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

Interuniversity Research Centre on Enterprise Networks, Logistics and Transportation

Time-Window Relaxations in Vehicle Routing Heuristics

Thibaut Vidal
Teodor Gabriel Crainic
Michel Gendreau
Christian Prins

July 2013

CIRRELT-2013-43

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

Time-Window Relaxations in Vehicle Routing Heuristics

Thibaut Vidal[1,2,*], Teodor Gabriel Crainic[1,3], Michel Gendreau[1,4], Christian Prins[2]

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

[2] Laboratoire d'optimisation des systèmes industriels (LOSI), Université de Technologie de Troyes, 12, rue Marie Curie, B.P. 2060, 10010 Troyes, Cedex, France

[3] Department of Management and Technology, Université du Québec à Montréal, P.O. Box 8888, Station Centre-Ville, Montréal, Canada H3C 3P8

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

Abstract. The contribution of infeasible solutions in heuristic searches for Vehicle Routing Problems (VRP) is not a subject of consensus in the metaheuristics community. Infeasible solutions may allow transitioning between structurally different feasible solutions, thus enhancing the search, but they also lead to more complex move evaluation procedures and wider search spaces. This paper introduces an experimental assessment of the impact of infeasible solutions on heuristic searches, through various empirical studies on local-improvement procedures, iterated local searches, and hybrid genetic algorithms for the VRP with time windows and other related variants with fleet mix, backhauls, and multiple periods. Four relaxation schemes are considered, allowing penalized late arrivals to customers, early and late arrivals, returns in time, or a flexible travel time relaxation. For all considered problems and methods, our experiments demonstrate the significant positive impact of penalized infeasible solution. Differences can also be observed between individual relaxation schemes. The "returns in time" and "flexible travel time" relaxations appear as the best options in terms of solution quality, CPU time, and scalability.

Keywords: Constraint relaxations, neighborhood search, vehicle routing, time windows.

Acknowledgements. Partial funding for this project has been provided by the Champagne-Ardennes regional council, France, the Natural Sciences and Engineering Council of Canada (NSERC), and by the Fonds de recherche du Québec - Nature et technologies (FQRNT). This support is gratefully acknowledged.

---

* Corresponding author: Thibaut.Vidal@cirrelt.ca

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

1 Introduction

The vehicle routing problem with time-windows (VRPTW) is one of the most intensively studied NP-hard combinatorial optimization problems in transportation logistics, due to its major practical applications and its remarkable difficulty. Most exact methods are still rarely able to solve instances of more than 100 customers. As a result, a wide range of heuristics and metaheuristics (see the surveys of Bräysy and Gendreau 2005b,a, Gendreau and Tarantilis 2010, Vidal et al. 2013b) have been proposed to address real-life settings.

Most efficient metaheuristics rely on local search-based improvement procedure and dedicate a large part of the computation effort to the serial exploration of neighborhoods. Efficient move evaluations are thus critical for algorithmic performance and scalability. Furthermore, even finding a feasible solution to the VRPTW is a NP-hard problem (Savelsbergh 1985). Hence, any method built on the postulate that an initial feasible solution can be rapidly found, e.g. by a constructive procedure, is bound to failure on tightly-constrained problem instances.

In this context, using intermediate infeasible solutions with relaxed time-window constraints is a simple alternative to guarantee the availability of some initial solutions. Several relaxation schemes have been used in previous works, such as penalized late arrival to customers (Taillard et al. 1997), early and late arrival (Ibaraki et al. 2005), or penalized returns in time (Nagata et al. 2010). We also introduce a "flexible travel time" relaxation scheme which allows to increase the speed on an arc up to a given limit. It is frequently conjectured that infeasible solutions enable to better transition during the search between structurally different feasible solutions (Cordeau et al. 2001). In particular, a purposeful management of penalties may enable to focus the search towards borders of feasibility, a place where high quality solutions are more likely to be located (Glover and Hao 2011). However, relaxations can also lead to more complex move evaluations (Ibaraki et al. 2005) and larger search spaces. Hence, it is necessary to assess whether the use of infeasible solutions contributes significantly to the search, and also if one relaxation scheme is more suitable to progress towards good feasible solutions.

This paper contributes towards answering these questions by means of extensive experimental analysis on different types of heuristic searches. We consider simple neighborhood-centered heuristics, a multi-start local-improvement procedures and an iterated local search, and a more complex population-based method, the hybrid generic search with advanced diversity control of Vidal et al. (2012a). Several variants of each algorithm, differing by their relaxation scheme, are compared on the well-known benchmark instances of Solomon (1987) and Gehring and Homberger (1999) for the VRPTW, as well as on several other routing variants with time windows. Sensitivity analyses are conducted to measure the contribution of relaxations in the search performance in presence of different objectives and solution initialization procedures. As a result, all four relaxation schemes have a significant positive impact on solution quality. Differences of solution quality can also be observed between individual relaxation schemes for some problems. Some relaxations allow for more efficient move evaluations within local searches, and thus smaller CPU times. Overall, the "flexible travel time" scheme outperforms the other relaxations w.r.t. both speed and solution quality.

The paper is organized as follows. Section 2 describes the VRPTW and the main relaxation schemes used in previous works. State-of-the-art move evaluation procedures for each relaxation scheme are recalled in Section 3. Section 4 describes the algorithms, parameter setting and benchmark instances used in our experiments. Our experimental analysis of relaxations is then reported, considering a simple local search for the VRPTW in Section 5.1, an iterated local search, and a hybrid genetic algorithm in Section 5.2. The impact of relaxations on other VRPTW variants is assessed in Section 5.3. Section 6 concludes.

2 VRP with time windows and relaxation schemes

The vehicle routing problem with time windows (VRPTW) can be defined on a complete undirected graph $G = (\mathcal{V}, \mathcal{E})$. The depot is modeled by a pair of nodes $(v_0, v_{n+1}) \in \mathcal{V}^2$, representing respectively the origin and the final destination. Vertices $\mathcal{V}^{\text{CST}} = \mathcal{V} \backslash \{v_0, v_{n+1}\}$ stand for n customers requiring service. Each edge $(i,j) \in \mathcal{E}$ represents a possible travel from vertex v_i to vertex v_j with travel time d_{ij} (assimilated to the distance). Each customer $v_i \in \mathcal{V}^{\text{CST}}$ is characterized by a non-negative demand q_i, a service duration τ_i, as well as an interval of allowed service times $[e_i, l_i]$, called time window. A fleet of m identical vehicles is located at the depot, the capacity of each vehicle being limited to Q.

The VRPTW aims to design at most m feasible routes $\sigma^k = (\sigma^k(1), \ldots, \sigma^k(|\sigma^k|))$ for $k \in \{1 \ldots m\}$, starting and ending at the depot, and respecting capacity constraints and arrival times at vertices for each route. The most common objective in the literature involves minimizing the number of routes in priority, and then distance. Some experiments are also conducted in this paper with a distance minimization objective.

$$\text{Minimize} \quad \sum_{v_i \in \mathcal{V}} \sum_{v_j \in \mathcal{V}} \sum_{k=1}^{m} d_{ij} x_{ijk} \tag{1}$$

$$\begin{aligned}
\text{Subject to:} \quad \sum_{v_j \in \mathcal{V}} \sum_{k=1}^{m} x_{ijk} &= 1 && v_i \in \mathcal{V}^{\text{CST}} && (2)\\
\sum_{v_j \in \mathcal{V} \backslash \{v_{n+1}\}} x_{jik} - \sum_{v_j \in \mathcal{V} \backslash \{v_0\}} x_{ijk} &= 0 && v_i \in \mathcal{V}^{\text{CST}} \ ; \ k \in \{1, \ldots, m\} && (3)\\
\sum_{v_j \in \mathcal{V} \backslash \{v_0\}} x_{0jk} &= 1 && k \in \{1, \ldots, m\} && (4)\\
\sum_{v_j \in \mathcal{V} \backslash \{v_{n+1}\}} x_{j,n+1,k} &= 1 && k \in \{1, \ldots, m\} && (5)\\
\sum_{v_i \in \mathcal{V}} \sum_{v_j \in \mathcal{V}} q_i x_{ijk} &\leq Q && k \in \{1, \ldots, m\} && (6)\\
t_{ik} + d_{ij} + \tau_i - t_{jk} &\leq (1 - x_{ijk}) T && v_i \in \mathcal{V} \ ; \ v_j \in \mathcal{V} \ ; \ k \in \{1, \ldots, m\} && (7)\\
e_i \leq t_{ik} &\leq l_i && v_i \in \mathcal{V} \ ; \ k \in \{1, \ldots, m\} && (8)\\
x_{ijk} &\in \{0, 1\} && v_i \in \mathcal{V} \ ; \ v_j \in \mathcal{V} \ ; \ k \in \{1, \ldots, m\} && (9)\\
t_{ik} &\in \Re^+ && v_i \in \mathcal{V} \ ; \ k \in \{1, \ldots, m\} && (10)
\end{aligned}$$

Equations (1-10) display a mathematical formulation of the VRPTW. T is an upper bound on the time horizon. The binary decision variables x_{ijk} are set to 1 if and only if vehicle k visits v_j immediately after v_i, and the linear variables t_{ik} stand for the service date to customer v_i, when serviced by vehicle k. Equations (2-5) force each customer to be visited and establish limits on the number of incoming and outgoing edges for each node. Equation (6) imposes the capacity restrictions on routes, while Equations (7-8) impose the time-window restrictions. Equation (7) also eliminates sub-tours.

