider those instances in which IGs is lesser than 10% for GMB (those above that figure are distorted because the method has not reached a significant solution), the average difference for both distributions is around 20%. In any case, no matter what are the differences between distributions and between groups of instances, GMB results in higher profits than DMB in all cases, an indication of robustness. The results show the superiority of the GMB strategy, because it maximizes the blending that results in milk of higher grades. The DMB strategy does not consider the SC count per milliliter of milk *a-priori*; hence, it tends to avoid the blending of milk of higher grades with lower grades, reducing the revenues. A remarkable instance (with near optimal solutions) is *att48*, whose solutions for DMB and GMB differ by a large amount for both distributions. This difference is due of the fact that GMB requires two of the three available vehicles, while DMB needs the three vehicles. DMB results in much higher transportation costs.

In distribution 1, the volumes of different types of milk (A, B and C) are similar, while in distribution 2, volume of A milk > volume of B milk > volume of C milk. The higher profits in distribution 2 are due to two facts: first, in distribution 2 the total volume of milk turns out to be higher, and so are the revenues; second, in distribution 2, because of the way the milk was distributed among farms, there are many farms that produce the same type of milk close to each other, which makes the collection more efficient in terms of cost, especially when using DMB. An exception is instance *eil31* in distribution 2, in which a few farms produce large amounts of lesser grade milk C. This results in lesser revenues for that distribution as compared to distribution 1 both because of the total volume of milk C and the reduced possibility of mixing small amounts of that milk into higher grade milk without changing its grade (in the case of GMB).

5.2. Effects of cuts in B&C

In Table 3 we compare the effect of each individual cut over the B&C. Note that we choose distribution 1 and the instances of Table (2) with more than 70 nodes, as well as instance *att48*. The smaller instances do not show significant differences in IG or Z when cuts are analyzed separately. Each instance was run with a time limit of four hours. The Table 3 has a first column indicating the instances. Then there are eight columns, grouped in pairs (A, B, C, D) displaying the objective value Z and the integer gap IG(%) for each test. In the first pair of columns (A) all our cuts are active, while default CPLEX cuts are deactivated. In the second pair of columns (B) *CPLEX* is used with all its default cuts activated, plus subtour-breaking constraints (7) added to make the solutions feasible. In the third pair of columns (C), only cuts (19) are used, while CPLEX cuts are not active. The fourth column pair (D) reports the results using cuts (20) and CPLEX cuts deactivated. The three last columns show the percent of deterioration in objective values when using CPLEX (column pair B) or individual cuts (column pairs C and D) as compared to using all the cuts (column pair A) ($\Delta_{AB}(\%)$,$\Delta_{AC}(\%)$,$\Delta_{AD}(\%)$).

As Table 3 shows, the instances with larger IGs are those with more than 100 nodes. For example, the cuts (19) could not provide any feasible solution for the test instance *f135* within four hours of computation. In addition, the instance *mn121* produces the most significant percentage differences between the different cut strategies. The mixture of different cuts (columns A) provides in most cases the best results.

Moreover, *CPLEX* results in worse IGs in most cases when compared with B&C with all its cuts (column pair A), except for instances $tai75B$ and $tai100A$. In terms of individual cuts, the cut (19) shows the best behavior, and cut (20) alone also reports better average IGs than the use of *CPLEX*.

The predominantly positive Δs in the last three columns show that the use of all cuts (column pair A) results in the best performance of all. In column Δ_{AB}, the 89% of the instances present a better performance of A as compared to CPLEX (column pair B). Column Δ_{AC} displays the 78% of the instances in which all cuts beat cut (19) (column pair C). This column presents the smallest difference in average percentage. Δ_{AD} column presents a complete dominance of the cut mixture A compared with cut (20).

5.3. The ILS algorithm and its performance compared to B&C

Since the ILS has a stochastic nature, its performance depends on its set of parameter values, and a random seed that controls the sequences of randomly generated numbers that the algorithm uses during its search process. Because of this stochastic nature, we execute the ILS 20 times, using a set of 20 initial seeds. We also report the results after 10 executions. In general, after 10 executions there is not much improvement.The ILS was coded in C++, and all of the experiments were executed considering a maximum execution time of $t_{\max} = 120$ seconds for small and medium-size instances and $t_{\max} = 3600$ seconds for the full 500-node instance. The stuck time was fixed to a quarter of the maximum execution time, i.e., $t_{stuck} = t_{\max}/4$. For small and medium-size instances it corresponds to $t_{stuck} = 30$ seconds and for the full instance it corresponds to $t_{stuck} = 900$ seconds. The restricted candidate list percentage was set to 0.4 following the literature Coelho et al. (2016).

Table 4 shows the comparison between ILS and B&C on the same test instances as in Table 2.

The three B&C columns indicate the figures for this method (Best Value Z, Integrality GAP *IG %*, and Time in seconds T). The next five columns show values for ILS after the first 10 executions. *Best Z(%)* is the percentage difference between the B&C objective value and the best objective of ILS. *Worst Z(%)* and *Av Z(%)* represent the same percentage, but using the worst and average objective values of ILS, respectively. Note that the negative percentages mean that B&C gives better results, while positive percentages indicate that ILS beats B&C. The column *Cv.T* indicates the time in seconds of convergence to the best solution, and *T.T.* indicates the total time in seconds required by ILS. The next five columns show the same values for ILS, after 20 executions. In the last row, we report the average taken over all instances.

It is remarkable that, considering average values, after the first 10 executions, the profit values of ILS differ in −1.94% as compared to B&C, but obtained in around 5.13% of the time. When executing 20 times, the performance does not change much, and the time doubles.

Note that there is no clear pattern of gaps between the B&C solutions and the ILS solutions. The only two factors that might

Table 3
B&C and cut tests.

