AN EMPIRICAL ANALYSIS OF TIME-WINDOW RELAXATIONS IN VEHICLE ROUTING HEURISTICS

Thibaut Vidal
ICD-LOSI - Université de Technologie de Troyes, France
12, rue Marie Curie, BP 2060, 10010 Troyes cedex, France
& CIRRELT - Université de Montréal
Thibaut.Vidal@cirrelt.ca

Teodor Gabriel Crainic
CIRRELT - École des sciences de la gestion, UQAM, Montréal, Canada
TeodorGabriel.Crainic@cirrelt.ca

Michel Gendreau
CIRRELT - École Polytechnique, Montréal, Canada
Michel.Gendreau@cirrelt.ca

Christian Prins
ICD-LOSI - Université de Technologie de Troyes, France
Christian.Prins@utt.fr

ABSTRACT

The contribution of infeasible solutions in heuristic searches for vehicle routing problems does not make, up to date, a consensus in the metaheuristics community. In one hand, infeasible solutions may allow to transition between structurally different feasible solutions, thus enhancing the search. In the other hand they also lead to more complex move evaluation procedures and in wider search spaces. Choices of constraint relaxations may this dramatically impact the method performance, both in terms of speed and solution quality. This paper analyzes the impact of infeasible solutions on heuristic searches through various empirical studies on local search improvement procedures for the vehicle routing problem with time windows (VRPTW). As a result of these analyses, relaxations appear clearly to have a positive impact on the solution quality. Especially, the time-window relaxation scheme of Nagata et al. (2010) clearly contributes to increase the performance of neighborhood searches for this problem.

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

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 being 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) have been proposed to address real-life settings.

Most efficient metaheuristics rely on some sort of local search-based improvement procedure, dedicating 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, may fail on tightly-constrained problem instances.

The use of intermediate infeasible solutions with relaxed time-window constraints appears in this context as an alternative to guarantee the availability of some initial solutions. Several relaxation schemes have been proposed through the years, 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). It is frequently conjectured that infeasible solutions enable to better transition during the search between structurally different feasible solutions (Cordeau et al. 2001, Vidal et al. 2011a). In particular, a clever management of penalties may enable to focus the search towards borders of feasibility, a place where high quality solutions are more susceptible to appear (Glover 2011). However, relaxations can also lead to more complex move evaluations (Ibaraki et al. 2005, Vidal et al. 2011b) and larger search spaces. Hence, it becomes necessary to determine whether 1) the use of infeasible solutions contributes significantly to the search 2) if one relaxation scheme is more suitable to progress towards good feasible solutions, and 3) to quantify the additional computational burden related to infeasibility evaluations on practical problems.

This paper contributes towards answering these questions by means of dedicated experimental analyses. Four local-search improvement procedures, differing only by their relaxation scheme, are compared on the well-known benchmark instances of Solomon (1987). Sensitivity analyses are conducted to assess on the computational effort related to relaxations, on the contributions of relaxations in the search performance, relatively to different objectives and initial solution procedures. As a result of these analyses, the "return in time" relaxation of Nagata et al. (2010) appears to contribute to reach higher quality solutions, mitigates the dependency upon a good initial solution, with only a minor impact on the algorithm speed.

The paper is organized as follows. A review of relaxation schemes for the VRPTW is conducted in Section 3, followed by a presentation of state-of-the-art neighborhood evaluation procedures for each relaxation in Section 4. Empirical analysis on the impact of relaxations are performed in Section 5, and finally Section 6 concludes.

2 Problem statement and notations

The vehicle routing problem with time windows (VRPTW) can be defined on a complete undirected graph $G = (\mathcal{V}, \mathcal{E})$, where $v_0 \in \mathcal{V}$ stands for a central depot, and vertices $\mathcal{V}^{\text{CST}} = \mathcal{V} \backslash \{v_0\}$ represent n geographically dispersed customers requiring service. Each edge $(i, j) \in \mathcal{E}$ represents a traveling possibility from vertex v_i to vertex v_j with distance d_{ij} (equal to the travel time w.l.o.g.). Each customer $v_i \in \mathcal{V}^{\text{CST}}$ is characterized by a non-negative demand q_i, a service time 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, where the product to be delivered to customers is kept. The capacity of each vehicle is limited to Q units of product. The VRPTW aims to design at most m sequences of visits σ^k for

$k \in \{1 \dots m\}$, starting from the depot $\sigma^k(1) = 0$, visiting customers $\sigma^k(2), \dots, \sigma^k(|\sigma^k| - 1)$ during their time-windows, and returning to the depot $\sigma^k(|\sigma^k|) = 0$. The traditional objective involves minimizing the number of routes in priority, and then distance.

