A general variable neighborhood search for the traveling salesman problem with time windows under various objectives

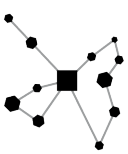
Working Paper DPO-2023-01 (version 0, 08.02.2023)

Mengdie Ye, Enrico Bartolini, Michael Schneider
{ye|bartolini|schneider}@dpo.rwth-aachen.de
Deutsche Post Chair – Optimization of Distribution Networks
RWTH Aachen University, Germany

Abstract

The traveling salesman problem with time windows (TSPTW) has wide practical applications in transportation and scheduling operations. We study the TSPTW in the deterministic case as well as under travel time uncertainty. We consider the three classical TSPTW variants with the objectives of minimizing cost, completion time, and tour duration. In addition, a new TSPTW variant that maximizes the minimum slack of a tour, i.e., the smallest time buffer between the arrival time and the end of the time window over all nodes visited on the tour, is introduced. We further address a variant of the TSPTW in which the arc travel times are uncertain. This problem is modeled by means of an uncertainty set, and the goal is to determine a tour that remains feasible in the worst case within a budget stipulating the sum of all travel time deviations from their nominal values. To solve all targeted problem variants, we develop a two-phase general variable neighborhood search (GVNS) that applies an efficient move evaluation approach within the local search. Extensive numerical experiments on benchmark instances from the literature show that our GVNS finds high-quality solutions that are competitive with those obtained by the state-of-the-art heuristics for all problem variants.

Keywords: traveling salesman problem, time windows, uncertainty, robust optimization, local search



Deutsche Post
Chair - Optimization of
Distribution Networks

RWTH AACHEN
UNIVERSITY

1 Introduction

The traveling salesman problem with time windows (TSPTW) is a very well-known NP-hard combinatorial optimization problem. It aims to find an optimal tour with respect to a certain objective function starting and ending at the depot and visiting a set of nodes within their given time windows exactly once. The objectives of the TSPTW depend on the goals of the specific application. The TSPTW is common in transportation applications such as school bus transportation and post deliveries, and it can also be viewed as a subproblem of the vehicle routing problem with time windows (VRPTW); in this case, the objective is typically to minimize the total cost (or travel time). The TSPTW also arises in scheduling applications: It is equivalent to model a single-machine scheduling problem, where the task is to schedule jobs on a single machine given the release and due time of each job and sequence-dependent setup times. The task of scheduling the jobs on every single machine in the job shop scheduling problem with sequence-dependent setup times, deadlines, and precedence constraints is also equivalent to solving a TSPTW (Balas et al. 2008). In this context, the so-called makespan, i.e., the maximum completion time of all jobs, is usually the objective to be minimized. Apart from the above two well-studied objectives, another variant in the routing context is to minimize the tour duration, i.e., the time difference between the arrival and departure times at the depot in case the departure time at the depot is not fixed in advance. For convenience, we refer to the TSPTW with cost minimization as TSPTW-C, to the TSPTW with makespan minimization as TSPTW-M, and to the TSPTW with tour duration minimization as TSPTW-D.

Inspired by the concept of earliness in scheduling applications, which is defined as the difference between the completion time of a job and its due time, we introduce a new concept called *slack* for the TSPTW. The slack of a node is defined as the time buffer between the arrival time at the node and the end of its time window. Its value indicates how far we can postpone the arrival time at the node without violating its time window. The slack of a tour is thus defined as the minimum slack over all nodes on the tour, and it indicates how far we can postpone the arrival times at all nodes without rendering the tour infeasible. The idea behind this objective is that a tour with a slack that is as large as possible can to some extent be robust against possible travel time fluctuations. Therefore, we also study this TSPTW variant maximizing the slack of the tour and we denote it as TSPTW-S.

Most papers on the TSPTW address deterministic models, in which the parameters defining the problem, such as the travel times, are assumed to be known a-priori. However, under uncertain travel times, the solutions obtained by deterministic models are likely to be infeasible when applied in practice. In this paper, we also study a TSPTW variant under travel time uncertainty and model it as a robust optimization problem. We consider the knapsack-constrained uncertainty set $\mathcal{T}_K$ that has been first introduced by Bartolini et al. (2021). $\mathcal{T}_K$ comprises all possible travel time realizations of interest, each one has a total amount of travel time deviations not larger than a predefined delay budget Δ. The resulting problem, denoted as RTSPTW($\mathcal{T}_K$), is to find a tour with minimum cost that remains feasible with respect to all travel time realizations described by $\mathcal{T}_K$, or in other words, that is feasible for up to Δ units of cumulated delay.

To the best of our knowledge, no solution method has addressed more than two TSPTW variants at the same time (e.g., Kara et al. 2013 and Karabulut and Tasgetiren 2014 both study the TSPTW-C and the TSPTW-M), and it is not clear whether existing heuristics achieve good performance on multiple problem variants. Contrary to this, the method presented in this paper is designed to target all four deterministic TSPTW and the RTSPTW variants described above.

To solve the variants, we develop a two-phase general variable neighborhood search (GVNS) heuristic. The heuristic is based on the variable neighborhood search (VNS) paradigm (see e.g., Mladenović and Hansen 1997) and consists of a constructive and an improvement phase. In the first phase, we build a feasible solution using a VNS heuristic. In the second phase, we try to improve this solution using a GVNS heuristic, i.e., a VNS using a variable neighborhood descent (VND) (Hansen and Mladenović 2001) as local search procedure. A preliminary version of the heuristic, which is designed only for the RTSPTW($\mathcal{T}_K$), was used by the exact algorithm of Bartolini et al. (2021) to obtain valid upper bounds for the problem. Our heuristic is inspired by the two-phase GVNS proposed by Da Silva and Urrutia (2010), which was originally developed to solve the TSPTW-C. Contrary to the original method, we replace the level-based random *OR-opt-1* move with a level-based destroy and repair procedure of Karabulut and Tasgetiren (2014) in the perturbation phase, and we use additional neighborhoods in the VND to be able to handle multiple problem variants. Moreover, we adapt the move evaluation approach from Vidal et al. (2013) to efficiently evaluate the feasibility and profitability of a local search move, and we extend it to the concept of robust feasibility required when solving the RTSPTW($\mathcal{T}_K$).

