CIRRELT

Centre interuniversitaire de recherche sur les réseaux d'entreprise, la logistique et le transport

Interuniversity Research Centre on Enterprise Networks, Logistics and Transportation

Solving a Real Vehicle Routing Problem in the Furniture and Electronics Industries

Jean-Philippe Gagliardi
Jacques Renaud
Angel Ruiz

June 2013

CIRRELT-2013-35

Document de travail également publié par la Faculté des sciences de l'administration de l'Université Laval, sous le numéro FSA-2013-008.

Bureaux de Montréal :
Université de Montréal
C.P. 6128, succ. Centre-ville
Montréal (Québec)
Canada H3C 3J7
Téléphone : 514 343-7575
Télécopie : 514 343-7121

Bureaux de Québec :
Université Laval
2325, de la Terrasse, bureau 2642
Québec (Québec)
Canada G1V 0A6
Téléphone : 418 656-2073
Télécopie : 418 656-2624

www.cirrelt.ca

Université Laval · UQÀM · HEC Montréal · École Polytechnique Montréal · Université de Montréal

Solving a Real Vehicle Routing Problem in the Furniture and Electronics Industries

Jean-Philippe Gagliardi*, Jacques Renaud, Angel Ruiz

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

Abstract. This article studies the distribution problem arising in a Quebec-based company, *Ameublement Tanguay* Inc., which operates in the furniture and electronics industries. Their distribution problem can be modeled as a variant of the heterogeneous vehicle routing problem with time windows, but it includes some additional and important real-life constraints which make the routing problem more challenging. Among them, let us mention that, due to marketing reasons linking products to the different brands of the company, each delivery can only be performed by a specific subset of the available vehicles. We described this new problem and developed a class-based insertion heuristic for solving it. The heuristic was tested on a set of real instances and compared with the results of the commercial solver used by our industrial partner. The results demonstrate that our routing algorithm performs better than the commercial solver on all instances. The number of routes and the daily distances are always reduced, and the weekly improvements are up to 23.7%. On an annual basis, these improvements represent cost savings of several hundred thousands of dollars.

Keywords. Rich vehicle routing problem, time windows, heuristics.

Acknowledgements. This research was partially supported by grants OPG 0293307 and OPG 0172633 from the Natural Sciences and Engineering Research Council of Canada (NSERC) and by Engage grant CGO 98069 from the NSERC. This support is gratefully acknowledged. We would also like to thank Louis Leclerc, Operations Manager, and Louis Thiboutot, Information Systems Manager, from Ameublement Tanguay, for their constant availability and comments and for providing us with relevant data. Their support was critical to the success of this research and is greatly appreciated.

---

* Corresponding author: Jean-Philippe.Gagliardi@cirrelt.ca

Dépôt légal – Bibliothèque et Archives nationales du Québec
Bibliothèque et Archives Canada, 2013

1. Introduction

The purpose of this article is to describe a complex heterogeneous vehicle routing problem arising in the furniture and electronics industries and present the algorithm that was used to solve it. In fact, we will show that our algorithm performs better than the commercial software currently in use. The problem was submitted to us by a retailer called *Ameublement Tanguay* Inc., which is a leader in their industries, with free shipping throughout the province of Quebec (eastern Canada). *Ameublement Tanguay* makes more than 120 000 deliveries per year (i.e. a visit to a specific client) with an average of 405 clients being delivered each day through 38 routes. During the peak periods (June, July and September), up to 700 deliveries can be performed per day. Most deliveries include installation services, which imply that the delivery personnel must spend some time at the customer's places. Other constraints (e.g., heterogeneous vehicles, deliveries limited to some labelled vehicles, time windows, lunch breaks) make this industrial context challenging and different from the ones already studied in the literature.

In this paper, we present the heuristic that was developed to solve *Ameublement Tanguay's* complex routing problem. The algorithm was embedded in a complete routing system and was tested using real data. It provided better solutions in shorter computing times than the commercial routing software that is currently used by the company.

The rest of this paper is structured as follows. Section 2 reviews the related literature and confirms the originality of the studied problem. Section 3 defines the problem and highlights its special features. Section 4 presents the specially designed routing heuristic and Section 5 reports the computational results on real data to assess its performance. Section 6 offers our conclusions.

2. Literature review

Research on the classical *vehicle routing problems* (VRP) has been very successful over the last fifty years (Laporte, 2009). Hundreds of articles and books have been published on the VRP (Laporte, 2007; Golden et al., 2008). However, the traditional VRP remains difficult to solve, and the most sophisticated algorithms can rarely solve instances with more than one hundred customers (Baldacci et al., 2011; Baldacci et al., 2012). Thus, many high-performing heuristics have been developed to solve larger instances (Nagata and Bräysy, 2009; Prins, 2009; Cordeau and Maischberger, 2012). Despite their excellent performance on traditional instances, with capacity and length constraints, they are not designed to handle real complex constraints, and their adaptation to industrial settings can be difficult.

There are few works addressing routing problems that include constraints limiting the selection of vehicles that can visit a given location or use a given road (Ma et al., 2012). Villegas et al. (2011) studied the *truck and trailer routing problem* (TTRP) in which a heterogeneous fleet of vehicles and trailers serves a set of customers. Some customers can be served only by a truck, while other customers can be served either by a truck or by a complete vehicle (i.e., a truck pulling a trailer). A slightly different version of this problem, called the partial accessibility constrained vehicle routing problem (PACVRP), was earlier introduced by Semet (1995).