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

$$\text{Subject to:} \sum_{v_j \in \mathcal{V}} \sum_{k=1}^{m} x_{ijk} = 1 \qquad v_i \in \mathcal{V}^{\text{CST}} \tag{2}$$

$$\sum_{v_j \in \mathcal{V} \setminus \{v_{n+1}\}} x_{jik} - \sum_{v_j \in \mathcal{V} \setminus \{v_0\}} x_{ijk} = 0 \qquad v_i \in \mathcal{V}^{\text{CST}} \,;\, k \in \{1, \dots, m\} \tag{3}$$

$$\sum_{v_j \in \mathcal{V} \setminus \{v_0\}} x_{0jk} = 1 \qquad k \in \{1, \dots, m\} \tag{4}$$

$$\sum_{v_j \in \mathcal{V} \setminus \{v_{n+1}\}} x_{j,n+1,k} = 1 \qquad k \in \{1, \dots, m\} \tag{5}$$

$$\sum_{v_i \in \mathcal{V}} \sum_{v_j \in \mathcal{V}} q_i x_{ijk} \leq Q \qquad k \in \{1, \dots, m\} \tag{6}$$

$$x_{ijk}(t_{ik} + d_{ij} + \tau_i - t_{jk}) \leq 0 \qquad v_i \in \mathcal{V} \,;\, v_j \in \mathcal{V} \,;\, k \in \{1, \dots, m\} \tag{7}$$

$$e_i \leq t_{ik} \leq l_i \qquad v_i \in \mathcal{V} \,;\, k \in \{1, \dots, m\} \tag{8}$$

$$x_{ijk} \in \{0, 1\} \qquad v_i \in \mathcal{V} \,;\, v_j \in \mathcal{V} \,;\, k \in \{1, \dots, m\} \tag{9}$$

$$t_{ik} \in \Re^+ \qquad v_i \in \mathcal{V} \,;\, k \in \{1, \dots, m\} \tag{10}$$

Equations (1-10) recall the mathematical formulation of the VRPTW as a multi-commodity network flow. For convenience, v_0 has been separated into two vertices v_0 and v_{n+1}, standing respectively for the depot at the origin and destination. The binary 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} provide the service date to customer v_i, when serviced by vehicle k. Equations (2-6) set up the basis VRP network structure, and Equations (7-8) formulate the choices of service times t_{ik}. Equation (7) also eliminates sub-tours and can be linearized by means of big M values.

The next Section reviews the use of these relaxations in the literature, and analyzes efficient methods to evaluate penalties in each case.

3 Time-window relaxations in VRPTW heuristics

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

Table 1 provides a review of recent state-of-the-art methods and their time-window relaxations. 13/24 of these state-of-the-art methods rely on time-window infeasible solutions. Three main relaxations 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” settings in the literature, and correspond to a Lagrangean 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 can be viewed as a Lagrangean relaxation of Equation (7). Up to this date, the hybrid genetic algorithms of Nagata et al. (2010) and Vidal et al. (2011a), which rely on this latter uncommon relaxation, have produced

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
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
Prescott-Gagnon et al. (2009)	Branch-and-price based Large Neighborhood Search	NO
Repoussis et al. (2009)	Evolutionary Algorithm & Local Search	Late service
Nagata et al. (2010)	Hybrid GA with edge assembly crossover	Return in time
Vidal et al. (2011a)	Hybrid GA with cost/diversity objective	Return in time

the best overall solutions. The contribution of this particular time-window relaxation in resolution approaches is an open question which is investigated in this paper.

4 Move evaluation methods and their computational complexity

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 practice, move evaluations make for the largest part of the overall computation effort in recent heuristics, and therefore must be very efficient.

It is well-known in the literature (Kindervater and Savelsbergh 1997b, Irnich 2008, Vidal et al. 2011a,b) 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 subsequences of consecutive visits (and reverse subsequences for 2-opt) from the incumbent solution. Since local-search neighborhoods generally involve recurrent subsequences of visits, the management of meaningful information on subsequences can save redundant computations and increase the local search performance.

