CIRRELT

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

Interuniversity Research Centre
on Enterprise Networks, Logistics and Transportation

# Heuristics for Time Slot Management: A Periodic Vehicle Routing Problem View

**Florent Hernandez**
**Michel Gendreau**
**Jean-Yves Potvin**

**November 2014**

**CIRRELT-2014-59**

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

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

www.cirrelt.ca

# Heuristics for Time Slot Management: A Periodic Vehicle Routing Problem View

**Florent Hernandez[1,2], Michel Gendreau[1,2], Jean-Yves Potvin[1,3,*]**

[1] Interuniversity Research Centre on Enterprise Networks, Logistics and Transportation (CIRRELT)

[2] Department of Mathematics and Industrial Engineering, Polytechnique Montréal, P.O. Box 6079, Station Centre-ville, Montréal, Canada H3C 3A7

[3] Department of Computer Science and Operations Research, Université de Montréal, P.O. Box 6128, Station Centre-Ville, Montréal, Canada H3C 3J7

**Abstract.** In this study, we consider a tactical problem where a time slot combination for delivery service over a given planning horizon must be selected in each zone of a geographical area. Based on the selected time slot combinations, routes are then constructed for the fleet of delivery vehicles for each period of the planning horizon. The routing plan serves as a blueprint for the following operational problem where the availability of the selected time slot combinations must be updated as real customer orders occur. We propose two heuristics for solving the tactical problem. The first heuristic is a three-phase approach: a periodic vehicle routing problem is first solved, followed by a repair phase and a final improvement phase where a vehicle routing problem with time windows is solved for each period of the planning horizon. The second heuristic tackles the problem as a whole by directly solving a periodic vehicle routing problem with time windows. Computational results compare the two heuristics under various settings, based on instances derived from benchmark instances for the vehicle routing problem with time windows.

**Keywords**: Time slot assignment, attended delivery service, vehicle routing, tabu search.

**Acknowledgements.** Financial support for this work was provided by the Natural Sciences and Engineering Research Council of Canada (NSERC). This support is gratefully acknowledged.



---

* Corresponding author: Jean-Yves.Potvin@cirrelt.ca

Dépôt légal – Bibliothèque et Archives nationales du Québec
Bibliothèque et Archives Canada, 2014

# 1 Introduction

The success of many retail businesses depends upon their ability to ensure efficient delivery services. To provide a high service level and to avoid delivery failures, companies often propose to their customers a choice of possible time slots for an attended delivery. Offering more time slots is typically perceived by customers as good service quality, although it might impact the transportation costs. Offering appropriate time slots is particularly complex in the context of home delivery of furniture or large appliances because "one-shot" customers are dealt with, that is, they are unlikely to be served again in the near future. In this context, it is common practice to group customers into geographical zones, based on their zip code for example.

The tactical optimization problem considered in this work is defined as follows. The service area is modeled as a complete directed graph $G = (V, A)$, with an arc set $A$ and a node set $V = \{0, \cdots, n\}$, where node 0 is the depot and the remaining nodes are the geographical zones. With each arc $(i, j) \in A$ is associated a cost $c_{ij}$ and a travel time $t_{ij}$.

The planning horizon $T$ is made of a number of time periods, for example days of the week, where each day has a fixed duration. The delivery route of each vehicle, starting from and ending at a central depot, must be performed within a single day. Each day $t \in T$ is divided into a number of shifts (or time slots). In this work, each day is divided as follow: either a single shift spans the full day ($t$ $s^d$); or two shifts span a half-day (HD) each, that is, ($t$ $s_1^{hd}$) followed by ($t$ $s_2^{hd}$); or three shifts span one-third-of-a-day (OTD) each, that is, ($t$ $s_1^{otd}$) followed by ($t$ $s_2^{otd}$) followed by ($t$ $s_3^{otd}$). With each zone $i$ is associated a service frequency $\eta_i$. A shift schedule for zone $i$ is then made of $\eta_i$ shifts, with the same estimated demand $d_i$ and service time $st_i$ for each shift in the shift schedule (as it is done in [3]). For example, if the service frequency of a given zone is equal to 3 and if we assume that each day is divided into three shifts and that the planning is made over Monday (MO), Tuesday (TU), Wednesday (WE), Thursday (TH) and Friday (FR), a shift schedule for the zone could be {MO $s_1^{otd}$, WE $s_3^{otd}$, FR $s_1^{otd}$}. Note that two shifts in the same day cannot both appear in the same shift schedule, as the idea is to cover different days.

Typically, many different shift schedules are proposed by the marketing department for each delivery zone. For example, we might have the following set $W_i$ of possible shift schedules for zone $i$:

{{MO $s_1^{otd}$, WE $s_3^{otd}$, FR $s_1^{otd}$}, {TU $s_2^{otd}$, WE $s_1^{otd}$, TH $s_2^{otd}$}, {TU $s_2^{otd}$, TH $s_3^{otd}$, FR $s_1^{otd}$}, {MO $s_1^{otd}$, TU $s_3^{otd}$, WE $s_2^{otd}$}}

The tactical time slot management problem (TSMP) consists in selecting a particular shift schedule for each zone, assigning the zones that must be served on any given day of the planning horizon to the fleet of vehicles and sequencing the zones assigned to any given vehicle on any given day, while minimizing the total travel cost

(travel time or distance). In the process, some side constraints must be satisfied: each route should take place within a single day and the capacity $Q$ of each vehicle should not be exceeded. This problem must be distinguished from its operational counterpart which is dealt with when the real customer orders become known. In this case, the previously selected shift schedules must be dynamically managed. It might simply consists in closing a shift assigned to a zone when a certain number of orders from that zone have been received. More advanced strategies can also be devised, based on future expected orders and dynamic pricing (delivery fees).

