

Please cite this paper as follows:

Fabien Cornillier, Gilbert Laporte, Fayed F. Boctor, Jacques Renaud, The petrol station replenishment problem with time windows, *Computers & Operations Research* 36 (2009) 919–935, <http://dx.doi.org/10.1016/j.cor.2007.11.007>

The petrol station replenishment problem with time windows

Fabien Cornillier^a, Gilbert Laporte^{a,b}, Fayez F. Boctor^a, Jacques Renaud^{a,*}

^a*Interuniversity Research Centre on Enterprise Networks, Logistics and Transportation (CIRRELT), Université Laval, Québec, Canada G1K 7P4*

^b*Canada Research Chair in Distribution Management, HEC Montréal, 3000 chemin de la Côte-Sainte-Catherine, Montréal, Canada H3T 2A7*

Abstract

In the *Petrol Station Replenishment Problem with Time Windows* (PSRPTW) the aim is to optimize the delivery of several petroleum products to a set of petrol stations using a limited heterogeneous fleet of tank-trucks. More specifically, one must determine the quantity of each product to deliver, the assignment of products to truck compartments, delivery routes, and schedules. The objective is to maximize the total profit equal to the sales revenue, minus the sum of routing costs and of regular and overtime costs. This article first proposes a mathematical formulation of the PSRPTW. It then describes two heuristics based on arc preselection and on route preselection. Extensive computational tests confirm the efficiency of the proposed heuristics.

Keywords: fleet management, fuel delivery, replenishment, routing and scheduling

1. Introduction

This article proposes a mathematical model and two heuristics for the *Petrol Station Replenishment Problem with Time Windows* (PSRPTW). This problem consists of optimizing the delivery of several petroleum products to a set of petrol stations which must be supplied once by a limited heterogeneous fleet of tank-trucks based at a terminal, within given time windows. This problem is motivated by a real case faced by a Quebec based transportation company. In North America most petrol companies subcontract the replenishment of their outlets to private transporters who are paid on a delivered quantity basis. The objective of the PSRPTW is to maximize the total profit, equal to the revenue which is a function of delivered quantities, minus the sum of routing costs and of regular and overtime costs. Decision variables specify how much of each product to deliver to stations subject to their minimum requirements, how to assign products to vehicle compartments, and how to design and schedule delivery routes. In this study, we formulate the PSRPTW as a mixed integer linear program. We then propose two heuristics based on the same formulation. The first heuristic consists of preselecting promising arcs and of solving the associated mathematical program to optimality. It can solve small instances of up to about 15

*Corresponding author

Email addresses: fabien.cornillier@cirrelt.net (Fabien Cornillier), gilbert@crt.umontreal.ca (Gilbert Laporte), fayez.boctor@fsa.ulaval.ca (Fayez F. Boctor), jacques.renaud@fsa.ulaval.ca (Jacques Renaud)

stations. The second heuristic makes a preselection of promising routes through a geographical decomposition method, and can be applied to larger instances.

A major difference between the PSRPTW and most vehicle routing problems with time windows is the loading component: in the PSRPTW one must simultaneously design vehicle routes and assign petroleum products to truck compartments for each trip. Related but different problems have been studied by a number of authors. Brown and Graves [1] have considered the problem of direct deliveries (i.e. single-customer trips) and time windows. Different algorithms have been proposed for other versions of the same problem without time windows. Brown et al. [2] and Malépart et al. [3] have generalized this problem by allowing the delivery of more than one station in a same trip. Heuristics for a multiperiod version of this problem have been developed by Taqa Allah et al. [4]. In Cornillier et al. [5], an exact algorithm is developed for a similar problem without time windows, where only one or two stations can be delivered within a same trip, while Avella et al. [6] have proposed a heuristic and an exact algorithm based on a route generation scheme and a branch-and-price algorithm to solve a similar problem. More recently, a heuristic for the multiperiod problem without time windows was put forward by Cornillier et al. [7] for the case where the number of stations on any given route is limited to two.

The remainder of this paper is organized as follows. The problem is defined and modeled in Section 2. The route generation procedure is described in Section 3. In Section 4, we propose two heuristics, one based on arc preselection, the other based on route preselection. Computational results are presented in Section 5 and conclusions follow in Section 6.

2. Problem definition and formulation

The PSRPTW can be formally defined as follows. Let $G = (V^*, A) = (V \cup \{0\}, A)$ be a directed graph where $V = \{1, \dots, n\}$ is the set of stations, vertex 0 is the terminal, and $A = \{(i, j) : i, j \in V^*, i \neq j\}$ is the arc set. Denote by t_{ij} the travel time associated with arc (i, j) and by s_i the service time of station i . The PSRPTW consists of maximizing a profit related function by designing delivery routes to replenish stations with a limited heterogeneous fleet of K tank-trucks based at the terminal. Service at station i must start and end within a given time window $[a_i, b_i]$ satisfying $b_i - a_i \geq s_i$. A working day contains H regular working hours which can be extended by using H' overtime hours. A regular wage rate applies to regular working time while a higher rate applies to overtime hours. Only the hours effectively worked are paid, i.e. the hours from the beginning of the vehicle first trip to the return of its last trip. The total variable cost is the sum of travel costs, regular and overtime wages. All trucks are assumed to travel at the same speed, and each is subdivided into several compartments of known capacities. Each petrol station has a given number of capacitated underground tanks. The PSRPTW consists of determining:

- the quantity of each product to be delivered to each station, which should lie between a minimum and a maximum;
- the loading of these products into vehicle compartments;
- feasible delivery routes to these stations;
- the selected routes assignment to available trucks;
- the departure time of each truck trip;

in order to maximize a profit function.

2.1. Assumptions

The following assumptions are made:

- only one working day is considered;
- the fleet is heterogeneous and limited;
- each station must be visited once and only once during the considered working day;
- several trips can be assigned to the same truck;
- each station requires delivery within a given time window;
- waiting time between stations is allowed;
- regular and overtime working hours are limited;
- regular and overtime wages are known and constant;
- only effectively worked hours are paid;
- the transporter is paid a given amount for each litre delivered which varies as a function of station location;
- the travel time between any two vertices (terminal and stations), service times at stations and loading times at the terminal are known;
- each station requires a minimum and a maximum quantity of one or more products that can be computed from initial inventories, expected consumptions, and the capacities of its underground tanks.

2.2. Mathematical formulation

Our mathematical model is based on the generation of all feasible routes a truck can follow. A route is feasible if it satisfies all time windows and constraints on delivered amounts. Given a route r , one can compute its earliest and its latest departure times, denoted by α_r and β_r , minimizing its total waiting time.

We first define the following parameters:

- ϕ regular wage per hour;
- ϕ' overtime wage per hour;
- α_r earliest departure time for route r ;
- β_r latest departure time for route r ;
- λ_r minimum duration of route r (including waiting time if any);
- a_{sr} a binary parameter equal to 1 if and only if station s is delivered within route r ;
- ρ_{rk} the profit of route r if performed by truck k . This parameter is equal to $-\infty$ if truck k is unable to carry out route r .