Keeping track of partial demands and distances, for example, on each $O(n^2)$ subsequence from the incumbent solution, provides the means to evaluate the demand and distance of any recombined route with a bounded number of sums. This opens the way to $O(1)$ time load-feasibility checks and distance computations during the local search, compared to $O(n)$ for a straightforward method that browses the routes and cumulates loads and distances.

In the literature, the sequence information is either pre-processed before 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). In addition, pre-processing can be limited to the $O(L_{max}n)$ subsequences that contain either the first node of the route, the last node, or of less than L_{max} nodes, while still allowing for efficient inter-route RELOCATE, SWAP, 2-OPT*, and CROSS exchanges. We opted in this paper to rely on pre-processing, since the computational effort required to manage the information on subsequences is generally negligible when compared to the effort required by move evaluations, and because it allows for a random exploration of local search moves which increases the solution variety.

In a similar manner, preprocessing meaningful information on subsequences 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. Still, as reviewed in the following, not all relaxation schemes allow for efficient move evaluation techniques, and some relaxations result in non-trivial sub-problems for determining service times (called timing problems in Vidal et al. 2011b) on each new route.

4.1 No infeasible solution

When no infeasible solution is used, checking route feasibility within a local search can be efficiently done with the approach of Savelsbergh (1985, 1992) and Kindervater and Savelsbergh (1997a). For any subsequence σ, the sum of travel and service times $T(\sigma)$, the earliest possible sequence completion time $E(\sigma)$, and the latest feasible starting date $L(\sigma)$ are pre-processed. These values can be obtained by induction on the concatenation operation, starting with the base case of a sequence $\sigma_0 = (v_i)$ containing a single visit where $T(\sigma_0) = \tau_i$, $E(\sigma_0) = e_i + \tau_i$ and $L(\sigma_0) = l_i$, and using Equations (11-14) to derive the same information on larger subsequences.

$$T(\sigma_1 \oplus \sigma_2) = T(\sigma_1) + d_{\sigma_1(|\sigma_1|)\sigma_2(1)} + T(\sigma_2) \tag{11}$$
$$E(\sigma_1 \oplus \sigma_2) = \max\{E(\sigma_1) + d_{\sigma_1(|\sigma_1|)\sigma_2(1)} + T(\sigma_2), E(\sigma_2)\} \tag{12}$$
$$L(\sigma_1 \oplus \sigma_2) = \min\{L(\sigma_1), L(\sigma_2) - d_{\sigma_1(|\sigma_1|)\sigma_2(1)} - T(\sigma_1)\} \tag{13}$$
$$isFeas(\sigma_1 \oplus \sigma_2) \equiv isFeas(\sigma_1) \wedge isFeas(\sigma_2) \wedge (E(\sigma_1) + d_{\sigma_1(|\sigma_1|)\sigma_2(1)} \leq L(\sigma_2)) \tag{14}$$

These equations, furthermore, enable to check in $O(1)$ time the feasibility of routes issued from moves, assimilated to a recombination of a bounded number of sequences.

4.2 Early/Late and Late service

Penalized late or early services have been mentioned in early works by Sexton and Choi (1986) in the context of a Pickup and Delivery Problem with Time-Windows. Such a relaxation represents a trade-off between service quality and routing costs which is frequently encountered in practice. It is also eventually used within heuristics to achieve better performance.

However, allowing both early and late deliveries leads to an additional decision problem, since upon an early arrival to a 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 $\{R,D|\phi\}$ timing problem (Vidal et al. 2011b) formulated in Equations (15-16). Coefficients α and β represent penalties for early and late service.

$$\min_{t_1,\dots,t_{|\sigma|}} \quad \sum_{i=1}^{n_k} \alpha(e_{\sigma(i)} - t_i)^+ + \sum_{i=1}^{n_k} \beta(t_i - l_{\sigma(i)})^+ \tag{15}$$
$$\text{s.t.} \quad t_i + \tau_{\sigma(i)} + d_{\sigma(i)\sigma(i+1)} \leq t_{i+1} \qquad 1 \leq i < |\sigma| \tag{16}$$

The previous model is encountered in various fields of operations research, in transportation logistics, project and machine scheduling literature, 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).

The special case where only late services are allowed ($\alpha = +\infty$) is straightforward to solve in $O(n)$ with a minimum idle time policy by servicing each customer as early as possible. The related route evaluation procedure will be denoted here as "straightforward evaluation".