Instance	$A = (7) + (19) + (20)$		B = CPLEX + (7)		C = Cut (19)		D = Cut (20)		Δ_{AB}(%)	Δ_{AC}(%)	Δ_{AD} (%)
	Z	IG (%)	Z	IG (%)	Z	IG (%)	Z	IG (%)			
a72	90,978.6	0.08	90,881.5	0.26	90,962.2	0.12	90,956.8	0.17	0.11	0.02	0.02
a80	37,346.0	1.28	36,984.0	11.84	36,791.0	2.84	36,111.0	4.88	0.97	1.49	3.31
att48	44,475.0	1.82	37,584.0	36.30	42,808.0	12.79	31,350.0	61.27	15.49	3.75	29.51
c75	105,589.0	0.35	105,568.0	0.38	94,723.0	11.86	95,011.0	11.53	0.02	10.29	10.02
eil76	107,161.0	0.36	107,091.0	0.45	106,702.0	0.81	106,368.0	1.13	0.07	0.43	0.74
eil101	106,342.0	15.10	104,576.0	8.80	100,785.0	12.88	101,081.0	28.99	1.66	5.23	4.95
f72	90,528.5	0.18	90,446.4	0.31	90,528.6	0.16	90,466.6	0.28	0.09	0.00	0.07
f135	8,006.4	45.58	6,398.2	78.44	–	–	6,669.6	75.53	20.09	–	16.70
tai75A	48,311.5	0.77	46,501.5	5.07	48,145.0	1.13	47,930.0	1.84	3.75	0.34	0.79
tai75B	56,270.5	1.92	56,683.5	1.53	57,026.0	0.56	53,099.5	13.99	−0.73	−1.34	5.64
tai75C	31,446.5	0.45	31,009.5	2.20	31,378.5	0.71	30,959.5	27.00	1.39	0.22	1.55
tai75D	52,084.5	0.06	51,487.5	2.14	52,094.0	6.57	51,759.0	1.68	1.15	−0.02	0.62
tai100A	8,432.1	46.61	8,502.1	46.46	8,542.1	44.84	8,305.1	49.63	−0.83	−1.30	1.51
tai100B	11,879.1	21.87	11,245.7	44.31	11,688.1	37.33	11,134.1	21.65	5.33	1.61	6.27
tai100C	15,055.9	9.80	13,124.3	34.22	14,560.1	22.56	13,666.8	28.74	12.83	3.29	9.23
tai100D	8,170.0	34.75	7,905.7	40.49	8,060.4	37.42	7,584.8	46.65	3.24	1.34	7.16
mn101	742.0	43.55	456.0	150.31	763.0	19.21	672.0	76.62	38.54	−2.83	9.43
mn121	285.5	111.28	−389.5	280.37	−48.5	155.54	−653.5	221.30	236.43	116.99	328.90
Average	–	18.66	–	41.33	–	21.61	–	37.38	18.87	8.21	24.24

Table 4
Test instances.