Considerable effort has been dedicated during the last decades on solving the VRPTW by means of metaheuristics, leading to a very large number of approaches, reviewed in Bräysy and Gendreau (2005a) and Gendreau and Tarantilis (2010) among others.

Table 1: Infeasible solutions in state of the art VRPTW heuristics

Authors	**Approach**	**TW Relax.**
Taillard et al. (1997)	Tabu Search	Late service
Gambardella et al. (1999)	Ant Colony Optimization & Local Search	NO
Homberger and Gehring (1999)	Evolution Strategies & Local Search	NO
Liu and Shen (1999)	Customers relocations with deteriorating moves	NO
Cordeau et al. (2001)	Unified Tabu search	Late service
Gehring and Homberger (2002)	Evolution Strategies & Tabu Search	NO
Bräysy (2003)	Node ejection chains & Variable neighborhood descent	NO
Berger et al. (2003)	GA & Local & Large Neighborhood Search	Late service
Bent and Van Hentenryck (2004)	Simulated Annealing & Large Neighborhood Search	Late service
Bräysy et al. (2004)	Injection Tree & Iterative Improvement	NO
Homberger and Gehring (2005)	Evolution Strategies & Tabu Search	NO
Ibaraki et al. (2005)	Iterated local search	Early/Late
Le Bouthillier and Crainic (2005a)	Cooperative GA and Tabu Searches	Late service
Le Bouthillier and Crainic (2005b)	Guided Cooperative GA and Tabu Searches	Late service
Mester and Bräysy (2005)	Active Guided Evolution Strategies	NO
Alvarenga et al. (2007)	Genetic Algorithm & Set Partitioning	NO
Lim and Zhang (2007)	Generalized Ejection Chains	NO
Pisinger and Ropke (2007)	Adaptive Large Neighborhood Search	NO
Hashimoto and Yagiura (2008)	Path Relinking	Return in time
Hashimoto et al. (2008)	Iterated Local Search	Early/Late
Ibaraki et al. (2008)	Iterated Local Search	Early/Late
Labadi et al. (2008)	Hybrid Genetic Algorithm	NO
Prescott-Gagnon et al. (2009)	Branch-and-price based Large Neighborhood Search	NO
Repoussis et al. (2009)	Evolutionary Algorithm & Local Search	Late service
Muter et al. (2010)	Tabu Search & Set Partitioning	NO
Nagata et al. (2010)	Hybrid GA with edge assembly crossover	Return in time
Kritzinger et al. (2012)	Variable Neighborhood Search	Late service
Vidal et al. (2013a)	Hybrid GA with cost/diversity objective	Return in time

Table 1 provides a review of recent state-of-the-art methods and their time-window relaxations. 14/28 of these state-of-the-art methods rely on time-window infeasible solutions. Three main relaxations and penalization schemes are used. The *Late service* relaxation allows linearly penalized late services but not early service to customers, while the *Early/Late* relaxation allows both early and late services. These two relaxations are often called "soft time windows" in the literature, and correspond to a relaxation of Equation (8). Finally, the relaxation of Nagata et al. (2010) allows the use of penalized *Returns in time* to reach customers in their time-windows, and corresponds to a relaxation of Equation (7). Up to this date, the hybrid genetic algorithms of Nagata et al. (2010) and Vidal et al. (2013a), which rely on this latter uncommon relaxation, have produced the best overall solutions. The contribution of this particular time-window relaxation is an open question, under investigation in this paper.

3 Move evaluation methods

Most metaheuristics for the VRPTW rely extensively on Local Search (LS) improvement procedures, exploring iteratively from an incumbent solution s a neighborhood $\mathcal{N}(s)$ of solutions defined relatively to a limited number of movements on the sequences of visits, called *moves*. In recent heuristics, the largest part of the overall computation effort is spent on evaluating moves. Move evaluation procedures must therefore be very efficient.

It is well-known in the literature (Kindervater and Savelsbergh 1997, Irnich 2008, Vidal et al. 2011, 2013a) that any classical VRP move based on a bounded number of edge exchanges or vertices relocations, such as Relocate, Swap, 2-opt, 2-opt*, or CROSS exchanges, can be assimilated to a recombination of a bounded number of partial routes, e.g. subsequences of consecutive visits, from the incumbent solution. Most local-search neighborhoods involve the same partial routes multiple times, such that managing meaningful information on them can save redundant computations and increase the local search performance.

The information on partial routes can be either pre-processed prior to move exploration and updated whenever a route change is performed, or computed on the fly if a *lexicographic order* is used for move evaluations (Savelsbergh 1985, 1992). We opted in this paper to rely on pre-processing, since the computational effort required to compute the information is generally negligible when compared to the effort required by move evaluations, and because it allows for an efficient random exploration of granular local search neighborhoods.

Pre-processing information on partial routes is frequently done in heuristics. Consider the example of the capacitated VRP. Keeping track of partial demands and distances on each partial route from the incumbent solution provides the means to evaluate the load of any recombined route with a bounded number of sums. This opens the way to $O(1)$ time load-feasibility checks during the local search, compared to $O(n)$ for a straightforward method that browses the routes and sums the loads. In a similar manner, pre-processing meaningful information on partial routes can contribute to reduce the computational complexity of time-window feasibility checks or, when applicable, time-penalties evaluations on the routes issued from the local search moves. Efficient move-evaluation methods exploiting these properties are presented in the following for different relaxation schemes.

3.1 No infeasible solution

When no infeasible solutions are used, checking route feasibility within a local search can be efficiently done with the approach of Savelsbergh (1985, 1992) and Kindervater and Savelsbergh (1997). For any partial route σ, the sum of travel and service times $T(\sigma)$, the earliest possible completion time $E(\sigma)$, and the latest feasible starting date $L(\sigma)$ are pre-processed. These values can be computed by induction on the concatenation operation, starting with the base case of a partial route $\sigma_0 = (v_i)$ containing a single visit where $T(\sigma_0) = \tau_i$, $E(\sigma_0) = e_i + \tau_i$ and $L(\sigma_0) = l_i$. Equations (11-13) are then used to derive this information on any larger partial route $\sigma_1 \oplus \sigma_2 = (\sigma_1(1), \dots, \sigma_1(|\sigma_1|), \sigma_2(1), \dots, \sigma_2(|\sigma_2|)$ obtained from a concatenation of two partial routes $\sigma_1 = (\sigma_1(1), \dots, \sigma_1(|\sigma_1|))$ and $\sigma_2 = (\sigma_2(1), \dots, \sigma_2(|\sigma_2|))$.

$$T(\sigma_1 \oplus \sigma_2) = T(\sigma_1) + d_{\sigma_1(|\sigma_1|)\sigma_2(1)} + T(\sigma_2) \quad (11)$$

$$E(\sigma_1 \oplus \sigma_2) = \max\{E(\sigma_1) + d_{\sigma_1(|\sigma_1|)\sigma_2(1)} + T(\sigma_2), E(\sigma_2)\} \quad (12)$$

$$L(\sigma_1 \oplus \sigma_2) = \min\{L(\sigma_1), L(\sigma_2) - d_{\sigma_1(|\sigma_1|)\sigma_2(1)} - T(\sigma_1)\} \quad (13)$$

Any such concatenation of two subsequences of visits σ_1 and σ_2 is feasible if and only if $E(\sigma_1) + d_{\sigma_1(|\sigma_1|)\sigma_2(1)} \leq L(\sigma_2)$. These equations enable to check in $O(1)$ time the feasibility of routes issued from moves, assimilated to a recombination of a bounded number of partial routes.

3.2 Late and Early/Late service

Penalized late or early services, also referred as *soft time windows*, are relevant in a variety of application cases in which a trade-off must be established between service quality and routing costs. This relaxation is also frequently used within VRPTW heuristics to achieve better performance (e.g. in Taillard et al. 1997). As discussed in Fu et al. (2007), various types of soft time windows, differing by the shape of the associated penalty function, are used in the literature. In this work we consider the two most frequent types: penalizing late services or penalizing both early and late services. In these cases, some waiting time can occur upon an early arrival.

When only late services are allowed, computing the total penalty on a route can be trivially done in $O(n)$ by servicing each customer as early as possible. This route evaluation procedure will be denoted here as “simple evaluation”.

In contrast, allowing both early and late deliveries leads to a combinatorial optimization problem. Indeed, upon an early arrival to any customer, choice must be made on either waiting or paying a penalty for early service. For a fixed route σ, these decisions can be modeled as a $\{R, D|\o\}$ timing problem (Vidal et al. 2011). A linear programming formulation is presented in Equations (14-15). Coefficients α and β represent unit penalties for earliness and lateness.

$$\min_{t_1,\dots,t_{|\sigma|}} \quad \sum_{i=1}^{n_k} \alpha(e_{\sigma(i)} - t_i)^+ + \sum_{i=1}^{n_k} \beta(t_i - l_{\sigma(i)})^+ \tag{14}$$

$$\text{s.t.} \quad t_i + \tau_{\sigma(i)} + d_{\sigma(i)\sigma(i+1)} \leq t_{i+1} \qquad 1 \leq i < |\sigma| \tag{15}$$

The model (14-15) is encountered in various operations research fields, in transportation logistics, project and machine scheduling, as well as in statistics as a generalization of the isotonic regression problem (Robertson et al. 1988). Therefore, various solution algorithms are available, some of which provide a solution in $O(n \log n)$ time (see Garey et al. 1988 and Dumas et al. 1990, among others).

