

A two-stage solution method for the annual dairy transportation problem

Renaud Masson^{(1,2)*}, Nadia Lahrichi^{(1,2)*}, Louis-Martin Rousseau^{(1,2)*}

(1) *CIRRELT, C.P. 6128, Succ. Centre-Ville, Montreal, Canada H3C 3J7*

(2) *Département de mathématiques et génie industriel, École Polytechnique de Montréal*

Abstract

The annual dairy transportation problem involves designing the routes that collect milk from farms and deliver it to processing plants. The demands of these plants can change from one week to the next, but the collection is fixed by contract and must remain the same throughout the year. While the routes are currently designed using the historical average demand from the plants, we show that including the information about plants demands leads to significant savings. We propose a two-stage method based on an adaptive large neighborhood search (ALNS). The first phase solves the transportation problem and the second phase ensures that the optimization of plant assignment is performed. An additional analysis based on period clustering is conducted to speed up the resolution.

Keywords: Dairy Transportation Problem, Two-Stage Method

*Corresponding author

Email addresses: renaud.masson@cirrelt.ca (Renaud Masson^(1,2)),
nadia.lahrichi@cirrelt.ca (Nadia Lahrichi^(1,2)),
louis-martin.rousseau@cirrelt.ca (Louis-Martin Rousseau^(1,2))

1. Introduction

In Canada, the dairy industry is ranked third in the agricultural sector (after grains & oilseeds and red meat). In 2011 the total net sales of dairy products generated \$13.7 billion, representing 16.4% of the Canadian food and beverage sector. Most Canadian dairy farms are located in Ontario and Quebec. The dairy processing sector is concentrated: 80% of the total raw milk produced in Canada is processed by three companies.

In Quebec, the *Producteurs de Lait du Québec* (PLQ), a coalition of dairy farmers, is responsible for planning the transportation of the milk produced in the province. The total annual cost of milk transportation from farms in Quebec to processing plants is more than \$70 million. The problem considered in this paper is the design of the routes that collect milk from farms and deliver it to processing plants.

Currently the PLQ solves two distinct problems: an annual problem to design the routes that collect the milk and a weekly problem to assign the routes to the plants depending on the quantity of milk requested. Obviously the routes have a large impact on the weekly assignments. In this paper, we integrate the two problems by considering the varying demand throughout the year as we design the routes. To our knowledge, the only analogous contribution in the literature deals with the dairy transportation problem introduced in Lahrichi et al. (2014). Although it handles the routing and the assignments to plants, it does not consider that the demand vary throughout the day therefore simplifying this aspect of the problem. In this paper, we aim to integrate this weekly variations to better plan the routes and the

assignment of routes to plants. This problem will be referred to as the Annual dairy transportation problem in the remainder of the paper.

This paper makes the following two contributions. First, we introduce a new routing and assignment problem, the ADTP. Second we improve on the state-of-the-art method for the dairy transportation problem (DTP). Average savings with the new approach to solve the DTP are as considerable as 4%. We also show that using the weekly information during the annual design of routes leads to additional savings. The proposed method is very efficient for the ADTP and may be adapted to similar problems in other industries.

The remainder of this article is structured as follows. We first define the problem and present a review of related problems. We then describe the proposed two-stage adaptive large neighborhood search (ALNS) and finally discuss our numerical experiments and provide concluding remarks.

2. Problem statement

The problem introduced in this article differs in several ways from classical vehicle routing problems. The dairy transportation problem was introduced in Lahrichi et al. (2014). It involves the design of minimum-cost routes that collect milk from producers and deliver it to processing plants. The PLQ faces this problem every year when designing the contracts that determine which routes will be performed by each transporter.

We introduce the annual version of the dairy transportation problem (ADTP), which consists of including the variation of the demand. Indeed, during each week of the year, each plant specifies its demand for milk. The total quantity of milk that it receives at the end of the week must be close to

the quantity requested (a tolerance is specified for each plant). This quantity is then split between each day (at least a predefined percentage of its weekly demand is assigned to ensure fairness between plants since most of them prefer to receive milk on weekdays rather than weekends).

The transportation problem is described as follows. We consider a heterogeneous fleet of capacitated vehicles located in various depots. Each producer is visited once by exactly one route that collects all its milk (split pickups are not allowed). The sequence of producers visited by a route has to be the same on every day of the planning horizon (fixed by contract), but the plants visited can change.

We present the mathematical model to formalize the problem the ADTP and the complicating constraints related to the weekly demand of the plants. Let F be the set of milk producers and P the set of processing plants. The set of weeks considered in the planning horizon is W . For each week $w \in W$, the demand of plant $p \in P$ is d_p^w . We denote by δ_p the tolerance for plant $p \in P$, and $\xi_p\%$ is the minimum percentage of its weekly demand that plant $p \in P$ must receive every day. Let K be the heterogeneous fleet of capacitated vehicles; each vehicle $k \in K$ has capacity q_k . Each producer $f \in F$ produces a quantity m_f of milk. Let D_w be the days of week $w \in W$; $D = \bigcup_{w \in W} D_w$ is the set of all the days of the planning horizon. O is the set of origin depots, and O' is the set of destination depots. The starting and ending depots of vehicle $k \in K$ are denoted $o(k)$ and $o'(k)$, and $N = F \cup O \cup O'$ is the set of all the nodes of the problem. The cost of the arc between nodes i and j is denoted $c_{i,j}$. We present this model to clearly state the problem that we are solving. To improve its readability, we do not linearize the nonlinear

constraints (we do not solve this model directly).