The decision variables are:

- x_{rkv} a binary variable equal to 1 if and only if route r corresponds to trip v of truck k ;
- d_{kv} the departure time of truck k for trip v ;
- h_k the number of regular working hours of truck k ;
- h'_k the number of overtime hours of truck k .

The model is then:

$$(PSRPTW) \quad \text{Maximize} \quad \sum_{(r,k,v)} \rho_{rk} x_{rkv} - \phi \sum_k h_k - \phi' \sum_k h'_k \quad (1)$$

subject to

$$\sum_{(r,k,v)} a_{sr} x_{rkv} = 1 \quad \forall s \quad (2)$$

$$\sum_r x_{rkv} \leq 1 \quad \forall(k, v) \quad (3)$$

$$\sum_r x_{rk,v+1} \leq \sum_r x_{rkv} \quad \forall(k, v) \quad (4)$$

$$\sum_r \alpha_r x_{rkv} \leq d_{kv} \leq \sum_r \beta_r x_{rkv} + M \left(1 - \sum_r x_{rkv} \right) \quad \forall(k, v) \quad (5)$$

$$d_{k,v+1} \geq d_{k,v} + \sum_r \lambda_r x_{rkv} \quad \forall(k, v) \quad (6)$$

$$d_{kv} - d_{k,1} + \sum_r \lambda_r x_{rkv} \leq h_k + h'_k \quad \forall(k, v) \quad (7)$$

$$h_k \leq H \quad \forall k \quad (8)$$

$$h'_k \leq H' \quad \forall k \quad (9)$$

$$d_{kv} \in \mathbb{N}^+ \quad \forall(k, v) \quad (10)$$

$$h_k \in \mathbb{N}^+, \quad h'_k \in \mathbb{N}^+ \quad \forall k \quad (11)$$

$$x_{rkv} \in \{0, 1\} \quad \forall(r, k, v). \quad (12)$$

In this formulation, the objective function (1) maximizes the total profit. Constraints (2) stipulate that each station is visited once and only once. Constraints (3) ensure that at most one route is assigned to the v^{th} trip of truck k . By constraints (4), trip $v + 1$ of truck k exists only if trip v exists. In constraints (5), M is a large positive number; these constraints require that route departure times lie within the computed windows $[\alpha_r, \beta_r]$. Constraints (6) state that trip departure occurs after the arrival time of the preceding trip. Since we do not know the number of trips of truck k , constraints (7) ensure that the total working hours, decomposed into regular and overtime hours is equal to the duration, calculated as the difference between its latest trip return time and its first trip departure time. Constraints (8) and (9) ensure that regular and overtime hours lie within the allowable limits.

Figure 1 illustrates a case with three routes r_1, r_2 and r_3 , where $[\alpha_1, \beta_1] = [1, 2]$, $[\alpha_2, \beta_2] = [1, 3]$, $[\alpha_3, \beta_3] = [0, 3]$, and $\lambda_1 = 1, \lambda_2 = 2$, and $\lambda_3 = 3$. There are two identical trucks generating the same profit, and the maximal working hours are $H = 3$ and $H' = 2$, for a total of five hours per truck. Figure 1a depicts an optimal solution using one overtime hour, while Figures 1b and 1c correspond to suboptimal solutions using two overtime hours. In Figure 1d, we show that the solution of Figure 1a cannot be improved by starting route r_1 at time $t = 0$ because r_1 would then need one additional waiting hour.

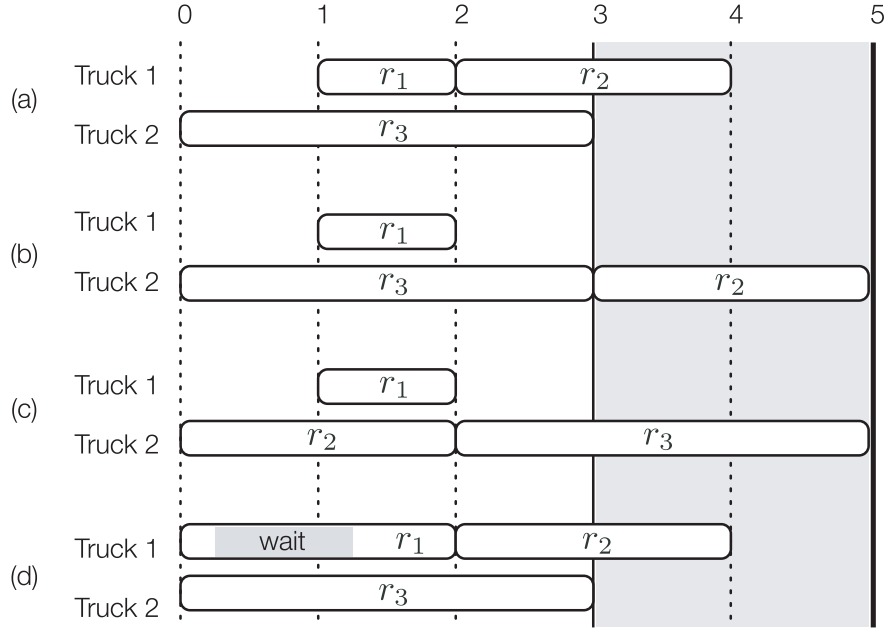


Figure 1: Illustration of the model with two tank-trucks and three routes

3. Route generation

In the above formulation, the number of potential routes is generally huge. We first propose ways of reducing the number of routes through feasibility and dominance criteria. A route can potentially visit as many stations as there are compartments in the truck. However, a two-stop limit per route is common practice in North America. This is explained by the fact that most trucks have from four to six compartments while stations generally require two or three products, one of which frequently requires two compartments. In this study, we consider the case where routes can visit up to four stations. If G is a complete graph, the number of feasible and infeasible routes visiting at most m stations is equal to $\sum_{i=1}^m \frac{n!}{(n-i)!}$ and can be rather large. Instead of making an explicit enumeration of all feasible and infeasible routes to be checked, we use an adaptation of Johnson's algorithm [8] to generate candidate routes from G , or a subgraph of G , from which all infeasible arcs are removed. These are arcs that cannot be included in a solution without violating a time or duration constraint. Given a directed graph, this algorithm consists of enumerating all or some of its elementary circuits. In our adaptation, routes are iteratively built starting from the terminal, station by station, until no more stations can be added without violating time or quantity constraints.

3.1. Infeasible arc deletion

In our problem, some infeasible arcs of G can be removed since some station pairs are incompatible in terms of time windows or requested quantities. Because these stations cannot belong to the same route, the number of feasible routes is reduced. We define the subgraph $G' = (V^*, A')$ of G where each arc of A' corresponds to a pair of compatible stations with respect to their time

windows. Also, for each truck k we define the subgraph $G'_k = (V^*, A'_k)$ of G' where each arc of A'_k corresponds to a pair of compatible stations with respect to their time windows and demand feasibility constraints. Demand feasibility of a route for a given truck is checked as shown in Section 3.2.