To formally specify our TSMP, the following sets must first be defined:

$P_t^d = \{(t\ s^d, t\ s^d)\},\ t \in T.$

$P_t^{hd} = \{(t\ s_1^{hd}, t\ s_1^{hd}), (t\ s_2^{hd}, t\ s_2^{hd}), (t\ s_1^{hd}, t\ s_2^{hd})\},\ t \in T.$

$P_t^{otd} = \{(t\ s_1^{otd}, t\ s_1^{otd}), (t\ s_2^{otd}, t\ s_2^{otd}), (t\ s_3^{otd}, t\ s_3^{otd}), (t\ s_1^{otd}, t\ s_2^{otd}), (t\ s_1^{otd}, t\ s_3^{otd}),$
$(t\ s_2^{otd}, t\ s_3^{otd})\},\ t \in T.$

Then, set $P$ is defined as the union over all days $t \in T$ of $P_t^d$ or (exclusive) $P_t^{hd}$ or (exclusive) $P_t^{otd}$. Set $P$ specifies admissible ordered pairs of shifts associated with two consecutive zones in a vehicle route. That is (1) two zones can be visited consecutively during the same shift or (2) a zone visited in a second HD shift can follow a zone visited in the first HD shift of the same day or (3) a zone visited in a second OTD shift can follow a zone visited in the first OTD shift of the same day, while a zone visited in a third OTD shift can follow a zone visited in the first or second OTD shift of the same day. In particular, pairs of shifts associated with two different days are forbidden through set $P$. It should be noted that the depot 0 is considered as a special zone which can be visited during any shift and can precede or follow any other zone, subject to the restrictions imposed by set $P$. Also, when zone $i$ in shift $s$ and zone $j$ in shift $s'$ are visited consecutively, the vehicle is allowed to wait to serve zone $j$ at the start time of shift $s'$ (if necessary).

With these assumptions, a mathematical programming model can be defined based on the following parameters and decision variables.

*Parameters*

- $I$ set of zones, that is, $I = V \backslash \{0\}$
- $K$ set of vehicles
- $T$ set of time periods (days)
- $W_i$ set of possible shift schedules for zone $i \in I$
- $W$ set of all possible shift schedules over all zones, that is, $W = \bigcup_{i \in I} W_i$
- $S_i$ set of possible shifts for zone $i \in I$, that is, $S_i = \bigcup W_i$

- $S$ set of all possible shifts over all zones, that is, $S = \bigcup_{i \in I} S_i$. Note that $S_0 = S$.
- $P_{ij}$ set of admissible ordered pairs of shifts for nodes $i, j \in V$, that is, $P_{ij} = \{(s, s') | (s, s') \in P, (i, j) \in A, s \in S_i, s' \in S_j\}$
- $N_{ij}^s$ set of possible shifts for the visit of node $j$ immediately after the visit of node $i$ in shift $s \in S_i$, that is, $N_{ij}^s = \{s' | (s, s') \in P_{ij}\}$
- $N_i^-$ set of immediate predecessors of node $i \in V$, that is, $N_i^- = \{j | (j, i) \in A\}$
- $N_i^+$ set of immediate successors of node $i \in V$, that is, $N_i^+ = \{j | (i, j) \in A\}$
- $c_{ij}$ travel cost on arc $(i, j) \in A$
- $d_i$ demand of node $i \in V$ (with $d_0 = 0$)
- $st_i$ service time of node $i \in V$ (with $st_0 = 0$)
- $a_t$ start time of period $t \in T$
- $b_t$ end time of period $t \in T$
- $e_s$ start time of shift $s \in S$
- $l_s$ end time of shift $s \in S$
- $Q$ vehicle capacity
- $M$ arbitrary large number
- $u_{sw}$ is 1 if shift $s \in S$ is in shift schedule $w \in W$, 0 otherwise
- $v_{st}$ is 1 if shift $s \in S$ is in period $t \in T$, 0 otherwise

*Decision variables*

- $x_{ij}^{kss'}$ is 1 if arc $(i, j)$ is used by vehicle $k$ to visit node $i$ during shift $s$ followed by node $j$ during shift $s'$, 0 otherwise, $(i, j) \in A$, $(s, s') \in P_{ij}$, $k \in K$.
- $y_{iw}$ is 1 if shift schedule $w$ is selected for zone $i$, 0 otherwise, $i \in I$, $w \in W_i$.
- $t_i^{ks}$ is the service start time of zone $i$ by vehicle $k$ in shift $s$, $i \in I$, $s \in S_i$, $k \in K$.