Although the literature contains many vehicle routing industrial applications and implementations, only a few works report results based on real implementation and data. Among them, the food and soft-drink industries have been studied by many authors (Tarantilis and Kiranoudis, 2001; Tarantilis and Kiranoudis, 2002; Golden et al., 2002; Privé et al., 2006; and Caramia and Guerriero, 2010). Lubrification oil distribution has been studied by Repoussi et al. (2009) and Uzar and Çatay (2012). Waste collection has been studied by Kim et al., (2006) and Benjamin and Beasley (2010). The distribution of industrial gases (Day et al., 2009; Furman et al., 2011) and petroleum products (Avella et al., 2004; Boctor et al., 2011) are also two well-documented research areas.

Despite all these contributions, to the best of our knowledge, this is the first time that the specific context of the vehicle routing problem in the furniture and electronics industries has been addressed in the literature.

3. Problem statement

The *Vehicle Routing Problem in the Furniture and Electronics Industries* (VRP–FEI) is defined on a directed graph $G = (V, A)$, where $V = \{0, 1, \dots, n\}$ is the vertex set and A is the arc set. Vertex 0 corresponds to the distribution center (DC), while the remaining vertices represent the customers needing a delivery on the current day. Each arc (i,j) is associated to a travel time t_{ij} and a travel distance d_{ij}. Each customer i is associated to a "delivery" defined by a service class c_i equal to 1, 2 or 3, a service time $s_{i,}$ a weight w_i and a volume v_i. Service classes are a special feature related to the quality of the expected service as it will be detailed later on. For now, we will refer to class–1, class–2 and class–3 customers: the higher the class, the higher the quality of the expected service. The service of a customer i must start within a time window $[a_i, b_i]$ corresponding, in most cases, to the morning or to the afternoon. For customers with no time windows, a_i and b_i are respectively set to the beginning and end of the driver's workday.

A heterogeneous fleet of K vehicles is located at the depot. Each vehicle k is associated to a volume capacity V_k, a weight capacity W_k, a maximum working time L_k, and a service qualification Q_k. The notions service qualification and service class are related. *Ameublement Tanguay* manages three different store banners: *Liquida Meuble*, which corresponds to low-cost stores; *Ameublement Tanguay*, which are the regular stores with the largest product offer; and *Signature Maurice Tanguay*, which offers high-end, prestigious and distinctive products. A customer buying a product at these three stores is associated to a service class 1, 2 or 3, respectively. In the fleet, the vehicles are labelled according to one of these banners ($Q_k = 1$ for *Liquida Meuble*, 2 for *Ameublement Tanguay* and 3 for *Signature Maurice Tanguay*) which imposes "branding" constraints on the delivery. A customer expects to be delivered by the right or "better" banner vehicle ($Q_k \geq c_i$). In other words, vehicles with $Q_k = 1$ can only perform delivery of service type $c_i = 1$, and vehicles with $Q_k = 3$ can perform deliveries for all three service classes.

Each vehicle is associated to a route which starts between 7:00am and 8:30am in the morning and must end before 6:00pm. Within these limits, each route must respect a maximum working time of ten hours ($L_k = 10$, for all k). Vehicles are allowed to wait between two customers, and the route must include a one-hour lunch break scheduled between 11:30am and 01:00pm.

In this context, the goal of *Ameublement Tanguay* was to minimize the total length of the routes in terms of travelling distance. Needless to say, achieving reductions in travelling distance should naturally lead to a reduction in the number of routes, which, in turn, should lead to other important cost savings on driver salaries and the fleet's fixed and operating fees. The next section describes the routing heuristic that we tailored to help *Ameublement Tanguay* manage its delivery fleet.

4. Routing algorithm

The routing algorithm encompasses five main steps. After an *Initialization* step, the procedure iterates between the *Routing customers with time windows*, *Routing customers without time windows*, and *Local improvement*. These steps are first run on the most restrictive class–3 customers. Afterwards, the class–2 and then the class–1 customers are dealt with. We use as many class–3 vehicles as required to visit the class–3 customers. Then, the remaining class–3 vehicles' capacity is used to visit the class–2 customers. Class–2 vehicles will be added if needed to complete the class–2 clients. Again, the remaining higher-class vehicle capacity is used to visit the class–1 customers along with the class–1 vehicles.

Due to the fleet limitations and customer time windows, it might be impossible to schedule one or several customers. These customers are referred to as "exceptions" and the algorithm puts them in an Exceptions List. The list is transferred to the customer service department who will contact the customers to schedule another delivery date.

As mentioned before, driver lunch times need to be plan by the algorithm. This is simply done by adding a *dummy* customer with a time window between 11:00am and 1:00pm with a service time of 60 minutes whenever a new route is created.

Once the algorithm has planned visits to all the customers or declared some of them as "exceptions", another improvement step, called *Route Crushing,* tries to reduce the duration of each route. These five general steps, illustrated by Figure 1, are now described, followed by a detailed explanation of the more specialized procedures.