Instance	B & C			ILS (10 executions)					ILS (20 executions)				
	Z	IG (%)	T	Best Z (%)	Worst Z (%)	Av.Z (%)	Cv. T.	T.T.	Best Z (%)	Worst Z (%)	Av. Z (%)	Cv. T.	T.T.
a32	30,898.0	0.00	122	−0.17	−3.16	−1.06	304	614	−0.17	−3.16	−1.06	564	1,184
a33	36,287.0	0.00	9,132	0.00	−15.39	−4.92	120	430	0.00	−15.40	−4.90	261	881
a34	35,466.0	0.07	14,400	−0.03	−1.35	−0.49	254	564	−0.03	−1.35	−0.71	452	1072
a36	33,476.0	0.69	14,400	0.22	−0.10	0.05	192	502	0.22	−0.13	0.03	402	1022
a37	30,032.0	0.00	1720	−0.09	−7.12	−1.47	310	620	−0.03	−7.26	−1.70	487	1107
a38	35,998.0	0.16	14,400	−0.09	−0.36	−0.22	156	466	−0.09	−0.46	−0.24	342	962
a39	35,720.0	0.25	14,400	−0.06	−6.83	−1.49	151	461	−0.06	−6.84	−1.17	291	911
a44	46,630.0	0.13	14,400	−0.11	−3.31	−2.32	303	614	−0.11	−3.31	−2.01	526	1147
a45	47,144.0	0.37	14,400	−0.07	−1.12	−0.76	272	582	−0.01	−1.12	−0.58	519	1139
a46	49,272.0	0.32	14,400	−0.01	−0.89	−0.52	342	652	−0.01	−0.89	−0.49	665	1285
a48	45,360.0	0.44	14,400	−3.28	−5.15	−4.24	247	557	−0.23	−5.15	−4.05	468	1088
a53	56,686.0	0.60	14,400	0.21	−0.13	0.04	236	546	0.21	−0.14	0.05	500	1116
a54	27,030.0	0.93	14,400	0.18	−0.25	−0.07	274	584	0.18	−0.25	−0.04	653	1273
a55	27,560.0	0.18	14,400	−6.40	−13.93	−8.13	247	557	−1.28	−13.93	−7.84	680	1288
a60	30,490.0	0.41	14,400	−0.89	−7.26	−1.88	316	626	−0.89	−7.26	−1.57	607	1227
a61	71,317.0	0.60	14,400	0.13	0.00	0.07	149	459	0.20	0.00	0.09	448	1068
a62	27,216.0	1.87	14,400	0.87	0.35	0.53	451	759	0.87	0.19	0.52	823	1442
a63	31,059.0	1.19	14,400	−0.44	−4.46	−2.56	582	875	−0.44	−4.46	−1.71	963	1555
a64	32,388.0	0.75	14,400	−0.48	−1.04	−0.75	326	637	−0.19	−1.04	−0.73	754	1375
a65	29,900.0	0.46	14,400	0.25	−0.16	0.09	271	582	0.26	−0.16	0.11	565	1187
a69	30,636.0	0.60	14,400	−1.08	−1.39	−1.19	592	859	−0.99	−1.46	−1.20	1017	1594
a72	90,978.6	0.08	14,400	−11.86	−11.90	−11.87	245	556	−11.86	−11.90	−11.87	647	1257
a80	37,346.0	1.28	14,400	0.21	−0.20	0.00	344	656	0.25	−0.20	0.05	649	1271
att48	44,475.0	1.82	14,400	−0.34	−4.32	−1.65	260	570	−0.34	−4.32	−1.49	398	1018
c50	59,065.0	0.34	14,400	0.10	−8.73	−1.73	320	631	0.16	−8.73	−1.30	619	1223
c75	105,589.0	0.35	14,400	−9.90	−10.18	−10.06	340	650	−9.85	−10.18	−10.07	652	1269
eil22	19,696.0	0.00	201	−1.21	−1.27	−1.25	22	332	−1.21	−1.27	−1.24	55	675
eil23	8,908.3	0.00	229	0.00	−2.32	−0.31	0	310	0.00	−2.64	−0.78	62	682
eil30	9,143.0	1.09	14,400	−0.30	−0.95	−0.37	225	535	−0.26	−1.20	−0.52	425	1045
eil31	73,720.0	0.09	14,400	0.00	−0.07	−0.04	68	378	0.00	−0.07	−0.03	266	886
eil33	23,916.0	0.00	4138	−1.29	−1.52	−1.38	89	399	−1.29	−1.52	−1.40	181	801
eil51	58,365.0	0.06	14,400	−7.18	−7.64	−7.58	208	519	−0.25	−7.64	−6.47	469	1090
eil76	107,161.0	0.36	14,400	0.08	−11.48	−8.01	572	864	0.14	−11.51	−8.58	971	1549
eil101	106,342.0	15.10	14,400	6.69	−4.73	−1.70	533	820	6.69	−4.73	−1.87	1163	1725
f45	28,535.5	0.01	14,400	−0.47	−3.87	−2.24	378	669	−0.29	−3.87	−2.20	547	1148
f71	90,528.5	0.18	14,400	0.03	−0.04	0.00	438	748	0.05	−0.04	0.00	1036	1625
tai75A	48,311.5	0.77	14,400	0.23	−2.76	−0.69	300	610	0.23	−2.95	−0.99	651	1272
tai75B	56,270.5	1.92	14,400	1.50	1.02	1.29	553	849	1.50	1.02	1.28	1022	1628
tai75C	31,446.5	0.45	14,400	−0.82	−13.87	−5.65	276	586	−0.82	−13.87	−5.13	588	1208
tai75D	52,084.5	0.06	14,400	−2.30	−6.43	−4.49	846	1033	−2.30	−6.43	−4.66	1282	1755
mn101	742.0	43.55	14,400	3.23	−10.24	−0.97	392	663	3.23	−10.24	−0.55	845	1371
mn121	286.0	111.28	14,400	−4.02	−41.78	−17.76	1142	1210	−2.97	−41.78	−18.79	2191	2395
f135	8,006.0	45.58	14,400	9.40	−27.62	−11.12	1086	1158	11.87	−27.97	−6.31	2194	2361
tai100A	8,432.0	46.61	14,400	14.58	12.58	13.77	917	1100	14.58	12.55	13.55	1852	2225
tai100B	11,879.0	21.87	14,400	5.47	4.69	5.12	671	946	5.65	4.20	5.05	1347	1856
tai100C	15,056.0	9.80	14,400	5.41	2.98	4.30	914	1118	5.41	2.98	4.30	1871	2274
tai100D	8,170.0	34.75	14,400	4.90	3.48	4.44	356	667	4.97	3.05	4.21	984	1530
Average	–	-	12,892.4	0.02	−4.69	−1.94	374.3	662.19	0.44	−4.74	−1.81	743.70	1,320.5

Table 5
Cluster instances.

Cluster	#F	#K	B & C				ILS (10 executions)					ILS (20 executions)				
			Z	IG (%)	T	#C	Best Z (%)	Worst Z (%)	Av Z (%)	Cv.T.	T.T.	Best Z (%)	Worst Z (%)	Av Z (%)	Cv.T.	T.T.
1	30	3	674.4	2.39	14,400	16,848	−0.13	−2.63	−1.25	88	398	−0.13	−2.63	−1.00	206	826
2	20	3	507.5	9.64	14,400	13,072	0.20	0.20	0.20	10	320	0.20	0.20	0.20	21	641
3	20	4	1,150.3	0.00	436	12,661	0.00	−0.38	−0.07	65	375	0.00	−0.38	−0.08	176	797
4	20	2	397.7	0.00	75	27,179	0.00	−0.08	−0.03	32	342	0.00	−0.25	−0.03	50	670
5	20	3	635.0	0.00	7161	13,655	0.00	0.00	0.00	47	357	0.00	−0.16	−0.02	60	680
6	39	3	966.9	3.63	14,400	14,364	0.10	0.10	0.10	69	379	0.10	0.10	0.10	108	728
7	28	1	65.9	0.00	0	3594	0.00	0.00	0.00	0	310	0.00	0.00	0.00	0	620
8	20	4	1,240.1	0.00	216	6382	0.00	0.00	0.00	11	321	0.00	0.00	0.00	21	641
9	36	4	1,030.3	0.00	13,322	8867	0.00	−0.97	−0.48	80	390	0.00	−0.97	−0.48	120	740
10	10	2	597.1	0.00	1	103	0.00	0.00	0.00	0	310	0.00	0.00	0.00	0	620
11	17	1	128.4	0.00	0	65	0.00	0.00	0.00	0	310	0.00	0.00	0.00	0	620
12	29	3	648.2	0.22	14,400	17,491	0.00	−6.85	−1.55	158	468	0.00	−7.47	−2.49	363	983
13	17	3	529.0	1.63	52	19,694	0.00	0.00	0.00	69	379	0.00	0.00	0.00	167	787
14	28	4	1,298.3	0.88	14,400	10,933	0.00	−2.37	−0.61	46	357	0.00	−3.43	−1.09	179	800
15	24	3	742.7	0.00	13,831	14,218	−0.49	−0.92	−0.70	101	411	−0.09	−0.92	−0.58	203	823
16	23	2	361.4	0.00	40	13,453	0.00	−1.85	−0.37	45	355	0.00	−1.85	−0.18	99	719
17	10	3	671.4	0.00	23	482	0.00	0.00	0.00	1	311	0.00	0.00	0.00	1	621
18	29	5	1,250.3	8.15	14,400	7646	4.45	4.06	4.36	158	468	4.45	3.90	4.35	245	865
19	21	4	947.7	0.00	8390	11,944	0.00	0.00	0.00	13	323	0.00	0.00	0.00	18	638
20	16	3	489.1	0.00	51	20,584	0.00	0.00	0.00	96	406	0.00	0.00	0.00	166	786
21	7	2	581.3	0.00	0	125	0.00	0.00	0.00	0	310	0.00	0.00	0.00	0	620
22	19	4	858.1	0.00	1213	3283	0.00	0.00	0.00	2	313	0.00	0.00	0.00	4	625
23	7	2	299.9	0.00	0	99	0.00	0.00	0.00	0	760	0.00	0.00	0.00	0	1250
24	7	2	314.2	0.00	0	256	0.00	0.00	0.00	0	310	0.00	0.00	0.00	0	620
25	3	2	−16.5	0.00	0	6	0.00	0.00	0.00	0	310	0.00	0.00	0.00	0	620
Average	–	–	–	1.06	–	–	0.17	−0.47	−0.02	–	–	0.18	−0.55	−0.05		