3.2. Demand feasibility check and quantity determination

A feasible route should allow the delivery of all minimal quantities required by its stations, and should visit these stations within the required time windows. In this section, we solve a *Tank-Truck Loading Problem* (TTLP) defined as follows. Let P be the set of demands of all stations on the route, and let g_p be the revenue associated with quantity q_p delivered to station p . This revenue is a function of the distance between the station at which the delivery takes place and the terminal. The TTLP consists of determining the quantity q_p to be delivered to each station p of the route in order to maximize the sum of revenues, while respecting the minimal and maximal requirements u_p and v_p , and without exceeding the capacity of any tank-truck compartment. Related problems using compartmented vehicles, generally referred to as *Loading Problems*, have been addressed with different objectives (Christofides et al. [9], Yuceer [10], Smith [11], Bukchin and Sarin [12]), and in different applications: bulk ship scheduling with flexible cargo holds (Fagerholt and Christiansen [13, 14]), livestock transportation (Oppen and Løkketangen [15]), grocery delivery (Eglese et al. [16]), and oil delivery (Brown et al. [2], Van der Bruggen et al. [17], Bausch et al. [18]).

In the PSRPTW, the loading problem can be formulated as follows. Let y_{pc} be a binary variable equal to 1 if demand p is assigned to compartment c , and 0 otherwise. Then the problem is:

$$(TTLP) \quad \text{Maximize} \quad \sum_{p \in P} g_p q_p \quad (13)$$

subject to

$$u_p \leq q_p \leq v_p \quad \forall p \quad (14)$$

$$q_p \leq \sum_c Q_c y_{pc} \quad \forall p \quad (15)$$

$$\sum_{p \in P} y_{pc} \leq 1 \quad \forall c \quad (16)$$

$$y_{pc} = 0 \text{ or } 1 \quad \forall (p, c). \quad (17)$$

In this formulation, the objective function maximizes the total revenue. Constraints (14) ensure that the delivered quantities lie between the requested minimum and maximum. Constraints (15) state that delivered quantity of demand p cannot be larger than the sum of compartment capacities in which it is loaded. By constraints (16), two distinct demands cannot be loaded in a same compartment.

This model is used to check the feasibility of each route with respect to a given truck and to obtain an optimal load by maximizing the corresponding revenue. Once the delivered quantities are known, one can compute the profit ρ_{rk} for route r and truck k , which is equal to the difference between the revenue generated by the delivered quantities and the travel cost. By convention, we set $\rho_{rk} = -\infty$ if truck k is unable to deliver the requirements of route r .

3.3. Route duration and departure window

Since the aim is to select a subset of feasible routes and to determine their optimal truck assignments and schedules, we must compute for each of these a time interval within which any departure time from the terminal minimizes the total duration including service time and waiting time, if any. Given a route r delivering all stations of the subset $V_r \subseteq V$, we index its stations according to the sequence in which they must be visited. Denote by $V_r^* = V_r \cup \{0\}$ the set of vertices including the terminal and all stations of route r . We check whether we can satisfy the time window constraint of each station and if so, we determine the departure window $[\alpha_r, \beta_r]$ for which the sum of waiting times is minimal. We then have to compute the sum w_r of all its necessary waiting times in order to determine its duration λ_r . First, for each vertex i we define a normalized time window $[a'_i, b'_i]$ representing the time interval within which the truck should leave the terminal in order to satisfy the time window constraint of station i if waiting times were not allowed:

$$[a'_i, b'_i] = \left[a_i - \sum_{u=0}^{i-1} t_{u,u+1} - \sum_{u=0}^{i-1} s_u, b_i - \sum_{u=0}^{i-1} t_{u,u+1} - \sum_{u=0}^i s_u \right]. \quad (18)$$

If waiting times were not allowed, the route would be feasible if and only if the intersection of all normalized time windows was not empty. However, waiting times may be needed in our problem and a new feasibility criterion is given by Proposition 1. The proofs of all propositions are given in the appendix.

PROP If waiting times are allowed, a route is feasible if and only if

$$\max_{0 \leq j < i} \{a'_j\} \leq b'_i, \quad \forall i \in V_r. \quad (19)$$

When a route r is feasible, we compute the sum w_r of all its minimal waiting times by means of Proposition 2. This waiting time is added to the sum of travel and service times in order to arrive at the route duration λ_r .

PROP If a route r is feasible, the sum of its minimal waiting times w_r is:

$$w_r = \max \left\{ 0, \max_{i \in V_r^*} \{a'_i\} - \min_{i \in V_r^*} \{b'_i\} \right\}. \quad (20)$$

Finally, we need to determine a departure window for the route r from the terminal, such that the time window constraint of each station is satisfied.

PROP If a route r is feasible, its departure time d_0 from the terminal has a time window $[\alpha_r, \beta_r]$ which minimizes the total waiting time, where

$$\alpha_r = \max_{i \in V_r^*} \{a'_i\} - w_r, \quad (21)$$

and

$$\beta_r = \min_{i \in V_r^*} \{b'_i\}. \quad (22)$$

Note that starting route r at any time before α_r only increases the embedded waiting times and does not allow the truck to return to the terminal earlier.

Proposition 1 is illustrated in Figure 2 where $b'_1 < \max\{a'_1, a'_2, a'_3\} = a'_2$; in this case there is no feasible solution. In Figure 3, $\max\{a'_1, a'_2, a'_3\} < \min\{b'_1, b'_2, b'_3\}$ and consequently, as implied

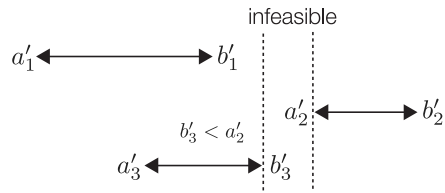


Figure 2: Three stations route with infeasible time windows

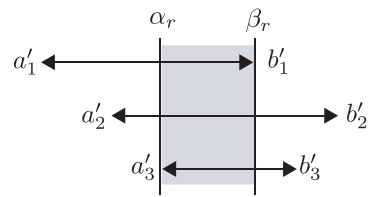


Figure 3: Three stations feasible route without waiting time

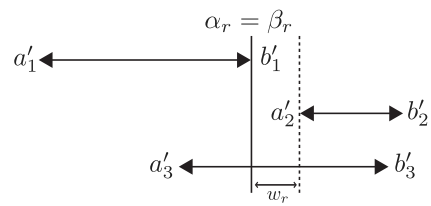


Figure 4: Three stations feasible route with waiting time

by Propositions 2 and 3, $w_r = 0$ and $\alpha_r \leq \beta_r$. Figure 4 shows the case where $w_r > 0$ and $\alpha_r = \beta_r$. In this case, the truck should leave the terminal exactly at β_r in order to minimize its waiting time w_r .