The model is then the following:

$$\min \sum_{k\in K} \sum_{(i,j)\in A} \sum_{(s,s')\in P_{ij}} c_{ij} x_{ij}^{kss'} \tag{1}$$

subject to

$$\sum_{k\in K} \sum_{j\in N_i^+} \sum_{(s,s')\in P_{ij}} x_{ij}^{kss'} = \eta_i, \qquad i \in I \tag{2}$$

$$\sum_{k\in K} \sum_{j\in N_i^+} \sum_{s'\in N_{ij}^s} x_{ij}^{kss'} \leq 1, \qquad i \in I, s \in S_i \tag{3}$$

$$\sum_{(i,j)\in A} \sum_{(s,s')\in P_{ij}} d_i x_{ij}^{kss'} \leq Q, \qquad k \in K \tag{4}$$

$$\sum_{j\in N_i^+} \sum_{(s,s')\in P_{ij}} x_{ij}^{kss'} = \sum_{j\in N_i^-} \sum_{(s,s')\in P_{ij}} x_{ji}^{kss'}, \qquad i \in V, k \in K \tag{5}$$

$$t_i^{ks} + st_i + t_{ij} - S_j^{ks'} \leq \left(1 - x_{ij}^{kss'}\right) M, \qquad (i,j) \in A, (i,j) \in I, k \in K, (s,s') \in P_{ij} \tag{6}$$

$$t_i^{ks} + st_i + t_{i0} - \sum_{t\in T} v_{s't} b_t \leq \left(1 - x_{i0}^{kss'}\right) M, \qquad (i,0) \in A, k \in K, (s,s') \in P_{i0} \tag{7}$$

$$\sum_{t\in T} v_{st} a_t + t_{0i} - t_i^{ks'} \leq \left(1 - x_{0i}^{kss'}\right) M, \qquad (0,i) \in A, k \in K, (s,s') \in P_{0i} \tag{8}$$

$$a_s u_{sw} y_{iw} \leq t_i^{ks} \leq b_s u_{sw} y_{iw}, \qquad i \in I, k \in K, s \in S_i, w \in W_i \tag{9}$$

$$\sum_{w\in W_i} y_i^w = 1, \qquad i \in I \tag{10}$$

$$x_{ij}^{kss'} \leq \sum_{w\in W_i} u_{sw} y_{iw} \qquad (i,j) \in A, i,j \in I, k \in K, (s,s') \in P_{ij} \tag{11}$$

$$x_{ij}^{kss'} \leq \sum_{w\in W_j} u_{s'w} y_{jw} \qquad (i,j) \in A, i,j \in I, k \in K, (s,s') \in P_{ij} \tag{12}$$

$$x_{i0}^{kss'} \leq \sum_{w\in W_i} u_{sw} y_{iw} \qquad (i,0) \in A, i \in I, k \in K, (s,s') \in P_{i0} \tag{13}$$

$$x_{0i}^{kss'} \leq \sum_{w\in W_i} u_{s'w} y_{iw} \qquad (0,i) \in A, i \in I, k \in K, (s,s') \in P_{0i} \tag{14}$$

$$x_{ij}^{kss'} \in \{0,1\} \qquad (i,j) \in A, k \in K, (s,s') \in P_{i,j} \tag{15}$$

$$y_{iw} \in \{0,1\} \qquad i \in I, w \in W_i \tag{16}$$

$$t_i^{ks} \geq 0 \qquad i \in I, k \in K, s \in S_i \tag{17}$$

The objective (1) minimizes the total cost. Constraints (2) ensure that the required number of shifts are assigned to each zone while constraints (3) state that every possible pair of zone and shift has to be visited by at most one vehicle (i.e., split delivery is forbidden). The vehicle capacity constraints are found in (4) while the flow conservation constraints are in (5). Constraints (6) to (9) ensure the coherence of the route and the satisfaction of the time bounds. Constraints (10) state that a shift schedule must be assigned to each zone. Constraints (11) to (14) establish the coherence between the values of decision variables $x$ and $y$. Finally, constraints (15) to (17) define the domain of the variables. It should be noted that this model would only be tractable for instances of very small size. Accordingly, heuristic approaches are proposed in the following.

The model, through the $x$ and $y$ variables, clearly underlines the two main issues, namely, the assignment of shift schedules to zones and the routing of the vehicles. The main contributions of this work come from the design of heuristics that simultaneous address these two issues, while exploiting similarities between the TSMP and the Periodic Vehicle Routing Problem (PVRP).

The paper is organized as follows. Section 2 summarizes the relevant literature. In Section 3, two heuristic approaches are proposed to solve our problem. Then, Section 5 reports and analyzes computational results obtained on different types of instances. Finally, Section 6 concludes the paper.

## 2 Literature review

To the best of our knowledge, the work in [3] is the contribution that mostly resembles ours, as it also addresses a tactical TSMP. Here, an expected number of customers is associated with each zone (zip code) which is distributed evenly among the time slots in the time slot schedules. The authors propose different ways to compute the expected travel time between two customers in the same zone, between two zones in the same time slot and between two zones in different time slots. Then, the authors describe two different problem-solving approaches. The first one uses a continuous approximation method to estimate the expected cost of a given assignment of time slot schedules to zip codes. A greedy iterative improvement heuristic is then used to improve this initial assignment. The second approach solves an integer programming model that integrates approximate delivery costs. In both cases, vehicle routes are not explicitly considered, as opposed to what is done in our work.

In [1], the authors explain the main challenges in attended home delivery applications, in particular the operational management of the selected time slots when the customer orders become know. In [2], the authors explain the main similarities and differences between demand management in home delivery and airline applications, from a pricing perspective.

In [10], the authors consider a dynamic time slot pricing problem, where the delivery fees differ depending on the time slot. By dynamically varying the fees, based on the observed demand, profits can be increased. The authors propose a customer choice model based on historical data to support the proposed dynamic pricing policy.