have an effect are size and structure of the instances. When the size of the instances increases, ILS has the highest difficulty in reaching good solutions. Analyzing the structure of the instances, it can be observed that a grouping of farms in small size clusters (as instances *mn*101 and *mn*121) decreases the quality of the solution. Both structure and size together might exert an influence on the quality and dispersion of the solutions obtained using different seeds. See, e.g., instances *mn*121 and *f*135. In instance *mn*121, for 20 runs, the best solution has a gap of -2,97% and the worst, −41.78%, with an average of −18.79%. This instance has six clusters of farms, distributed uniformly over the region. The best solution of instance *f*135 beats B&C by 11.87%, while the worst is worse than B&C in 27.97%. This instance, as opposed to most, has a strong clusterization of all farms on the east side of the region. In general, the large instances (the set *mn* and the set *tai*) offer the highest difficulty to the ILS.

5.4. Full instance

The real full instance considers 500 farms producing three milk grades: 313 farms produce milk A, 159 farms milk B, and 28 farms milk C. There is a homogeneous truck fleet composed of 65 trucks. The milk volume to be collected per farm varies between 57 and 25,018 liters. The revenue, in monetary units per liter of milk is 0.0150, 0.0105, and 0.0045 for milk A, B and C, respectively. The data details are available in the repository (Montero, 2021).

The milk production region is divided into the same clusters as in Paredes-Belmar et al. (2016). We first compare B&C and ILS to solve the collection problem for each cluster, because B&C was not able to find a feasible solution for the full instance. The 25 clusters were found using geographical accidents (rivers, lakes, mountains) and main highways to divide the collection region. B&C was given a time limit of 14,400 seconds of CPU time for each cluster. Whenever the time limit was reached, we report the best incumbent and the integrality gap; otherwise, we report the optimal solution.

Table 6
Results for Full Instance, ILS. B&C with added 25 clusters' profit and time.

Method	Best profit	Worst profit	Average profit	Total time [s]
ILS (10 executions)	17,395.1	16,801.8	17,123.5	36,239
ILS (20 executions)	17,395.1	16,801.8	17,105.6	72,437
B&C	16,368.7			131,211

Table 5 shows the detailed results for each cluster. For B&C, the column *Cluster* indicates the number of the cluster, *#F* is the number of farms per cluster. *#K* is the number of available trucks in each cluster.*Z* is the objective value reached in at most four hours. *IG* reports the integrality gap, *T* is the CPU time, and *#C* represents the number of cuts required by B&C. The remaining columns have the same meaning as in Table 4.

The B&C approach took a CPU time of around 36 hours for solving all the 25 clusters. The average integrality gap is close to 1%. Note that the ILS results are very close to those of B&C in most clusters. The average percentage difference (taking average values) between ILS and B&C is −0.02% with 10 executions, and −0.05% with 20 executions. However, for the 25 clusters, 10 executions of ILS required only 7.1% of the time required by B&C. For 20 executions, the result is not significantly different, but the time increases by a factor of 2.

We next present the results for the full 500-node instance using ILS. Table 6 shows the profit and run times of the best, worst and average solutions for 10 and 20 executions. For comparison purposes, we also include the profit and run time values of B&C when the clustered version is solved, i.e., adding up the values in Table 5. As the Table shows, ILS performs better than B&C even if the worst solution is considered. The time it takes ILS to solve the problem running 10 executions is 28% of the time required by B&C on separate clusters. The ILS time for 20 executions is doubled, without significant changes in objective value. If we consider the CPU time of one execution, the time is 2.8% of the B&C time, and considering as a benchmark the worst execution, there is an improvement in profit over B&C of 2.65%.

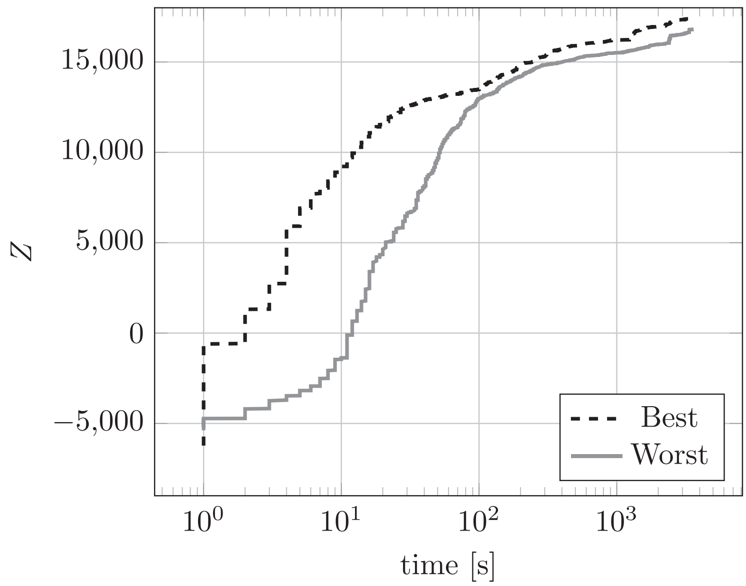

Fig. 3. Best and worst execution convergence.