The variables are as follows: $x_{i,j}^k$ is a binary variable indicating whether or not vehicle $k \in K$ goes from $i \in O \cup O' \cup F$ to $j \in O \cup O' \cup F$. The delivery cost of route $k \in K$ for day $d \in D$ is represented by continuous variable b_d^k . The quantity delivered by vehicle $k \in K$ to plant $p \in P$ on day $d \in D$ is represented by variable $l_{p,k}^d$. Finally, the binary variable $y_{p,d}^k$ indicates whether or not vehicle $k \in K$ visits plant $p \in P$ on day $d \in D$.

$$\min \sum_{k \in K, i \in \{o(k)\} \cup F, j \in F} x_{i,j}^k c_{i,j} |D| + \sum_{k \in K, d \in D} b_d^k \quad (1)$$

The objective function (1) minimizes the distance covered every day to collect and deliver the milk.

$$\sum_{f \in F} x_{o(k),f}^k = 1 \quad \forall k \in K \quad (2)$$

$$\sum_{i \in N} x_{i,o(k)}^k = 0 \quad \forall k \in K \quad (3)$$

$$\sum_{i \in N} x_{o'(k),i}^k = 0 \quad \forall k \in K \quad (4)$$

$$\sum_{f \in F} x_{f,o'(k)}^k = 1 \quad \forall k \in K \quad (5)$$

Constraint (2) ensures that each vehicle leaves its starting depot only once. Constraint (3) ensures that no vehicle enters its starting depot, and constraint (4) ensures that no vehicle leaves its ending depot. Constraint (5) forces the

vehicles to end their routes at their ending depots.

$$\sum_{k \in K, j \in F \cup \{o'(k)\}} x_{f,j}^k = 1 \quad \forall f \in F \quad (6)$$

Constraint (6) ensures that each producer is visited exactly once.

$$\sum_{j \in F \cup \{o'(k)\}} x_{f,j}^k = \sum_{j \in F \cup \{o(k)\}} x_{j,f}^k \quad \forall f \in F \quad (7)$$

Constraint (7) ensures vehicle flow conservation.

$$\sum_{f \in F, j \in F \cup \{o(k)\}} x_{f,j}^k m_f \leq q_k \quad \forall k \in K \quad (8)$$

Constraint (8) ensures that the vehicle capacities are respected.

$$y_{p,d}^k = 0 \implies l_{p,k}^d = 0 \quad \forall p \in P, d \in D, k \in K \quad (9)$$

Constraint (9) prevents a plant from receiving milk from a vehicle on a given day, if the vehicle does not visit that plant.

$$\sum_{p \in P} l_{p,k}^d = \sum_{f \in F, j \in F \cup \{o'(k)\}} x_{f,j}^k m_f \quad \forall d \in D, k \in K \quad (10)$$

Constraint (10) ensures that all the collected milk is delivered.

$$\sum_{k \in K} l_{p,k}^d \geq \xi_p \sum_{d' \in D_w, k \in K} l_{p,k}^{d'} \quad \forall p \in P, w \in W, d \in D_w \quad (11)$$

$$\sum_{d \in D_w, k \in K} l_{p,k}^d \leq (1 + \delta_p) d_p^w \quad \forall p \in P, w \in W \quad (12)$$

$$\sum_{d \in D_w, k \in K} l_{p,k}^d \geq (1 - \delta_p) d_p^w \quad \forall p \in P, w \in W \quad (13)$$

Constraint (11) ensures that each plant receives each day at least a given percentage of the weekly delivery. Constraints (12) and (13) ensure that the quantity received each week by each plant is sufficiently close to the quantity requested.

$$\sum_{i \in S, j \in S} x_{i,j}^k \leq |S| - 1 \quad \forall k \in K, S \subseteq F \cup O \cup O', |S| \geq 2 \quad (14)$$

Constraint (14) ensures subtour elimination.

$$b_d^k \geq x_{f,o'(k)}^k y_{p,d}^k (c_{f,p} + c_{p,o'(k)}) \quad \forall k \in K, d \in D, f \in F \quad (15)$$

Constraint (15) determines the delivery cost of each route on each day.

$$x_{i,j}^k \in \{0, 1\} \quad \forall k \in K, i \in \{o(k)\} \cup F, j \in F \cup \{o'(k)\} \quad (16)$$

$$y_{p,d}^k \in \{0, 1\} \quad \forall k \in K, p \in P, d \in D \quad (17)$$

$$b_d^k \in \mathbb{R}^+ \quad \forall k \in K, \forall d \in D \quad (18)$$

$$l_{p,k}^d \in \mathbb{R}^+ \quad \forall k \in K, \forall d \in D, \forall p \in P \quad (19)$$

Constraints (16)–(19) are domain constraints.

We illustrate in Figure 1 how the annual routes are constructed and how they are modified weekly. Figure 1 (a) shows a route starting at depot d_1 , visiting farms f_1 through f_5 and delivering the milk collected in processing plant p_2 . The daily cost of a route is the sum of i) the distance covered by the vehicle during the collection (i.e, from d_1 to f_5), ii) the distance between the last producer visited and the plant that is the farthest from the last producer (i.e, between f_5 and p_2), and iii) the distance between that plant and the depot (i.e, between p_2 and d_1). The total cost of the solution is the sum of the daily route costs.

Figure 1 (b) illustrates the modification to the routes to better serve the demand of the plants. Due to the amount of milk requests by plants p_1 and p_2 , the route has to be diverted to p_1 . The sequence of producers stay identical in (a) and (b).