In [9], the authors deal with a related problem known as the Time Window Assignment VRP (TWAVRP). In this problem, potential individual customers are known but not their demand. Each customer has an exogenous time window which can be seen as a shift in our problem. Different demand scenarios are defined with an associated probability. The objective is to assign a time window within the exogenous time window of each customer, to minimize the expected routing costs. A branch-and-price-and-cut algorithm is proposed to solve the problem. As mentioned

by the authors, even if the customers are grouped by zip codes, the problem differs from the TSMP because the latter is defined over a number of periods and a time window or shift must be defined for each zip code.

# 3 Heuristics

This section proposes two heuristic approaches to solve our tactical TSMP. Both heuristics exploit the similarities between our problem and the PVRP. The latter is a generalization of the classical VRP where vehicle routes are constructed for each time period over a number of consecutive time periods known as the planning horizon. A service frequency is associated with each customer over the horizon, as well as a number of possible time period schedules. The problem then consists in selecting a time period schedule for each customer and to construct routes for each time period over the horizon, while satisfying the vehicle capacity constraints. The objective is to minimize the total travel costs.

We can model our problem as a PVRP by associating a geographical zone with a customer. Furthermore, a shift is associated with a PVRP time period in the first heuristic, while a day is associated with a PVRP time period in the second heuristic. Based on these assumptions, the two heuristics can be described as follows.

## 3.1 Heuristic 1

The first problem-solving methodology is a three-phase heuristic, where both a PVRP and a number of VRPTWs (one for each day) must be solved. More precisely:

*Phase 1*

In this phase, a PVRP is solved over the shifts, using the set of shift schedules associated with each geographical zone. At the end, a particular shift schedule is assigned to each zone and a set of routes is constructed over the zones that must be visited during a given shift, for each shift of the planning horizon. It should be observed that the routes obtained for each shift start and end at the depot.

*Phase 2*

Given that a vehicle route should extend not over a shift, but over a day, the routes obtained in shifts associated with the same day must be merged. In the case of HD shifts, we must evaluate the cost of connecting the last zone of a route executed during a first HD shift with the first zone of a route executed during a second HD shift, for every possible pair of routes. To this end, a simple greedy heuristic is used. That is, the connection of least cost between two routes is first performed. These two routes are then put aside and the procedure is repeated for the remaining routes until all routes are done. In the case of OTD shifts, the procedure is first applied between the routes of the first and second OTD shifts, and then, between the routes

of the second and third OTD shifts. At the end, a set of routes covering a full day is obtained. Clearly, there is nothing to do in the case of a day shift, because a route already extends over a full day in this case.

*Phase 3*

At the end of Phase 2, we have a set of routes for each day. In Phase 3, the routes of each individual day are optimized by applying a VRPTW heuristic. In this case, the time window of each zone corresponds to the start time and end time of the shift where it must be visited. Clearly, the number of VRPTWs to be solved in Phase 3 corresponds to the number of days in the planning horizon.

This methodology is quite attractive because of its generic nature. That is, any problem-solving methodology previously developed for the well-studied PVRP and VRPTW, either heuristic or exact, can be used in Phase 1 and Phase 3, respectively. In this work, we have chosen the Unified Tabu Search (UTS) because it is simple and fast. Furthermore, UTS has proven to be efficient for solving both the PVRP and VRPTW [4, 5]. But, any other methodology could be used as well to obtain a different algorithmic behavior. In Section 4, the main characteristics of UTS are summarized.

### 3.2 Heuristic 2

The second heuristic does not use a decomposition approach like Heuristic 1, but rather solves the problem directly as a PVRP with time windows (PVRPTW), where the time windows correspond to the start time and end time of the shifts and the PVRP time periods correspond to full days. In this case, the shift schedules must first be transformed into full day schedules. For example, if the set of shift schedules for a zone is :

{{MO $s_1^{otd}$, WE $s_3^{otd}$, FR $s_1^{otd}$}, {TU $s_2^{otd}$, WE $s_1^{otd}$, TH $s_2^{otd}$}, {TU $s_2^{otd}$, TH $s_3^{otd}$, FR $s_1^{otd}$}, {MO $s_1^{otd}$, TU $s_3^{otd}$, WE $s_3^{otd}$}}

then the following set of full day schedules would be obtained (where $t$ is used instead of ($t$ $s^d$) for brevity):

{{MO, WE, FR}, {TU, WE, TH}, {TU, TH, FR}, {MO, TU, WE}}.

A previously reported variant of UTS for solving the PVRPTW is chosen again [5], although some adaptation is required here because we have to deal with dynamic time windows. More precisely, let us consider the previous example, where the shift schedules are transformed into day schedules. Given that UTS applies moves that modify the shift schedule assigned to a zone (as explained in Section 4), if the schedule {MO, WE, FR} is replaced by {TU, WE, TH}, then the time window on Wednesday must now correspond to the shift $s_1^{otd}$ instead of $s_3^{otd}$ to ensure that the zone is visited at the right time.

Thus, it is not possible to take an algorithm for the PVRPTW from the shelves

and apply it to our problem, because an adaptation is required depending on the algorithm. In other words, the generic nature of Heuristic 1 is lost. However, by avoiding the decomposition approach of Heuristic 1, better solutions should be obtained.