Table 7
Results for the full real instance.

Instance	Best	Worst	Average	T.T
GMB (20 executions)	17,395.1	16,801.8	17,105.6	72,437
DMB (20 executions)	13,379.0	12,933.0	13,090.7	158,871

Fig. 3 shows the convergence of the ILS algorithm for the worst and best execution. Both processes start their search from low profit initial solutions (−8,692.89 for the worst execution and -8,412.7 for the best execution). The best execution quickly improves the quality of solutions during the first 100 seconds while the worst execution shows a slower convergence. Anyway, both search processes obtain solutions with a profit higher than 15,000 in the first 1000 seconds. After that, minor improvements are obtained in both cases.

For comparison purposes, we solve the same instance using the best previously known routing problem strategy: the *discrete* milk blending (DMB) (Paredes-Belmar et al., 2016). We use the ILS algorithm of Villagrán et al. (2020) to solve the DMB, as it is the best one known for that problem. As we mentioned before, both their construction heuristic and search procedure are not applicable to our problem. Vice-versa, adapting our algorithm to their problem, requires designing a special construction heuristic for DMB that works only on the feasible space. Table 7 shows the profits and times for both. We set the maximum number of trucks to 65, all with 30,000 liters of capacity for these experiments.

Table 7 summarizes the results of 20 executions for the 500-farm problem, for GMB, and DMB. We remark that in this instance, the corrected grades, assigned to the milk in each truck according to its actual content of SC per milliliter at the plant, turned out to be identical to the grades assigned to these trucks by the DMB strategy. For each case, the Table shows the best, worst and average profit, as well as the total run time.

As the Table shows, no matter if the best, the worst or the average solutions are considered, the GMB strategy is around 30% better than the DMB strategy and takes 46% of the time of the DMB-ILS of Villagrán et al. (2020).

The best gradual blending solution implemented in this paper is a 30% better and takes only one hour versus the 2 hours taken by the DMB-ILS algorithm of Villagrán et al. (2020).

6. Conclusions

As a first contribution, we present a new vehicle routing strategy for a heterogeneous-grade milk collection problem, in which the routing depends not only on the features of the network but also on how the collected products can be blended (transformed) during the collection process. Three grades of milk, defined by their maximum SC per milliliter count, are collected from a number of scattered farms, and controlled volumes of the three grades are blended in the trucks. The trade-off between the volumes of different grades of milk and the transportation costs is managed in such a way as to optimize the profit. Our strategy is a generalization of the best previously known one, the DMB of Paredes-Belmar et al. (2016), over which we obtain significant gains in profit: around 30% in our real case, and between 20 and 28% in test instances. Note that, as opposed to the DMB, our GMB strategy allows increasing the volume of milk A (highest grade) received at the plant, as compared to the collected volume of milk A, by blending small amounts of lower grades milk into the trucks carrying milk grade A, increasing the SC per milliliter count, but without exceeding the limits for which the grade is affected. In Paredes-Belmar et al. (2016) this is not possible.

Our second contribution is a MILP model and a B&C method for small instances of up to 75–100 nodes. We use this exact method on test problems first, and then to solve a large 500-node instance by dividing the area in smaller zones or clusters, to compare the GMB and DMB strategies, and analyze the performance of the different cuts in the B&C method.

The third contribution is an ILS metaheuristic for large instances. It uses RVND as its local search procedure, that includes four different well-known local search operators. The ILS procedure uses a construction procedure based on the SC per milliliter content and volumes of milk at each farm, and a local search stage as a means to optimize the profit obtained by the blending/collection process.

We solve 47 test instances from the literature and a real case of a milk company in the south of Chile. For the test instances, we compare the GMB and DMB strategies, and the performance of B&C against ILS. For the full instance, we first divide the instance into clusters. We then solve each cluster with B&C and with ILS. In most cases, ILS reports a solution that is very close to the B&C

solution in terms of profit but uses a fraction of the time. We finally solve the full instance at once using ILS.

It can be seen from Tables 4–7 that the results of 10 and 20 executions of ILS are very similar in quality, which allows to conclude that it is not worth running ILS for more than 10 iterations. However, ILS could be run only once, at the expense of the quality of the obtained solution. If ILS is executed only once on the real instance, it would be fair to say that the worse solution obtained in our tests should be considered for comparison purposes. In this case, the comparison with B&C shows an improvement of 2.6%, and it takes a CPU time of only 2.8% of that of B&C. If ILS is executed 10 times, it is fair to use the best value obtained, which is 6.3% better than B&C, with a CPU time of 27.6% of that of B&C.

Our results could be easily applied to trucks with compartments (Chokanat et al., 2019; Lahrichi et al., 2015; Sethanan & Pitakaso, 2016), within which the milk can be blended, as these can be regarded simply as chains of lower-capacity trucks, forced to go together by the appropriate constraints. Similarly, although we do not deal with maximum route times or workload equity, constraints can be added to those effects. Finally, this new routing strategy can be applied to other industries, e.g., gas with different octane content, cooking oil, or even different grade produce (fruits and vegetables).

Future research can be extended in several directions. First, Column Generation and Branch and-Price (Dayarian et al., 2015a) could provide optimal solutions for large-size instances. Second, it would be interesting to consider probability distributions in milk production and qualities at individual farms, as in Polat & Topaloğlu (2021).