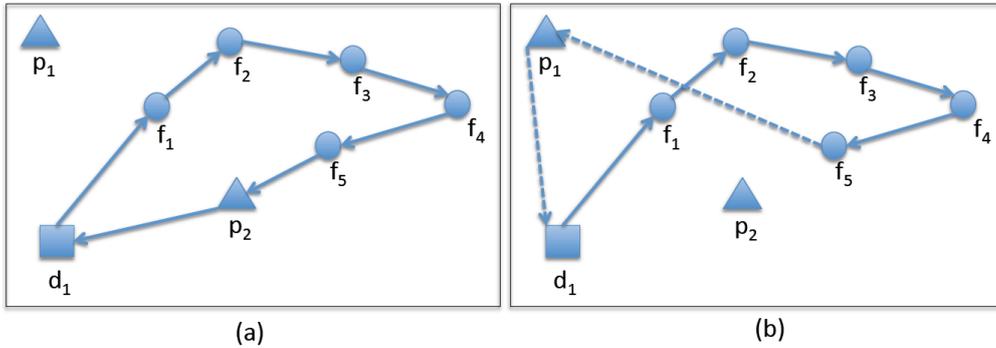


Figure 1: Illustration of a typical route collecting milk at farms and delivering at a designated plant

3. Related literature

The literature on the DTP is rather limited.

Several studies have been carried out in the context of dairy transportation, most of which consider only a subproblem of the DTP: Sankaran and Ubgade (1994), Basnet et al. (1997), Caramia and Guerriero (2010), and Dooley et al. (2005) present heuristic methods for routing problems in which the assignment of the producer to the plant is not considered. Exact methods have been devised by Basnet et al. (1999) and Claassen and Hendricks (2007) where only one plant and uniform demand were considered respectively. Finally, Foulds and Wilson (1997) present heuristic methods for the assignment of producers to plants. They do not consider the routing: the cost of serving each plant from each farm is known beforehand. The reader may refer to Lahrichi et al. (2014) for an extensive literature on the DTP.

The closest version of the DTP is tackled in Lahrichi et al. (2014). They consider a similar version of the problem but the demands of the processing plants are fixed and cannot change on a daily basis. Their method is based on unified tabu search (UTS) (Cordeau et al., 2001) and is called generalized UTS (GUTS). Experiments are performed on real-life instances with up to 226 customers. The authors compare the current practice, which assigns each producer to the same route and the same plant each year, with an approach that can modify these assignments. The problem is by nature stochastic (the amount to collect each week and the plant demands are not known a year in advance), but even solving the static case is challenging. Moreover, in the ADTP the plant demands change weekly, and this impacts the assignment of routes to plants. We must determine the assignment of the generated routes throughout the year, so the ADTP is much more difficult than the DTP.

Periodic vehicle routing problems occurring outside the dairy industry

share some similarities with our problem. One should note however that the dairy transportation problem is much more complex to solve than other periodic perishable food transportation problems (see Tarantilis and Kiranoudos (2001); Golden et al. (2002); Tarantilis and Kiranoudos (2007)).

When focusing on the multi-commodity or one-commodity pickup and delivery features of the ADTP, similar problems occurs in forestry and car sharing. For instance, Flisberg et al. (2009) investigate a vehicle routing problem in a forestry context. It is a variant of a multi-commodity pickup and delivery problem with a heterogeneous fleet. The planning horizon is one week, and on each day the quantity received by each demand node must lie within a predefined range. The authors present a two-stage solution method. The first stage creates transport nodes using linear programming, and the second stage designs the routes using a modified version of UTS. Instances with 26 to 665 supply nodes and 8 to 113 demand nodes are solved.

The single-vehicle version of the one-commodity pickup and delivery problem has been studied (Hernández-Pérez and Salazar-González, 2004; Hernández-Pérez et al., 2009; Zhao et al., 2009), but the multivehicle version has received little attention. Dror et al. (1998) study the problem of car redistribution in a car-sharing system. At any given time, some of the stations where the cars are parked may need more cars while others have too many. The objective is to redistribute the cars so as to meet the target level for each station while minimizing the cost of the redistribution. The authors present a mathematical model for the problem, but they fail to solve instances for which more than six cars need to be redistributed. The authors also propose a local-search approach based on constraint programming. It obtains good

results in less than five minutes on the instances they were able to solve to optimality.

Finally, one common constraint in the ADTP is that some of the producers cannot be visited by heavily loaded vehicles, because of access constraints. This so-called draft limit has been defined in the context of maritime transportation by Rakke et al. (2012). They propose an exact method for the traveling salesman problem with draft limits.

The ADTP is thus different from these related problems, as the impact of the operational truck to mills assignment decisions when we perform the strategic routing decisions.

4. Solution method

Addressing the ADTP is challenging and it cannot be directly solved in a reasonable time. Therefore, we develop a two-stage method. At the first stage we compute high-quality routes using ALNS for the DTP, which assumes that each plant has the same demand every day of the year and that a route can visit only one plant. At the second stage the resulting routes form an initial solution for a method that considers the assignment of the routes to the plants on a daily basis. We hope that the first-stage solution needs only minor changes to become a good solution for the ADTP.

The ALNS extends the large neighborhood search (LNS), introduced by Shaw (1998) in a constraint programming framework, to the vehicle routing problem with time windows (VRPTW). The ALNS includes an adaptive layer that enables some parameters to be automatically adjusted based on the performance of the heuristic in the most recent iterations. It was intro-

duced by Pisinger and Ropke (2007) for a variety of vehicle routing problems including the pickup and delivery problem with time windows (Ropke and Pisinger, 2006). In their assessment of advanced heuristics for the large-scale VRPTW, Gendreau and Tarantilis (2010) observe that the ALNS has good effectiveness, efficiency, simplicity, and flexibility. These features make it a perfect candidate for the DTP. The reader interested in an extensive description of the ALNS and its application to combinatorial problems is referred to the review by Pisinger and Ropke (2010).

Algorithm 1 presents the general outline of the two-stage method. The DTP is first solved by the ALNS (line 1). At the second stage we compute the variations of the routes of the DTP (line 2) and select the best routes and assign them to the plants (line 3).