In addition, Hendel and Sourd (2006) and Ibaraki et al. (2005, 2008) proposed a more efficient move evaluation procedure, generally applicable to any vehicle routing problem with separable piecewise linear service costs $c_i(t_i)$ as a function of time. This "advanced evaluation" requires managing two types of information on subsequences during the search: a function $F(\sigma)(t)$ representing the minimum cost to service the sequence 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 represented explicitly in the method using appropriate data structures.

For a sequence $\sigma_0 = (v_i)$ 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 sequences using Equations (17-18).

$$F(\sigma \oplus v_i)(t) = \min_{0 \leq x \leq t} \{c_i(x) + F(\sigma)(x - \tau_{\sigma(|\sigma|)} - d_{\sigma(|\sigma|),i})\} \tag{17}$$

$$B(v_i \oplus \sigma)(t) = \min_{x \geq t} \{c_i(x) + B(\sigma)(x + \tau_i + t_{i,\sigma(1)})\} \tag{18}$$

Equation (19) then provides the optimal service cost $Z^*(\sigma_1 \oplus \sigma_2)$ for a route issued of the concatenation of two subsequences σ_1 and σ_2.

$$Z^*(\sigma_1 \oplus \sigma_2) = \min_{x \geq 0} \{F(\sigma_1)(x) + B(\sigma_2)(x + \tau_{\sigma_1(|\sigma_1|)} + t_{\sigma_1(|\sigma_1|)\sigma_2(1)})\} \tag{19}$$

This equation allows, among other, for efficient evaluations of 2-OPT* neighborhoods. Moreover, any route resulting from the concatenation of three subsequences $(\sigma_1 \oplus \sigma_L \oplus \sigma_2)$, where σ_L is a sequence of bounded size, can be evaluated by relying on Equation (17) $|\sigma_L|$ successive times to yield the information on $\sigma' = \sigma_1 \oplus \sigma_L$, and computing $Z^*(\sigma' \oplus \sigma_2)$ with Equation (19). This latter strategy can be used to account for inter-route moves such as RELOCATE, SWAP or CROSS.

Route evaluations based on this methodology achieve a time complexity of $O(\Sigma_i \xi(c_i))$, where $\xi(c_i)$ represents the number of pieces in each function $c_i(t_i)$. The methodology applies for the wide majority of intra-route VRP neighborhoods. In the particular case where all functions $c_i(t_i)$ are convex, which is the case in soft time windows settings, more advanced implementations based either on heap data structures (Hendel and Sourd 2006), or on search trees (Ibaraki et al. 2008), achieve a route evaluation complexity of $O(\log \Sigma_i \xi(c_i))$. In soft time-windows settings, $\xi(c_i) = 3$ for any customer v_i, and thus moves can be evaluated in amortized $O(\log n)$. Hence, when considering such advanced neighborhood evaluation procedures, the two relaxations schemes *Early/Late* and *Late Service* lead to the same move evaluation complexity despite the additional decisions related to allowable earliness.

4.3 Returns in time

The relaxation proposed by Nagata et al. (2010) is based on linearly penalized "time warps" to reach time windows upon a late arrival. As shown in the following, despite its very limited practical significance, the relaxation proves to be particularly useful to allow intermediate infeasible solution in heuristics while still allowing for amortized constant time move evaluations.

A possible way to efficiently perform move evaluations (Vidal et al. 2011a) requires computing on any subsequence 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 minimum time-warp use. For a sequence $\sigma_0 = (v_i)$ 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 then be computed on larger subsequences by induction on the concatenation operator using Equations (20-26).

$$D(\sigma_1 \oplus \sigma_2) = D(\sigma_1) + D(\sigma_2) + d_{\sigma_1(|\sigma_1|)\sigma_2(1)} + \Delta_{WT} \tag{20}$$
$$TW(\sigma_1 \oplus \sigma_2) = TW(\sigma_1) + TW(\sigma_2) + \Delta_{TW} \tag{21}$$
$$E(\sigma_1 \oplus \sigma_2) = \max\{E(\sigma_2) - \Delta, E(\sigma_1)\} - \Delta_{WT} \tag{22}$$
$$L(\sigma_1 \oplus \sigma_2) = \min\{L(\sigma_2) - \Delta, L(\sigma_1)\} + \Delta_{TW} \tag{23}$$
$$where\ \Delta = D(\sigma_1) - TW(\sigma_1) + d_{\sigma_1(|\sigma_1|)\sigma_2(1)} \tag{24}$$
$$\Delta_{WT} = \max\{E(\sigma_2) - \Delta - L(\sigma_1), 0\} \tag{25}$$
$$\Delta_{TW} = \max\{E(\sigma_1) + \Delta - L(\sigma_2), 0\} \tag{26}$$

The previous equations lead to amortized $O(1)$ time move evaluations. This complexity is identical to the case where no infeasible solutions are used (Section 4.1).