Some dynamic programming methods (Yano and Kim 1991, Hendel and Sourd 2006, Ibaraki et al. 2005, 2008) provide the means to pre-process information on partial routes to speed-up the search: a function $F(\sigma)(t)$ representing the minimum cost to service the partial route σ while arriving at the last customer before time t, and a function $B(\sigma)(t)$ stating the minimum cost of servicing σ after time t. These functions are algebraically represented in the algorithm using appropriate data structures. For a partial route $\sigma_0 = (v_i)$ with a single vertex, $F_{\sigma_0}(t) = \min_{x \leq t} c_i(x)$ and $B_{\sigma_0}(t) = \min_{x \geq t} c_i(x)$. These values can then be computed by forward dynamic programming, or backward dynamic programming, respectively, on longer routes made of a concatenation of a route σ and an additional delivery v_i using Equations (16-17).

$$F(\sigma \oplus v_i)(t) = \min_{0 \leq x \leq t} \{c_i(x) + F(\sigma)(x - \tau_{\sigma(|\sigma|)} - d_{\sigma(|\sigma|),i})\} \tag{16}$$

$$B(v_i \oplus \sigma)(t) = \min_{x \geq t} \quad \{c_i(x) + B(\sigma)(x + \tau_i + d_{i,\sigma(1)})\} \tag{17}$$

Equation (18) then provides the optimal service cost $Z^*(\sigma_1 \oplus \sigma_2)$ for a route issued of the concatenation of two partial routes σ_1 and σ_2.

$$Z^*(\sigma_1 \oplus \sigma_2) = \min_{x \geq 0}\{F(\sigma_1)(x) + B(\sigma_2)(x + \tau_{\sigma_1(|\sigma_1|)} + d_{\sigma_1(|\sigma_1|)\sigma_2(1)})\} \tag{18}$$

This equation allows, among other, for efficient evaluations of 2-OPT* neighborhoods. Moreover, any route resulting from the concatenation of three partial routes $(\sigma_1 \oplus \sigma_L \oplus \sigma_2)$ can be evaluated by relying $|\sigma_L|$ successive times on Equation (16) to yield the information on $\sigma' = \sigma_1 \oplus \sigma_L$, and computing $Z^*(\sigma' \oplus \sigma_2)$ with Equation (18). This strategy leads to efficient evaluations of moves such as RELOCATE, SWAP, for which $|\sigma_L| \leq 1$, and CROSS, for which $|\sigma_L|$ stands for the maximum size of the exchanged segment (limited to 2 in our experiments). The resulting move evaluation complexity is $O(|\sigma_L|\Sigma_i \xi(c_i))$, where $\xi(c_i)$ represents the number of pieces in each function $c_i(t_i)$. In soft time-windows settings, $\xi(c_i) = 3$ for any customer v_i, and thus moves are evaluated in amortized $O(|\sigma_L|n)$.

In the particular case where all functions $c_i(t_i)$ are convex (e.g. in soft time windows settings), *advanced implementations* based on heap or search tree data structures (Hendel and Sourd 2006, Ibaraki et al. 2008) achieve a route evaluation complexity of $O(\log n)$. The two relaxations schemes *Early/Late* and *Late Service* thus lead to the same best known logarithmic move evaluation complexity.

3.3 Returns in time

The relaxation proposed by Nagata et al. (2010) is based on linearly penalized "time warps", which are used upon a late arrival to "return in time" to the end of the time windows. Any unit of time warp is penalized by a factor α. As demonstrated in the following, despite its limited practical significance, the relaxation proves to be particularly useful to allow intermediate infeasible solution in heuristics while still allowing for amortized $O(1)$ move evaluations.

A possible way to efficiently perform move evaluations (Vidal et al. 2013a) requires computing on any partial route σ the minimum duration $D(\sigma)$ to perform the services, the minimum time warp usage $TW(\sigma)$, and the earliest $E(\sigma)$ and latest visit $L(\sigma)$ to the first vertex allowing a schedule with minimum duration and time-warp use. For a partial route $\sigma_0 = (v_i)$ containing a single vertex $D(\sigma_0) = \tau_i$, $TW(\sigma_0) = 0$, $E(\sigma_0) = e_i$ and $L(\sigma_0) = l_i$. The same information can be computed on larger routes by induction on the concatenation operator with Equations (19-25).

$$D(\sigma_1 \oplus \sigma_2) = D(\sigma_1) + D(\sigma_2) + d_{\sigma_1(|\sigma_1|)\sigma_2(1)} + \Delta_{WT} \tag{19}$$
$$TW(\sigma_1 \oplus \sigma_2) = TW(\sigma_1) + TW(\sigma_2) + \Delta_{TW} \tag{20}$$
$$E(\sigma_1 \oplus \sigma_2) = \max\{E(\sigma_2) - \Delta, E(\sigma_1)\} - \Delta_{WT} \tag{21}$$
$$L(\sigma_1 \oplus \sigma_2) = \min\{L(\sigma_2) - \Delta, L(\sigma_1)\} + \Delta_{TW} \tag{22}$$
$$\textit{where } \Delta = D(\sigma_1) - TW(\sigma_1) + d_{\sigma_1(|\sigma_1|)\sigma_2(1)} \tag{23}$$
$$\Delta_{WT} = \max\{E(\sigma_2) - \Delta - L(\sigma_1), 0\} \tag{24}$$
$$\Delta_{TW} = \max\{E(\sigma_1) + \Delta - L(\sigma_2), 0\} \tag{25}$$

These equations lead to amortized $O(1)$ time move evaluations. This complexity is identical to the case where no infeasible solutions are used (Section 3.1).

3.4 Flexible service and travel times

The amount of time warp is not limited in the "return in time" relaxation. Routes servicing a customer i at time t_i, paying for a time warp, and servicing the next customer j at time $t_j < t_i$ are thus allowed. To avoid this issue, we investigate another relaxation alternative based on flexible travel times, which allows speed-ups under some limits and forbids negative travel durations. This relaxation is a very simple case of flexible travel and service time where the penalty $p_{ij}(\delta t)$ as a function of the service and travel duration is a piecewise linear function given in Equation (26). In our experiments, the minimum duration allowed for a service to a customer v_i is $\tau_i^{min} = \tau_i/2$, and the minimum travel time for driving from any vertex v_i to any vertex v_j is set to $d_{ij}^{min} = d_{ij}/2$.

$$p_{ij}(\delta t) = \begin{cases} +\infty & \text{if } \delta t < d_{ij}^{min} + \tau_i^{min} \\ \alpha \times (d_{ij} - \delta t) & \text{if } d_{ij}^{min} + \tau_i^{min} \leq \delta t < d_{ij} + \tau_i \\ 0 & \text{if } d_{ij} + \tau_i \leq \delta t \end{cases} \tag{26}$$

Route evaluations in presence of this relaxation can be managed by means of a combination of the previous methodologies. First, Equations (11-13) are used to check whether the path is time-window feasible when the maximum speed and minimum service time is used. If it is feasible, Equations (19-25) are used to measure the necessary amount of *return in time* along the path, which is equivalent to the necessary speedups, otherwise the route is declared infeasible. Thus, the relaxation cost related to flexible travel times can be measured in amortized $O(1)$ operations.

4 Vehicle routing heuristics

Table 2 recalls the different relaxation alternatives and their respective move evaluation complexities. The impact of these different relaxation schemes is investigated in the following. We investigate the related solution quality, CPU time and number of iterations to reach a local minimum. We also consider three different heuristics and metaheuristics, a multi-start local-improvement procedure, an iterated local search and a hybrid genetic search, two solution initialization procedures, random or I1-insertion of Solomon (1987), two different objectives, distance minimization or fleet-size minimization, and five VRPTW variants with multi-depots, multi-periods, backhauls, or fleet mix.

Table 2: Relaxations schemes for the VRPTW

	Relaxation	**Complexity**	**First introduced in**
NoInf	No infeasible solution	$O(1)$	Kindervater and Savelsbergh (1997)
Late	Late arrival	$O(\log n)$	Cordeau et al. (2001) or Ibaraki et al. (2008)
E/L	Early/Late arrival	$O(\log n)$	Ibaraki et al. (2005) or Ibaraki et al. (2008)
Return	Return in time	$O(1)$	Nagata et al. (2010)
Flex	Flexible service and travel time	$O(1)$	This paper

For each alternative relaxation, the move evaluation strategy presented in Section 3 has been implemented. It should be noted that the simpler $O(n)$ move evaluation procedures are used for the Late and E/L relaxations. Faster $O(\log n)$ evaluations are known to be achievable, but the required algorithm development is high, and the purpose of this paper is not to propose a horse-racing algorithm, but rather to investigate solution quality on a fixed number of iterations in presence of different relaxations. The three heuristics and metaheuristics considered in this work are described in the following:

The Multi-Start Local Improvement (MS-LI) procedure is based on classic vehicle routing neighborhoods: 2-OPT, 2-OPT* as well as CROSS and I-CROSS exchanges restricted to sequences of size smaller than $L_{max} = 2$. Moves are explored in random order, any improving move being directly applied, until no improvement can be found in the whole neighborhood. The resulting improvement procedure is described in Algorithm 1. In this algorithm, the partial route information is preprocessed and updated by means of functions "updateData" (lines 2, 11, 13 and 16). The procedure is run N_{DESCENT} times with penalty values of $\alpha = \beta = 2$. This value has been chosen in order to obtain feasible solutions as output in around 50% of cases. Any infeasible solution undergoes a second local improvement with penalties $\alpha = \beta = 100$ to restore feasibility.