We report the results of extensive computational experiments conducted on different benchmark sets from the literature for each of the considered variants. We compare our results with the solutions obtained by the state-of-the-art heuristics and the best-known solutions (BKS) from the literature. The results show that our two-phase GVNS is able to achieve high-quality solutions on the benchmark sets for all variants considered. In particular, our heuristic finds 13 new BKS for the TSPTW-D and one new BKS for the RTSPTW($\mathcal{T}_K$).

The paper is structured as follows. We survey the related literature in Section 2. In Section 3, we provide a technical description of the TSPTW variants with different objective functions. The two-phase GVNS heuristic for solving all variants is described in Section 4. We describe the computational experiments in Section 5. Finally, Section 6 presents our conclusion.

2 Literature review

The TSPTW-C, the TSPTW-M, and the TSPTW-D have been extensively studied, and we review exact and heuristic solution methods in Section 2.1. In Section 2.2, we review papers that address the RTSPTW with a budgeted uncertainty set.

2.1 Papers on deterministic TSPTWs

Among the three deterministic TSPTW variants, the TSPTW-C is the most studied one. A variety of exact solution methods can be found in the literature. These methods are mainly based on dynamic programming (DP, see, e.g., Dumas et al. 1995, Balas and Simonetti 2001, Baldacci et al. 2012), branch-and-cut algorithms (Ascheuer et al. 2001, Dash et al. 2012, Boland et al. 2017), and constraint programming (CP, see, e.g., Pesant et al. 1998, Focacci et al. 2002). The TSPTW-M has also been well studied. Many exact solution methods, from branch-and-bound algorithms (Christofides et al. 1981, Baker 1983, Langevin et al. 1993) to mixed integer programming (MIP) formulations (Kara et al. 2013 and Kara and Derya 2015), have been developed. The TSPTW-D has not received as much attention as the other two variants. There are two recent papers solving the TSPTW-D in an exact fashion. One is from Tilk and Irnich (2017), they propose a new DP algorithm that generalizes the approach presented by Baldacci et al. (2012) and provide for the first time the

BKS for four benchmark sets. The other one is from Lera-Romero et al. (2022), they extend the algorithm of Tilk and Irnich (2017) to deal with time-dependent costs and provide 31 new BKS for the problem. For an extensive review of the exact solution methods for deterministic TSPTW variants, we refer to Baldacci et al. (2012) and Pralet (2023).

Numerous heuristics have been developed for solving the TSPTW-C. Carlton and Barnes (1996) solve the TSPTW-C using a tabu search (TS) heuristic that allows infeasible solutions that are handled by a static penalty function. Gendreau et al. (1998) propose a construction heuristic to gradually build a feasible solution and improve it through the removal and reinsertion of all nodes. Ohlmann and Thomas (2007) use compressed annealing, which is a variant of simulated annealing that relaxes the time window constraints by integrating a penalty function using a variable penalty multiplier within a stochastic search procedure. López-Ibáñez and Blum (2010) propose a hybrid method called Beam-ACO that combines ant colony optimization with a tree search method called beam search. Their method is able to improve the best solutions found by Ohlmann and Thomas (2007) on several benchmark sets. Beam-ACO is also adapted to the TSPTW-M by the same authors (Manuel López-Ibáñez et al. 2013) and it provides the BKS for the TSPTW-M on several benchmark sets.

Several papers opt for the metaheuristic GVNS embedded into a two-phase heuristic framework to solve the TSPTW-C and the TSPTW-M. Such a two-phase heuristic is usually composed of a constructive phase, in which a feasible solution is built using a VNS, and an improvement phase, in which a GVNS is called to improve the solution returned by the previous phase. Da Silva and Urrutia (2010) are the first to apply a two-phase GVNS to the TSPTW-C. Their results show the great potential of the GVNS by providing better solutions for several benchmark sets compared to those reported in López-Ibáñez and Blum (2010). Mladenović et al. (2013) extend this approach by using more neighborhoods in the VND in the improvement phase. They also introduce a vector that is used to effectively verify the feasibility of a move. They are able to improve the BKS for some instances, provide better average solution quality, and reduce the average runtimes on all tested benchmarks compared to Da Silva and Urrutia (2010). Recently, Amghar et al. (2019) have adapted this approach to the TSPTW-M. They apply a different order of exploring the neighborhoods in the VND than Mladenović et al. (2013) to handle the makespan objective more effectively. Their computational results show that the GVNS is able to generate at least as good results as the Beam-ACO for the tested benchmarks, and provides new BKS for several instances.

An alternative approach that is competitive with the two-phase GVNS heuristic of Mladenović et al. (2013) is the variable iterated greedy (VIG) algorithm of Karabulut and Tasgetiren (2014) for the TSPTW-C. The algorithm of Karabulut and Tasgetiren (2014) is able to provide new BKS for several benchmark instances compared to Da Silva and Urrutia (2010). For the TSPTW-M, Pralet (2023) has recently developed an algorithm called iterated maximum large neighborhood search (ImaxLNS) which dominates the two-phase GVNS of Amghar et al. (2019). ImaxLNS is a combination of iterated local search (ILS) and large neighborhood search (LNS). It uses a parameter called insertion-width to control the maximum destroy size such that the repair method can remain applicable with a limited complexity. The algorithm is able to return the BKS for all tested benchmark sets in less than 1 second in the worst case for the TSPTW-M and can be applied to the time-dependent version of the TSPTW.

Heuristic approaches for the TSPTW-D are rather limited. Savelsbergh (1992) is the first to address tour duration minimization in the routing context. He introduces the concept of forward time slack to help compute the tour duration and solves the problem by a local search heuristic based on edge exchanges. Favarreto et al.