4.4 Flexible service and travel times

A drawback of the relaxation of Section 4.3 lies in the fact that the amount of time warp is not limited, opening the way to routes that can potentially serve some customers with early time windows, move on towards other customers later in the day, and pay for a large time warp to start again deliveries to early customers. To avoid this issue, we investigate another relaxation alternative based on flexible travel and service times, but which forbids too small and negative travel durations.

Flexible travel times are usually computationally expensive to deal with (Hashimoto et al. 2006). Yet, the relaxation we propose is a very simple case, since the penalty $p_{ij}(\delta t)$ as a function of the service and travel duration is a simple piecewise linear function given in Equation (27), which is equivalent to pay linearly for a duration gain in presence of a minimum duration constraint. 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 duration for driving from any vertex v_i to any vertex v_j is set to $d_{ij}^{min} = d_{ij}/2$.

$$p_{ij}(\delta t) = \begin{cases} +\infty & \text{if } \delta t < d_{ij}^{min} + \tau_i^{min} \\ \alpha \times (d_{ij} - \delta t) & \text{if } d_{ij}^{min} + \tau_i^{min} \le \delta t < d_{ij} + \tau_i \\ 0 & \text{if } d_{ij} + \tau_i \le \delta t \end{cases} \tag{27}$$

Route evaluations in presence of this relaxation can be managed by means of a combination of the previous methodologies. First, Equations (11-14) are used to check whether the path is time-window feasible when the maximum speed is used. If it is feasible, the Equations (20-26) are used to measure the necessary amount of *return in time* along the path, which also corresponds to the necessary speedups, otherwise the route is declared infeasible. Thus, the relaxation cost related to flexible travel times can be measured in amortized $O(1)$ time.

5 Empirical comparison of relaxations

In this Section, experimental investigations are conducted on several important questions related to relaxations schemes for the VRPTW, and especially:

1. What is the impact of relaxation schemes on heuristics performance, does one relaxation lead to solutions of higher quality ?
2. Do relaxation schemes mitigate the need for good starting solutions in heuristics ?
3. What is the practical computational burden related to different relaxations for problems of practical sizes ?

To answer to these questions, four variants of a simple local-search improvement procedure, differing only by their relaxation scheme, have been compared on the well-known benchmark instances of Solomon (1987) with 100 customers. These 56 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 longer time horizon, larger vehicle capacities, and lead to longer routes.

Experiments have been conducted on two different problem objectives: a hierarchical objective involving first the minimization of fleet size and then distance, and the distance-minimization objective. Experiments have been conducted with two different types of initial solutions to analyze their impact, 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).

Three alternative relaxations, with either late services "LATE", returns in time "RETURN", and flexible travel times "FLEX" have been tested and compared with a strategy where infeasible solutions are forbidden "NO INF". The efficient move evaluation strategies presented in Section 4 are implemented for each relaxation scheme. The *straightforward evaluation* strategy in $O(n)$ time is used for the LATE relaxation. Faster $O(\log n)$ evaluations are known to be achievable (Section 4.2), but the price to pay in terms of algorithm development is very high.

5.1 Performance of relaxations with regards to distance minimization

The local search procedure used in our experiments 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 $L_{max} = 2$ (see Vidal et al. 2012 for a detailed presentation of these neighborhoods). Moves are explored in random order, any improving move being directly applied, until no improvement can be found in the whole neighborhood.

This procedure is applied several times from different initial solutions, the best overall final solution being returned. Each run involves two phases as described in Algorithm 1. First, an initial solution is created with either the random construction procedure, or the I1 insertion heuristic of Solomon (1987), and the local search is applied with moderately small penalty coefficients ($\alpha = \beta = 1$). This process can lead to an infeasible solution, such that a second local search run with higher penalty coefficients is performed to restore the solution feasibility ($\alpha = \beta = 100$).