Algorithm 1 Local Improvement(s_{CURR})

1: isEnd = false
2: **for** each route $r \in s_{\text{CURR}}$ **do** updateData(r)
3: **while not** isEnd **do**
4:    isEnd = true
5:    **for** $i = 1, \ldots, n$ **and** $j = 1, \ldots, n$ **do**
6:       $c_i \leftarrow$shuffledNodeOrder(i) ; $c_j \leftarrow$ shuffledNodeOrder(j) ;
7:       $r_i \leftarrow$getRoute(c_i) ; $r_j \leftarrow$ getRoute(c_j) ;
8:       **if** $r_i \neq r_j$ **and** {isImprovingCROSS(c_i,c_j) **or** isImproving2opt*(c_i,c_j)} **then**
9:          $s_{\text{CURR}} \leftarrow$ performMove(s_{CURR};c_i,c_j) ; updateData(r_i,r_j); isEnd = false
10:       **if** $r_i == r_j$ **and** {isImprovingOrOpt(c_i,c_j) **or** isImproving2Opt(c_i,c_j)} **then**
11:          $s_{\text{CURR}} \leftarrow$ performMove(s_{CURR};c_i) ; updateData(r_i); isEnd = false;
12: **return** s_{CURR}

The Multi-Start Iterated Local Search (MS-ILS) considered in our experiments, depicted in Algorithm 2, is similar to the one described in Prins et al. (2009) The method starts from a randomly generated solution. Subsequently, from an incumbent solution, MS-ILS generates n_{C} child solutions by applying a shaking operator and the local-improvement procedure of Algorithm 1, the best child solution being taken as new incumbent solution for the next iteration. For this reason, the method is sometimes called Evolutionary Local Search. As in Prins et al. (2009), shaking operations are performed on the giant-tour solution representation, a polynomial Split algorithm being applied to obtain the associated complete solution. Also, the penalty coefficients are adapted during the search relatively to the proportion of feasible solutions as in Vidal et al. (2012a). The method is applied n_{P} times. Each run is terminated after n_{I} consecutive iterations without improvement of the best solution. The overall best solution is finally returned.

The Hybrid Genetic Algorithm with advanced diversity control of Vidal et al. (2012a, 2013a) relies on four main successful concepts: 1) A hybridization of genetic algorithms with local search procedures; 2) The use of penalized infeasible solutions, managed through two distinct sub-populations during the search; 3) A solution representation as a giant tour *without*

Algorithm 2 MS-ILS()

for $i_{\text{P}} = 1$ to n_{P} **do**
  $s_{\text{CURR}} \leftarrow$ getInitialSolution() ; $s_{\text{BEST}} \leftarrow s_{\text{CURR}}$; $i_{\text{ILS}} = 0$
  while $i_{\text{ILS}} < n_{\text{I}}$ **do**
    $S_{\text{CHILDREN}} \leftarrow \emptyset$
    for $i_{\text{C}} = 1$ to n_{C} **do**
      $s \leftarrow$Shaking(s_{CURR})
      $s \leftarrow$Split(s)
      $S_{\text{CHILDREN}} \leftarrow$localImprovement($s$)
    $s_{\text{CURR}} \leftarrow$bestElement($S_{\text{CHILDREN}}$)
    if cost(s_{CURR}) $<$ cost(s_{BEST}) **then** $s_{\text{BEST}} \leftarrow s_{\text{CURR}}$; $i_{\text{ILS}} = 0$
    else $i_{\text{ILS}} = i_{\text{ILS}} + 1$
    adaptPenaltyCoefficients()
return bestSolutionEver()

trips delimiters with an optimal *Split* procedure for delimiter computation (Prins 2004); 4) A bi-criteria fitness individual evaluation function computed as a weighted sum of *cost* and *contribution to diversity* measures. Starting from initial population, new solutions are iteratively generated by means of a binary-tournament selection, PIX crossover, *Split*, and the local-improvement procedure of Algorithm 1. Any new solution is inserted in the population, and a survivor selection phase is triggered whenever the population reaches a maximum size of $\mu + \lambda$ to select out the worst λ individuals according to our fitness measure. Diversification and decomposition phases (Vidal et al. 2013a), triggered periodically after a given number of iterations, enable to enhance the search towards good solutions and unknown areas of the search space.

5 Empirical comparison of relaxations

The impact of relaxations, in presence of different metaheuristic schemes and construction methods, is assessed on the VRPTW benchmark instances of Solomon (1987) with 100 customers and Gehring and Homberger (1999) with 200 and 400 customers. These instances are grouped in 6 categories which differ by the characteristics of the geographical distribution of customers. Customers are uniformly distributed in the R1 and R2 problem classes, clustered in the C1 and C2 classes, whereas RC1 and RC2 mix both uniform and clustered customer distributions. Class C1, R1 and RC1 contain problems with short time horizon and small vehicle capacities, while C2, R2 and RC2 have larger vehicle capacities and lead to longer routes. Two different objectives are considered: either distance minimization, or minimization of fleet size and then distance. The impact of initial solutions is also examined, by testing either a random solution obtained by randomly assigning and positioning customers into routes, or a solution produced by the I1 insertion heuristic of Solomon (1987).

The parameters of HGA (Vidal et al. 2012a) and MS-ILS (Prins et al. 2009), with $(n_{\text{P}}, n_{\text{I}}, n_{\text{C}}) = (5, 50, 10)$, are maintained identical to the original papers. The next sections report in turn the experiments on the MS-LI (Section 5.1), MS-ILS and HGA for the VRPTW (Section 5.2). Further tests on different time-window constrained VRP variants and benchmark instances are reported in Section 5.3.

5.1 Multi-start local-improvement heuristic

Distance minimization. Table 3 displays the average solutions retrieved by each method in 10 runs, using either the random initialization procedure or the I1 heuristic of Solomon (1987). The initial solution provided by the I1 heuristic is also displayed in Column 2 as well as the best known solution. Distances are aggregated by problem classes. The last four lines indicate the cumulated distance (CTD), the distance gap (%) to the best solution, the number of local search iterations and CPU time, averaged on all instances. Double precision numbers have been used for distance computations.

Table 3: Distance minimization on the VRPTW benchmark instances of Solomon (1987)

Inst.	Sol I1	NoInf	Late	E/L	Return	Flex	No Inf	Late	E/L	Return	Flex	BKS
		Random Initial Solution					Solomon1 Initial Solution					
R1	1431.97	1242.52	1212.87	1211.76	1211.17	1211.56	1208.05	1204.38	**1203.07**	1203.36	1203.28	1178.98
R2	1326.64	918.97	915.40	914.98	914.57	915.41	915.61	907.84	907.21	909.33	**906.81**	877.20
C1	936.48	**828.38**	828.39	**828.38**	**828.38**	**828.38**	828.40	828.40	828.41	**828.38**	**828.38**	828.38
C2	696.57	592.63	591.49	591.44	**591.09**	591.39	592.01	591.73	591.55	591.33	591.29	589.86
RC1	1578.28	1440.39	1388.62	1386.57	1386.14	1385.61	1382.52	1377.88	1376.90	**1375.08**	1375.67	1338.18
RC2	1653.61	1056.67	1046.00	1045.42	1044.19	1046.80	1053.07	1040.95	1041.13	**1038.77**	1040.91	1003.95
CTD	71633	57192	56288	56249	56221	56254	56245	55979	55948	55940	**55933**	54708
D(%)	30.99%	3.91%	2.67%	2.61%	2.56%	2.62%	2.66%	2.20%	2.16%	2.14%	**2.13%**	0.00
IT-LS		43222	44081	43864	43676	44936	7437	8498	8597	8636	8643	
T(min)		0.08	0.20	6.06	0.09	0.13	0.07	0.21	6.34	0.09	0.11	

Table 3 demonstrates the significant contribution of time-window relaxations to the solution quality. Relaxations lead to a Gap reduction of -1.24% to -1.35% when random initial solutions are used, otherwise of -0.46% to -0.53% when the I1 construction procedure is used. Furthermore, relaxations tend also to mitigate the impact of low quality initial solutions. When opting for a random construction instead of an I1 constructive procedure, the Gap increases by $+0.42\%$ to $+0.49\%$ in presence of relaxations, whereas it strongly increases by $+1.25\%$ if only feasible solutions are used.

It should be noted that evaluating neighborhoods with penalized infeasible solutions only leads to a slight difference of computational effort, with 0.07 and 0.08 minutes for NoInf compared to 0.09 to 0.13 minutes if a computationally efficient relaxation scheme such as Return or Flex is used. This observation goes in accordance with the computational complexity results of Section 3, and motivates the choice of one of these two relaxations to achieve a good trade-off between solution quality and CPU time.

Fleet-size minimization. When relaxations are used, addressing the hierarchical objective of fleet size minimization and distance can be simply done by iteratively applying Algorithm 1 and decrementing the fleet size as long as a final feasible solution is found. Fleet-size minimization with only feasible solutions is more intricate. Any such method has to start from a feasible solution, which has inevitably a too large number of routes, and reduce the number of routes during the search. Several tailored procedures have been proposed to that extent, using auxiliary objectives to progress towards empty routes as in Gendreau et al. (1996) and Bent and Van

Hentenryck (2004), or ejection chains and route-removal operations (Nagata and Bräysy 2009). However, these procedures are complex and require different types of neighborhoods, thus not allowing for a fair comparison with the simple heuristics that we study. For this reason, only the relaxed heuristic variants are compared on the fleet-minimization objective.

Table 4 displays the results of the fleet minimization algorithm for each relaxation scheme, using either the constructive heuristic I1 or the random procedure for solution initialization. The results of the three-phase local and large neighborhood search of B02 (Bräysy 2002) are displayed. B02 was identified as the "best route construction and local search heuristic" in the survey of Bräysy and Gendreau (2005a). For each problem class, the average number of vehicles and distance on 10 runs is displayed. The last six lines display the cumulated number of vehicles (CNV), cumulated distance (CTD), average fleet size gap V(%), distance gap D(%), LS iterations and CPU time on all 56 instances.