Algorithm 1 Two-Stage Method

- 1: $DTPsolution \leftarrow ALNSsolveDTP()$
 - 2: $routesPool \leftarrow computeVariations(DTPsolution)$
 - 3: $ADTPsolution \leftarrow selectBestRoutesAndAssign(routesPool)$
 - 4: **return** $ADTPsolution$
-

We first recall the general principles of the ALNS and then present the ALNS used at the first stage to solve the DTP. We then discuss the method used to solve the ADTP.

4.1. Overview of the ALNS

The underlying principle of the ALNS is that we can improve a solution by iteratively partially destroying and repairing it. ALNS relies on heuristic operators that either destroy the solution (by removing producers from routes) or repair it (by reinserting producers). Algorithm 2 presents an overview of

the ALNS.

Algorithm 2 ALNS

Require: *InitialSolution*

```
1: BestSolution  $\leftarrow$  InitialSolution
2: CurrentSolution  $\leftarrow$  InitialSolution
3: while the termination criterion is not satisfied do
4:   Select a Destroy and a Repair operator based on past performance
5:   S  $\leftarrow$  CurrentSolution
6:   S  $\leftarrow$  Destroy(S)
7:   S  $\leftarrow$  Repair(S)
8:   if S  $\prec$  BestSolution then
9:     BestSolution  $\leftarrow$  S
10:    CurrentSolution  $\leftarrow$  S
11:   else
12:     if AcceptanceCriterion(S, CurrentSolution) then
13:       CurrentSolution  $\leftarrow$  S
14:     end if
15:   end if
16: end while
17: return BestSolution
```

The algorithm starts from an initial solution. At each iteration we select a destroy operator and a repair operator from a pool of operators and use them to create a new solution by modifying the current one (line 4). The selection is done via a roulette wheel algorithm, which uses a score computed for each operator based on its performance in the most recent iterations. See Ropke and Pisinger (2006) for a description of the operator-selection mechanism.

In our implementation the solution is destroyed by removing some of its producers (line 6), and the repair step reinserts them (line 7). If the resulting solution is worse than the current solution, an acceptance criterion (e.g., simulated annealing (Kirkpatrick et al., 1983) or record-to-record travel (Dueck, 1993)) is used to determine whether the new solution should replace

the current one (lines 12–13). We use simulated annealing for the acceptance criterion and when the termination criterion is satisfied, the algorithm returns the best solution encountered (line 17).

4.1.1. *ALNS for the DTP*

We now describe the objective function and operators used by the ALNS to solve the DTP.

Objective Function

One of the complications of the DTP is that there are bounds on the quantity of milk received by each plant. When we use a construction heuristic the solution may not satisfy these constraints until the last producers are inserted. To drive the search toward a feasible solution we penalize the plants that do not receive the desired quantity of milk. The objective function is therefore the sum of the distance traveled by the fleet plus the penalty due to undersupply or oversupply of milk at the plants.

Destroy operators

We define three destroy operators described as follows:

1. *Random removal*: This operator selects at random the producers to be removed and helps to ensure proper diversification.
2. *Worst removal*: This operator removes the nodes for which removal gives the greatest savings. After each removal, we reevaluate the savings due to the removal of the other nodes of the current route.
3. *History removal*: This heuristic is an adaptation of the two history-removal heuristics proposed by Pisinger and Ropke (2007). The goal is to remove the nodes that are currently poorly placed in comparison with their

locations in the best-known solutions. For the best fifty solutions, let $B_{p,p'}$ be the number of times that node p is directly followed by node p' in a route. Let $p-1$ be the predecessor of node p in the current solution and $p+1$ its successor. For each node p' we define a score $S_{p'}$ as follows: $S_{p'} = B_{p'-1,p'} + B_{p',p'+1}$. Low values of $S_{p'}$ indicate nodes that are more likely to be removed.

Repair operators

Two repair operators are used and are described as follows:

1. *Best insertion*: At every iteration, we consider every route to compute the best insertion cost for every node that is not yet inserted. The node for which insertion gives the smallest increase in the objective function is inserted in its best position.

2. *K-Regret*: This heuristic relies on the notion of regret, which is used for example by Potvin and Rousseau (1993) for the VRPTW. Let U be the set of nodes not yet inserted, and for each $i \in U$ let Δf_i^j be the cost of inserting i in the j^{th} best route in its best position. At every iteration, we set $i^* = \arg \max_{i \in U} (\sum_{j=2}^k \Delta f_i^j - \Delta f_i^1)$ and insert the corresponding node in its best position. The heuristic stops when U is empty or none of the remaining nodes can be inserted. We consider values of k in the range 2 to 5.

4.1.2. Efficient handling of the draft limit in the insertion heuristics

Our repair operators rely on an insertion heuristic, which must quickly compute whether or not the insertion of a node in a given position is consistent. We will not discuss vehicle capacity constraints, which can easily be verified in constant time, but we will explain how the draft limit can also be verified in constant time. This feasibility check is inspired by the

forward-time slack approach (Savelsbergh, 1992) used to efficiently handle time-window constraints in the VRPTW. We first introduce some notation.

Let:

- N_k : be the set of nodes currently served by route $k \in K$;
- $\bar{L}_i = \max(L_i, \bar{L}_{s(i)} - m_i)$: be the propagated maximum load at node i where $s(i)$ indicates the successor of i if it has one; otherwise $\bar{L}_i = L_i$. \bar{L}_i is the maximum load that a vehicle entering node i can have without violating the draft limit on i or one of its successors;
- \bar{q}_i : be the load of the vehicle visiting node i after its visit.