(2006) call the problem temporal TSPTW. They develop an ant colony approach to solve the problem and report solutions for benchmark sets originally proposed for the TSPTW-C. Tilk and Irnich (2017) develop a VND heuristic for the problem to generate upper bounds that can be used in their DP approach. The VND starts from a known best tour to the TSPTW-C or TSPTW-M and calls the Balas-Simonetti neighborhoods (see Section 4.2) in the local search iteratively. To the best of our knowledge, the proposed VND is the state-of-the-art heuristic for the TSPTW-D.

2.2 Papers on robust routing problems

For robust routing problems considering uncertain parameters, we mainly review the solution methods based on the budget of uncertainty model of Bertsimas and Sim (2004), in which the uncertainty set encodes a budget of uncertainty defined by a cardinality constraint specifying the number of uncertain parameters that are allowed to deviate from their nominal value.

Based on this model, the robust VRPTW under travel time and (or) demand uncertainty have been extensively addressed. Exact solution methods, such as branch-and-cut algorithms (Agra et al. (2013), Munari et al. (2019)) and DP approaches (Lee et al. 2012) have been proposed to solve small instances to optimality. Several heuristic approaches have also been developed. Braaten et al. (2017) consider a robust VRPTW in maritime transportation in which travel times are uncertain. They use a similar uncertainty set as in Agra et al. (2013) and propose an efficient heuristic based on adaptive LNS to solve the resulting problem. Hu et al. (2018) address the VRPTW under both demand and travel time uncertainty and design a two-stage algorithm based on a modified adaptive VNS heuristic. We refer to Bartolini et al. (2021) and Zhang et al. (2022) for a more detailed overview of robust routing problems.

All the above robust models consider uncertainty sets that are parameterized by means of a budget value Γ that is cardinality-based and that models the maximum number of travel times whose value can simultaneously deviate from the nominal one. Contrary to this, Bartolini et al. (2021) study the robust TSPTW under travel time uncertainty and introduce a knapsack-constrained uncertainty set in which the budget value is denoted by Δ and represents the maximum cumulative delay that can be incurred on a tour. They solve the resulting problem, denoted as RTSPTW($\mathcal{T}_K$), using an exact algorithm based on column generation and DP with a state-space relaxation adapted from Baldacci et al. (2012). In the paper at hand, we study the same problem as in Bartolini et al. (2021), i.e., the RTSPTW($\mathcal{T}_K$), and aim for an effective heuristic for solving it.

3 Problem description

The TSPTW can be formulated on a complete directed graph $G = (N, A)$ with $N = \{0, 1, 2, ..n\}$ as the node set and A as the arc set. Each arc $(i, j) \in A$ is associated with a cost $c_{ij} > 0$ and a travel time $t_{ij} > 0$ (it is assumed in most applications that $c_{ij} = t_{ij}$ if not specified). The travel time t_{ij} generally includes the service time at node i. Each node $i \in N$ is associated with a time window $TW_i = [e_i, l_i]$, $e_i < l_i$, which represents a time interval within which node i must be visited. Node 0 represents the depot, and we assume that $TW_0 = [0, T]$, where $T \geq l_i, \forall i = 1, ..., n$.

We call a path that starts from the depot 0, visits each node $i \in N \backslash \{0\}$ exactly once, and finally returns to the depot 0, a tour, and refer to it as $P = (i_0 = 0, i_1, ...i_n, i_{n+1} = 0)$. As commonly done in most formulations

of the problem, we allow waiting times at the nodes, which means that if the arrival time at a node i is before e_i, waiting occurs until the start of its time window. Therefore, given a tour P, the departure time from a node i_h of P with $h = 0, ..., n+1$, is calculated as $\tau_h^D = \max\{\tau_h^A, e_{i_h}\}$, where $\tau_h^A = \tau_h^D + t_{i_{h-1}\, i_h}$ is the arrival time at node i_h. It is convenient to assume that $\tau_0^A = \tau_0^D$ and $\tau_{n+1}^A = \tau_{n+1}^D$. A tour P is feasible if it allows to visit each node i_h of P within its time window, i.e., $e_{i_h} \leq \tau_h^A \leq l_{i_h}$, $\forall h = 0, ..., n+1$. The problem is to find a feasible tour with respect to a specific objective function as discussed in the following.

3.1 The TSPTW with classical objectives

The TSPTW-C can be formulated as follows (see e.g., López-Ibáňez and Blum 2010, Karabulut and Tasgetiren 2014):

$$\begin{aligned} min \quad & f_c(P) = \sum_{h=0}^{n} c_{i_h\, i_{h+1}} \\ s.t. \quad & \Omega(P) = \sum_{h=0}^{n+1} \omega(i_h) = 0, \end{aligned}$$

where $\omega(i_h) = 1$ if $\tau_h^A > l_{i_h}$, and 0 otherwise. $\Omega(P)$, which denotes the total number of violated time window constraints of tour P, must be zero for feasible solutions. $f_c(P)$ is the objective function minimizing the total cost of arcs traversed along tour P.

The TSPTW-M and the TSPTW-D only differ from the TSPTW-C with respect to the objective function. Given a tour P, the objective functions of minimizing its makespan or tour duration can be respectively written as:

$$\min\ f_m(P) = \tau_{n+1}^A,$$

or

$$\min\ f_d(P) = \tau_{n+1}^A - \tau_0^D.$$

Note that in the TSPTW-D, the departure time τ_0^D at the depot is not fixed and thus is a decision variable that has to be determined for a given tour.

3.2 The TSPTW with slack maximization

In this section, we provide a formal definition of the slack of a tour. Given a path $P = (i_0, i_1, ..., i_p)$ starting from node $i_0 = 0$ and ending at a node $i_p \in N \backslash \{0\}$, we define the slack s_h of the h-th node i_h ($0 \leq h \leq p$) of P, as the gap between the end of the time window l_{i_h} and its arrival time τ_h^A, i.e.,

$$s_h = l_{i_h} - \tau_h^A.$$

The value of s_h indicates how far the arrival time at node i_h can be shifted forward in time without causing a time window violation at i_h. We then define the slack of P, denoted as $\underline{s}(P)$, as the minimum slack over

all nodes of P. More precisely, we define:

$$\underline{s}(P) = \min_{h=0,\ldots,p} s_h.$$

It is not hard to see that P can remain feasible if the arrival time at each node i_h with $0 \leq h \leq p$ is postponed by an amount that is at most $\underline{s}(P)$ units.