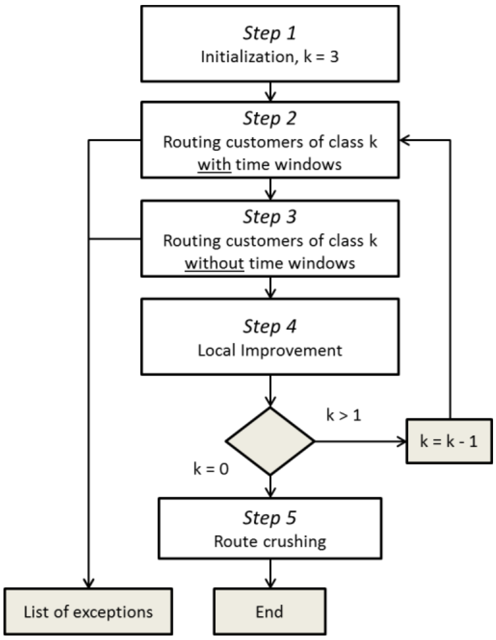

Figure 1: The 5 steps of the routing algorithm

Step 1: Initialization

Initialization creates and initializes the different data sets that will be used during the solving process. Let D be the ordered set of available vehicles sorted in descending order of their class. The set of exceptions E is initially empty. Let R be defined as the set of active routes; initially $R = \{\}$. Let C_1 be defined as a set containing all class–k customers with time windows, sorted according to the increasing width of their time window; initially $k = 3$. Let C_2 be defined as a set containing the remaining class–k customers (i.e., those without time windows). Execute the *Start_Routes* procedure, which is detailed later, in order to open one or several routes, assign vehicles to them and update set D accordingly.

Step 2: Routing customers with time windows

Routing customers with time windows inserts customers of C_1 into open routes sequentially, starting with the one having the tightest time window through the customer having the widest one. At the end of Step 2, all the customers in C_1 will be assigned to a route or the exception set E. Step 2 uses the *Best_Insertion* procedure to find the best route and the best place in it to insert a given customer. Step 2 executes the following activities:

Step 2a: Consider customer i, the first customer of C_1.

Step 2b: Call the *Best_Insertion* procedure to choose the best place to insert customer i, within the existing routes. Update the chosen route adequately, set $C_1 = C_1 / \{i\}$, and go to *Step 2d*. If the *Best_Insertion* routine is not able to route customer i the procedure goes to *Step 2c*.

Step 2c: This step is only entered if customer i cannot be inserted in any of the existent routes. If there is no vehicle of class k available in D, it is impossible to route customer i. Therefore, customer i is added to the exception set, and the algorithm goes to *Step 2d*. If D contains at least one vehicle of class–k (or higher), call the *Start_Routes* procedure, and update sets D and R accordingly. Return to *Step 2a*.

Step 2d: If C_1 is not empty, go to *Step 2a* to assign the following customer. If C_1 is empty, all class–k customers with time windows have been assigned to either a route or to E, go to *Step 3*.

Step 3 – Routing customers without time windows

This step performs the routing of customers in C_2. Unlike Step 2, which considers one customer in C_1 at each iteration, this step considers all the possible insertions for every customer in C_2 simultaneously and chooses the one leading to the lowest increase in a route's total length. At the end of Step 3, all *class–k* customers of C_2 will have been assigned to a route or to the exception set E. *Step 3* executes the following activities:

Step 3a: Use the *Best_Insertion* procedure with each customer i in C_2 to find its best insertion position. If *Best_Insertion* is not able to insert any customer, go to *Step 3b*; otherwise, let i^* be the customer leading to the best insertion. Insert customer i^* in the best identified route and position, set $C_2 = C_2$ / $\{i^*\}$, and go to *Step 3c*.

Step 3b: This step is only entered if none of the customers in C_2 can be routed into any of the available routes. If the set of available vehicles of *class-k* in *D* is empty, it is impossible to route the remaining customers of C_2 which are added to the exception set, $E = E \cup C_2$. Go to *Step 4*. If *D* contains one or more available vehicles of *class–k* (or higher), call the *Start_Routes* procedure, update *D* and *R*, and return to *Step 3a*.

Step 3c: If C_2 is not empty, go to *Step 3a* to route another customer. If C_2 is empty, all class–*k* customers have been assigned to either a route or to the exception set *E*, go to *Step 4*.

Step 4 – Local improvement

The Local improvement step encompasses two different procedures, called *Intra_Route* and *Inter_Routes*, and is applied to all the current routes.

Step 4a: The *Intra_Route* improvement procedure tries to reduce the total distance of a single route. It is applied to each route with a best improvement strategy until no more improvements can be reached. The *Intra_Route* improvement procedure is based on the 3-opt algorithm of Lin (1965) in which the objective is to minimize the total traveling distance. The original algorithm is adapted to check for time window feasibility before each improving move.

Step 4b: The *Inter_Routes* procedure considers the set of existing routes $R = \{r_1, r_2, \ldots, r_m\}$ and tries to improve sequentially each pair of routes r_i ($i = 1, \ldots, m-1$) and r_j ($j = i+1, \ldots, m$) by exchanging customers between them.