Knowing the total minimum waiting time of a route r , we are able to compute its total duration λ_r , including travel and service times:

$$\lambda_r = w_r + \sum_{u=0}^{n-1} (t_{u,u+1} + s_{u+1}) + t_{n,0}. \quad (23)$$

3.4. Route dominance

When several routes visit the same subset of stations but in a different order, we only retain Pareto optimal routes. More precisely, given two routes r_1 and r_2 , both visiting the same set of stations with the same truck, route r_1 can be eliminated if $\alpha_1 \geq \alpha_2, \beta_1 \leq \beta_2, \lambda_1 \geq \lambda_2$ and $\rho_1 \leq \rho_2$.

4. Heuristics

As the number of stations grows, the problem becomes more difficult to solve, even if there are fewer candidate routes, since the number of vehicles increases proportionally. In practice, it is often difficult to solve problems to optimality with more than 15 stations. For larger problems, we propose two heuristic procedures in which we solve the proposed mathematical model with only a preselected subset of all feasible routes instead of the whole set. The aim of the first heuristic is to reduce the number of routes by preselecting a subset of all feasible arcs of G'_k . It can be applied to relatively small instances. In the second heuristic, we decompose the geographical space in order to iteratively construct a candidate set of locally optimal routes which is then used to solve the global problem. This heuristic is more appropriate for larger instances.

4.1. A heuristic based on arc preselection

The first heuristic preselects an arc subset A''_k of A'_k . In the first version of the heuristic, this subset includes the arcs linking each vertex to its η nearest neighbours, where η is a parameter to be determined. Note that η is at most $n - 1$ because a station has $n - 1$ neighbours. In the second version, the arc subset includes all arcs of at most ν successive minimum spanning tree, where ν is a parameter; this way of reducing the arc set is inspired by the work of Helsgaun [19, 20] for the *Traveling Salesman Problem* and of Toth and Vigo [21] for the *Vehicle Routing Problem*. The procedure first generates a minimum spanning tree on the initial graph. It then removes the selected edges and repeats itself as long as the graph is connected. The value of ν can be at most $\lfloor n/2 \rfloor$ because the spanning trees sequential generation procedure uses $n - 1$ of the $n(n - 1)/2$ potential edges for each tree. In addition to these selected arcs, all arcs linking the terminal to customers in both directions are included. Nearest neighbours and minimum spanning trees are not based on distances, but on travel times. Since minimum spanning trees are constructed on undirected graphs, we construct them while setting the value of each edge equal to the minimum travel time for each of the two directions. All routes are generated from each A''_k as described in Section 3 and the optimal routes selection is made by solving the proposed PSRPTW model.

If the parameters η or ν are too small, the problem may be infeasible. On the other hand, for large instances and larger values of the parameters, the number of generated routes can become prohibitive and make the formulation unsolvable. We found that this heuristic becomes inefficient as soon as n reaches 20.

4.2. A decomposition heuristic based on route preselection

To solve larger instances, we propose a decomposition of the geographical space into sectors. Since any given decomposition would be arbitrary, we consider successive random partitions. Each generated random sector s corresponds to a different subset of stations $V^s \subset V$. The decomposition is such that no sector appears in two different partitions. Theoretically, any partition of V could be used, but since in practice distances are Euclidean, we have only used partitions induced by non-overlapping sectors centered at the terminal. Once the partitions are generated, the problem associated with each sector is solved exactly as a separate PSRPTW and, each time, the corresponding optimal routes are added to the preselected route set. Thus, the idea of this decomposition scheme is to generate a set of locally optimal routes which will be used to define the decision variables of the whole problem model.

4.2.1. Sector generation

Each sector includes a small number of stations, so that the associated problem can easily and quickly be solved to optimality while allowing the generation of good locally optimal routes (in our heuristic, sectors contain a random number of stations between five and ten). Each time a partition is generated, we make sure that none of its sectors has previously been selected. We iteratively generate new partitions until a given limit of κ preselected routes is reached, or until a given number of iterations have been executed. Note that in a non-Euclidean space, we would have to partition the set of stations in a different way, based for example on a measure of geographical and time windows distances. The rest of the method would otherwise be identical.

4.2.2. Optimal routes for a given sector

For each sector, we solve the corresponding PSRPTW to optimality by means of a branch-and-bound algorithm in order to generate a set of preselected routes. Since identical routes can be generated from different partitions, an exponential penalty on the number π_r of times a preselected route r has appeared in previous partitions is added to the objective function in order to prevent cyclic generation of routes from one partition to another. This penalty is equal to $\delta_r(\exp(\pi_r/2) - 1)$, where δ_r is proportional to the length of route r . The penalized objective function of the subproblem is then:

$$\text{Maximize } \sum_{(r,k,v)} (\rho_{rk} - \delta_r(e^{\pi_r/2} - 1))x_{rkv} - \phi \sum_k h_k - \phi' \sum_k h'_k. \quad (24)$$

To solve the subproblem associated with a sector V^s , we generate all its feasible routes using a restricted fleet in which $\lceil K|V^s|/n \rceil$ trucks are randomly chosen from the whole fleet. Each time a subproblem is solved, we add all of its optimal routes not already included to the set of preselected routes, up to the given limit of κ routes.

4.2.3. Recomposition procedure

After locally optimal routes have been extracted for all generated sectors, the recomposition procedure consists of determining the best routes in order to obtain a global solution to the whole PSRPTW. The resulting route selection problem (1)-(12) is solved using the entire fleet.

Figure 5 illustrates the route preselection heuristic. Figure 5a shows a first partition. The problems associated with all sectors (1.1), (1.2), (1.3) and (1.4) are independently solved to optimality, and the resulting optimal routes are added to the preselected routes set. Sectors (1.1) and (1.2) each give three locally optimal routes, and sectors (1.3) and (1.4) each yield only one