Consider the occurrence of a delay that will cause a forward shifting of the arrival time at each node of a tour. If the amount of delay is smaller than the value of the slack, the tour can remain feasible. Hence, we believe that the slack of a tour can in fact to a certain degree reflect the robustness of a tour under travel time uncertainty. Therefore, we consider a new TSPTW variant in which the objective function is to maximize the slack of a tour. The resulting problem, referred to as the TSPTW-S, can be formulated analogously to the TSPTW-C with the following objective function:

$$\max\ f_s(P) = \max\ \underline{s}(P) = \max \min_{h=0,\ldots,p} s_h.$$

3.3 The robust TSPTW with a knapsack-constrained uncertainty set

In the RTSPTW($\mathcal{T}_K$), the travel time t_{ij} of each arc (i,j) is no longer deterministic but, instead, can take any value within an interval $T_{ij} = [t^0_{ij}, t^0_{ij} + \delta_{ij}]$, where t^0_{ij} is called the nominal travel time and δ_{ij} is called the maximum delay of arc (i,j). In this problem, we assume that the total amount of delay incurred by a tour P, i.e., the quantity $\sum_{(i,j)\in P} t_{ij} - t^0_{ij}$, cannot exceed an upper bound called delay budget Δ. The corresponding knapsack-constrained uncertainty set $\mathcal{T}_K$ is thus defined as follows:

$$\mathcal{T}_K = \left\{ \mathbf{t} \in \mathbb{R}^{|A|} : t_{ij} = t^0_{ij} + \xi_{ij}\delta_{ij},\ 0 \leq \xi_{ij} \leq 1,\ \forall (i,j) \in A,\ \sum_{(i,j)\in A} \delta_{ij}\xi_{ij} \leq \Delta \right\}.$$

The goal of the RTSPTW($\mathcal{T}_K$) is to find a least-cost tour that remains feasible with respect to all travel time vectors belonging to $\mathcal{T}_K$.

Consider a path $P = (i_0, i_1, ..., i_p)$ with $i_0 = 0$ and $i_p \in N\backslash\{0\}$ and a travel time vector $\mathbf{t} \in \mathcal{T}_K$. We define $\tau^{\mathbf{t}}_h$ as the departure time from a node i_h of P with respect to $\mathbf{t}$. It is calculated as:

$$\tau^{\mathbf{t}}_h = \max\{e_{i_h}, \tau^{\mathbf{t}}_{h-1} + t_{i_{h-1}i_h}\},\ \ \forall h = 1, \ldots, p.$$

Clearly, a path P is robust feasible with respect to $\mathcal{T}_K$ if

$$\max_{\mathbf{t}\in\mathcal{T}_K} \tau^{\mathbf{t}}_h \leq l_{i_h}, \quad \forall h = 1, \ldots, p. \tag{1}$$

Bartolini et al. (2021) provide an explicit characterization for the robust feasibility of a tour with respect to the uncertainty set $\mathcal{T}_K$, and they develop an efficient algorithm to verify the robust feasibility condition (1). Based on this, we are able to evaluate the robust feasibility of a RTSPTW($\mathcal{T}_K$) tour in the local search procedure of the proposed heuristic. A simplified version of the algorithm is available in Appendix A, and we refer to Bartolini et al. (2021) for a more detailed discussion.

4 A two-phase GVNS for the TSPTW under various objectives

We propose a two-phase heuristic for solving all the considered problem variants. The heuristic is inspired by the GVNS heuristic presented in Da Silva and Urrutia (2010). It features a different perturbation procedure and involves additional neighborhoods in the VND. We use the efficient move evaluation approach from Vidal et al. (2013) in the local search procedure, which has the advantage to track multiple values that are related to different objectives and to speed up the search process. In particular, we introduce an extension of it that can be applied to the RTSPTW($\mathcal{T}_K$).

As commonly done, the algorithm starts with a preprocessing step, in which incompatible arcs that cannot belong to any feasible solution are identified. Specifically, an arc (i, j) is incompatible if $e_i + t_{ij} > l_j$. After the preprocessing, the algorithm executes two phases. Algorithm 1 provides a pseudocode overview of our two-phase GVNS heuristic. The first is a constructive phase, which tries to find a feasible solution using a VNS (Section 4.1). This phase iteratively calls a combination of a perturbation and a local search procedure until a feasible solution is found or the time limit is reached. The second is an improvement phase, which aims to improve the feasible solution obtained by the first phase using a GVNS, i.e., a variant of VNS with a VND as the local search (Section 4.2). The perturbation procedure of the GVNS is the same as in the VNS in the constructive phase. The GVNS iterates to update the overall best solution until a given iteration limit k_{max} or a time limit is reached. Additionally, we explain the efficient move evaluation approach and its extension to the robust problem in Section 4.3, and we discuss several speedup techniques in Section 4.4.

Algorithm 1: Two-phase GVNS

```
1  X ← ConstructionVNS();
2  X* ← X;
3  repeat
4      k ← 1;
5      repeat
6          X' ← Perturbation(X, k);
7          X'' ← VND(X');
8          NeighborhoodChange(X, X'', k);
9      until k > k_max;
10 until time limit is reached;
```

4.1 Constructive phase

To build a feasible solution, we follow the VNS-based procedure proposed in Da Silva and Urrutia (2010) but apply some modifications.