Table 4: Fleet minimization on the VRPTW benchmark instances of Solomon (1987)

		Random Initial Solution				**SolomonI1 Initial Solution**					
Inst.	**SolI1**	**Late**	**E/L**	**Return**	**Flex**	**Late**	**E/L**	**Return**	**Flex**	**B02**	**BKS**
R1	13.42	12.52	12.45	12.47	12.47	12.51	12.46	12.48	12.44	**12.17**	11.92
	1431.97	1224.24	1230.79	1225.21	1229.28	1219.68	1222.69	1219.51	1225.74	**1253.24**	1210.33
R2	3.18	2.76	2.75	2.77	2.75	2.75	2.75	**2.74**	2.75	2.82	2.73
	1326.64	987.91	986.55	984.70	988.56	993.44	988.78	**992.73**	991.36	1039.56	951.03
C1	10.00	10.00	10.00	10.00	10.00	10.00	**10.00**	10.00	10.00	10.00	10.00
	936.48	830.69	830.51	831.27	830.71	828.54	**828.42**	828.47	828.44	832.88	828.38
C2	3.13	3.00	3.00	3.00	3.00	3.00	3.00	3.00	**3.00**	3.00	3.00
	696.57	590.76	590.56	590.99	590.99	591.03	590.55	590.64	**590.54**	593.49	589.86
RC1	13.50	12.06	12.09	12.05	12.06	12.04	12.03	12.05	12.04	**11.88**	11.50
	1578.28	1405.86	1401.81	1407.78	1404.09	1403.59	1397.71	1400.73	1402.24	**1408.44**	1384.16
RC2	3.75	3.25	3.26	3.26	3.25	**3.25**	3.25	3.25	3.25	3.25	3.25
	1653.61	1183.05	1181.11	1176.11	1177.11	**1176.92**	1177.22	1179.14	1179.74	1244.96	1119.24
CNV	449.0	417.1	416.5	416.6	416.3	416.7	415.9	416.3	415.8	**412.0**	405
CTD	71633	58471	58483	58415	58480	58393	58329	58374	58450	**59945**	57187
V(%)	12.41%	2.28%	2.17%	2.32%	2.03%	2.09%	1.90%	1.89%	1.90%	**1.47%**	0.00%
D(%)	25.19%	2.22%	2.21%	2.11%	2.22%	2.12%	2.00%	2.09%	2.19%	**3.15%**	0.00%
IT-LS		50273	45049	49078	54794	10577	10643	10117	10427		
T(min)		0.74	24.71	0.29	0.41	0.52	18.03	0.20	0.26	4.60	

According to these results, as long as one of the considered relaxation scheme is introduced, the specific choice of relaxation scheme has no significant major impact on the results of the fleet minimization. There is no "clear winner", but a tendency for the Late relaxation to produce slightly worst results in terms of fleet size. This can be related to the fact that infeasible insertions at the beginning of routes are likely to result in massive penalties in the Late relaxation scheme, leading to imbalanced insertion capabilities, and thus a search bias. In this setting, we also suggest to rely on the less computationally expensive relaxation scheme such as Return or Flex. The relaxation Return, in particular, leads to the best average results in both settings, distance and fleet-size minimization.

These simple local searches compare favorably with B02, yielding better solutions on 4 sets out of 6, and in general producing lower overall distances. For the two remaining sets, the better results of B02 are a normal consequence of the more advanced route-minimization techniques which are used, such as large neighborhoods and ejection chains. Even if the goal of these experiments with these basic local searches was to investigate the relaxations rather than competing with intricate state-of-the-art methods, it is noteworthy that this simple process of decrementing the fleet size and solving relaxed problems produces solutions of fair quality.

5.2 Iterated Local Search and Hybrid Genetic Algorithm

Similar experiments have been conducted with two richer metaheuristic frameworks, HGA and MS-ILS described in Section 4, using the instances of Solomon (1987) and Gehring and Homberger (1999). As previously, double precision numbers have been used for distance computations. There was also no clear difference related to the use of the I1 constructive procedure for solution initialization, perhaps because the proposed metaheuristics themselves already produce higher quality and more diverse solutions than any construction procedure in a few seconds. For the sake of simplicity, a random initial solution is thus used in the remainder of this section.

The next two tables report the solutions retrieved by MS-ILS and HGA with different relaxation schemes, with a distance minimization objective (Table 5), or with a fleet-size minimization objective (Table 6). 10 runs were performed for Late, Return and Flex relaxations, and one single run for E/L because of the prohibitively high CPU time required for these tests. Average solution values are reported in all cases. Results are aggregated by problem size and classes. We also display the results reported by the current best metaheuristic for the VRPTW with distance minimization (Labadi et al. 2008 – LPR08), for the VRPTW with fleet-size minimization (Nagata et al. 2010 – NB10), and the best solution found in our experiments during all runs. The last lines indicate the cumulated number of vehicles (CNV), cumulated distance (CTD), average fleet size gap V(%), distance gap D(%), LS iterations and CPU time on all 176 instances.

Table 5: HGA and MS-ILS for distance minimization on the larger VRPTW benchmark instances of Solomon (1987) and Gehring and Homberger (1999)

		HGA					**MS-ILS**						
Inst	n	**NoInf**	**Late**	**E/L**	**Return**	**Flex**	**No Inf**	**Late**	**E/L**	**Return**	**Flex**	**LPR08**	**Best**
R1	100	1181.25	1180.03	**1178.98**	1179.93	1179.90	1188.27	1181.59	1181.24	1182.13	1181.14	1184.16	1178.98
R2	100	879.26	877.43	**877.23**	877.24	877.24	884.88	879.00	878.97	879.21	878.61	879.51	877.20
C1	100	**828.38**	**828.38**	**828.38**	**828.38**	**828.38**	**828.38**	**828.38**	**828.38**	**828.38**	**828.38**	**828.38**	828.38
C2	100	**589.86**	**589.86**	**589.86**	**589.86**	**589.86**	**589.86**	**589.86**	**589.86**	**589.86**	**589.86**	**589.86**	589.86
RC1	100	1350.79	1339.07	**1338.32**	1338.80	1338.86	1365.76	1347.98	1345.65	1350.14	1347.59	1352.02	1338.18
RC2	100	1007.27	1004.03	**1004.00**	1004.20	1004.16	1012.42	1006.62	1008.35	1006.94	1007.06	1009.37	1003.95
R1	200	3610.89	3595.61	**3593.81**	3596.40	3595.85	3681.07	3623.87	3615.07	3624.12	3618.22		3588.45
R2	200	2657.51	2636.57	**2635.12**	2636.57	2636.04	2698.52	2648.30	2650.73	2646.21	2644.06		2633.59
C1	200	2680.78	**2678.96**	**2678.96**	2679.00	2678.97	2685.58	2683.80	2684.09	2684.22	2683.41		2678.96
C2	200	1832.12	1830.94	1830.11	1830.08	**1829.80**	1858.74	1833.55	1833.42	1833.72	1831.99		1828.01
RC1	200	3169.58	3165.05	3166.28	**3164.75**	3165.82	3213.04	3182.80	3188.00	3184.67	3181.13		3158.83
RC2	200	2318.49	2300.27	2301.34	**2300.23**	2300.28	2350.74	2313.37	2314.48	2310.13	2306.96		2293.88
R1	400	8471.08	8414.10	8415.73	8411.10	**8409.35**	8763.45	8531.72	8546.83	8535.81	8514.34		8378.08
R2	400	5808.77	5711.50	5714.42	5715.88	**5707.83**	6079.31	5783.10	5795.39	5774.01	5751.82		5681.16
C1	400	7063.14	7043.86	7047.07	**7041.94**	7042.00	7101.20	7071.73	7079.89	7074.78	7067.44		7037.47
C2	400	3914.42	3859.04	3856.24	**3855.54**	3857.36	4028.82	3899.53	3903.50	3896.37	3880.36		3839.97
RC1	400	7944.93	7921.38	7925.84	7918.99	**7917.30**	8148.93	8037.51	8034.35	8037.74	8012.35		7875.81
RC2	400	5022.01	4939.59	4935.83	4938.82	**4935.04**	5250.40	4985.22	4987.37	4977.27	4960.62		4911.09
CTD		599823	595700	595718	595620	**595483**	613791	600804	601181	600678	599377		593761
D(%)		0.77%	0.23%	0.21%	0.22%	**0.20%**	2.42%	0.81%	0.85%	0.80%	0.65%		0.00%
T(min)		17.41	27.71	460.33	14.01	18.64	10.53	15.04	262.49	8.06	10.63		

Table 6: HGA and MS-ILS for fleet size minimization on the larger VRPTW benchmark instances of Solomon (1987) and Gehring and Homberger (1999)