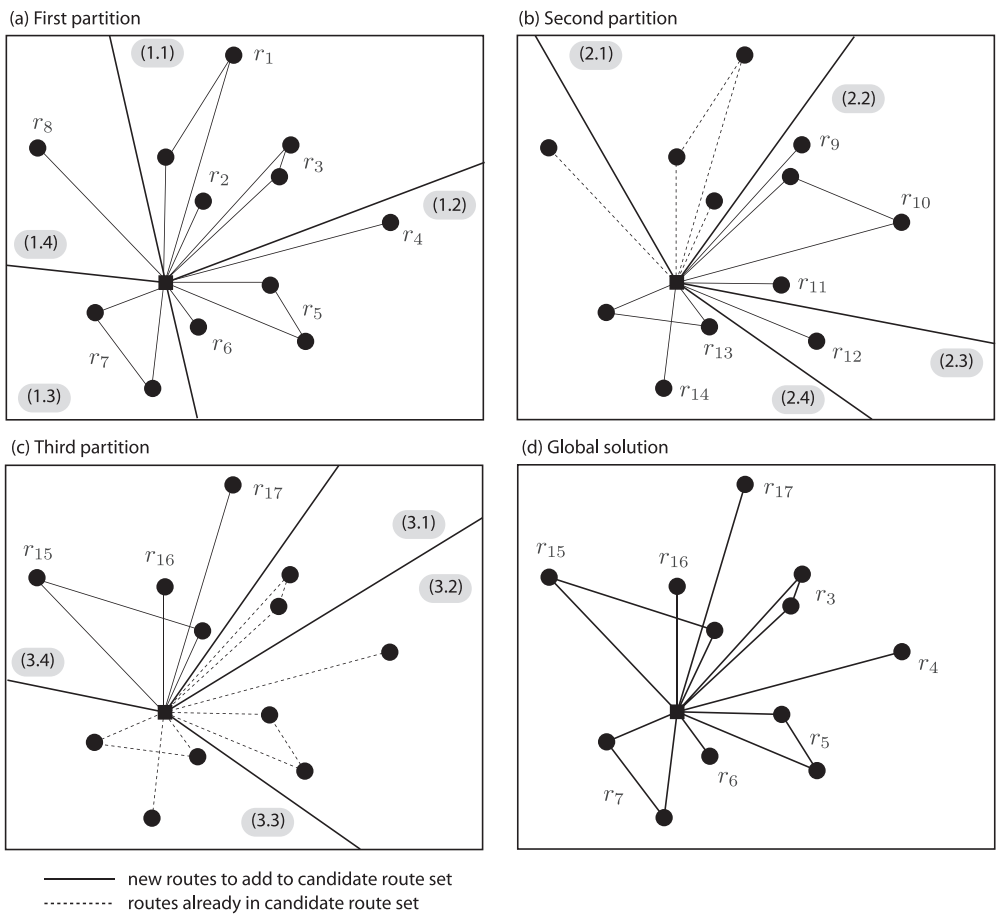


Figure 5: The route preselection heuristic using three successive partitions

route. We generate a second partition (Figure 5b) and a third one (Figure 5c) which respectively give five and three new locally optimal routes to add to the preselected routes set (r_9 to r_{14} for the second partition, and r_{15} to r_{17} for the third). After three successive partitions, there are 16 routes in the preselected routes set. The problem is then to select the best routes from the preselected route set in order to visit each station once and only once. In this example, the solution (Figure 5d) uses five routes from the first partition (r_3, r_4, r_5, r_6 and r_7), and three from the third one (r_{15}, r_{16} and r_{17}).

5. Computational results

The two heuristics just described were coded in Objective-C and used the CPLEX 10.0 Callable Library. They were run on dual AMD Opteron 250 processors 2.4GHz computers with Linux operating system. We present two sets of tests. We have first solved a set of randomly generated 15 stations instances in order to evaluate the performance of the arc preselection and route preselection heuristics. The route preselection heuristic was then assessed on randomly generated instances with 50 stations.

5.1. Test instances

We have generated test instances similar to real-life problems using a set of real data extracted from Malépart et al. [22]. From these data, we have determined a discrete random distribution on six categories of stations in function of their daily sales (Table 1). Station categories are randomly drawn from this distribution and daily sales are then randomly determined within the lower and upper limits of the obtained categories.

Table 1: Daily sales distribution

Category	Daily sales (litres)	Percentage (%)
1	0–1 350	21.7
2	1 350–2 700	22.6
3	2 700–5 400	29.8
4	5 400–8 100	13.6
5	8 100–10 800	6.2
6	10 800–16 200	6.1

The sales of regular, intermediate and super petrol grades are 76%, 7% and 17% of the total, respectively. Because daily sales and underground tank capacities are generally correlated, we present in Table 2 the typical observed tank capacities as a function of the total daily sales. For our test instances, the underground tanks configuration for each station is randomly selected among these three typical configurations. However, the probability of choosing the configuration corresponding to the station daily sales is 80% and the probability of choosing each of the other configurations is 10%.

We consider three tank-truck configurations among those commonly used in practice (Table 3).

Table 2: Underground tank typical configurations as a function of daily sales

Daily sales (litres)	Tank	Tank size (litres)
0–2 700	1	25 000
	2	15 000
	3	15 000
700–8 100	1	35 000
	2	22 700
	3	25 000
8 100–16 200	1	50 000
	2	25 000
	3	35 000

Table 3: Configurations of the tank-trucks

Type	Total capacity (1000 litres)	Number of compartments	Capacities (1000 litres)
1	60	6	17, 6, 10, 10, 7, 10
2	54	5	16, 6, 6, 10, 16
3	50	4	16, 8, 12, 14

The terminal coordinates are (50,50) for all instances, while stations coordinates are integer and randomly drawn from a uniform distribution in the 100km×300km Euclidean space. The fleet compositions as a function of the problem size are given in Table 4.

Table 4: Fleet compositions

Number of stations	Type I	Type II	Type III	Fleet size
15	2	2	1	5
50	8	5	5	18

In all tests, the number of stations that can be visited in a same route is limited to four, except in tests in which we evaluate the impact of this limit.

We have used the following data for all instances:

driver wage per regular working hour: \$15.00;
overtime hourly cost: \$30.00;
variable travel cost per kilometer: \$1.70;
average travel speed (km/h): 60;
truck loading time (minutes): 15;
station delivery time (minutes): 30;
daily regular working hours: 9;
daily maximum overtime hours: 3.

The revenue per delivered litre is a function of distance from the terminal. Rates are given in Table 5.

Table 5: Per litre revenue as a function of the distance from terminal

Distance	Revenue per delivered litre
0–50	\$0.004
50–100	\$0.007
100–150	\$0.010
150–200	\$0.013
> 200	\$0.016

5.2. Performance of the proposed heuristics

In this section, we study the performance of the proposed heuristics. We first analyze the results given by the arc preselection heuristic used. We then evaluate the impact of limiting the number of delivered stations per route. Finally, we analyze the performance of the route preselection heuristic. Average results over 20 instances of 15 or 50 stations are given.

5.2.1. Performance of the arc preselection heuristics

In Table 6, we evaluate the performance of the arc preselection heuristic which uses $\eta \in \{3, \dots, 6\}$ nearest neighbours on instances of 15 stations. The last row corresponds to the case where all arcs are selected and the solution is therefore optimal. We can see that with three nearest neighbours the profit of 504.24 is 98.88% of the optimum, and an optimal solution is found 13 times out of 20. With $\eta = 3$, the arc preselection heuristic generates only 299 of all 3 060 feasible routes, i.e. it eliminates 90.2% of all feasible routes limited to four stations; it also reduces computation time by 88.6%, from 350 to 40 seconds. A tangible improvement can be observed when four nearest neighbours are considered: we then attain 99.2% of the optimal profit while eliminating 81.5% of all feasible routes. An optimal solution is found in 90% of the cases. Further marginal improvements are obtained by using a larger number of nearest neighbours.