The objective function in this phase used by Da Silva and Urrutia (2010) is to minimize the sum of the time window violations, i.e., the sum of all positive differences between the arrival time at each node and the end of its time window. In their algorithm, finding a feasible solution is equivalent to finding a solution that has a zero objective function value. To also possibly guide the search process towards high-quality solutions, we instead consider the original objective function of each problem variant and permit time window violations that will be penalized. To this end, we use a generalized objective function which is defined as follows:

$$f_{gen}(X) = f(X) + \alpha \cdot P_{tw}(X),$$

where $f(X)$ is the objective value of a solution X depending on the problem variant, $P_{tw}(X)$ is the total time window violations of X, and α is the corresponding penalty weight. Thus, our algorithm in this phase aims at finding a solution that is both feasible ($P_{tw}(\cdot) = 0$) and of good quality.

Algorithm 2 shows the constructive phase. The VNS iterates until it finds a feasible solution or a time limit is reached. After a random solution X^0 (possibly infeasible) is generated, a local search procedure using the *OR-opt-1* neighborhood (a neighboring solution is obtained by relocating a node, see Section 4.1.2) is applied to X^0. The variable k is used to control the strength of the perturbation, and the incumbent solution X is iteratively improved until the maximum perturbation strength k_{max} is reached. In each iteration, X is first destroyed and rebuilt by calling the perturbation procedure (Section 4.1.1), and then improved by a local search using the *OR-opt-1* neighborhood. At the end of each iteration, the incumbent solution and the variable k are updated. If no feasible solution can be found for some consecutive iterations, the penalty weight α is updated.

Algorithm 2: ConstructionVNS

```
X^0 ← RandomSolution();
repeat
    X ← OrOpt1LS(X^0);
    k ← 1;
    repeat
        X' ← Perturbation(X, k);
        X'' ← OrOpt1LS(X');
        NeighborhoodChange(X, X'', k);
    until k > k_max;
    UpdatePenalty(X, α);
until X is feasible or time limit is reached;
```

4.1.1 Perturbation procedure

In the perturbation, we use the destroy and repair procedure similar to that of the VIG algorithm presented in Karabulut and Tasgetiren (2014). This procedure consists of two parts. The first part destroys a solution by removing d nodes from it, and the second part rebuilds a new complete solution with a constructive insertion heuristic by inserting each removed node into the partial solution. In our implementation, the destroy size d is controlled by the perturbation strength k, and we consider the insertion of a node to the position which yields the smallest increase in the objective value.

4.1.2 Local search procedure

In the local search procedure of the VNS, we consider only the *OR-opt-1* neighborhood, which consists in relocating one node to another position. Our local search implementation is of first-improvement type. This means that the incumbent solution is updated as soon as an improvement in the current neighborhood is found, and a change of the neighborhood happens only if there is no possible improvement in the current one.

To evaluate as few neighboring solutions as possible, we first investigate the moves that are more likely to yield a feasible solution. To this end, we partition the nodes of an incumbent solution into two subsets: the set of violated nodes, i.e., the nodes visited after the end of their time windows, and the set of non-violated

nodes, i.e., the nodes visited before the end of their time windows. An *OR-opt-1* move can also be divided into two types: forward or backward move of a node. As a result, we can derive four types of the *OR-opt-1* move: backward move of a violated node, forward move of a violated node, backward move of a non-violated node, and forward move of a non-violated node. Da Silva and Urrutia (2010) present a discussion that shows that exploring the *OR-opt-1* neighborhood in the following order leads to finding a feasible solution more efficiently than exploring the complete neighborhood without partitioning: backward move of a violated node, forward move of a non-violated node, backward move of a non-violated node backward, and forward move of a violated node. We follow this guideline and use the same order of the *OR-opt-1* moves in our implementation.

4.2 Improvement phase

If a feasible solution X can be found after the constructive phase, the heuristic tries to improve it using a GVNS (see lines 2 to 10 in Algorithm 1). The perturbation calls the same destroy and repair procedure as in the constructive phase, and the local search procedure of the GVNS uses a VND that considers more neighborhoods than the one in Da Silva and Urrutia (2010) to address multiple problem variants more effectively.

In our VND procedure, we consider six neighborhoods for all problem variants in the following order: *OR-opt-2 backward*, *OR-opt-2 forward*, *1-opt*, *OR-opt-1 backward*, *OR-opt-1 forward*, *2-opt*. All our neighborhoods are searched in best-improvement fashion. We have tested several other orders, especially the orders proposed in Mladenović et al. (2013) and in Amghar et al. (2019), but the order shown above gives the best results for all considered problem variants.

In addition to these neighborhoods, we include the Balas-Simonetti neighborhoods in our VND to improve its performance in finding high-quality solutions. Balas (1999) introduced a family of large-scale neighborhoods $\mathcal{N}_{BS}^k$ for $k > 2$ for the TSP that each one defined by a given k has an exponential number of neighboring solutions, but can be searched efficiently. Their use in the TSPTW context was first discussed in Balas and Simonetti (2001). To be self-contained, we briefly summarize the Balas-Simonetti neighborhoods of the TSP.

Let $X = (x_0, x_1, ..., x_{+1})$ be a TSP tour. For a parameter $k \geq 2$, $\mathcal{N}_{BS}^k(X)$ consists of all solutions $X' = (x_{\pi(0)}, x_{\pi(1)}, ..., x_{\pi(n+1)})$, where π is a permutation of $\{0, 1, ..., n+1\}$ that satisfies: i) $\pi(0) = 0$, $\pi(n+1) = n + 1$; ii)$\pi(i) \leq \pi(j)$ for any pair of two indices $i, j \in \{1, ..., n\}$ such that $i + k \leq j$. This means that if a node x_i precedes a node x_j by at least k positions in solution X, x_i must still precede x_j in the neighboring solution X'. A best neighboring solution $X' \in \mathcal{N}_{BS}^k(X)$ can be found by solving a shortest path problem with the help of an auxiliary graph G_k^* in $\mathcal{O}(nk^2 2^k)$. Figure 1 shows an example of an auxiliary graph for $k = 3$ given a solution $X = (x_0, ..., x_{n+1})$ with $n = 4$. In the auxiliary graph G_k^*, each position on the tour represents a stage, each stage has states (vertices) that are associated with different values α, and the states of consecutive stages are connected by arcs. Each state of a stage corresponds to a restricted permutation of the nodes around position i. Thus, a state associated with α at stage i represents that the node $x_{i+\alpha}$ is moved from position $i + \alpha$ to position i in the neighboring solution X', i.e., $x'_i = x_{i+\alpha}$. Every path in the graph starting from state o and ending at state d corresponds to a neighboring solution X'. In the example, the path associated with the neighboring solution $X' = (x_0, x_3, x_1, x_4, x_2, x_5)$ is depicted in bold. A detailed discussion of the general structure of the auxiliary graph can, e.g., be found in Balas and Simonetti (2001) or Tilk and Irnich (2017).