Acknowledgments

We deeply thank two anonymous referees who did thorough reviews and made positive suggestions. They helped significantly improve the quality of this paper. Paredes-Belmar gratefully thanks FONDECYT grant number 1210183. Marianov gratefully acknowledges grants FONDECYT 1190064 and CONICYT-PIA-AFB180003. Lüer-Villagra gratefully thanks FONDECYT grant number 1200706.

References

Augerat, P., Naddef, D., Belenguer, J., Benavent, E., Corberan, A., & Rinaldi, G. (1995). Computational results with a branch and cut code for the capacitated vehicle routing problem. *Technical report INPG-RR-949-m*. https://www.osti.gov/etdeweb/biblio/289002

Babin, G., Deneault, S., & Laporte, G. (2007). Improvements to the or-opt heuristic for the symmetric travelling salesman problem. *Journal of the Operational Research Society, 58*(3), 402–407. https://doi.org/10.1057/palgrave.jors.2602160.

Basnet, C., Foulds, L. R., & Wilson, J. M. (1999). An exact algorithm for a milk tanker scheduling and sequencing problem. *Annals of Operations Research, 86*, 559–568. https://doi.org/10.1023/A:1018943910798.

Brandão, J. (2020). A memory-based iterated local search algorithm for the multi-depot open vehicle routing problem. *European Journal of Operational Research, 284*(2), 559–571. https://doi.org/10.1016/j.ejor.2020.01.008. ISSN 0377-2217

Brandão, J. (2018). Iterated local search algorithm with ejection chains for the open vehicle routing problem with time windows. *Computers and Industrial Engineering, 120*, 146–159. https://doi.org/10.1016/j.cie.2018.04.032. ISSN 0360-8352

Butler, M., Herlihy, P., & Keenan, R. B. (2005). Integrating information technology and operational research in the management of milk collection. *Journal of Food Engineering, 70*(3), 341–349. https://doi.org/10.1016/j.jfoodeng.2004.02.046. ISSN 0260-8774

Butler, M., Williams, H. P., & Yarrow, L. A. (1997). The two-period travelling salesman problem applied to milk collection in Ireland. *Computational Optimization and Applications, 7*, 291–306.

Caramia, M., & Guerriero, F. (2010). A milk collection problem with incompatibility constraints. *Interfaces, 40*(2), 130–143. https://doi.org/10.1287/inte.1090.0475.

Caria, M., Todde, G., & Pazzona, A. (2018). Modelling the collection and delivery of sheep milk: A tool to optimise the logistics costs of cheese factories. *Agriculture, 8*(1), 5. https://doi.org/10.3390/agriculture8010005.

CEPAL (1998). El clúster lácteo en Chile. https://repositorio.cepal.org/bitstream/handle/11362/31119/S9890743_es.pdf?sequence=1.[Online; accessed 24-September-2021].

Consorcio Lechero de Chile (2015). Indicadores del sector lechero de Chile. http://www.consorciolechero.cl/industria-lactea/wp-content/uploads/2015/12/Indicadores-del-Sector-Lechero.pdf. [Online; accessed 24-September-2021].

Chokanat, P., Pitakaso, R., & Sethanan, K. (2019). Methodology to solve a special case of the vehicle routing problem: A case study in the raw milk transportation system. *AgriEngineering, 1*(1), 75–93. https://doi.org/10.3390/agriengineering1010006.

Christofides, N., & Eilon, S. (1969). An algorithm for the vehicle-dispatching problem. *Journal of the Operational Research Society, 20*(3), 309–318. https://doi.org/10.2307/3008733.

Christofides, N., Mingozzi, A., & Toth, P. (1979). The vehicle routing problem. In N. Christofides, A. Mingozzi, P. Toth, & C. Sandi (Eds.), *Combinatorial optimization*. London: John Wiley & Sons.

Coelho, V., Grasas, A., Ramalhinho, H., Coelho, I., Souza, M., & Cruz, R. (2016). An ILS-based algorithm to solve a large-scale real heterogeneous fleet VRP with multi-trips and docking constraints. *European Journal of Operational Research, 250*(2), 367–376. https://doi.org/10.1016/j.ejor.2015.09.047. ISSN 0377-2217

Cook, W., Espinoza, D., & Goycoolea, M. (2007). Computing with domino-parity inequalities for the traveling salesman problem (TSP). *INFORMS Journal on Computing, 19*, 356–365. https://doi.org/10.1287/ijoc.1060.0204.

Cuervo, D. P., Goos, P., Sörensen, K., & Arráiz, E. (2014). An iterated local search algorithm for the vehicle routing problem with backhauls. *European Journal of Operational Research, 237*(2), 454–464. https://doi.org/10.1016/j.ejor.2014.02.011. ISSN 0377-2217

Dayarian, I., Crainic, T. G., Gendreau, M., & Rei, W. (2016). An adaptive large-neighborhood search heuristic for a multi-period vehicle routing problem. *Transportation Research Part E: Logistics and Transportation Review, 95*, 95–123. https://doi.org/10.1016/j.tre.2016.09.004. ISSN 1366-5545

Dayarian, I., Crainic, T. G., Gendreau, M., & Rei, W. (2015a). A branch-and-price approach for a multi-period vehicle routing problem. *Computers and Operations Research, 55*, 167–184. https://doi.org/10.1016/j.cor.2014.06.004. ISSN 0305-0548

Dayarian, I., Crainic, T. G., Gendreau, M., & Rei, W. (2015b). A column generation approach for a multi-attribute vehicle routing problem. *European Journal of Operational Research, 241*(3), 888–906. https://doi.org/10.1016/j.ejor.2014.09.015.

Dooley, A. E., Parker, W. J., & Blair, H. T. (2005). Modelling of transport costs and logistics for on-farm milk segregation in New Zealand dairying. *Computers and Electronics in Agriculture, 48*(2), 75–91. https://doi.org/10.1016/j.compag.2004.12.007.

FAO (2012). Agricultural cooperatives: Paving the way for food security and rural development. URL http://www.fao.org/3/ap088e/ap088e00.pdf.[Online; accessed 24-September-2021].