Table 6: Average results as a function of the number of nearest neighbours.

η	#CR	Dist	Qty	Rev	RT	OT	#Rtes	%Rtes(s)				Profit	#O	CPU
								1	2	3	4			
3	299	2 067	568 485	4 743	41.36	3.46	10.75	70.7	20.5	7.4	1.4	504.24	13	40
4	565	2 051	564 712	4 711	41.18	3.35	10.65	69.5	21.6	7.5	1.4	505.69	17	61
5	888	2 052	564 705	4 713	41.21	3.34	10.65	69.5	21.6	7.5	1.4	506.18	18	92
6	1 271	2 052	564 705	4 713	41.21	3.34	10.65	69.5	21.6	7.5	1.4	506.18	18	155
-	3 060	2 047	563 282	4 709	40.93	3.51	10.60	68.9	22.2	7.5	1.4	509.94	20	350

η number of nearest neighbours;
 #CR number of preselected routes;
 Dist distance travelled;
 Qty delivered quantity in litres;
 Rev revenue;
 RT regular hours used ;
 OT overtime hours used ;
 #Rtes number of selected routes in the solution;
 %Rtes(s) percentage of routes visiting s stations;
 Profit profit corresponding to the best solution;
 #O number of times the optimal solution has been obtained;
 CPU computing time in seconds;
 - all arcs are selected.

Table 7 shows the average results of the arc preselection heuristic using $\nu \in \{3, \dots, 6\}$ successive minimum spanning trees. We observe that the arc preselection heuristic based on the computation of three successive minimum spanning trees yields a profit equal to 99.3% of the optimum while eliminating 79.7% of all feasible routes. Further improvements are obtained if five successive minimum spanning trees are generated, yielding a profit equal to 99.83% of the optimum. This procedure uses 30.6% less than the CPU time needed to obtain an optimal solution. From Tables 6 and 7, we can see that there is no tangible performance difference between the two versions of the arc preselection heuristic for a similar number of candidate routes.

5.2.2. Impact of limiting the number of delivered stations per route

It is possible to reduce the number of generated routes by reducing the maximal number of stations per route. In this section, we evaluate the impact of this parameter. Average results are presented in Table 8. A significant improvement over the common practice discussed in

Table 7: Average results as a function of the number of minimum spanning trees.

ν	#CR	Dist	Qty	Rev	RT	OT	#Rtes	%Rtes(s)				Profit	#O	CPU
								1	2	3	4			
3	621	2052	564705	4713	41.21	3.34	10.65	69.5	21.6	7.5	1.4	506.18	18	54
4	1317	2052	564705	4713	41.21	3.34	10.65	69.5	21.6	7.5	1.4	506.18	18	156
5	2080	2050	564530	4714	41.04	3.47	10.65	69.5	21.6	7.5	1.4	509.06	19	243
6	2589	2050	564530	4714	41.04	3.47	10.65	69.5	21.6	7.5	1.4	509.06	19	287
–	3060	2047	563282	4709	40.93	3.51	10.60	68.9	22.2	7.5	1.4	509.94	20	350

ν maximum number of generated minimum spanning trees.

Section 3, which consists of limiting to two the number of stations per route, can indeed be obtained by increasing this limit. A relatively large profit improvement of 10.2% is obtained when we increase the limit to three stations per route. A marginal profit improvement of 0.58% is obtained by further raising this limit to four stations. We note that the optimal profit with a limit of three stations per route is slightly better than that obtained by the arc preselection heuristic with four successive minimum spanning trees or six nearest neighbors, when limiting the number of stations per route to four.

Table 8: Average results as a function of the maximal number of stations per route.

#S/R	#CR	Dist	Qty	Rev	RT	OT	#Rtes	%Rtes(s)				Profit	#O	CPU
								1	2	3	4			
2	140	2223	585279	5025	42.67	4.83	11.05	64.3	35.7	–	–	460.15	5	182
3	807	2062	564979	4732	41.44	3.27	10.65	68.5	22.1	9.4	–	507.01	17	113
4	3060	2047	563282	4709	40.93	3.51	10.60	68.9	22.2	7.5	1.4	509.94	20	350

#S/R maximal number of stations per route.

5.2.3. Performance of the route preselection heuristic

To evaluate the performance of the route preselection heuristic, we set κ as a multiple of the instance size: $\kappa \in \{45, 90, \dots, 315\}$ for the 15 stations instances, and $\kappa \in \{150, 300, \dots, 900\}$ for the 50 stations instances. In the last row, all feasible routes are generated, yielding the optimum. Table 9 shows the average results obtained with the route preselection heuristic for the 15 station instances. For three preselected routes per station (45 routes, i.e. 1.47% of all feasible routes), the profit is about 96.2% of the optimum. The largest improvement is obtained between six and 12 preselected routes per station (90 and 180 routes, i.e. 2.94% and 5.88%) with a profit of about 99.5% of the optimum.

Table 10 shows the average results of the route preselection heuristic for instances with 50 stations. For each instance, the allowed computation time was limited to two hours of CPU time. We can see that a major profit improvement of 4.9% can be obtained by increasing the number of preselected routes per station from three to nine (150 to 450 routes). A slight improvement can be obtained by further increasing this number, but we observe that the profit and the MIP best

Table 9: Average results as a function of the number of preselected routes for the 15 stations instances.

κ	Dist	Qty	Rev	RT	OT	#Rtes	%Rtes(s)				Profit	#O	CPU
							1	2	3	4			
45	2077	564 455	4 752	41.15	3.80	10.65	68.1	23.9	7.0	0.9	490.56	11	131
90	2082	567 680	4 764	41.25	3.81	10.70	68.2	24.3	6.5	0.9	491.08	12	241
135	2066	564 765	4 742	41.08	3.69	10.65	69.5	21.6	7.5	1.4	502.72	14	308
180	2070	568 161	4 753	41.25	3.60	10.70	69.6	22.0	7.0	1.4	507.32	18	472
225	2070	568 161	4 754	41.25	3.60	10.70	69.6	22.0	7.0	1.4	508.54	19	579
270	2070	568 161	4 754	41.25	3.60	10.70	69.6	22.0	7.0	1.4	508.54	19	792
315	2070	568 161	4 754	41.25	3.60	10.70	69.6	22.0	7.0	1.4	508.54	19	954
3 060	2 047	563 282	4 709	40.93	3.51	10.60	68.9	22.2	7.5	1.4	509.94	20	350

κ number of preselected routes.

Table 10: Average results as a function of the number of generated routes for the 50 stations instances.