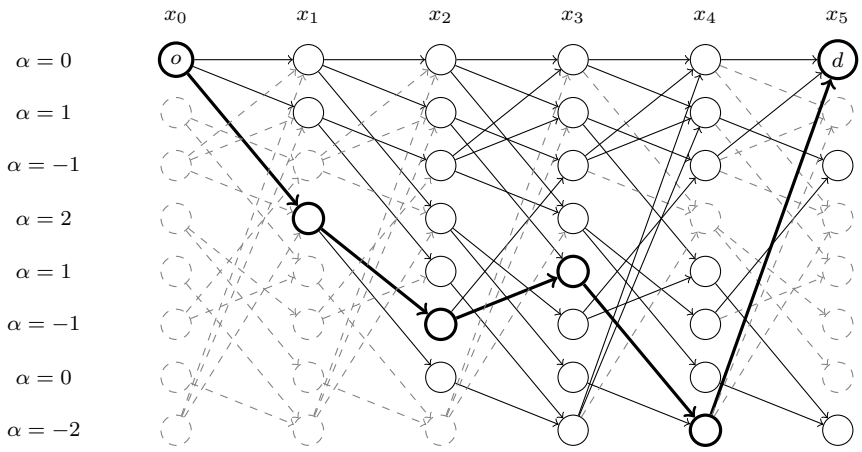


Figure 1: Auxiliary Graph G_k^* for $k = 3$ for $\mathcal{N}_{BS}^k(X)$ of $X = (x_0, ...x_5)$ (source: Tilk and Irnich 2017).

For a given k, the computational effort required to explore $\mathcal{N}_{BS}^k$ is linear in n, yet it increases rapidly for larger values of k. Therefore, we investigate two versions of the GVNS heuristic to solve all deterministic TSPTW variants: One puts more emphasis on the solution quality by including $\mathcal{N}_{BS}^k$ with a value of $k = 6$ as the last neighborhood in the VND, and we denote it as GVNS-Q; the other one considers only the six classical neighborhoods in the VND, is denoted as GVNS-S, and aims to find good solutions with a smaller amount of runtime.

4.3 Efficient move evaluation approach

In the local search procedure, every move needs to be evaluated with respect to feasibility and profitability (a move is profitable if it improves the current solution). Checking the profitability of a move for the TSPTW-C or the RTSPTW($\mathcal{T}_K$) is easy, one only needs to compare the sum of the cost of the deleted arcs and that of the inserted arcs. For the other three problem variants, checking the profitability of a move is of the same difficulty as checking the feasibility: in principle, we need to update the arrival times at all the nodes of the tour that are involved in the move, which is time-consuming.

To speed up the evaluation of solutions in the neighborhood search, we adapt the move evaluation approach proposed by Vidal et al. (2013) to our setting. It is a sequence-based approach that allows to efficiently evaluate the solutions with respect to their duration and time window violations. We first describe the approach adapted to the deterministic TSPTW variants and then discuss the modifications that are needed in the robust setting.

4.3.1 Move evaluation in deterministic TSPTWs

The approach of Vidal et al. (2013) exploits the fact that all moves based on a constant number of arc exchanges or sequence relocations can be viewed as a separation of a tour into subsequences of visits, which are concatenated into a new tour (see the formal property in, e.g., Kindervater and Savelsbergh 1997, Irnich 2008, Vidal et al. 2013). For instance, the new tour obtained after applying an *OR-opt-1* move by relocating a node i to a forward position of j can be described using the concatenation of four subsequences of visits: $(0, ..., i-1) \oplus (i+1, ..., j) \oplus (i) \oplus (j+1, ..., n+1)$, where $\oplus$ represents the concatenate operation. Each operation is evaluated using the stored values for the involved subsequences.

The effectiveness of the approach depends substantially on the concatenation operation. Vidal et al. (2013) show that in the VRPTW, many objective functions and constraints can be evaluated in amortized constant time for inter-route and intra-route moves which corresponds to moves in the TSPTW. Therefore, we can directly apply their concatenations of the distance, duration, and time window violations to our deterministic TSPTW variants. However, we still need to account for the objective value of the makespan and the slack. In the remainder, we first briefly present the definitions of the concatenations of distance $Dist$, duration Dur, time window violations TW, and corresponding earliest and latest departure times from the first node E and L in Vidal et al. (2013). We then describe the concatenations of the makespan M and the slack S that are derived from the previous definitions.

Let $\sigma = (i, ..., j)$ and $\sigma' = (i', ..., j')$ be two subsequences of visits. The following operations are presented by Vidal et al. (2013):

$$Distance : Dist(\sigma \oplus \sigma') = Dist(\sigma) + d_{ji'} + Dist(\sigma'), \tag{2}$$

$$Duration : Dur(\sigma \oplus \sigma') = Dur(\sigma) + Dur(\sigma') + t_{ji'} + \Delta_{WT}, \tag{3}$$

$$TW\ violation : TW(\sigma \oplus \sigma') = TW(\sigma) + TW(\sigma') + \Delta_{TW}, \tag{4}$$

$$Earliest : E(\sigma \oplus \sigma') = \max\{E(\sigma') - \Delta_D, E(\sigma)\} - \Delta_{WT}, \tag{5}$$

$$Latest : L(\sigma \oplus \sigma') = \min\{L(\sigma') - \Delta_D, L(\sigma)\} - \Delta_{TW}, \tag{6}$$

where $\Delta_D = Dur(\sigma) + t_{ji'} - TW(\sigma)$ is an auxiliary variable for calculating the time needed to reach the first node of σ', $\Delta_{WT} = \max\{e_{i'} - \Delta_D - L(\sigma), 0\}$ is the additional waiting time, and $\Delta_{TW} = \max\{E(\sigma) + \Delta_D - l_{i'}, 0\}$ is the additional time window violation after reaching the first node of σ'.