Let us assume that (i_1, i_2, i_3, i_4) and (j_1, j_2, j_3, j_4) are two chains of four consecutive vertices on routes r_i and r_j. Figure 2 illustrates the 11 exchanges that are tested. Moves 1 to 3 correspond to the 1–interchange procedure of Osman (1993) and moves 4 to 11 represent a subset of the exchanges tested by the Osman 2–interchange, as proposed by Renaud and Boctor (2002) for solving the fleet size and mixed vehicle routing problem. These moves are applied to all possible chains of four consecutive vertices between routes r_i and r_j. The first improving move is applied and the procedure is repeated as long as improvements are possible between the two routes. Then, the procedure is repeated, considering all the remaining pairs of routes. Once all the possible pairs of routes have been considered, the procedure restarts but this time evaluating only pairs of routes for which at least one of the routes have been modified during the previous iteration.

Three feasibility checks need to be satisfied before evaluating a move. First, it is required that the service classes of the exchanged customers are compatible with the vehicle classes of the considered routes. If both routes use vehicles of the same category, this check is not necessary. The second feasibility check deals with truck capacity, both in terms of weight and volume, as well as total route duration. The last feasibility check concerns the satisfaction of customer time windows. If the feasibility checks are successful, new distances are calculated to see if an improvement is obtained.

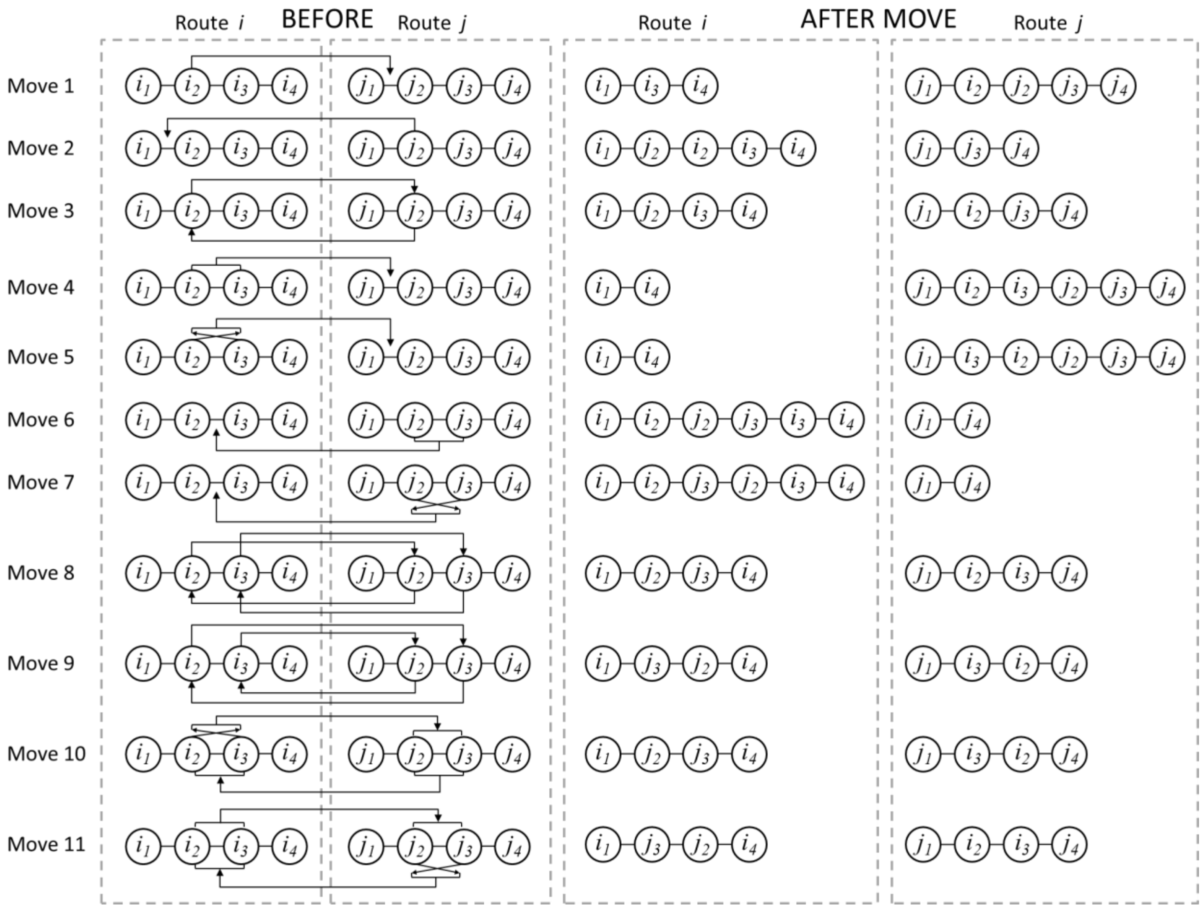

Figure 2: The 11 moves explored by the *Inter_Routes* procedure

Step 5 – Route crushing

Route crushing tries to reduce the total time duration of a route. Since the previous steps of the algorithm try to insert customers following a "visit as soon as possible" strategy, it is possible that some routes include unnecessary waiting time.

The Route crushing step tries to "delay" the visits to customers as much as possible. It starts by setting the visit to the last customer as soon as possible in order to finish the workday earlier. Then, going back from the second last customer to the first one, each visit time is rescheduled to as late as possible that still respects the time window constraints. In addition to the potential reduction in the total route length, this simple "route-crushing" procedure also leads to two interesting advantages. The last visit is performed as soon as possible, which is usually appreciated by customers, and the first visit is performed as late as possible, which helps drivers avoid morning traffic.

Figure 3 shows the whole routing algorithm.