# 4 Unified Tabu Search

Tabu search was first proposed by Glover in 1986 [7] and has quickly become one of the best and most widespread local search method for solving combinatorial optimization problems. Tabu search is a metaheuristic that provides a good compromise between solution quality and computation time. At each iteration, the search moves from the current solution to the best solution in a neighborhood defined through various local modifications to the current solution. To avoid cycling, a tabu list forbids the search to come back to recently visited solutions. This list also forces the search to explore new regions in the solution space.

The PVRP and VRPTW in Heuristic 1, as well as the PVRPTW in Heuristic 2, are solved with different variants of UTS reported in [4, 5]. In the following, we do not discuss at length the specifics of each variant, but rather the general problem-solving approach. It should be noted that the starting solution is generated with a randomized construction heuristic where the customers are considered one by one in random order and inserted at least cost in the current solution.

The first important idea of UTS is to define neighborhood structures based on simple moves. For the PVRP, two different types of moves are proposed to modify the assignment of schedules to zones and to modify the routes (without modifying the schedules). These two moves are described below, where a time period corresponds either to a shift or to a day depending if Heuristic 1 or Heuristic 2 is applied:

1. Remove zone $i$ from route $k$ in time period $t$ and insert it in another route $k'$ in the same time period.
2. (a) Replace the time period schedule $w$ assigned to zone $i$ by another schedule $w' \in W_i$.
   (b) For each time period $t \in w$ do :
      - if $t \notin w'$, remove zone $i$ from its route in period $t$.
   (c) For each time period $t \in w'$ do :
      - if $t \notin w$, insert zone $i$ at least cost into a route of period $t$.

When UTS is applied to a VRPTW over each time period in the third phase of Heuristic 1, only moves of type 1 are used. In the case of Heuristic 2, where a PVRPTW is solved, time windows also need to be reset when a move of type 2 is applied. If a zone is removed from its original route and inserted into another

route through a move of type 1, this zone cannot go back to its original route for a given number of iterations. Similarly, if a zone is removed from a route in a given time period and inserted into another route in another time period through a move of type 2, this zone cannot go back to its original route in its original time period for a number of iterations. This tabu status can however be revoked if a new best solution is obtained in this way.

Another important idea at the core of UTS is the possibility to explore infeasible solutions. It means here that the search allows (1) excess vehicle load, (2) excess route duration and, in the case of the PVRPTW and VRPTW, (3) lateness. The latter occurs when a vehicle visits a zone after its time window's upper bound. Weights $\alpha$, $\beta$ and $\gamma$ are associated with each type of violation, in this order. Each weight multiplies the value obtained by summing up the corresponding constraint violations over all vehicles routes and over all time periods. Then, the resulting value is added to the original objective value. The three weighting parameters are dynamically increased or decreased depending if the last visited solutions are feasible or not (see [4, 5] for details).

The third main idea of UTS is a sophisticated diversification mechanism, where a solution is penalized through a factor that accounts for the number of times the so-called attributes of a solution have been involved in previous moves. In the case of the VRPTW, an attribute corresponds to the visit of zone in a given route, while for the PVRP and PVRPTW it corresponds to the visit of a zone in a given route in a given time period. A solution is then favored if this factor is low. That is, given that its attributes have seldom been modified, the solution is more likely to lead to an unexplored region of the search space.

The reader is referred to [4, 5] for additional details on these variants of UTS.

# 5 Computational results

In this section, we first explain how we designed our test instances. Then, the parameter settings for both heuristics are introduced. Comparisons between Heuristic 1 (H1) and Heuristic 2 (H2) under different scenarios are finally reported.

## 5.1 Test instances

Our test instances are derived from Solomon's Euclidean VRPTW instances [8] where travel times and distances are the same. Different types of instances were produced by varying the number of zones and their spatial distribution, the number of periods over the planning horizon, the number of shifts per period, the number of possible shift schedules for each zone and the location of the depot. Furthermore, instances are distinguished by setting the shift length to a multiple $r^s$ of the maximum time needed to visit a single zone, where the latter is defined as:

$$\max_{i \in I} \left( t_{0i} + st_i + t_{i0} \right).$$

The following values are considered for each characteristic:

- The number of zones and their spatial distribution are based on Solomon's instances. Basically, instances with 50 and 100 zones are considered. These zones can be clustered ($C$ type), randomly distributed ($R$ type), or both clustered and randomly distributed ($RC$ type).
- The number of periods over the planning horizon is set to 3 or 5.
- The number of shifts per period is set to 1, 2 or 3.
- The number of possible shift schedules for each zone depends on the number of periods $|T|$ over the planning horizon and number of shifts per period. First, the set of all possible schedules with 1 to $|T|$ shifts is generated. Then, for each zone, a certain number of possible shift schedules is randomly selected from that set. Overall, instances are created by selecting only one possible schedule per zone, half of all possible schedules per zone or all possible schedules per zone. With our parameter values, a maximum of 216 different shift schedules can be associated with a given zone.
- The location of the depot is either at the center (as taken from Solomon's instances) or at one of the corners of the service area.
- the multiple $r^s$ used for generating the shift length is set to 1 or 4.

Table 1 summarizes the values of each characteristic. A total of $2 \times 3 \times 2 \times 3 \times 3 \times 2 \times 2 = 432$ instances is obtained (since all Solomon's instances of a given distribution type, either C, R or RC, have the same coordinates and demand values).

| # Zones | Distribution | # Periods | # Shifts | # Schedules | Depot | $r^s$ |
|---|---|---|---|---|---|---|
| 50, 100 | C, R, RC | 3, 5 | 1, 2, 3 | 1, half, all | center, corner | 1, 4 |

Table 1: Characteristics of test instances

The expected number of customers $n_i$ per zone $i$ and per shift is randomly selected between 1 and 3 and this number is the same for every shift in a shift schedule. The service frequency $\eta_i$ of each zone $i$ is also randomly selected between 1 and 3. Note that possible shift schedules for a zone must necessarily contain a number of shifts corresponding to the selected frequency for that zone.