When we evaluate the insertion of a node j after node i and before node i' , we must check that $\bar{q}_i + m_j \leq L_j$ and $\bar{q}_i + l_j + l_i \leq \bar{L}_{i'}$. When an insertion is actually performed, the values of \bar{L} and \bar{q} must be updated for all the nodes of the route; this update is done in linear time. Moreover, the number of actual insertions is likely to be much less than the number of insertion evaluations. Therefore, this method is more efficient than doing a linear-time check whenever we check the consistency of an insertion.

4.1.3. Phase 2: Assignment-based reoptimization

The set of routes produced by the DTP is consistent, in the sense that the producers visited by a given route are likely to be close to one another. However, the last producer visited has been selected to minimize the delivery cost to a given processing plant, that associated with the route in the DTP solution. In the ADTP a route is unlikely to visit the same plant every day of the year. Selecting a producer farther from the plant visited in the

DTP solution but closer on average to the plants usually visited in the ADTP solution can lead to a more efficient solution. We therefore propose a method that explores the neighborhood of the DTP solution to identify a better solution for the ADTP.

We compute variations of each route of the DTP solution. For each producer served by the route, we build a route where this producer is the last served, and we carry out a local search to minimize the pickup cost of this route. We then use this pool of variations to form a set partitioning problem that selects the routes that give the most efficient solution.

Let's introduce: Ω containing the DTP route variations, Ω_k containing the route variations performed by vehicle $k \in K$, c_ω is the pickup cost of route $\omega \in \Omega$, c_ω^p is the cost for route $\omega \in \Omega$ to visit plant $p \in P$, and q^ω is the quantity of milk transported by route $\omega \in \Omega$. The binary variable z_ω indicates whether or not route $\omega \in \Omega$ is used in the solution.

$$\min \sum_{\omega \in \Omega} z_{\omega} c_{\omega} |D| + \sum_{k \in K, d \in D} b_d^k \quad (20)$$

s.t.

$$\sum_{\omega \in \Omega_k} z_{\omega} = 1 \quad \forall k \in K \quad (21)$$

$$l_{p,\omega}^d \leq y_{p,d}^{\omega} q^{\omega} \quad \forall d \in D, p \in P, \omega \in \Omega \quad (22)$$

$$\sum_{p \in P} l_{p,\omega}^d = z_{\omega} q^{\omega} \quad \forall d \in D, \omega \in \Omega \quad (23)$$

$$\sum_{\omega \in \Omega} l_{p,\omega}^d \geq \xi_p \sum_{d' \in D_w, \omega \in \Omega} l_{p,\omega}^{d'} \quad \forall p \in P, w \in W, d \in D_w, \omega \in \Omega \quad (24)$$

$$\sum_{d \in D_w, \omega \in \Omega} l_{p,\omega}^d \leq (1 + \delta_p) d_p^w \quad \forall p \in P, w \in W \quad (25)$$

$$\sum_{d \in D_w, \omega \in \Omega} l_{p,\omega}^d \geq (1 - \delta_p) d_p^w \quad \forall p \in P, w \in W \quad (26)$$

$$b_d^k \geq \sum_{\omega \in \Omega_k} y_{p,d}^{\omega} c_{\omega}^p \quad \forall k \in K, d \in D, p \in P \quad (27)$$

$$z_{\omega} \in \{0, 1\} \quad \forall \omega \in \Omega \quad (28)$$

$$y_{p,d}^{\omega} \in \{0, 1\} \quad \forall \omega \in \Omega, p \in P, d \in D \quad (29)$$

$$b_d^k \in \mathbb{R}^+ \quad \forall k \in K, d \in D \quad (30)$$

The objective function (20) minimizes the total cost of the routing plan. Constraint (21) ensures that each vehicle is assigned to a single route. Constraint (22) makes sure that a route delivers milk to a plant only if it visits that plant. Constraint (23) enforces that the selected routes deliver all the milk they have collected. Constraints (24)–(26) requires that the quantities of milk received by the plants each day and each week are close to the

requested quantities. Constraint (27) determines the delivery cost for each vehicle and each day. Finally, constraints (28)–(30) are domain constraints.

4.2. Clustering of the weeks

The ADTP is a large size problem: we must plan the routes for 365 days. For large real-life instances it takes more than thirty minutes to compute the optimal assignment of a predetermined set of routes to the plants for the fifty-two weeks of the year. However, many of the weeks have similar plant demands. We can therefore carry out the optimization for some representative weeks, associating to each week a weight that corresponds to the number of similar weeks it represents.

To determine how to group the weeks we solve a clustering problem. Our goal is to divide a set of $|W|$ observations (the weeks) composed of $|P|$ dimensions (the plants) into a given number of clusters, such that the intra-cluster sum of squared distances is minimized. To build these clusters we use the k-means algorithm of Hartigan and Wong (1979). Figure 2 shows for a given instance the optimal objective value for different numbers of clusters. In our computational experiments we will study how the number of clusters impacts the cost of the solution.

5. Computational Experiments

In this section we present the results of several computational experiments. We first compare the ALNS developed to solve the DTP in the first stage with the state-of-the-art method, GUTS, developed by Lahrichi et al. (2014). We then investigate the impact of the number of clusters used. The