Note that in the routing context, the makespan represents the completion time of ending the tour, which can be seen as a special case of the tour duration when the starting time at the original depot is fixed. Therefore, we can define the concatenation of the makespan M using the duration Dur and the earliest departure E that ensures this minimum duration, i.e.:

$$Makespan :\ M(\sigma \oplus \sigma') = Dur(\sigma \oplus \sigma') + E(\sigma \oplus \sigma'). \tag{7}$$

As for the slack, we recall that it indicates how far we can postpone the arrival time at each node such that the tour remains feasible. Hence, its value for the concatenated sequence can be easily calculated using the earliest and latest departure E and L, and the time window violation TW, i.e.:

$$Slack :\ S(\sigma \oplus \sigma') = L(\sigma \oplus \sigma') - E(\sigma \oplus \sigma') - TW(\sigma \oplus \sigma'). \tag{8}$$

Based on operations (2)–(8), the evaluation of a move is to first calculate the values for the concatenations on relevant subsequences and then to evaluate feasibility and profitability with respect to different objective functions of the new sequence after the move. Hence, if the values for the concatenations on each relevant subsequence can be computed in a preprocessing step, we can perform any move evaluation in constant time.

4.3.2 Move evaluation in the RTSPTW($\mathcal{T}_K$)

In the RTSPTW($\mathcal{T}_K$), the determination of cost can easily be done using the concatenation of distance as mentioned above. To check the robust feasibility of a given tour $P = (i_0, ..., i_p)$, we need to calculate for

each node i_h on P its worst-case departure time, which can be computed by considering the scenarios in which P starts to accumulate as much delay as possible from all possible nodes i_k with $0 \leq k < p$ (see Appendix A). This means that we need to keep track of the amount of delay that is cumulated from any possible i_k, which is non-trivial when concatenating two subsequences.

However, if we concatenate one arbitrary subsequence and another one that contains only one single node, i.e., we extend the first sequence by inserting a new node at the end of it, the values for the above concatenations in the deterministic setting can be modified in such a way that the position k of the starting node of delay on the subsequence and the cumulated delay starting from node i_k can also be stored.

Let $\sigma = (i_0, ..., i_h)$ be the first subsequence and $\sigma' = (i_v)$ be the second one consisting of one node i_v. We next introduce the minimum duration $Dur_k(\sigma)$, minimum time window violation $TW_k(\sigma)$, and the earliest and latest departure times $E_k(\sigma)$ and $L_k(\sigma)$ when the delay starts to accumulate from the k-th position ($0 \leq k < h$) on the subpath represented by σ. Note that these values for σ' are computed as: $Dur_k(\sigma') = 0$, $TW_k(\sigma') = 0$, $E_k(\sigma') = e_{i_v}$, and $L_k(\sigma') = l_{i_v}$. The following operations can be used to evaluate the time window violations of the new sequence:

$$Dur_k(\sigma \oplus \sigma') = Dur_k(\sigma) + Dur_k(\sigma') + t^0_{i_h i_v} + \min\{\delta_{i_h i_v}, r(\sigma, k, h)\} + \Delta^k_{WT}, \tag{9}$$

$$TW_k(\sigma \oplus \sigma') = TW_k(\sigma) + TW_k(\sigma') + \Delta^k_{TW}, \tag{10}$$

$$E_k(\sigma \oplus \sigma') = \max\{E_k(\sigma') - \Delta^k_D, E_k(\sigma)\} - \Delta^k_{WT}, \tag{11}$$

$$L_k(\sigma \oplus \sigma') = \min\{L_k(\sigma') - \Delta^k_D, L_k(\sigma)\} + \Delta^k_{TW}, \tag{12}$$

where $\delta_{i_h i_v}$ is the maximum delay on arc (i_h, i_v), $r(\sigma, k, h)$ is the residual delay that is allowed if the delay starts to accumulate from i_k to i_h under the control of the delay budget Δ (see details in Appendix A). In fact, the most important change with respect to the calculations in the deterministic case is that we use the actual travel time $t^0_{i_h i_v} + \min\{\delta_{i_h i_v}, r(\sigma, k, h)\}$ on arc (i_h, i_v) when delay starts from i_k to replace the nominal travel time that we use in the deterministic setting. The calculation of the additional waiting Δ^k_{WT} and time window violations Δ^k_{TW} after reaching the new node, as well as the auxiliary variable Δ^k_D can also be extended easily as follows: $\Delta^k_D = Dur_k(\sigma) + t^0_{i_h i_v} + \min\{\delta_{i_h i_v}, r(\sigma, k, h)\} - TW_k(\sigma)$, $\Delta^k_{WT} = \max\{E_k(\sigma') - \Delta^k_D - L_k(\sigma), 0\}$, and $\Delta^k_{TW} = \max\{E_k(\sigma) + \Delta^k_D - L_k(\sigma'), 0\}$.

Based on these operations, the robust feasibility of any move considered in this paper can be evaluated efficiently provided all concatenation values associated with the corresponding subsequences can be retrieved in constant time for all possible positions k of the starting node of delay.

4.4 Speedup techniques

As mentioned in Section 4.2, we only allow feasible solutions during the search in the improvement phase. This allows us to use specific techniques to avoid evaluating unpromising neighboring solutions.