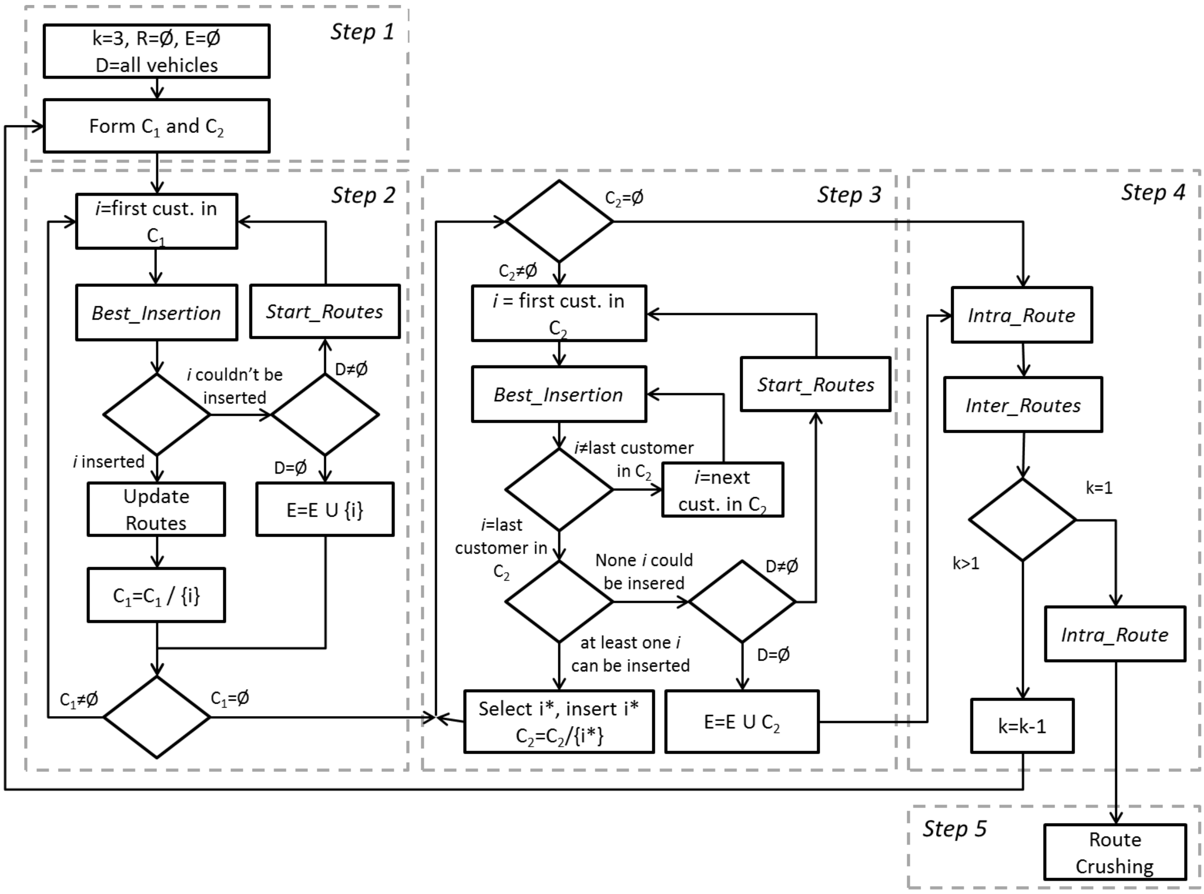

Figure 3: Detailed scheme of the routing algorithm

The next paragraphs are devoted to describing *Start_Routes* and *Best_Insertion* procedures that are used by steps *Routing customers with time windows* and *Routing customers without time windows*.

Start_Routes

The *Start_Routes* procedure tries to open new routes whenever the existing routes cannot accommodate the clients remaining unvisited. Unlike other classic approaches, this procedure may open up to π new routes each time it is called, depending on the number and the needs of the remaining customers to visit. In particular, the procedure computes the total weight and volume to deliver as well as an estimation of the total time required to do it. This estimation is computed as the average traveling time between each unvisited customer i and all the other customers in C' (C' being C_1 if *Start_Routes* is called during *Step 2* or C_2 if *Start_Routes* is called during *Step 3*) and the distribution center. Based on these calculations, and assuming the loading capacity of an average vehicle and maximum work time, the minimum number of vehicles required is estimated. This estimation is compared to both the available number of vehicles in D and the parameter π, and the smallest value is used as the number of new routes to create. Also, the procedure initializes each new route by inserting a customer into each of them. The reason for opening several routes is to try to avoid situations in which the routes are fairly concentrated in a specific area, and faraway customers are visited at a very high cost. Thus, if two or more routes are open, the procedure tries to "launch" them in rather different geographic directions.

To this end, the assignment of the first customer to a route is done as follows. If only one route is created, we include in this route the furthest customer from the depot in C'. If two routes are opened, two customers i and j in C' are selected in such a way that the total distance $d_{0i} + d_{ij} + d_{j0}$ is maximized. In other words, the routes will tend to cover opposite regions. Finally, if three routes are created, the three customers i, j and k are selected to maximize $d_{0i} + d_{0j} + d_{0k} + d_{ij} + d_{jk} + d_{ki}$.

The new routes are added to the set of existing routes R, and the customers that have been assigned to them are deleted from C'. Preliminary numerical experiments led us to set $\pi = 3$.