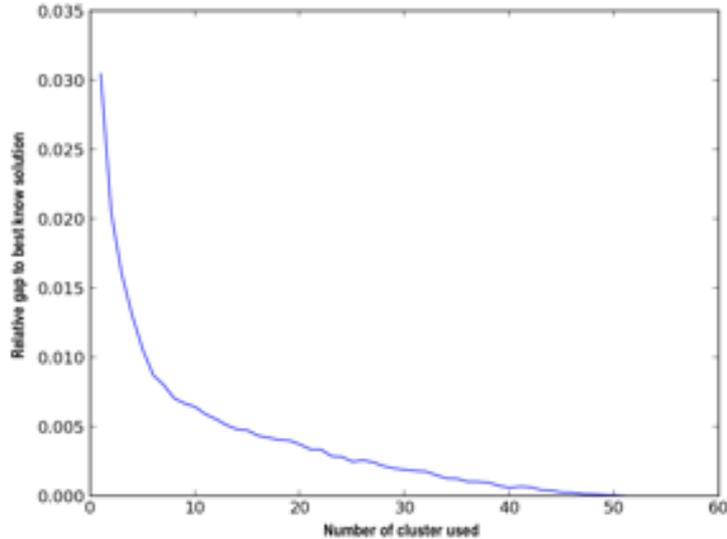


Figure 2: Optimal objective value for different numbers of clusters

experiments are performed on an Intel Xeon X5675 computer with 96 Go of RAM running Red Hat Enterprise Linux Server release 6.4. The code is written in C++ and compiled using g++ 4.7.2. Cplex 12.5.0.1 is used to solve the mixed integer problems, and it is restricted to a single thread.

Note that the total annual routing cost is around \$70 million. Therefore, if 1% can be saved on every contract, it represents a total saving of \$700,000 per year.

5.1. Performance of ALNS on the DTP

Since no benchmark instances are available for the DTP, we built a set of 23 instances by adapting the multi-depot vehicle routing instances of Cordeau et al. (1997). There are two classes of instances, A and B:

- The instances of class A have a total vehicle capacity close to the actual milk production;

- the instances of class B have more capacity but there is a much smaller tolerance on the difference between the milk quantity requested and the quantity received.

The reader may refer to Cordeau et al. (1997) for the details on the number of depots, vehicles and distances.

For these tests we used a five-minute time limit imposed by the application. The planning is done once a year, but on a contract basis (not every contract is renewed at the same time). The planner has just one day in which to evaluate the various scenarios and cannot afford to spend more than five minutes solving each possibility.

The class-A results are presented in Table 1 and the class-B results are presented in Table 2. For instance 11 in class A neither GUTS nor ALNS finds a feasible solution; we omit this instance from Table 1.

The ALNS clearly outperforms GUTS on class A and is more efficient on class B, however the improvement is smaller.

5.2. Performance on the Annual DTP

In order to assess the efficiency of this approach, we evaluate two sets of instances are used. The first set of instance consists of 12 randomly generated instances for which various parameters were used. The second set of instances consists of 4 real life instances provided by the PLQ.

5.2.1. Performance on generated instances

Once again no benchmark instances are available for the ADTP, we built a set of twelve instances to analyze the impact of the number of clusters.

Table 1: Results for class-A instances

Instance	ALNS		GUTS	
	Min.	Avg.	Min.	Avg.
1	1065.91	1066.89	1115.09	1117.71
2	1080.38	1086.70	1089.7	1110.246
3	1562.67	1568.38	1670.39	1686.13
4	1687.04	1692.43	1784.75	1795.926
5	1670.69	1672.82	1760.08	1772.69
6	1681.13	1691.81	1794.97	1832.216
7	1672.15	1685.82	1779.86	1811.882
8	5146.52	5166.81	5397.09	5471.694
9	5240.68	5265.35	5689.15	5784.458
10	5307.72	5364.55	6108.49	6233.056
12	2142.65	2162.05	2217.02	2265.432
13	2055.12	2067.75	2155.93	2200.466
14	2102.48	2105.60	2263.16	2293.874
15	3794.30	3814.27	3994.96	4079.89
16	3768.36	4000.00	4669.51	4925.426
17	3570.72	3591.21	3825.47	4037.146
18	6034.73	6077.18	6563.33	6663.114
19	5319.37	5397.75	5970.53	6058.686
20	5405.98	5459.60	6035.76	6146.382
21	9723.61	9751.44	11105.3	11285.56
22	9666.96	9748.29	10912.5	11100.74
23	9633.21	9697.34	10751.0	10858.18
Avg.	4060.56	4097.00	4484.27	4569.59

Table 2: Results for class-B instances

Instance	ALNS		GUTS	
	Min.	Avg.	Min.	Avg.
1	964.63	968.10	964.63	967.48
2	980.61	984.86	1008.29	1028.38
3	1406.46	1414.74	1415.67	1425.59
4	1543.78	1546.59	1567.36	1570.13
5	1534.39	1534.39	1548.16	1551.79
6	1548.53	1549.74	1553.52	1564.72
7	1526.95	1528.67	1530.35	1538.26
8	4778.37	4793.71	4905.92	5014.02
9	5173.91	5188.77	5280.39	5487.32
10	4783.34	5842.87	5237.88	5377.55
11	5009.67	6027.86	5095.22	5196.67
12	1949.68	1950.42	1950.22	1954.86
13	1974.73	1974.73	1975.49	1982.19
14	1939.04	1943.82	1944.12	1959.93
15	3265.50	3279.78	3420.62	3495.61
16	3071.42	3075.91	3146.86	3276.52
17	3171.16	3173.01	3315.51	3335.37
18	5768.20	5773.37	5908.74	5958.55
19	5736.59	5749.26	6006.70	6175.59
20	5212.73	5226.31	5529.20	5625.27
21	8899.43	8950.87	9653.72	9680.85
22	8790.15	9058.66	9222.31	9277.82
23	8537.60	8580.62	9154.89	9286.91
Avg.	3807.26	3918.13	3971.12	4031.80

Table 3 gives the characteristics of the twelve generated instances used for this experiment. We consider 100 producers and 2 depots. We use as many vehicles as needed to answer the demand, considering that only 80% of the capacity of the vehicles is used. We vary the number of plants and the variability of their demand, and how producers are located in the territory. Three distinct producers distributions are considered:

- R in which the producers are randomly distributed,
- C where the producers are clustered and
- RC where half the producers is randomly distributed while the other half is clustered.