Algorithm 1 Duration Minimization

```
for t = 1,...,nbTry do
    penalties = 1
    sol = LocalSearch(newInitialSol())
    if not isFeasible(sol) then
        penalties = 100
        sol = LocalSearch(sol)
returnBestFeasibleSolution()
```

Table 2 reports the solutions retrieved by each method during a sample run, 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. Distances are aggregated by problem classes. The last two lines indicate the Cumulated Total Distance (CTD) for all the 56 instances and the average computation time per instance.

Table 2 demonstrates the significant contribution of time-window relaxations to the solution quality. Relaxations lead to an improvement in distance of −1.30% to −1.65% when the constructive initialization procedure is used, otherwise it reaches −4.65% to −4.76% when random

Table 2: Distance minimization on the VRPTW benchmark instances of Solomon (1987)

Inst.	SolI1	No Inf	Late	Return	Flex	No Inf	Late	Return	Flex
		SolomonI1 Initial Solution				Random Initial Solution			
R1	1431.97	1225.25	1219.11	1220.13	**1219.41**	1268.97	1231.70	1226.08	1229.73
R2	1326.64	963.53	957.11	947.77	942.87	982.44	**940.38**	947.41	940.79
C1	936.48	844.00	835.17	835.67	**834.26**	860.82	835.14	840.73	835.32
C2	696.57	605.62	603.62	603.08	**600.59**	709.37	650.52	645.45	649.61
RC1	1578.28	1401.49	1399.11	**1389.83**	1396.54	1482.79	1406.57	1401.53	1404.57
RC2	1653.61	1139.12	1077.93	1093.02	1079.40	1130.21	1072.63	1075.60	**1070.59**
CTD	71633	58067	57319	57275	**57125**	60361	57679	57678	57621
T(sec)	0.03	3.41	20.06	6.59	7.90	6.25	17.55	8.03	10.00

initial solutions are used. Furthermore, relaxations tend also to mitigate the impact of low quality initial solutions. When passing from a random initialization to a constructive procedure, the distance deteriorates by $+0.63\%$ to $+0.87\%$ in presence of relaxations, whereas the distance strongly deteriorates by $+3.95\%$ if only feasible solutions are used.

The different relaxations appear to perform equally well with regards to distance minimization, however the RETURN relaxation is less time-consuming. This observation goes in accordance with the theoretical results of Section 4, and would motivate the choice of this particular relaxation for distance minimization.

5.2 Performance of relaxations with regards to fleet minimization

When relaxations are used, addressing the hierarchical objective of fleet size minimization and distance can be simply done by iteratively decrementing the fleet size limit and running the local search algorithm of the previous section, as described in Algorithm 2.

Algorithm 2 Fleet Size Minimization

```
1: t = 0 ;
2: while t < nbTry do
3:     sol = LocalSearch(newInitialSol())
4:     if isFeasible(sol) then fleetSize = fleetSize - 1; t = 0 ; else t = t+1 ;
5: fleetSize = fleetSize + 1;
6: for t = 1,...,nbTry do
7:     LocalSearch(newInitialSol())
8: returnBestFeasibleSolution()
```

However, when only feasible solutions are explored, minimizing the number of vehicles leads to new theoretical issues. Indeed, the method must 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, involving eventually the use of auxiliary objectives to progress towards empty routes as in Gendreau et al. (1996) and Bent and Van Hentenryck (2004), or route-removal operations (Nagata and Bräysy 2009). However, these procedures are too different in their behavior to conduct a fair comparison with relaxation-based heuristics. For this reason, only the results for heuristics relying on infeasible solutions are reported in the following experiments.

Table 3 reports computational results on one run of the fleet minimization algorithm for each relaxation scheme, using either the constructive heuristic I1 or the random procedure for solution initialization. For each problem class, the average number of vehicles and distance is indicated. The last two lines indicate the Cumulated Total Distance (CTD) for all the 56 instances and the average computation time per instance.