Once these values are fixed, the expected service time $st_i$ and the expected demand $d_i$ of each zone $i$ can be computed. Then, the shift length, period length, number of vehicles and vehicle capacity are computed, as it is explained below.

**Expected service time**

The expected service time is :

$$st_i = n_i \tilde{st}_i + (n_i - 1)\tilde{t}_i$$

where $\tilde{st}_i$ and $\tilde{t}_i$ are the average service time of a single customer in zone $i$ and the average travel time between two customers in zone $i$, respectively. Here, $\tilde{st}_i$ is set to the service time of the corresponding customer in Solomon's instances. The average travel time (distance) between two customers is approximated as follow. First, it is assumed that each zone $i$ is a square of side $a_i$. To calculate $a_i$, the center of each zone corresponds to the coordinates of each customer in Solomon's files. Then, the length of the diagonal of zone $i$ is set to twice the average distance between the center of zone $i$ and the center of its four closest zones. The length of the diagonal is finally divided by $\sqrt{2}$ to obtain $a_i$. Second, it is assumed that customer service requests appear in a zone according to a uniform law. The two previous assumptions lead to the following approximation for the average travel time (distance) between two customers in zone $i$:

$$\tilde{t}_i = \int_0^{a_i}\int_0^{a_i}\int_0^{a_i}\int_0^{a_i} \sqrt{(x-x')^2 + (y-y')^2}\,\mathrm{d}x\,\mathrm{d}y\,\mathrm{d}x'\,\mathrm{d}y'.$$

**Expected demand**

The expected demand of zone $i$ is

$$d_i = \tilde{d}_i \times n_i,$$

where $\tilde{d}_i$ is the average demand of a customer in zone $i$, as obtained from the demand values in Solomon's files.

**Shift and period lengths**

The shift length $D^S$ is based on parameter $r^s$ and on the maximum time required to visit a single zone, that is:

$$D^S = r^s \times \max_{i \in I}(t_{0i} + st_i + t_{i0})$$

Then, the length of a period is simply $D^P = s \times D^S$, where $s$ is the number of shifts in a period.

### Number of vehicles

Once the duration of a period is known, the number of vehicles can be defined. A coarse approximation $\tilde{K}$ if first calculated by considering the ratio of the total time required to serve all zones over the planning horizon to the total time available, that is:

$$\tilde{K} = \lceil \frac{(|I|+1) \times tz_i \times \tilde{\eta} + \sum_{i \in I} st_i \times \eta_i}{D^P \times |T|} \rceil$$

where $tz_i$ is the average travel time between the center of two zones and $\tilde{\eta}$ is the average service frequency over all zones. The first component in the numerator estimates the total travel time needed to visit all zones over the planning horizon along a single large tour, including the departure from and the return to the depot, while the second component estimates the time needed to travel inside the zones. To be sure to have a feasible solution, $\tilde{K}$ is multiplied by two to obtain the final number of vehicles. It is seldom advantageous to use a vehicle to serve a single zone, unless there is no way to do otherwise, due to the back and forth trip to the depot. Consequently, the vehicles in excess are not used.

### Vehicle capacity

Clearly, the vehicle capacity must be greater than or equal to the maximum demand over all zones. Also, there should be enough capacity to cover the total demand. The vehicle capacity $Q$ is thus defined as follow:

$$Q = \max \left( \max_{i \in I} \{d_i\}, \frac{\sum_{i \in I} d_i \eta_i}{\tilde{K} \times \tilde{\eta}} \right)$$

## 5.2 Parameter settings

In heuristic H1, the first and third phases perform similar moves. One important issue is to determine the number of iterations in each phase. Based on the convergence curve shown in Figure 1, the number of iterations of the third phase was fixed to 1,000 iterations per period (a VRPTW must be solved in each period). Figure 1 was obtained on a subset of instances by running heuristic H1 with 1,000, 5,000 and 10,000 iterations during the first phase. It was observed that the number of iterations in the first phase had a marginal impact on the convergence behavior (i.e, the three curves are similar for all practical purposes). In the remainder, the number of iterations of the third phase is set to 1,000 iterations per period.

Otherwise, the UTS parameter values for heuristics H1 and H2 have been set as in [5].