		HGA				**MS-ILS**					
Inst.	n	**Late**	**E/L**	**Return**	**Flex**	**Late**	**E/L**	**Return**	**Flex**	**NB10**	**Best**
R1	100	11.93	11.92	11.92	11.93	11.95	11.92	11.94	11.92	**11.92**	11.92
		1210.23	1211.98	1211.45	1210.52	1211.36	1213.16	1213.73	1213.52	**1210.34**	1210.33
R2	100	2.73	2.73	2.73	2.73	2.73	2.73	2.73	2.73	**2.73**	2.73
		951.93	952.30	952.23	952.32	954.02	954.27	953.85	954.36	**951.71**	951.03
C1	100	**10.00**	**10.00**	**10.00**	**10.00**	**10.00**	**10.00**	**10.00**	**10.00**	**10.00**	**10.00**
		828.38	**828.38**	**828.38**	**828.38**	**828.38**	**828.38**	**828.38**	**828.38**	**828.38**	828.38
C2	100	**3.00**	**3.00**	**3.00**	**3.00**	**3.00**	**3.00**	**3.00**	**3.00**	**3.00**	3.00
		589.86	**589.86**	**589.86**	**589.86**	**589.86**	**589.86**	**589.86**	**589.86**	**589.86**	589.86
RC1	100	11.53	**11.50**	11.50	11.53	11.62	11.75	11.71	11.58	11.50	11.50
		1382.30	**1384.16**	1384.20	1382.15	1375.19	1367.21	1370.68	1378.88	1384.30	1384.16
RC2	100	3.25	**3.25**	3.25	3.25	3.25	3.25	3.25	3.25	3.25	3.25
		1119.44	**1119.24**	1119.31	1119.31	1121.44	1122.77	1121.59	1120.85	1119.43	1119.24
R1	200	18.20	18.20	18.20	18.20	18.20	18.20	18.20	18.20	**18.20**	18.20
		3619.53	3620.66	3618.94	3621.25	3659.90	3689.07	3670.46	3661.07	**3614.06**	3611.93
R2	200	4.00	**4.00**	4.00	4.00	4.00	4.00	4.00	4.00	4.00	4.00
		2930.85	**2929.58**	2932.04	2931.56	2935.28	2933.79	2936.33	2936.62	2930.63	2929.41
C1	200	18.90	18.90	18.90	18.90	18.90	18.90	18.90	18.90	**18.90**	18.90
		2720.34	2718.56	2720.95	2720.46	2724.82	2723.19	2727.34	2725.90	**2718.44**	2718.41
C2	200	6.00	**6.00**	6.00	6.00	6.00	6.00	6.00	6.00	6.00	6.00
		1831.73	**1831.65**	1831.73	1831.72	1832.78	1833.35	1833.03	1832.71	1831.73	1831.59
RC1	200	18.00	18.00	18.00	18.00	18.00	18.00	18.00	18.00	**18.00**	18.00
		3190.75	3193.17	3193.74	3196.37	3270.74	3283.79	3283.78	3267.50	**3181.27**	3178.41
RC2	200	4.30	4.30	4.30	4.30	4.30	4.30	4.30	4.30	**4.30**	4.30
		2538.38	2552.66	2537.86	2538.38	2544.63	2550.90	2544.45	2543.28	**2536.46**	2536.12
R1	400	36.40	36.40	36.40	36.40	36.46	36.50	36.45	36.41	**36.40**	36.40
		8429.15	8436.28	8432.65	8430.84	8692.86	8720.75	8744.92	8672.16	**8420.11**	8388.34
R2	400	8.00	8.00	8.00	8.00	8.00	8.00	8.00	8.00	**8.00**	8.00
		6168.58	6172.60	6172.05	6171.18	6211.57	6224.50	6219.46	6195.97	**6156.47**	6147.21
C1	400	37.60	**37.60**	37.60	37.61	37.68	37.80	37.87	37.68	37.60	37.60
		7184.34	**7175.07**	7184.66	7178.03	7333.86	7313.20	7280.62	7330.44	7185.88	7167.67
C2	400	11.70	11.70	11.70	11.69	11.75	11.80	11.75	11.73	**11.70**	11.60
		3913.91	3908.48	3910.66	3913.93	3947.47	3919.29	3944.51	3950.87	**3906.34**	3951.62
RC1	400	**36.00**	36.00	36.00	36.00	36.02	36.00	36.04	36.00	36.00	36.00
		7941.83	7945.38	7947.67	7948.13	8230.12	8276.59	8261.11	8202.62	7948.69	7884.94
RC2	400	8.50	8.50	8.52	8.53	8.56	8.70	8.62	8.50	**8.48**	8.50
		5248.73	5254.70	5235.23	5224.66	5298.46	5387.09	5277.38	5282.83	**5237.31**	5206.18
CNV		2481.3	2481.0	2481.2	2481.6	2485.1	2489.0	2488.2	2482.9	**2480.8**	2480.0
CTD		614363	614609	614396	614253	624003	625704	624404	623252	**613871**	612705
V(%)		0.06%	0.05%	0.06%	0.08%	0.21%	0.39%	0.31%	0.12%	**0.04%**	0.00%
D(%)		0.14%	0.19%	0.15%	0.13%	1.00%	1.17%	1.04%	0.96%	**0.08%**	0.00%
T(min)		43.73	924.18	19.99	28.04	31.29	569.67	13.91	21.51	15.45	

The results of Table 5, with the distance-minimization objective, emphasize the notable contribution of infeasible solutions in HGA and MS-ILS. For HGA, the gap is reduced from +0.77% to less than +0.23%, while for MS-ILS it goes down from +2.42% to less than +0.85%. This is a significant reduction of a factor three for only a minor difference in CPU time. We conducted a group of two-tailed paired-samples t-tests on the average results, expressed as gaps, to investigate for each relaxation X the hypothesis that *Relaxation X leads to results which are* *significantly different than NoInf.* For both HGA and MS-ILS, a p-value $p < 10^{-20}$ is retrieved, thus confirming the contribution of infeasible solutions with high confidence. It should also be noted that the proposed methods yield the best results for the distance minimization objective in the literature.

The results of Table 6 shows that HGA yields solutions of good quality, with a total number of vehicles between 2481.0 and 2481.6, very close to Nagata et al. (2010) with 2480.8. From previous papers, NB10 has been shown to be slightly better than Vidal et al. (2013a) for the smaller instance sets, while the inverse tendency is observable on the largest instances. Now, to assess on the difference between relaxation schemes for both distance minimization or fleet minimization objective, we conducted for each relaxation X among LATE, RETURN or E/L two-tailed paired-samples t-tests on the average gap to investigate the hypothesis that *the results of MS-ILS with* FLEX *relaxation are significantly different from those of MS-ILS with* X. For all X, the hypothesis is confirmed with high confidence, with a p-value $p < 0.008$. Thus, using FLEX as a relaxation with MS-ILS is the best option, since it leads to a better solution quality in similar CPU time. For HGA, the differences among different relaxation schemes, for both the distance and fleet size objectives are smaller and not statistically significant (p-values are higher than 0.2). HGA is fairly robust and performs well with any relaxation scheme. This observation goes in accordance with Vidal et al. (2012b), which demonstrate that with a single parameter setting and no particular calibration, HGA can work well on many problems with different evaluation procedures and objectives. Again, the choice of relaxation should be oriented towards the less computationally expensive one, such as RETURN or FLEX.

5.3 Other time-window constrained VRPs

To complement these experiments on the classic VRPTW, we also consider the impact of relaxation schemes on four time-window constrained vehicle routing variants with vehicle fleet mix (VFMPTW), multiple periods (PVRPTW), and backhauls (VRPBTW). We consider the VFMPTW instances of Liu and Shen (1999), the PVRPTW benchmark instances of Pirkwieser and Raidl (2009) without duration constraints and the VRPBTW instances of Ropke and Pisinger (2006). The standard objective for the first two problems is distance minimization, while the VRPBTW has been usually addressed with fleet-size minimization in priority. The best current methods in the literature, other than HGA, are included in the comparison : the Guided Local Search of Bräysy et al. (2009) (Bal09), the GA and neighborhood-based search hybrid of Nguyen et al. (2011) (NCT11), and the Adaptive Large Neighborhood Search of Ropke and Pisinger (2006) (RP06). The results of these tests are reported in Tables 7 to 9, using the same format as previously.

The same conclusions arise from these tests. For the VFMPTW (Table 7), a significant difference is observed between the methods using infeasible solutions, with a 0.06% gap for HGA and 0.18% for MS-ILS, and those which only allow feasible solutions during the search, with 0.79% and 1.18% gap respectively. This is so far the most considerable difference observed in our tests. This difference may be related to the additional effect of vehicle-type selections, which require the ability to build longer routes with tightly-scheduled customers for vehicles with higher capacity, and shorter routes for the others. Transitioning between solutions with different route types may lead to several time-window infeasibilities to be resolved.

No significant difference was observed, on the VFMPTW and PVRPTW, when comparing the individual relaxations together. For the VRPBTW still, the relaxation FLEX leads to

Table 7: HGA and MS-ILS with different relaxation schemes on the VFMPTW instances of Liu and Shen (1999)

		HGA					MS-ILS						
Inst	n	**NoInf**	**Late**	**E/L**	**Return**	**Flex**	**No Inf**	**Late**	**E/L**	**Return**	**Flex**	**Bal09**	**Best**
R1	100	1534.54	**1530.18**	1530.30	1530.21	1530.27	1536.96	1532.27	1531.37	1531.82	1531.82	1539.90	1529.40
R2	100	1146.34	**1128.90**	1129.53	1129.12	1129.00	1157.30	1131.50	1133.63	1131.97	1130.85	1149.06	1127.20
C1	100	1639.39	1615.46	1615.49	1615.45	1615.48	1652.35	1615.42	**1615.39**	1615.43	**1615.39**	1615.40	1615.39
C2	100	1186.75	**1185.19**	**1185.19**	**1185.19**	**1185.19**	1186.91	**1185.19**	**1185.19**	**1185.19**	**1185.19**	1185.70	1185.19
RC1	100	1745.04	1734.37	**1734.04**	1734.53	1734.45	1745.29	1735.89	1736.41	1736.28	1735.66	1749.66	1734.00
RC2	100	1365.04	1359.15	**1358.91**	1359.20	1359.25	1367.55	1360.68	1360.88	1361.41	1360.56	1372.82	1358.24
CTC		80153	**79549**	79553	79553	79553	80443	79627	79645	79636	79611	80122	79510
D(%)		0.79%	**0.05%**	0.06%	0.06%	0.05%	1.18%	0.15%	0.18%	0.16%	0.13%	0.82%	0.00%
T(min)		2.97	4.09	85.36	2.19	2.89	2.17	4.87	114.69	2.48	3.52	0.06	