We consider two settings for the variability of the demand of the plants:

- L, where the variability is *low*: demands follow a normal distribution with a standard deviation equal to 10% of the mean;
- H, where the variability is *high*: standard deviation is equal to 100% of the mean.

Table 3: Generated instances characteristics

Instance	I-1	I-2	I-3	I-4	I-5	I-6	I-7	I-8	I-9	I-10	I-11	I-12
Plants	3	3	3	3	3	3	4	4	4	4	4	4
Producer distribution	R	C	RC	R	C	RC	R	C	RC	R	C	RC
Demand Variability	L	L	L	H	H	H	L	L	L	H	H	H

As mentioned earlier in the paper, solving the problem for 365 days is very time consuming. Therefore we recommend clustering. As the number of clusters increases, we get the closer to the original problem specification,

but each iteration will take longer. Given the same time limit, the milk collection is likely to be optimized faster when the number of clusters is smaller, although the demand considered during the optimization is not as close to the actual demand over the planning horizon. Note that the optimal solution of problems with differing numbers of clusters can be associated with the same set of pickup routes. In this case it is clearly better to solve the problem with as few clusters as possible. Results will thus be presented with a varying number of clusters.

Using the results of the first phase that designs the routes to collect the milk, we compare the results obtained by the algorithm with a single cluster to those with 5, 10, 15 and 20 clusters. We have used running time as stopping criteria rather than the number of iterations. Indeed, the running time of each iteration grows with the number of clusters. Results are presented in Tables 4.

Table 4: Comparison of the Two Stage Method to the DTP results on generated instance: Improvement (degradation of solution quality, by number of cluster, over single cluster results)

	Clusters			
	5	10	15	20
I-1	4.93%	1.59%	0.35%	0.32%
I-2	-1.00%	-3.70%	-3.97%	-6.15%
I-3	0.29%	-0.10%	-0.97%	-1.49%
I-4	2.29%	1.71%	-3.48%	-3.48%
I-5	0.92%	-0.43%	0.15%	-0.43%
I-6	0.64%	-1.16%	-4.88%	-4.31%
I-7	0.58%	-0.75%	-9.48%	-8.30%
I-8	-0.88%	-1.90%	-5.15%	-8.55%
I-9	-0.14%	-6.02%	-4.29%	-6.24%
I-10	1.74%	-1.72%	-1.86%	-1.86%
I-11	6.36%	-4.49%	-6.50%	-5.03%
I-12	2.60%	-7.56%	-6.78%	-5.06%

It appears clearly that the best results are obtained when we use 5 clus-

ters. Up to 6% savings can be obtained when comparing to the results obtained with one single cluster. We also note that on instances I-2, I-8 and I-9, the results with the single cluster are better but only by 1.00% in the worst case. One may note that results get worse when the number of clusters increases, this is due to allowed computing time which is set to 3 hours.

5.2.2. Performance of the two-stage method on real-life instances

We evaluated the performance of our approach using real data provided by the PLQ to evaluate the savings when our two-phase approach is used to solve the ADTP. Table 5 gives the characteristics of the four instances used for these experiments.

Table 5: Real life instance characteristics

Instance	I-165	I-210	I-224	I-280
Producers	139	94	40	229
Plants	2	5	2	7
Vehicles	10	10	5	23
Depots	4	4	2	11

The number of producers varies from 40 to 229, number of plants from 2 to 7, number of vehicles from 5 to 23 and number of depots from 2 to 11.

To evaluate our two-stage method, we first compare the solutions obtained by solving only the first stage. This is equivalent to comparing our method to the current practice, which solves the DTP although the vehicles are likely to serve more than one plant during the year. Table 6 gives the percentage improvement achieved by our method for five to twenty clusters. Depending on the instance and the number of clusters, the savings achieved by the second stage range from negligible to a few percentage points.

Table 6: Comparison of two-stage and DTP results on real-life instances

	Clusters			
	5	10	15	20
I-165	-0.21%	1.51%	0.97%	-1.97%
I-210	0.00%	0.00%	0.01%	0.00%
I-224	0.00%	0.07%	0.00%	0.00%
I-280	1.42%	1.42%	1.43%	1.47%

6. Conclusions

We have developed a two-stage method to solve a strategic vehicle routing problem arising in the dairy industry. We have shown that for some instances, considering the different demands of different weeks can lead to significant savings in comparison with the approach currently used. We have also shown that the first-stage ALNS for the DTP outperforms the GUTS algorithm previously proposed for this problem. Average savings with the new approach to solve the DTP are as considerable as 4% on 23 instances. We have also shown that using the weekly information during the annual design of routes leads to additional savings.

Extensions of this work could involve removing some or all of the simplifications that we have made. First, we have assumed that the cost of a route is linearly dependent on the number of kilometers traveled. The true cost is given by a nonlinear formula that depends on the total distance driven, the number of stops, the quantity of milk transported, and the number of kilometers below a predefined bound between two consecutive producers. Second, we have assumed that the production of each farm is constant throughout the year.

We have studied real-life instances. However, the position of the plant and the variability of the demand may have a large impact on the effectiveness

of the second stage. The instances studied in this article may not have had sufficient variety. It would therefore be interesting to study real-life instances for which these parameters vary considerably to investigate their impact on the efficiency of the method. Finally, it would be interesting to integrate the computation of the variations and the set partitioning method within a heuristic framework that iteratively adjusts the allocation of customers to routes. We would thus no longer require each route to serve the customer that it serves in the DTP solution.

Acknowledgments

Work supported by Mitacs acceleration, Prompt Québec and Producteurs de Lait du Québec.