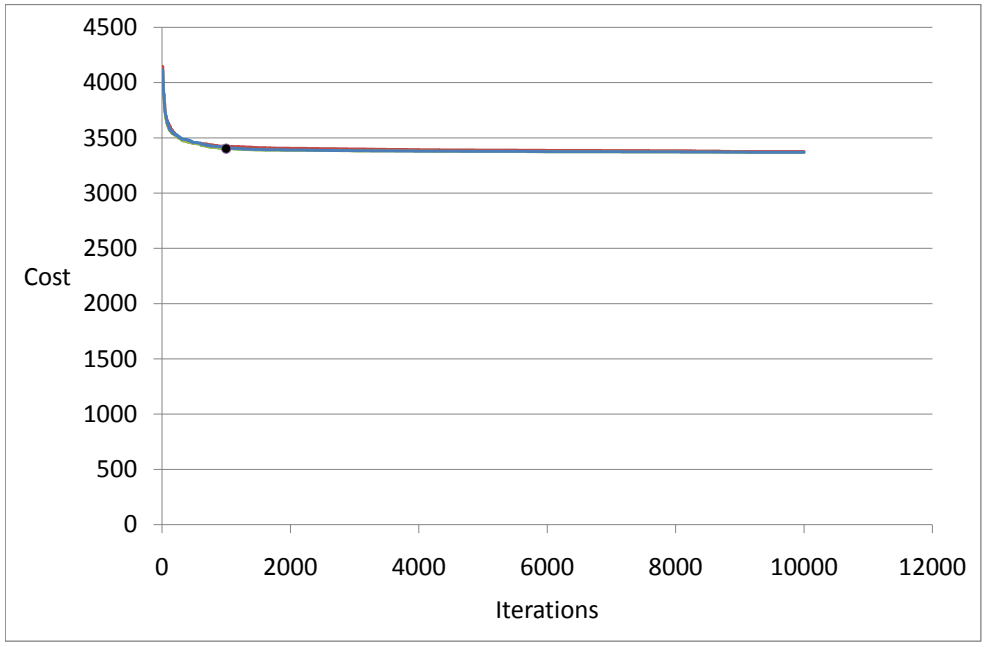


Figure 1: Convergence over 10,000 iterations

### 5.3 Comparison procedure between heuristics H1 and H2

Heuristic H1 clearly performs less calculations during one iteration, as compared with heuristic H2 (consider, for example, the management of the dynamic time windows in H2). Hence, we chose to compare both heuristics on the basis of computation time instead of number of iterations. To this end, the stopping criterion in the first phase of H1 corresponds to a fixed number of iterations. The computation time of H1 on a given instance, which is the time needed to perform the fixed number of iterations in the first phase plus the time to perform 1,000 iterations per period in the third phase, is registered and set as the time limit for heuristic H2.

### 5.4 Number of restarts

Due to the stochastic nature of the construction heuristic in UTS, restarts can be beneficial by allowing the heuristics to explore different regions of the search space. Accordingly, the total number of iterations in the first phase of H1 was set to 50,000 iterations and was evenly distributed among a variable number of runs as follow: 25 runs with 2,000 iterations, 10 runs with 5,000 iterations, 5 runs with 10,000 iterations and a single run of 50,000 iterations. We recall that each run of heuristic H1 is completed with 1,000 iterations per period in the third phase.

Table 2 shows, for each configuration, the average CPU time in seconds, the average solution value obtained by heuristics H1 and H2 and the gap in percentage between the two heuristics (a negative gap means that H2 outperforms H1).

| # iterations | 25x2000 | 10x5000 | 5x10000 | 1x50000 |
|---|---|---|---|---|
| CPU (s) | 125.4 | 105.6 | 99.3 | 94.1 |
| H1 | 3372.4 | 3399.7 | 3425.0 | 3490.0 |
| H2 | 3066.0 | 3054.5 | 3053.8 | 3058.6 |
| Gap (%) | -7.4 | -11.3 | -12.2 | -14.1 |

Table 2: Impact of number of iterations

We note that the computation time increases with the number of restarts because each run of heuristic H1 includes 1,000 iterations per period in the third phase. Thus, more runs mean more iterations in the third phase. Of particular interest is the improvement in solution quality of heuristic H1 with an increasing number of restarts (which is not the case for H2). We observed that H2 converges after approximately 1,000 iterations. The same is true for the first phase of H1. However, each iteration of H2 is more costly in terms of computation time. In the case of the 25x2000 configuration, we observed that H2 performed less than 1,000 iterations on 216 instances out of 432 and performed less than 2,000 iterations for most of the remaining instances within the allotted time (which is the time required by heuristic H1 to perform 2,000 iterations plus 1,000 iterations per period in the third phase). It means that H2 failed to fully converge on half of the test instances. This effect

is mitigated when the number of runs decreases and the number of iterations per run increases, as H2 is more likely to complete at least 1,000 iterations and fully converge. But, it might explain why the performance of H2 stagnates, and even slightly deteriorates, from the 5x10000 to the 25x2000 configurations. Considering the trade off between CPU time and solution quality for both heuristics, we chose the configuration based on 5 runs with 10,000 iterations per run in the remainder of the tests.

### 5.5 Number of zones

Results obtained with 50 and 100 zones are shown in Table 3, using the format of the previous table. An increase of 2.5% in the gap between the two heuristics is observed when the instance size doubles. This is quite modest when compared with the variations observed in Table 2 for the number of restarts. Hence, the number of zones is not such an important factor when comparing both heuristics. However, the average computation time itself increases from about 55 seconds to 143 seconds.

| # zones | 50 | 100 |
|---|---|---|
| CPU (s) | 55.3 | 143.2 |
| H1 | 2448.6 | 4401.5 |
| H2 | 2215.1 | 3892.5 |
| Gap (%) | -10.6 | -13.1 |

Table 3: Impact of number of zones

### 5.6 Number of periods

Table 4 shows the results obtained with 3 and 5 periods. Again, a slight increase of 1.7 % in the gap between the two heuristics is observed. This is an indication that the number of periods does not impact much the behavior of heuristics H1 and H2.