κ	Dist	Qty	Rev	RT	OT	#Rtes	%Rtes(s)				Profit
							1	2	3	4	
150	7 003	1863 858	16 075	141.27	9.66	34.55	60.3%	34.6%	5.1%	0.0%	1760.54
300	6 862	1849 960	15 843	138.91	9.57	34.20	60.1%	33.9%	5.7%	0.3%	1807.58
450	6 686	1830 191	15 531	135.64	9.81	33.80	60.5%	32.0%	6.7%	0.9%	1835.91
600	6 632	1818 840	15 438	134.62	9.86	33.55	61.1%	30.1%	7.5%	1.3%	1848.42
750	6 542	1806 543	15 256	133.81	9.11	33.30	61.0%	29.6%	7.8%	1.7%	1853.71
900	6 571	1808 900	15 307	134.81	8.60	33.35	61.2%	29.4%	7.8%	1.6%	1856.91

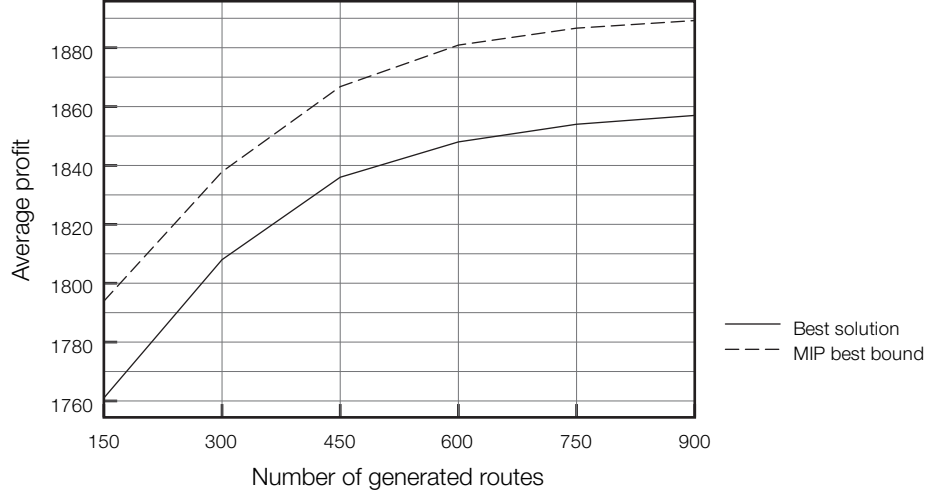


Figure 6: Average profits as a function of the number of generated routes for the 50 stations instances

bound grow in an asymptotic manner (Figure 6). When going from 15 to 18 preselected routes per station (750 to 900 routes), the profit improvement is only 0.17%. These results show that this route preselection heuristic is capable of generating a set of good preselected routes and can solve much larger instances than any of the two versions of the arc preselection heuristic.

6. Conclusions

We have proposed a mathematical formulation of the Petrol Stations Replenishment Problem with Time Windows. Based on this formulation, an arc preselection heuristic was developed in order to reduce the number of candidate routes. Computational results show that this heuristic considerably reduces computation time while yielding near-optimal solutions. For larger instances, a decomposition heuristic based on route preselection was proposed. On small instances, it was compared to an exact algorithm. Computational results show that this decomposition heuristic succeeds in finding near-optimal solutions while using a very small proportion of all feasible routes. The effect of generating more routes was analyzed on larger instances.

Appendix

PROOF. Proof of Proposition 1 Let $T_i = \sum_{u=0}^{i-1} t_{u,u+1}$, $S_i = \sum_{u=0}^i s_u$ and $Y_i = \sum_{u=0}^i y_u$, where $y_i \geq 0$ denotes a minimal waiting time between stations i and $i + 1$. Then $a'_i = a_i - T_i - S_{i-1}$ and $b'_i = b_i - T_i - S_i$. The route is feasible if and only if for each station i there exists a departure

time, denoted by $d_i \in \mathbb{R}$, such that

$$\forall i \in V_r^*, a_i + s_i \leq d_i \leq b_i, \quad (25)$$

which is equivalent to

$$\forall i \in V_r^*, \exists d_i \in \mathbb{N} : a'_i + T_i + S_{i-1} + s_i \leq d_i \leq b'_i + T_i + S_i \quad (26)$$

$$\Leftrightarrow \forall i \in V_r^*, \exists d_i \in \mathbb{N} : a'_i + T_i + S_i \leq d_i \leq b'_i + T_i + S_i \quad (27)$$

$$\Leftrightarrow \forall i \in V_r^*, \exists d_i \in \mathbb{N} : a'_i \leq d_i - T_i - S_i \leq b'_i. \quad (28)$$

But as $d_i = d_0 + T_i + S_i + Y_i$, we have

$$(28) \Leftrightarrow \forall i \in V_r^*, \exists (Y_i \geq 0, d_0 \in \mathbb{N}) : a'_i \leq d_0 + Y_i \leq b'_i. \quad (29)$$

Then, the route is feasible if and only if there exists $d_0 \in \mathbb{N}$ and $Y_i \geq 0$ such that for all $i \in V_r^*$:

$$a'_i - Y_i \leq d_0 \leq b'_i - Y_i. \quad (30)$$

For all $(i, j) \in (V_r^*)^2$ such that $j < i$, we need to show that there exists a sum of minimal non-negative waiting times between stations j and i , and a departure time $d_0 \in \mathbb{N}$ from the terminal such that

$$a'_j - Y_j \leq d_0 \leq b'_j - Y_j \text{ and } a'_i - Y_i \leq d_0 \leq b'_i - Y_i, \quad (31)$$

whenever $a'_j \leq b'_i$.

We get

$$\exists (Y_i - Y_j \geq 0, d_0 \in \mathbb{N}) :$$

$$\left[a'_j - Y_j \leq d_0 \leq b'_j - Y_j \text{ and } a'_i - Y_i \leq d_0 \leq b'_i - Y_i \right] \quad (32)$$

$$\Leftrightarrow \exists (Y_i - Y_j \geq 0) : \left[a'_j - Y_j \leq b'_i - Y_i \text{ and } a'_i - Y_i \leq b'_j - Y_j \right] \quad (33)$$

$$\Leftrightarrow \exists (Y_i - Y_j \geq 0) : a'_i - b'_j \leq Y_i - Y_j \leq b'_i - a'_j \quad (34)$$

$$\Leftrightarrow b'_i - a'_j \geq 0. \quad (35)$$

Thus $b'_i \geq \max_{0 \leq j < i} \{a'_j\}$. □

PROOF. Proof of Proposition 2 We have shown in the proof of Proposition 1 that a route is feasible if and only if there exists for all vertices i and $j > i$ a non-negative sum of waiting times $Y_j - Y_i$ between i and j within the interval $[a'_j - b'_i, b'_j - a'_i]$ (Eq. 34). Then there exists a non-negative sum of waiting times $w = Y_n$ which can be decomposed as follows:

$$w = Y_n \quad (36)$$

$$= (Y_{j_2} - Y_0) + (Y_{j_1} - Y_{j_2}) + (Y_n - Y_{j_1}), \quad (37)$$

where $j_1 = \arg \max_{i \in V_r^*} \{a'_i\}$ and $j_2 = \arg \min_{i \in V_r^*} \{b'_i\}$.