Best_Insertion

The *Best_Insertion* procedure tries to find the best route and the best place to insert the visit of customer i. Since the routing procedure starts by routing the clients requiring the highest class service, we can be sure that all the routes in R can perform a service of the class required by customer i.

The procedure tries to insert customer i in each possible position of all the current routes. If the insertion of i between two customers verifies that the time window of customer i is respected, the visiting time to i is set as soon as possible and the visiting time of the subsequent customers in the same route are checked to ensure that their time window requirements are maintained. Two other checks are performed to ensure that the total route duration is respected and that the vehicle can return to the depot before the end of the day (set to 6:00pm). If all these tests are satisfied, the added traveling distance of inserting customer i is computed.

When all the routes and all the insertion positions have been tested, the procedure returns either the best insertion position or an indication that no feasible insertion was possible for customer *i*.

This version of the class-based insertion heuristic is deterministic and will lead to a reproducible solution in short computing time, which is a prerequisite for a commercial implementation. However, introducing some randomness in the heuristic is possible in order to benefit from a multi-start implementation and eventually obtain better and more robust solutions.

Multi-start version

The multi-start version of the routing algorithm repeats the whole procedure for a given number of times. The first repetition corresponds to the routing procedure described above. In order to build different routes, the multi-start version applies a perturbation to the ordered set C_1 in Step 1. By changing the order in which customers with time windows are ordered in C_1, they are inserted differently into the solution, which may help to explore different solution spaces. The best solution found over all the executions is retained.

5. Computational results

We implemented the heuristic under Microsoft Visual Studio 2010 using VB.net (.net framework 4.0). The software developed included a vehicle administration system to manage vehicle availability and services. All the numerical experiments were run on a desktop Intel Core 2 Duo (T810 @ 2.10Ghz) with 4.00 Go RAM.

In order to evaluate the performance of the developed system, we used real delivery data covering a complete week of operation. Table 1 reports some characteristics of the available vehicle types. To manipulate and unpack the goods, and bring packing as well as old appliances back when requested, the volume capacity of the vehicles was reduced by 22%. This parameter was provided and used by *Ameublement Tanguay* distribution managers and is based on long-term observations.

Table 1: Fleet composition and characteristics

Vehicle type	Vehicle Length (feet)	Number of vehicles	Weight capacity (pounds)	Volume capacity (ft^3)	Service class
1	18	2	5 000	900	1
2	20	7	8 179	1 134	2
3	22	3	8 999	1 199	3
4	24	24	13 423	1 612	2
5	26	7	13 500	1 704	2
6	28	3	17 000	1 840	2

We compared the quality of the solutions produced by the commercial solver used by *Ameublement Tanguay* and the ones produced by our algorithm, called Class–Based Insertion Heuristic (CBIH), in terms of total travelled distance, total driving time, and the number of routes. The commercial solver is based on a genetic algorithm, which is run usually for 40 000 iterations.

Complete numerical results are reported in Tables 2 to 6, each table shows the results for one weekday. The first line in each table refers to the *Ameublement Tanguay* solution, while the other 4 lines refer to the CBIH initial solution and the multi-start solution with 10, 20 and 30 repetitions. Each table reports the *Number of routes* planned, the *Total distance* (in kilometres) for these routes, the *Driving time* (in minutes) and the computing time, *Comp. time* (in seconds) to obtain the solution by our algorithm. Tables 2 and 3 refer to delivery days with 121 and 108 customers, which represent slow business days. In both cases, the algorithm reduces the number of routes from 10 to 8. In Tables 2 and 3, the CBIH initial solution produces distance reduction of 4.4% and 19.7%, respectively. After 30 repetitions in less than 3 minutes of computing time, the distance reduction for these two days is of around 257 and 420 km, which correspond to improvements of about 21.0% and 20.1%.

Table 2: Delivery day with 121 customers

Solution	Number of routes	Total distance (km)	Driving time (min.)	Comp. time (seconds)
Ameublement Tanguay	10	1 227.2	1 412.0	N/A
CBIH – Initial solution	9	1 173.6	1 339.6	7.03
CBIH – 10 repetitions	8	977.5	1 157.1	54.96
CBIH – 20 repetitions	8	976.7	1 169.2	102.79
CBIH – 30 repetitions	8	969.3	1 158.2	148.99

Table 3: Delivery day with 108 customers

Solution	Number of routes	Total distance (km)	Driving time (min.)	Comp.time (seconds)
Ameublement Tanguay	10	2 096.3	2 136.0	N/A
CBIH – Initial solution	9	1 684.2	1 851.6	6.13
CBIH – 10 repetitions	9	1 675.5	1 848.6	39.24
CBIH – 20 repetitions	9	1 675.5	1 848.6	72.55
CBIH – 30 repetitions	9	1 675.5	1 848.6	105.01

With 290 deliveries, Table 4 corresponds to an average business day. In this case, three routes were saved, and the distance was reduced by 724 km, which represents a 21% distance reduction.