| # periods | 3 | 5 |
|---|---|---|
| CPU (s) | 60.4 | 138.1 |
| H1 | 3309.7 | 3540.4 |
| H2 | 2973.5 | 3134.1 |
| Gap (%) | -11.3 | -13.0 |

Table 4: Impact of number of periods

### 5.7 Number of shifts

Table 5 shows the results obtained with 1, 2 and 3 shifts per period. A substantial increase in the gap between heuristics H1 and H2 is observed when going from a

single shift to 3 shifts. Heuristic H1 even outperforms H2, although slightly, when a period is made of a single shift. These results can be explained by the three-phase decomposition approach of heuristic H1. When there is only one shift per period, the decomposition does not take place, which is clearly beneficial.

| # shifts | 1 | 2 | 3 |
|---|---|---|---|
| CPU (s) | 53.7 | 75.1 | 168.9 |
| H1 | 4311.1 | 3090.4 | 2873.6 |
| H2 | 4341.3 | 2561.5 | 2258.6 |
| Gap (%) | 0.7 | -20.7 | -27.2 |

Table 5: Impact of number of shifts

### 5.8 Depot location

Table 6 shows the results obtained when the depot is at the center of the service area or at one of the corners. The results indicate that this characteristic does not have any significant impact when comparing both heuristics.

| Depot location | Center | Corner |
|---|---|---|
| CPU (s) | 105.1 | 93.4 |
| H1 | 2789.2 | 4060.9 |
| H2 | 2493.3 | 3614.3 |
| Gap (%) | -11.9 | -12.4 |

Table 6: Impact of depot location

### 5.9 Shift length

Table 5 shows the results with the multiplier $r^s$ set to 1 and 4.

| Shift length | $r^s = 4$ | $r^s = 1$ |
|---|---|---|
| CPU (s) | 78.8 | 119.7 |
| H1 | 1872.7 | 4977.4 |
| H2 | 1839.1 | 4268.5 |
| Gap (%) | -1.9 | -16.6 |

Table 7: Impact of shift length

When $r^s$ is equal to 1, the shifts are tighter and the instances become more difficult to solve. Clearly, heuristic H2 behaves better in this setting with a gap of about 16% with H1 (as compared with 2% when $r^s$ is equal to 4). We also noted a maximum gap of 96% in solution cost when $r^s$ is set to 1 and 26% when $r^s$ is set to 4. Again, these results are related to the decomposition approach of heuristic

H1. If the shift lenght is tight, H1 struggles because each route in the first phase must return to the depot at the end of a shift, thus consuming some (scarce) time resource for this purpose instead of using that time for service purposes. This is a clear disadvantage when compared to H2.

# 6 Conclusion

This study has considered the tactical time slot management problem, a problem largely ignored in the literature, as a generalization of the well known periodic vehicle routing problem. Two heuristics based on UTS have been proposed. The first one decomposes the problem into a PVRP and a VRPTW. This decomposition allows the integration of any known methodology for solving the PVRP or the VRPTW. The second heuristic solves the problem directly. A new set of benchmark instances have been generated to compare the performance of the two heuristics.

The computational results indicate that heuristic H2 generally improves upon heuristic H1 for the same computation time. However, H1 remains competitive in certain circumstances, for example when the shift length is large. Thus, H1 could be of interest to companies with some PVRP and VRPTW software.

Future research will be oriented towards the operational problem, which is the dynamic assignment of time slots to customers as they called in. This dynamic assignment will be guided by the tactical plan generated either through heuristic H1 or H2.

# References

[1] Agatz N.A.H., Campbell A.M., Fleischmann M., Savelsbergh M.W.P., Challenges and opportunities in attended home delivery. In: The Vehicle Routing Problem, B. Golden, S. Raghavan, E. Wasil (eds), Springer: New York, pp. 379-396, 2008.

[2] Agatz N.A.H., Campbell A.M., Fleischmann M., van Nunen J.A.E.E., Demand management opportunities in E-fulfillment: What internet retailers can learn from revenue management. ERIM Report Series Research in Management No. ERS-2008-021-LIS, Erasmus Research Institute of Management, The Netherlands, 2008.

[3] Agatz N.A.H., Campbell A.M., Fleischmann M., Savelsbergh M.W.P., Time slot management in attended home delivery. Transportation Science 45, 435-449, 2011.

[4] Cordeau J.-F., Gendreau M., Laporte G., A tabu search heuristic for periodic and multi-depot vehicle routing problems. Networks 30, 105-119, 1997.

[5] Cordeau J.-F., Laporte G., Mercier A., A unified tabu search heuristic for vehicle routing problems with time windows. Journal of the Operational Research Society 52, 928-936, 2001.

[6] Francis P.M., Smilowitz K.R., Tzur M., The period vehicle routing problem and its extensions. In: The Vehicle Routing Problem, B. Golden, S. Raghavan, E. Wasil (eds), Springer: New York, pp. 73-103, 2008.

[7] Glover F., Future paths for integer programming and links to artificial intelligence. Computers & Operations Research 13, 533549, 1986.

[8] Solomon M.M., Algorithms for the vehicle routing and scheduling problems with time window constraints. Operations Research 35, 254265, 1987.

[9] Spliet R., Gabor A.F., The time window assignment vehicle routing problem. Forthcoming in Transportation Science.

[10] Yang X., Strauss A.K., Curriez C., Eglese R., Choice-based demand management and vehicle routing in E-fulfilment. Technical Report, University of Warwick, UK. Forthcoming in Transportation Science.