FAO (2019). Gateway to dairy production and products, collection and transport. URL http://www.fao.org/dairy-production-products/processing/collection-and-transport/en/. [Online; accessed 24-September-2021].

Feo, T. A., & Resende, M. G. (1989). A probabilistic heuristic for a computationally difficult set covering problem. *Operations Research Letters, 8*(2), 67–71. https://doi.org/10.1016/0167-6377(89)90002-3. ISSN 0167-6377

Fisher, M. L. (1994). Optimal solution of vehicle routing problems using minimum k-trees. *Operations Research, 42*(4), 626–642. https://doi.org/10.1287/opre.42.4.626.

Gendreau, M., & Potvin, J. Y. (2005). Tabu search. In M. Gendreau, & J. Y. Potvin (Eds.), *Handbook of metaheuristics* (pp. 37–55). Cham, Switzerland: Springer International Publishing, Springer. https://doi.org/10.1007/978-3-319-91086-4_2.

Heßler, K. (2021). Exact algorithms for the multi-compartment vehicle routing problem with flexible compartment sizes. *European Journal of Operational Research, 294*(1), 188–205. https://doi.org/10.1016/j.ejor.2021.01.037. ISSN 0377-2217

Hoff, A., & Løkketangen, A. (2007). A tabu search approach for milk collection in Western Norway using trucks and trailers. In *TRISTAN VI, Phuket Island, Thailand,*. URL https://himolde.brage.unit.no/himolde-xmlui/bitstream/handle/11250/2488977/WP_2008_06.pdf?sequence=1

Huang, K., Wu, K.-F., & Ardiansyah, M. N. (2019). A stochastic dairy transportation problem considering collection and delivery phases. *Transportation Research Part E: Logistics and Transportation Review, 129*(C), 325–338. https://doi.org/10.1016/j.tre.2018.01.018.

Irnich, S., Toth, P., & Vigo, D. (2014). Chapter 1: The family of vehicle routing problems. In P. Toth, & D. Vigo (Eds.), *Vehicle routing problems, methods, and applications, chapter 1* (pp. 1–33). SIAM Publications. https://epubs.siam.org/doi/abs/10.1137/1.9781611973594.ch1

Lahrichi, N., Crainic, T. G., Gendreau, M., Rei, W., & Rousseau, L. M. (2015). Strategic analysis of the dairy transportation problem. *Journal of the Operational Research Society, 66*(1), 44–56. https://doi.org/10.1057/jors.2013.147.

Laporte, G., Ropke, S., & Vidal, T. (2014). Chapter 4: Heuristics for the vehicle routing problem. In *Vehicle routing problems, methods, and applications, chapter 4* (pp. 87–116). SIAM Publications. https://doi.org/10.1137/1.9781611973594.ch4.

Lourenço, H. R., Martin, O. C., & Stützle, T. (2019). Iterated local search. In M. Gendreau, & J. Y. Potvin (Eds.), *Handbook of metaheuristics* (pp. 129–168). Cham, Switzerland: Springer International Publishing. https://doi.org/10.1007/978-3-319-91086-4_6.

Martins, S., Ostermeier, M., Amorim, P., Hübner, A., & Almada-Lobo, B. (2019). Product-oriented time window assignment for a multi-compartment vehicle routing problem. *European Journal of Operational Research, 276*(3), 893–909. https://doi.org/10.1016/j.ejor.2019.01.053. ISSN 0377-2217

Masson, R., Lahrichi, N., & Rousseau, L. M. (2015). A two-stage solution method for the annual dairy transportation problem. *European Journal of Operational Research*. https://doi.org/10.1016/j.ejor.2015.10.058. ISSN 0377-2217

Ministerio de Agricultura (1979). Reglamento especifico para la determinacion de la calidad de la leche cruda. URL https://www.leychile.cl/Navegar?idNorma=229646.[Online; accessed 24-September-2021].

Montero, E. (2021). Gradual-milk-blending-instances. https://github.com/elimail/Gradual-Milk-Blending-Instances, [Online; accessed 24-September-2021].

Montero, E., Canales, D., Paredes-Belmar, G., & Soto, R. (2019). A prize collecting problem applied to a real milk collection problem in Chile. *IEEE Congress on Evolutionary Computation (CEC)*, 1415–1422.

Morais, V. W., Mateus, G. R., & Noronha, T. F. (2014). Iterated local search heuristics for the vehicle routing problem with cross-docking. *Expert Systems with Applications, 41*(16), 7495–7506. https://doi.org/10.1016/j.eswa.2014.06.010. ISSN 0957-4174

More, S., Clegg, T., Lynch, P., & O'Grady, L. (2013). The effect of somatic cell count data adjustment and interpretation, as outlined in European Union legislation, on herd eligibility to supply raw milk for processing of dairy products. *Journal of Dairy Science, 96*, 3671–3681. https://doi.org/10.3168/jds.2012-6182. ISSN 0022-0302

Mumtaz, M. K., Jalil, M. N., & Chatha, K. A. (2014). Designing the milk collection network using integrated location routing approach. In *International conference on industrial engineering and operations management, Bali, Indonesia*. URL http://ieomsociety.org/ieom2014/pdfs/245.pdf

O'Callaghan, S., O'Connor, D., & Goulding, D. (2018). Distance optimisation of milk transportation from dairy farms to a processor over a national road network. In *Agriculture and food*. URL https://www.scientific-publications.net/get/1000028/1532007129523016.pdf

ODEPA (2018). Estudio de caracterización de la cadena de producción y comercialización de la industria de lácteos: Estructura, agentes y prácticas. In *Chile: Ministerio de agricultura*. URL https://www.odepa.gob.cl/wp-content/uploads/2019/01/InformeCadenaLactea2018.pdf. [Online; accessed 24-September-2021]

Ostermeier, M., Henke, T., Hübner, A., & Wäscher, G. (2021). Multi-compartment vehicle routing problems: State-of-the-art, modeling framework and future directions. *European Journal of Operational Research, 292*(3), 799–817. https://doi.org/10.1016/j.ejor.2020.11.009. ISSN 0377-2217

Ostermeier, M., & Hübner, A. (2018). Vehicle selection for a multi-compartment vehicle routing problem. *European Journal of Operational Research, 269*(2), 682–694. https://doi.org/10.1016/j.ejor.2018.01.059. ISSN 0377-2217

Palhazi Cuervo, D., & Hendriks, T. H. B. (2007). An application of special ordered sets to a periodic milk collection problem. *European Journal of Operational Research, 180*(2), 754–769. https://doi.org/10.1016/j.ejor.2006.03.042.