Relaxations appear to have a large impact on the method performance when dealing with the

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

Inst.	SolI1	Late	Return	Flex	Late	Return	Flex
		SolomonI1 Initial Solution			Random Initial Solution		
R1	13.42	12.67	12.50	**12.42**	13.17	12.50	12.58
	1431.97	1231.65	1241.02	1246.45	1257.77	1258.19	1242.94
R2	3.18	3.09	**2.91**	**2.91**	4.27	**2.91**	3.00
	1326.64	1006.06	1019.56	1017.65	998.95	1026.30	1011.49
C1	10.00	**10.00**	**10.00**	**10.00**	**10.00**	**10.00**	**10.00**
	936.48	840.41	838.63	838.90	859.98	858.21	861.25
C2	3.13	**3.00**	**3.00**	**3.00**	3.75	**3.00**	**3.00**
	696.57	604.90	603.34	603.11	674.27	646.22	629.17
RC1	13.50	12.25	**12.13**	**12.13**	12.25	12.25	**12.13**
	1578.28	1415.26	1414.58	1435.54	1417.72	1420.09	1431.37
RC2	3.75	3.50	**3.25**	**3.25**	3.38	**3.25**	**3.25**
	1653.61	1246.05	1228.97	1241.17	1218.00	1238.56	1254.65
CNV	459	426	419	**418**	450	420	421
CTD	71633	59539	59630	59940	60301	60550	60314
T(sec)	0.03	42.61	14.02	17.10	59.97	19.94	24.89

hierarchical objective of fleet and distance minimization. In particular, the RETURN and FLEX relaxations produce solutions of better quality, with a 418 and 419 cumulated vehicles respectively, than the LATE relaxation, which leads to 426 vehicles.

RETURN and FLEX relaxations are also less dependent upon the availability of a good initial solution. Indeed, the cumulated fleet size reaches 420 and 421 vehicles when a random initial solution is used, whereas with a LATE relaxation the fleet size increases to 450 vehicles. A potential explanation is that infeasible insertions at the beginning of routes are likely to result in massive penalties in the LATE relaxation scheme, leading to imbalanced insertion capacities within the routes, thus reducing the ability to progress towards feasible solutions in tightly constrained settings with few routes.

6 Conclusions

In this paper, four simple neighborhood search procedures, differing only by their relaxation scheme, have been tested on the vehicle routing problem with time windows. Experimental results demonstrate the positive contribution of time-window relaxations in the method performance for two different objectives, as well as their ability to mitigate the impact of low quality starting solutions. The relaxation of Nagata et al. (2010), especially, based on the concept of returns in time, yields the best results in terms of both solution quality and computation efficiency, and thus appear as a promising option for VRPTW metaheuristics.

Finally, as a perspective of research, this study will be completed by comparison of relaxations within an iterated local search and the hybrid genetic algorithm of Vidal et al. (2011a), thus providing the means to challenge the previous observations on more advanced metaheuristic procedures.

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

Bent, R., P. 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. 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.

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.

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*. 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. 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. 2011. The case for strategic oscillation. *Annals of Operations Research* **183**(1) 163–173.

Hashimoto, H., T. Ibaraki, S. Imahori, M. Yagiura. 2006. The vehicle routing problem with flexible time windows and traveling times. *Discrete Applied Mathematics* **154**(16) 2271–2290.

Hashimoto, H., M. Yagiura. 2008. A Path Relinking Approach with an Adaptive Mechanism to Control Parameters for the Vehicle Routing Problem with Time Windows. *EvoCOP'08 Proceedings of the 8th European conference on Evolutionary computation in combinatorial optimization*, *LNCS*, vol. 4972. Springer, 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, M Savelsbergh. 1997a. Vehicle routing: Handling edge exchanges. E. Aarts, J.K. Lenstra, eds., *Local Search in Combinatorial Optimization*. Wiley, New York, 337–360.

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

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.

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.

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.

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, Wiley, New York.

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.

Sexton, T.R., Y.-M. Choi. 1986. Pickup and delivery of partial loads with "soft" time windows. *American Journal of Mathematical and Management Sciences* **6**(3-4) 369–398.

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, C. Prins. 2011a. A Hybrid Genetic Algorithm with Adaptive Diversity Management for a Large Class of Vehicle Routing Problems with Time Windows. Tech. rep., CIRRELT 2011-61.

Vidal, T., T.G. Crainic, M. Gendreau, C. Prins. 2011b. A Unifying View on Timing Problems and Algorithms. Tech. rep., CIRRELT 2011-43.

Vidal, T., T.G. Crainic, M. Gendreau, C. Prins. 2012. Heuristics for Multi-Attribute Vehicle Routing Problems : A Survey and Synthesis. Tech. rep., CIRRELT 2012-05.