Table 4: Delivery day with 290 customers

Solution	Number of routes	Total distance (km)	Driving time (min.)	Comp. time (seconds)
Ameublement Tanguay	25	3 449.8	3 900.0	N/A
CBIH – Initial solution	22	3 232.0	3 506.6	61.40
CBIH – 10 repetitions	23	2 733.9	3 159.6	655.12
CBIH – 20 repetitions	22	2 725.1	3 125.6	1 284.00
CBIH – 30 repetitions	22	2 725.1	3 125.6	1 974.60

Tables 5 and 6 are representative of two heavily loaded days, with respectively 382 and 421 customers, the latter being one of the tenth busiest days of the year. Again, in both cases, the number of routes was reduced and important reductions in travelling distances (1 282 and 1 494 km) were obtained, respectively 24.9% and 26.1%. Computing times remain affordable, under 47 minutes for the 382 customer instance. For the larger, 421 customer instance, most of the improvement was already obtained by CBIH after 20 repetitions, requiring only 71 minutes of computing time. However, since the calculations are often performed overnight, it is still affordable to run CBIH with 30 repetitions on this instance, as it requires less than two hours of computing time.

Table 5: Delivery day with 382 customers

Solution	Number of routes	Total distance (km)	Driving time (min.)	Comp. time (seconds)
Ameublement Tanguay	31	5 138.6	5 862.0	N/A
CBIH – Initial solution	27	4 147.2	4 686.6	97.20
CBIH – 10 repetitions	26	3 953.4	4 521.3	987.00
CBIH – 20 repetitions	27	3 856.6	4 420.0	1 872.00
CBIH – 30 repetitions	27	3 856.6	4 420.0	2 811.60

For all the studied instances, CBIH final solutions are much better than those produced by the commercial solver used by *Ameublement Tanguay*. All the produced solutions were feasible and respected all the constraints, and all customers were assigned to a route (i.e., no “exception” customers were identified). Thus, we can consider CBIH as a serious alternative to the commercial software currently used by *Ameublement Tanguay*.

Table 6: Delivery day with 421 customers

Solution	Number of routes	Total distance (km)	Driving time (min.)	Comp. time (seconds)
Ameublement Tanguay	30	5 736.7	6 183.0	N/A
CBIH – Initial solution	29	4 267.1	4 685.6	200.40
CBIH – 10 repetitions	29	4 267.1	4 685.6	2 211.60
CBIH – 20 repetitions	28	4 242.2	4 642.1	4 262.40
CBIH – 30 repetitions	28	4 242.2	4 642.1	6 351.00

Table 7 gives a macroscopic view of the results over a complete one-week planning horizon. CBIH reduces the number of routes from 106 to 94. With two employees by vehicle, this translates into a reduction of 24 workers. If an average day of eight hours is considered, this is a reduction of 192 working hours. At an average hourly rate of 20 CAD per hour, this is a one-week saving of $ 3 840 (in Canadian currency). Moreover, it is generally assumed that the long-term operating rate of a delivery vehicle ranges from $ 1.50 to $ 2.00 per kilometre, including an average fuel consumption of 27.7 l / 100 km (Table RO8, Transport Canada, 2012). If a 4 179 km reduction is considered at $ 1.50 per km, this is another $ 6 268. This leads to a cost reduction of $ 10 108 over a single week, which, if transposed to a 50-week year, corresponds to a global cost reduction of above $ 500 000.

Table 7: Global results over a one-week planning horizon

Solution	Number of routes	Total distance (km)	Distance reduction(km)	Distance reduction (%)
Ameublement Tanguay	106	17 648.6		
CBIH – Initial solution	96	14 504.1	3144.5	17.82%
CBIH – 10 repetitions	95	13 607.4	4041.2	22.90%
CBIH – 20 repetitions	94	13 476.1	4172.5	23.64%
CBIH – 30 repetitions	94	13 468.7	4179.9	23.68%

6. Conclusion

We described and solved a complex real-life problem in the furniture and electronics industries. In addition to traditional constraints (e.g., capacity, volume, route duration, time windows), this routing problem presents many interesting features, such as service category, vehicle qualification and lunch breaks. We proposed a Class-Based Insertion Heuristic (CBIH) which is designed to efficiently handle customer and vehicle class constraints.

Our heuristic was tested on a five-day data set obtained from our industrial partner and compared to their solutions, produced by a well-known commercial solver. Our results show that, in all the instances received from our industrial partner, our heuristic improved the solutions produced by the commercial solver, leading to a total distance reduction of 4 179 km over five days, which represents an improvement of 23.68%. Moreover, the computing time needed by the heuristic to produce such impressive results ranges from several seconds to less than two hours, which is comparable to the time required by the commercial software.

From an economical point of view, a conservative calculation estimates that the use of the solutions produced by the heuristic could lead to annual savings of more than 500 000 CAD.

We believe that our algorithm offers a very attractive alternative to the commercial solver currently in use. According to Mr. Louis Leclerc, Operations Manager at *Ameublement Tanguay*, “The new system provides better quality solutions in terms of distribution time and is more flexible to handle the company constraints. Computing time is also faster than the commercial solution.”

The next step in the partnership development is to connect the prototype with a commercial mapping system that can ensure fast distance calculation, graphical maps interaction and customer geocoding functionalities. Presently, the prototype uses Google maps and depends on external maps and internet reliability, which is not acceptable for a commercial company like *Ameublement Tanguay*.

Acknowledgements – This research was partially supported by Grants OPG 0293307 and OPG 0172633 from the Canadian Natural Sciences and Engineering Research Council (NSERC) and by Engage Grant CGO 98069 from the NSERC. This support is gratefully acknowledged. We would also like to thank Louis Leclerc, Operations Manager, and Louis Thiboutot, Information Systems Manager, from *Ameublement Tanguay*, for their constant availability and comments and for providing us with relevant data. Their support was critical to the success of this research and is greatly appreciated.