Table 8: HGA and MS-ILS with different relaxation schemes on the PVRPTW instances of Pirkwieser and Raidl (2009)

		HGA				MS-ILS					
Inst.	n	**Late**	**E/L**	**Return**	**Flex**	**Late**	**E/L**	**Return**	**Flex**	**NCT11**	**Best**
R4	100	**3440.34**	3441.30	3440.92	3441.27	3446.07	3446.32	3446.13	3445.32	3441.86	3433.92
C4	100	2768.66	**2768.22**	2768.45	2768.69	2775.52	2778.34	2773.76	2774.56	2778.19	2765.70
RC4	100	3635.01	3638.82	3631.97	3630.61	3641.08	3640.16	3638.47	3639.89	**3628.41**	3617.46
R6	100	4446.08	**4438.04**	4450.06	4447.06	4451.57	4457.74	4450.96	4452.66	4445.81	4428.40
C6	100	3732.05	3731.66	3730.93	**3729.45**	3738.91	3739.16	3737.91	3739.65	3742.74	3722.90
RC6	100	4973.89	**4966.12**	4972.24	4974.38	4991.39	4992.20	4987.99	4989.65	4967.34	4941.22
R8	100	5455.98	5455.64	5448.45	5464.17	5466.12	5472.68	5469.53	5468.23	**5443.08**	5423.92
C8	100	**4832.26**	4849.66	4833.75	4833.49	4844.32	4839.36	4850.11	4849.81	4860.52	4803.12
RC8	100	5889.99	**5885.04**	5889.58	5886.27	5904.95	5909.72	5902.83	5903.28	5902.67	5848.12
CTD		195871	195872	**195832**	195877	196300	196378	196289	196315	196053	194924
D(%)		0.46%	0.46%	**0.44%**	0.46%	0.69%	0.72%	0.68%	0.69%	0.57%	0.00%
T(min)		8.15	155.59	5.12	6.72	6.82	130.61	4.09	5.58	97.51	

Table 9: HGA and MS-ILS with different relaxation schemes on the VRPBTW instances of Ropke and Pisinger (2006)

		HGA				MS-ILS					
Inst.	n	**Late**	**E/L**	**Return**	**Flex**	**Late**	**E/L**	**Return**	**Flex**	**RP06**	**Best**
Bhr101	100	**23.00**	**23.00**	**23.00**	**23.00**	**23.00**	**23.00**	**23.00**	**23.00**	**23.00**	23.00
		1905.83	**1905.83**	**1905.83**	**1905.83**	**1905.83**	**1905.83**	**1905.83**	**1905.83**	NC	1905.83
Bhr102	100	**21.00**	**21.00**	**21.00**	**21.00**	**21.00**	**21.00**	**21.00**	**21.00**	**21.00**	21.00
		1726.55	**1726.55**	**1726.55**	**1726.55**	**1726.55**	**1726.55**	**1726.55**	**1726.55**	NC	1726.55
Bhr103	100	15.70	**15.67**	15.70	**15.67**	15.67	**15.67**	**15.67**	15.67	15.67	15.67
		1409.21	**1410.73**	1409.21	**1410.73**	1410.79	**1410.73**	**1410.73**	1410.76	NC	1410.73
Bhr104	100	11.00	11.00	10.83	**10.73**	11.00	11.00	10.83	10.73	11.00	10.67
		1143.72	1148.68	1163.55	**1175.61**	1143.46	1144.19	1166.27	1178.07	NC	1182.73
Bhr105	100	15.67	15.67	15.67	15.67	15.67	**15.67**	15.67	15.67	15.97	15.67
		1618.51	1623.10	1618.04	1618.15	1617.99	**1617.88**	1618.13	1618.42	NC	1617.71
CNV		259.1	259.0	258.6	**258.2**	259.0	259.0	258.5	258.2	259.9	258
CTD		23411	23445	23470	**23511**	23414	23416	23483	23519	NC	23531
V(%)		0.71%	0.67%	0.38%	**0.13%**	0.67%	0.67%	0.33%	0.13%	1.05%	0.00%
D(%)		-0.66%	-0.49%	-0.34%	**-0.11%**	-0.65%	-0.64%	-0.27%	-0.07%	NC	0.00%
T(min)		4.91	57.40	2.79	3.97	6.91	62.86	3.75	5.40	1.90	

solutions of significantly higher quality in similar CPU time. Finally, as a consequence of these tests some new best known solutions in the literature were generated. These new solutions are listed in Table 10 for further research.

Table 10: New Best Known Solutions

VRPTW – dist		VRPTW – fleet size			VFMPTW – dist	
R104	976.61	R1-400-7	36	7640.96	R104	1354.29
R106	1239.37	R1-400-8	36	7273.07	R110	1443.33
R107	1072.12	R1-400-9	36	8741.92	R201	1427.39
R108	938.20	R1-400-10	36	8102.78	R210	1149.31
R112	953.63	C1-400-9	36	7043.10	RC101	2040.61
R201	1147.80	RC1-400-2	36	7898.37		
R202	1034.35	RC1-400-4	36	7309.64	**PVRPTW – dist**	
R204	735.80	RC1-400-7	36	7948.51		
R205	955.82	RC1-400-9	36	7746.03	T6-RC1-1	5778.7
R207	797.99	RC1-400-10	36	7601.90	T6-RC1-3	4267.0
R208	705.33	RC2-400-8	8	4792.69		
R210	904.78					
R211	753.15	**VRPBTW – fleet size**				
RC101	1623.59					
RC102	1461.23	BHR104C	11	1188.78		
RC103	1261.67					
RC104	1135.48					
RC105	1518.58					
RC106	1376.26					
RC107	1211.11					
RC201	1265.56					
RC202	1095.65					
RC203	926.82					
RC204	786.38					
RC205	1157.55					
RC207	966.08					
RC208	778.93					

6 Conclusions

We introduced a comprehensive assessment of the impact of penalized infeasible solutions during heuristic search for VRPTW. Three different heuristic procedures have been considered (a multi-start local-improvement procedure, an iterated local search, and a hybrid genetic algorithm) in the presence of four different relaxation schemes (penalized late arrivals, both early and late arrivals, returns in time, or flexible speed). Our experimental results demonstrate that, for all three heuristics, any considered relaxation scheme leads to solutions of significantly higher quality. Observed differences can be considerable in some cases, the gap to the best known solutions, being decreased by a factor 3 for the VRPTW, and a factor up to 8 for the VFMPTW. Moreover, our experiments on simple local searches show that relaxations tend to mitigate the impact of low-quality starting solutions.

The individual differences between the four relaxation schemes may not be observed on all problems, heuristic frameworks, and benchmark instances. We noticed a significant improvement of solution quality when using the proposed "flexible speed" relaxation for the VRPTW and VRPBTW. In light of computational complexity results and CPU time measures, the returns in time and flexible speed relaxations are the fastest and most scalable methods. These two latter relaxations are a promising choice for further heuristics.

Acknowledgments

Partial funding for this project has been provided by the Champagne-Ardennes regional council, France, the Natural Sciences and Engineering Council of Canada (NSERC), and by the Fonds québécois de la recherche sur la nature et les technologies (FQRNT). This support is gratefully acknowledged.

References

Alvarenga, G.B., G.R. Mateus, G. De Tomi. 2007. A genetic and set partitioning two-phase approach for the vehicle routing problem with time windows. *Computers & Operations Research* **34**(6) 1561–1584.

Bent, Russell, Pascal Van Hentenryck. 2004. A Two-Stage Hybrid Local Search for the Vehicle Routing Problem with Time Windows. *Transportation Science* **38**(4) 515–530.

Berger, J., M. Barkaoui, O. Bräysy. 2003. A route-directed hybrid genetic approach for the vehicle routing problem with time windows. *INFOR* **41**(2) 179–194.

Bräysy, O. 2002. Fast local searches for the vehicle routing problem with time windows. *INFOR* **40**(4) 319–330.

Bräysy, O. 2003. A Reactive Variable Neighborhood Search for the Vehicle-Routing Problem with Time Windows. *INFORMS Journal on Computing* **15**(4) 347–368.

Bräysy, O., M. Gendreau. 2005a. Vehicle routing problem with time windows, Part I : Route construction and local search algorithms. *Transportation Science* **39**(1) 104–118.

Bräysy, O., M. Gendreau. 2005b. Vehicle routing problem with time windows, Part II : Metaheuristics. *Transportation Science* **39**(1) 119–139.

Bräysy, O., G. Hasle, W. Dullaert. 2004. A multi-start local search algorithm for the vehicle routing problem with time windows. *European Journal of Operational Research* **159**(3) 586–605.

Bräysy, O., P.P. Porkka, W. Dullaert, P.P. Repoussis, C.D. Tarantilis. 2009. A well-scalable metaheuristic for the fleet size and mix vehicle routing problem with time windows. *Expert Systems with Applications* **36**(4) 8460–8475.

Cordeau, J.-F., G. Laporte, A. Mercier. 2001. A unified tabu search heuristic for vehicle routing problems with time windows. *Journal of the Operational Research Society* **52**(8) 928–936.