Since the route is feasible, for each $i < j_2$, there exists a non-negative value $Y_{j_2} - Y_i$ such that $a'_{j_2} - b'_i \leq Y_{j_2} - Y_i \leq b'_{j_2} - a'_i$, and we have $b'_{j_2} - a'_i \geq 0$. Since $a'_{j_2} - b'_{j_2} \leq 0$ is true by definition, $Y_{j_2} - Y_i$ can always take a zero value for each $i < j_2$ and a fortiori for $i = 0$. On the other hand, for each $i > j_1$, there exists a non-negative value $Y_i - Y_{j_1}$ such that $a'_i - b'_{j_1} \leq Y_i - Y_{j_1} \leq b'_i - a'_{j_1}$,

and we have $b'_i - a'_{j_1} \geq 0$. Since $a'_{j_1} - b'_{j_1} \leq 0$ is true by definition, $Y_i - Y_{j_1}$ can always take a zero value for each $i > j_1$ and a fortiori for $i = n$. Then, $w_r = Y_{j_1} - Y_{j_2}$ and, because the route is feasible, there exists $w \geq 0$ such that

$$a'_{j_1} - b'_{j_2} \leq w \quad (38)$$

$$\Leftrightarrow \max_{0 \leq i \leq n} \{a'_i\} - \min_{0 \leq i \leq n} \{b'_i\} \leq w \quad (39)$$

$$\Leftrightarrow \max_{i \in V_r^*} \{a'_i\} - \min_{i \in V_r^*} \{b'_i\} \leq w, \quad (40)$$

and

$$w_{min} = \max_{i \in V_r^*} \{a'_i\} - \min_{i \in V_r^*} \{b'_i\} \quad (41)$$

$$= w_r. \quad (42)$$

When $\max_{i \in V_r^*} \{a'_i\} - \min_{i \in V_r^*} \{b'_i\} \leq 0$, there exists a feasible time departure from the terminal which is common to all stations without waiting time. In this case, we have $w_r = 0$. \square

PROOF. Proof of Proposition 3 Since waiting time always delays the return to the terminal even if departure occurs ε time units before β_r , it is always preferable to start from the terminal as late as possible, i.e. at $d_0 = \beta_r = \min_{i \in V_r^*} \{b'_i\}$. Any other departure time $\beta_r - \varepsilon$ just makes the route duration ε longer. Then, if waiting is needed, we have $\alpha_r = \max_{i \in V_r^*} \{a'_i\} - w_r = \beta_r$. \square

Acknowledgements

This work was partially supported by the Canadian Natural Sciences and Engineering Research Council (NSERC) under grants OGP0036509, OGP0039682 and OGP0172633. This support is gratefully acknowledged.

References

- [1] G. G. Brown and G. W. Graves. Real-time dispatch of petroleum tank trucks. *Management Science*, 27:19–32, 1981.
- [2] G. G. Brown, C. J. Ellis, G. W. Graves, and D. Ronen. Real-time, wide area dispatch of mobil tank trucks. *Interfaces*, 17:107–120, 1987.
- [3] V. Malépart, F. F. Boctor, J. Renaud, and S. Labilois. Nouvelles approches pour l’approvisionnement des stations d’essence. *Revue Française de Gestion Industrielle*, 22:15–31, 2003.
- [4] D. Taqa allah, J. Renaud, and F. F. Boctor. Le problème d’approvisionnement des stations d’essence. *APII-JESA, Journal Européen des Systèmes Automatisés*, 34:11–33, 2000.
- [5] F. Cornillier, F. F. Boctor, G. Laporte, and J. Renaud. An exact algorithm for the petrol station replenishment problem. *Journal of the Operational Research Society*, forthcoming, 2007.
- [6] P. Avella, M. Boccia, and A. Sforza. Solving a fuel delivery problem by heuristic and exact approaches. *European Journal of Operational Research*, 152:170–179, 2004.
- [7] F. Cornillier, F. F. Boctor, G. Laporte, and J. Renaud. A heuristic for the multiperiod petrol station replenishment problem. Submitted for publication, 2006.
- [8] D. B. Johnson. Finding all the elementary circuits of a directed graph. *SIAM Journal on Computing*, 4:77–84, 1975.
- [9] N. Christofides, A. Mingozzi, and P. Toth. Loading problems. In P. Toth, N. Christofides, and C. Sandi, editors, *Combinatorial Optimization*, pages 339–369. Wiley, Chichester, 1979.
- [10] U. Yuceer. A multi-product loading problem: a model and solution method. *European Journal of Operational Research*, 101:519–531, 1997.

- [11] J. C. Smith. A genetic algorithm approach to solving a multiple inventory loading problem. *International Journal of Industrial Engineering*, 10:7–16, 2003.
- [12] Y. Bukchin and S.C. Sarin. Discrete and dynamic versus continuous and static loading policy for a multi-compartment vehicle. *European Journal of Operational Research*, 174:1329–1337, 2006.
- [13] K. Fagerholt and M. Christiansen. A combined ship scheduling and allocation problem. *Journal of the Operational Research Society*, 51:834–842, 2000.
- [14] K. Fagerholt and M. Christiansen. A travelling salesman problem with allocation, time window and precedence constraints — an application to ship scheduling. *International Transactions in Operational Research*, 7:231–244, 2000.
- [15] J. Oppen and A. Løkketangen. Finding solutions for the livestock collection and inventory problem using tabu search. *Computers & Operations Research*, forthcoming, 2007.
- [16] R. W. Eglese, A. Mercer, and B. Sohrabi. The grocery superstore vehicle scheduling problem. *Journal of the Operational Research Society*, 56:902–911, 2005.
- [17] L. Van der Bruggen, R. Gruson, and M. Salomon. Reconsidering the distribution structure of gasoline products for a large oil company. *European Journal of Operational Research*, 81:460–473, 1995.
- [18] D. O. Bausch, G. G. Brown, and D. Ronen. Scheduling short-term marine transport of bulk products. *Maritime Policy and Management*, 25:335–348, 1998.
- [19] K. Helsgaun. An effective implementation of the Lin-Kernighan traveling salesman heuristic. *Datalogiske Skrifter* No. 81, Roskilde University, 1998.
- [20] K. Helsgaun. An effective implementation of the Lin-Kernighan traveling salesman heuristic. *European Journal of Operational Research*, 126:106–130, 2000.
- [21] P. Toth and D. Vigo. The granular tabu search and its application to the vehicle routing problem. *INFORMS Journal on Computing*, 14:333–346, 2003.
- [22] V. Malépart, J. Renaud, and F. F. Boctor. La distribution des produits pétroliers au Québec : État de la situation. Technical report, Université du Québec, 1998.