References

Applegate D. L., Bixty R. E., Chvátal V. & Cook W. J., *The traveling salesman problem. A computational study*. Princeton University Press, **2007**, Princeton, NJ.

Avella P., Boccia M. & Sforza A., Solving a fuel delivery problem by heuristic and exact approaches. *European Journal of Operational Research*, **2004**, 152, 170-179.

Baldacci R., Mingozzi A. & Roberti R., New route relaxation and pricing strategies for the vehicle routing problem. *Operations Research*, **2011**, 59, 1269-1283.

Baldacci R., Mingozzi A. & Roberti R., Recent exact algorithms for solving the vehicle routing problem under capacity and time window constraints. *European Journal of Operational Research*, **2012**, 218, 1-6.

Benjamin A. M. & Beasley J. E., Metaheuristics for the waste collection vehicle routing problem with time windows, driver rest period and multiple disposal facilities. Computers & Operations Research, **2010**, 37, 2270-2280.

Boctor F., Renaud J. & Cornillier F., Trip Packing In Petrol Stations Replenishment. *Omega*, 39, **2011**, 86-98.

Caramia M. & Guerriero F., A milk collection problem with incompatibility constraints. *Interfaces*, **2010**, 40, 130-143.

Cook W. J. *In pursuit of the traveling salesman: Mathematics at the limits of computation*, **2012**, Princeton University Press.

Cordeau J.-F. & Maischberger M., A parallel iterated tabu search heuristic for vehicle routing problems. *Computers & Operations Research*, **2012**, 39, 2033-2050.

Day J. M., Wright P. D., Schoenherr T., Venkataramanan M. & Gaudette K., Improving routing and scheduling decisions at a distributor of industrial gasses. *Omega*, **2009**, 37, 227-237

Furman K. C., Song J.-H., Kocis G. R., McDonald M. K. & Warrick P. H., Feedstock routing in the ExxonMobil downstream sector. *Interfaces*, **2011**, 41, 149-163.

Golden B. L., Assad A. A. & Wasil E. A., Routing vehicles in the real world: Applications in the solid waste, beverage, food, dairy, and newspaper industries. In: Toth P. and Vigo D. (eds). *The vehicle routing problem*. SIAM Monographs on Discrete Mathematics and Applications, **2002**, Philadelphia, 245-286.

Golden B. L., Raghavan S. & Wasil E. A., eds. *The vehicle routing problem: Latest advances and new challenges*. Springer, **2008**, New York.

Kim B.-I., Kim S. & Sahoo S., Waste collection vehicle routing problem with time windows. *Computers and Operations Research*, **2006**, 33, 3624-3642.

Laporte G., What you should know about the vehicle routing problem. *Naval Research Logistics*, **2007**, 54, 811-819.

Laporte G. Fifty years of vehicle routing. *Transportation Science*, 43, **2009**, 408-416.

Lin S., Computer solutions of the traveling salesman problem. *The Bell System Technical Journal*, **1965**, 2245-2269

Ma H., Cheang B, Lim A, Zhang L. & Zhu Y, An investigation into the vehicle routing problem with time windows and link capacity constraints, Omega, 40, **2012**, 336-347.

Nagata Y. & Bräysy O., Edge assembly-based memetic algorithm for the capacitated vehicle routing problem. *Networks*, **2009**, 54, 205-215.

Prins C., A GRASP ×evolutionary local search hybrid for the vehicle routing problem. In *Bio-inspired algorithms for the vehicle routing problem*, **2009**, 161, 35-53, Berlin, Springer.

Privé J., Renaud J., Boctor F. & Laporte G., Solving a vehicle-routing problem arising in soft-drink distribution. *Journal of the Operational Research Society*, **2006**, 57, 1045-1052.

Renaud J. & Boctor F. F., A sweep based algorithm for the fleet size and mix vehicle routing problem. *European Journal of Operational Research*, **2002**, 140, 618-628.

Repoussis P. P., Paraskevopoulos D. C., Zobolas G., Tarantilis C.D. & Ioannou G., A web-based decision support system for waste lube oils collection and recycling. *European Journal of Operational Research*, **2009**, 195, 676-700.

Semet F., A two-phase algorithm for the partial accessibility constrained vehicle routing problem. *Annals of Operations Research*, 61, **1995**, 45-66.

Tarantilis C. D. & Kiranoudis C. T., A meta-heuristic algorithm for the efficient distribution of perishable foods. *Journal of Food Engineering*, **2001**, 50, 1-9.

Tarantilis C. D. & Kiranoudis C. T., Distribution of fresh meat. *Journal of Food Engineering*, **2002**, 51, 85-91.

Transport Canada, Les transports au Canada 2011, Addenda statistique, TP 14816, **2012**.

Uzar M. F. & Çatay B., Distribution planning of bulk lubricants at BP Turkey. *Omega*, **2012**, 40, 870-881.

Villegas J. G., Prins C., Prodhon C., Medaglia A. L. & Velasco N., A GRASP with evolutionary path relinking for the truck and trailer routing problem. *Computers & Operations Research*, **2011**, 38, 1319-1334.