References

- Chuda Basnet, Leslie R. Foulds, and John M. Wilson. A decision aid for milk tanker run allocation. Journal of the Operational Research Society, 48:786–792, 1997.
- Chuda Basnet, Leslie R. Foulds, and John M. Wilson. An exact algorithm for a milk tanker scheduling and sequencing problem. Annals of Operations Research, 86:559–568, 1999.
- Massimiliano Caramia and Francesca Guerriero. A milk collection problem with incompatibility constraints. Interfaces, 40(2):130–143, 2010.
- G.D.H. Claassen and Th.H.B Hendricks. An application of special ordered

- sets to a periodic milk collection problem. European Journal of Operational Research, 180:754–769, 2007.
- Jean-François Cordeau, Michel Gendreau, and Gilbert Laporte. A tabu search heuristic for periodic and multi-depot vehicle routing problems. Networks, 30(2):105–119, 1997.
- Jean-François Cordeau, Gilbert Laporte, and Anne Mercier. A unified tabu search heuristic for vehicle routing problems with time windows. Journal of the Operational research society, pages 928–936, 2001.
- A.E. Dooley, W.J. Parker, and H.T. Blair. Modelling of transport costs and logistics for on-farm milk segregation in new zealand dairying. Computers and Electronics in Agriculture, 48:75–91, 2005.
- Moshe Dror, Dominique Fortin, and Catherine Roucairol. Redistribution of self-service electric cars: a case of pickup and delivery. Technical report, INRIA, 1998.
- Gunter Dueck. New optimization heuristics: The great deluge algorithm and the record-to-record travel. Journal of Computational Physics, 104:86–92, January 1993. ISSN 0021-9991. doi: 10.1006/jcph.1993.1010.
- Patrik Flisberg, Bertil Lidén, and Mikael Rönnqvist. A hybrid method based on linear programming and tabu search for routing of logging trucks. Computers & Operations Research, 36:1122–1144, 2009.
- L.R. Foulds and J.M. Wilson. A variation of the generalized assignment

- problem arising in the new zealand dairy industry. Annals of Operations Research, 69:105–114, 1997.
- Michel Gendreau and Christos D Tarantilis. Solving large-scale vehicle routing problems with time windows: The state-of-the-art. Technical report, 2010.
- B.L. Golden, A.A. Assad, and E.W. Wasil. Routing vehicles in the real world: Applications in the solid waste, beverage, food, dairy, and newspaper industries. In P. Toth and D. Vigo, editors, The Vehicle Routing Problem, pages 245 – 286. SIAM, Philadelphia, PA, 2002.
- J.A. Hartigan and M.A. Wong. Algorithm as 136: A k-means clustering algorithm. Journal of the Royal Statistical Society, Series C, 28(1):100–108, 1979.
- Hipólito Hernández-Pérez and Juan José Salazar-González. Heuristics for the one-commodity pickup-and-delivery traveling salesman problem. Transportation Science, 38(2):245 – 255, 2004.
- Hipólito Hernández-Pérez, Inmaculada Rodríguez-Martín, and Juan José Salazar-González. A hybrid grasp/vnd heuristic for the one-commodity pickup-and-delivery traveling salesman problem. Computers & Operations Research, 36(5):1639 – 1645, 2009.
- S. Kirkpatrick, C. D. Gelatt, and M. P. Vecchi. Optimization by simulated annealing. Science, 220:671–680, 1983.
- Nadia Lahrichi, Teodor Gabriel Crainic, Michel Gendreau, Walter Rei, and

- Louis-Martin Rousseau. Strategic analysis of the dairy transportation problem. Journal of Operational Research Society, 66:44–56, 2014.
- David Pisinger and Stefan Ropke. A general heuristic for vehicle routing problems. Computers & Operations Research, 34(8):2403 – 2435, 2007. ISSN 0305-0548. doi: DOI: 10.1016/j.cor.2005.09.012.
- David Pisinger and Stefan Ropke. Large neighborhood search. In Michel Gendreau and Jean-Yves Potvin, editors, Handbook of Metaheuristics, volume 146 of International Series in Operations Research & Management Science, pages 399–419. Springer US, 2010. ISBN 978-1-4419-1665-5.
- Jean-Yves Potvin and Jean-Marc Rousseau. A parallel route building algorithm for the vehicle routing and scheduling problem with time windows. European Journal of Operational Research, 66(3):331 – 340, 1993. ISSN 0377-2217. doi: DOI: 10.1016/0377-2217(93)90221-8.
- J. G. Rakke, M. Christiansen, K. Fagerholt, and G. Laporte. The traveling salesman problem with draft limits. Computers & Operations Research, 39(9):2161 – 2167, 2012.
- Stefan Ropke and David Pisinger. An adaptive large neighborhood search heuristic for the pickup and delivery problem with time windows. Transportation Science, 40:455–472, November 2006. ISSN 1526-5447.
- Jayaram K. Sankaran and Rahul R. Ubgade. Routing tankers for dairy milk pickup. Interfaces, 24(5):59–66, 1994.
- Martin WP Savelsbergh. The vehicle routing problem with time windows:

- Minimizing route duration. ORSA Journal on Computing, 4(2):146–154, 1992.
- Paul Shaw. Using constraint programming and local search methods to solve vehicle routing problems. In Proceedings of the 4th International Conference on Principles and Practice of Constraint Programming, pages 417–431, 1998.
- C.D. Tarantilis and C.T. Kiranoudos. A meta-heuristic algorithm for the efficient distribution of perishable foods. Journal of Food Engineering, 50: 1–9, 2001.
- C.D. Tarantilis and C.T. Kiranoudos. A flexible adaptive memory-based algorithm for real-life transportation operations. European Journal of Operational Research, 179:806–822, 2007.
- Fanggeng Zhao, Sujian Li, Jiangsheng Sun, and Dong Mei. Genetic algorithm for the one-commodity pickup-and-delivery traveling salesman problem. Computers & Industrial Engineering, 56(4):1642 – 1648, 2009.