Paredes-Belmar, G., Lüer-Villagra, A., Marianov, V., Cortés, C. E., & Bronfman, A. (2017). The milk collection problem with blending and collection points. *Computers and Electronics in Agriculture, 134*, 109–123. https://doi.org/10.1016/j.compag.2017.01.015.

Paredes-Belmar, G., Marianov, V., Bronfman, A., Obreque, C., & Lüer-Villagra, A. (2016). A milk collection problem with blending. *Transportation Research Part E: Logistics and Transportation Review, 94*, 26–43. https://doi.org/10.1016/j.tre.2016.07.006.

Paredes-Belmar, G., Montero, E., & Leonardini, O. (2021). A milk transportation problem with milk collection centers and vehicle routing. *ISA Transactions*. https://doi.org/10.1016/j.isatra.2021.04.020. ISSN 0019-0578

Penna, P. H. V., Subramanian, A., & Ochi, L. S. (2013). An iterated local search heuristic for the heterogeneous fleet vehicle routing problem. *Journal of Heuristics, 19*(2), 132–201. https://doi.org/10.1007/s10732-011-9186-y. ISSN 1572-9397

Polat, O., Kalayci, C. B., Kulak, O., & Günther, H. O. (2015). A perturbation based variable neighborhood search heuristic for solving the vehicle routing problem with simultaneous pickup and delivery with time limit. *European Journal of Operational Research, 242*(2), 369–382. https://doi.org/10.1016/j.ejor.2014.10.010. ISSN 0377-2217

Polat, O., & Topaloğlu, D. (2021). Collection of different types of milk with multi-tank tankers under uncertainty: A real case study. *TOP*, 1–33. https://doi.org/10.1007/s11750-021-00598-x.

Prasertsri, P., & Kilmer, R. L. (2004). Scheduling and routing milk from farm to processors by a cooperative. *Journal of Agribusiness, 22*(2), 1–14. https://doi.org/10.22004/ag.econ.59380.

Reinelt, G. (1991). Tspliba traveling salesman problem library. *ORSA Journal on Computing, 3*(4), 376–384. https://doi.org/10.1287/ijoc.3.4.376.

Resende, M. G. C., & Ribeiro, C. C. (2019). Greedy randomized adaptive search procedures: Advances and extensions. In M. Gendreau, & J. Y. Potvin (Eds.), *Handbook of metaheuristics* (pp. 169–220). Cham, Switzerland: Springer International Publishing. https://doi.org/10.1007/978-3-319-91086-4_6. ISBN 978-3-319-91086-4

Rochat, Y., & Taillard, É. D. (1995). Probabilistic diversification and intensification in local search for vehicle routing. *Journal of Heuristics, 1*(1), 147–167. https://doi.org/10.1007/BF02430370.

Sankaran, J. K., & Ubgade, R. R. (1994). Routing tankers for dairy milk pickup. *Interfaces, 24*(5), 59–66. https://doi.org/10.1287/inte.24.5.59.

Sebastino, K. B., Uribe, H., & González, H. H. (2020). Effect of test year, parity number and days in milk on somatic cell count in dairy cows of los Ríos region in Chile. *Austral Journal of Veterinary Sciences, 52*, 1–7. https://doi.org/10.4067/S0719-81322020000100102.

Sethanan, K., & Pitakaso, R. (2016). Differential evolution algorithms for scheduling raw milk transportation. *Computers and Electronics in Agriculture, 121*, 245–259. https://doi.org/10.1016/j.compag.2015.12.021.

Soria-Alcaraz, J. A., Ochoa, G., Sotelo-Figueroa, M. A., Carpio, M., & Puga, H. (2017). Iterated VND versus hyper-heuristics: Effective and general approaches to course timetabling. In P. Melin, O. Castillo, & J. Kacprzyk (Eds.), *Nature-inspired design of hybrid intelligent systems*. In *Studies in computational intelligence: 667* (pp. 687–700). Springer. https://doi.org/10.1007/978-3-319-47054-2_45.

de Sousa, M. M., González, P. H., Ochi, L. S., & de Lima Martins, S. (2021). A hybrid iterated local search heuristic for the traveling salesperson problem with hotel selection. *Computers and Operations Research, 129*, 105229–105243. https://doi.org/10.1016/j.cor.2021.105229. ISSN 0305-0548

Stützle, T., & Ruiz, R. (2018). Iterated local search. In R. Martí, P. M. Pardalos, & M. G. Resende (Eds.), *Handbook of heuristics* (pp. 579–605). Cham: Springer International Publishing. https://doi.org/10.1007/978-3-319-07124-4_8. ISBN 978-3-319-07124-4

United States Department of Agriculture (2018). Determining U.S. milk quality using bulk-tank somatic cell counts. URL https://www.aphis.usda.gov/animal_health/nahms/dairy/downloads/dairy_monitoring/BTSCC_2017infosheet.pdf. [Online; accessed 24-September-2021].

Villagrán, J., Montero, E., & Paredes-Belmar, G. (2020). An iterated local search approach to solve the milk collection problem with blending. In *2020 IEEE congress on evolutionary computation (CEC)* (pp. 1–8). https://doi.org/10.1109/CEC48606.2020.9185888.

Yaman, H. (2006). Formulations and valid inequalities for the heterogeneous vehicle routing problem. *Mathematical Programming, 106*(2), 365–390. https://doi.org/10.1007/s10107-005-0611-6.