Dumas, Y., F. Soumis, J. Desrosiers. 1990. Optimizing the Schedule for a Fixed Vehicle Path with Convex Inconvenience Costs. *Transportation Science* **24**(2) 145–152.

Fu, Z., R. Eglese, L.Y.O. Li. 2007. A unified tabu search algorithm for vehicle routing problems with soft time windows. *Journal of the Operational Research Society* **59**(5) 663–673.

Gambardella, L.M., E. Taillard, G. Agazzi. 1999. MACS-VRPTW: A multiple ant colony system for vehicle routing problems with time windows. F. Glover D. Corne, M. Dorigo, ed., *New Ideas in Optimization*. McGraw-Hill, 63–76.

Garey, M.R., R.E. Tarjan, G.T. Wilfong. 1988. One-processor scheduling with symmetric earliness and tardiness penalties. *Mathematics of Operations Research* **13**(2) 330–348.

Gehring, H., J. Homberger. 1999. A parallel hybrid evolutionary metaheuristic for the vehicle routing problem with time windows. *Proceedings of EURO-GEN'99*. 57–64.

Gehring, H., J. Homberger. 2002. Parallelization of a two-phase metaheuristic for routing problems with time windows. *Journal of heuristics* **8**(2) 251–276.

Gendreau, M., G. Laporte, R. Seguin. 1996. A Tabu Search Heuristic for the Vehicle Routing Problem with Stochastic Demands and Customers. *Operations Research* **44**(3) 469–477.

Gendreau, M., C.D. Tarantilis. 2010. Solving large-scale vehicle routing problems with time windows: The state-of-the-art. Tech. rep., CIRRELT.

Glover, F., J.-K. Hao. 2011. The case for strategic oscillation. *Annals of Operations Research* **183**(1) 163–173.

Hashimoto, H., M. Yagiura. 2008. A Path Relinking Approach with an Adaptive Mechanism to Control Parameters for the Vehicle Routing Problem with Time Windows. J. Hemert, C. Cotta, eds., *Evolutionary Computation in Combinatorial Optimization*, *LNCS*, vol. 4972. Springer Berlin Heidelberg, 254–265.

Hashimoto, H., M. Yagiura, T. Ibaraki. 2008. An iterated local search algorithm for the time-dependent vehicle routing problem with time windows. *Discrete Optimization* **5**(2) 434–456.

Hendel, Y., F. Sourd. 2006. Efficient neighborhood search for the one-machine earliness-tardiness scheduling problem. *European Journal of Operational Research* **173**(1) 108–119.

Homberger, J., H. Gehring. 1999. Two evolutionary metaheuristics for the vehicle routing problem with time windows. *INFOR* **37**(3) 297–318.

Homberger, J., H. Gehring. 2005. A two-phase hybrid metaheuristic for the vehicle routing problem with time windows. *European Journal Of Operational Research* **162**(1) 220–238.

Ibaraki, T., S. Imahori, M. Kubo, T. Masuda, T. Uno, M. Yagiura. 2005. Effective Local Search Algorithms for Routing and Scheduling Problems with General Time-Window Constraints. *Transportation Science* **39**(2) 206–232.

Ibaraki, T., S. Imahori, K. Nonobe, K. Sobue, T. Uno, M. Yagiura. 2008. An iterated local search algorithm for the vehicle routing problem with convex time penalty functions. *Discrete Applied Mathematics* **156**(11) 2050–2069.

Irnich, S. 2008. A Unified Modeling and Solution Framework for Vehicle Routing and Local Search-Based Metaheuristics. *INFORMS Journal on Computing* **20**(2) 270–287.

Kindervater, G.A.P., M.W.P. Savelsbergh. 1997. Vehicle routing: Handling edge exchanges. E.H.L. Aarts, J.K. Lenstra, eds., *Local Search in Combinatorial Optimization*. Princeton Univ Pr, 337–360.

Kritzinger, S., F. Tricoire, K.F. Doerner, R.F. Hartl, T. Stützle. 2012. A Unified Framework for Routing Problems with Fixed Fleet Size. Tech. rep., Johannes Kepler University, Linz, Austria.

Labadi, N., C. Prins, M. Reghioui. 2008. A memetic algorithm for the vehicle routing problem with time windows. *RAIRO Operations Research* **42**(3) 415–431.

Le Bouthillier, A., T.G. Crainic. 2005a. A cooperative parallel meta-heuristic for the vehicle routing problem with time windows. *Computers & Operations Research* **32**(7) 1685–1708.

Le Bouthillier, A., T.G. Crainic. 2005b. A guided cooperative search for the vehicle routing problem with time windows. *Intelligent Systems, IEEE* **20**(4) 36–42.

Lim, A., X. Zhang. 2007. A two-stage heuristic with ejection pools and generalized ejection chains for the vehicle routing problem with time windows. *INFORMS Journal on Computing* **19**(3) 443–57.

Liu, F.-H., S.-Y. Shen. 1999. The fleet size and mix vehicle routing problem with time windows. *Journal of the Operational Research Society* **50**(7) 721–732.

Mester, D., O. Bräysy. 2005. Active guided evolution strategies for large-scale vehicle routing problems with time windows. *Computers & Operations Research* **32**(6) 1593–1614.

Muter, I., S. I. Birbil, G. Sahin. 2010. Combination of Metaheuristic and Exact Algorithms for Solving Set Covering-Type Optimization Problems. *INFORMS Journal on Computing* **22**(4) 603–619.

Nagata, Y., O. Bräysy. 2009. Edge Assembly-Based Memetic Algorithm for the Capacitated Vehicle Routing Problem. *Networks* **54**(4) 205–215.

Nagata, Y., O. Bräysy, W. Dullaert. 2010. A penalty-based edge assembly memetic algorithm for the vehicle routing problem with time windows. *Computers & Operations Research* **37**(4) 724–737.

Nguyen, P.K., T.G. Crainic, M. Toulouse. 2011. A Hybrid Genetic Algorithm for the Periodic Vehicle Routing Problem with Time Windows. Tech. rep., CIRRELT.

Pirkwieser, S., G.R. Raidl. 2009. Multiple Variable Neighborhood Search Enriched with ILP Techniques for the Periodic Vehicle Routing Problem with Time Windows. M. Blesa, C. Blum, L. Di Gaspero, A. Roli, M. Sampels, A. Schaerf, eds., *Hybrid Metaheuristics*, *LNCS*, vol. 5818. Springer Berlin / Heidelberg, 45–59.

Pisinger, D., S. Ropke. 2007. A general heuristic for vehicle routing problems. *Computers & Operations Research* **34**(8) 2403–2435.

Prescott-Gagnon, E., G. Desaulniers, L.M. Rousseau. 2009. A branch-and-price-based large neighborhood search algorithm for the vehicle routing problem with time windows. *Networks* **54**(4) 190–204.

Prins, C. 2004. A Simple and Effective Evolutionary Algorithm for the Vehicle Routing Problem. *Computers & Operations Research* **31**(12) 1985–2002.

Prins, C., N. Labadi, M. Reghioui. 2009. Tour splitting algorithms for vehicle routing problems. *International Journal of Production Research* **47**(2) 507–535.

Repoussis, P.P., C.D. Tarantilis, G. Ioannou. 2009. Arc-Guided Evolutionary Algorithm for the Vehicle Routing Problem With Time Windows. *IEEE Transactions on Evolutionary Computation* **13**(3) 624–647.

Robertson, T., F.T. Wright, R.L. Dykstra. 1988. *Order Restricted Statistical Inference*. Wiley Series in Probability and Statistics, John Wiley & Sons, New York.

Ropke, S., D. Pisinger. 2006. A unified heuristic for a large class of Vehicle Routing Problems with Backhauls. *European Journal of Operational Research* **171**(3) 750–775.

Savelsbergh, M.W.P. 1985. Local Search in Routing Problems with Time Windows. *Annals of Operations Research* **4**(1) 285–305.

Savelsbergh, M.W.P. 1992. The Vehicle Routing Problem with Time Windows: Minimizing Route Duration. *ORSA Journal on Computing* **4**(2) 146–154.

Solomon, M.M. 1987. Algorithms for the vehicle routing and scheduling problems with time window constraints. *Operations Research* **35**(2) 254–265.

Taillard, E., P. Badeau, M. Gendreau, F. Guertin, J.-Y. Potvin. 1997. A tabu search heuristic for the vehicle routing problem with soft time windows. *Transportation Science* **31**(2) 170–186.

Vidal, T., T.G. Crainic, M. Gendreau, N. Lahrichi, W. Rei. 2012a. A Hybrid Genetic Algorithm for Multidepot and Periodic Vehicle Routing Problems. *Operations Research* **60**(3) 611–624.

Vidal, T., T.G. Crainic, M. Gendreau, C. Prins. 2011. A Unifying View on Timing Problems and Algorithms. Tech. rep., CIRRELT.

Vidal, T., T.G. Crainic, M. Gendreau, C. Prins. 2012b. A Unified Solution Framework for Multi-Attribute Vehicle Routing Problems. Tech. rep., CIRRELT.

Vidal, T., T.G. Crainic, M. Gendreau, C. Prins. 2013a. A hybrid genetic algorithm with adaptive diversity management for a large class of vehicle routing problems with time-windows. *Computers & Operations Research* **40**(1) 475–489.

Vidal, T., T.G. Crainic, M. Gendreau, C. Prins. 2013b. Heuristics for Multi-Attribute Vehicle Routing Problems: A Survey and Synthesis. *European Journal of Operational Research* **231**(1) 1–21.

Yano, C.A., Y.-D. Kim. 1991. Algorithms for a class of single-machine weighted tardiness and earliness problems. *European Journal of Operational Research* **52**(2) 167–178.