Because the evaluation of cost can be done in $\mathcal{O}(1)$, we can discard those neighboring solutions with a larger cost in the TSPTW-C before evaluating their feasibility. This also applies to the RTSPTW($\mathcal{T}_K$). Additionally, as discussed in the move evaluation approach in Section 4.3.1, the feasibility of a TSPTW tour can be checked in constant time. Hence, in the RTSPTW($\mathcal{T}_K$), before evaluating the robust feasibility of a solution, we can filter out those that are already infeasible with respect to the deterministic case, i.e., those with positive time window violations.

In the TSPTW-M and TSPTW-D, because the objective functions are related to the arrival times at the nodes, we cannot separate the evaluation of profitability and feasibility. However, we observe that given a tour $P = (i_0, i_1, i_2, ...i_n, i_{n+1})$, if there exists positive waiting time at a node i_h, i.e., the arrival time $\tau_h^A < e_{i_h}$, relocating it to a backward position on the tour would never decrease its departure time, i.e., $\tau_h^D = e_{i_h}$, and thus would not be profitable. Hence, a neighboring solution in the TSPTW-M or TSPTW-D that is produced by moving a node with positive waiting time to a previous position on the tour can be discarded.

5 Computational experiments

We introduce the benchmark sets in Section 5.1, and we describe the computational environment and the parameter setting of our algorithm in Section 5.2. We carry out numerical experiments to (i) investigate the impact of the efficient move evaluation approach described in Section 4.3 (see Section 5.3), and (ii) compare GVNS-S and GVNS-Q to the state-of-the-art heuristics from the literature for each variant (see Section 5.4).

5.1 Benchmark instances

For the deterministic variants, we consider the following seven TSPTW benchmark sets:

- The *Dumas* set proposed by Dumas, Desrosiers, Gelinas, and Solomon (1995) contains 135 instances with a number of nodes ranging from 20 to 200. The instances are grouped into 27 classes, each of which has five instances with the same number of nodes and the same maximum time window width.
- The *GDE* set consists of 130 instances which were obtained from the *Dumas* instances with 20 to 100 nodes by extending the time window width. This set was introduced by Gendreau, Hertz, Laporte, and Stan (1998).
- The *OT* set, generated by Ohlmann and Thomas (2007), contains 25 instances that were obtained from the *Dumas* instances with 150 and 200 nodes in the same way as the *GDE* instances.
- The *DaSilva* set was proposed by Da Silva and Urrutia (2010) and consists of 125 instances with a number of nodes ranging from 200 to 400. The instances are grouped into 25 classes in the same way as the *Dumas* instances.
- The *AFG* set contains 50 asymmetric instances with 10 to 233 nodes introduced by Ascheuer (1996).
- The *Pesant* set was proposed by Pesant, Gendreau, Potvin, and Rousseau (1998) and consists of 27 instances with 19 to 44 nodes. These instances were derived from Solomon's RC2 VRPTW instances (Solomon 1987).
- The *Potvin* set introduced by Potvin and Bengio (1996) contains 30 instances that were also derived from Solomon's RC2 instances. The number of nodes per instance ranges from 3 to 45.

All benchmarks are available at http://lopez-ibanez.eu/tsptw-instances.

For the RTSPTW($\mathcal{T}_K$), the only available benchmark set has been proposed by Bartolini et al. (2021). These instances are derived from the *GDE* instances with 20 to 80 nodes. They are divided into two groups, called

GDE-D and *GDE-I*, which differ with respect to the definition of the maximum arc travel time. For each instance in the former group, the maximum arc travel time is proportional to its nominal value, while for each instance in the latter group, it equals the nominal value plus a random independent number. In each group, there are three different uncertainty levels that are represented by three different budget values $\Delta = 20$, 40, and 60. The resulting set contains a total of 630 instances and is available at https://pubsonline.informs.org/doi/abs/10.1287/trsc.2020.1011.

5.2 Computational environment and parameter setting

Both GVNS-S and GVNS-Q are implemented in C++ and compiled with GCC release 9.1. All experiments were conducted on a single core of a computing cluster with an Intel(R) Xeon(R) E5-2430v2 processor at 2.50 GHz with 64 GB RAM under CentOS 7. For each instance, GVNS-S and GVNS-Q were run 15 times, and each run was executed sequentially in a single thread. In each run, we set the destroy size $d = k \times 5$ as in the VIG algorithm of Karabulut and Tasgetiren (2014), where k is the perturbation strength and its maximum value defined by the parameter k_{max} is set to $(n-1)/5$. The penalty weight α is initialized to 10 and multiplied (or divided) by $\delta = 2$ every $\eta = 5$ iterations in which the feasibility status of the best found solution has not changed after the local search procedure. The maximum value of α is restricted to 1000. The time limit for finding a feasible solution in the constructive phase is set to 30 seconds, and the total time limit is set to 60 seconds.

5.3 Impact of the efficient move evaluation approach

To investigate the effect of the adapted move evaluation approach, we compare the runtimes of its efficient implementation (see Section 4.3) with a straightforward implementation of the move evaluation for various problem variants. The straightforward move evaluation tentatively performs a move and tests its feasibility and profitability by recalculating the related information, e.g., arrival time at each node, of the solution obtained after the move.

The comparison is conducted on the *Dumas*, *GDE* and *OT* instances for the deterministic TSPTW variants and on the *GDE-D* and *GDE-I* instances with a medium uncertainty level for the RTSPTW($\mathcal{T}_K$). Figure 2 presents the results in terms of runtimes of the two implementations. The runtimes are given in seconds as the average over 15 runs for all instances with the same number of nodes. Because for different objective functions, the average runtimes of the respective implementation share the same increasing trend for an increasing number of nodes, the runtimes for the deterministic variants were further aggregated over all considered objective functions. The comparison in Figure 2 demonstrates the efficiency of the adapted move evaluation approach, and as expected, its advantage clearly increases with the number of nodes in the instances.

5.4 Performance comparison on the different TSPTW variants

In this section, we compare the performance of GVNS-S and GVNS-Q to the state-of-the-art heuristics from the literature on the benchmark sets described in Section 5.1. Sections 5.4.1–5.4.5 provide summary tables of the comparisons for the four deterministic TSPTW variants and the RTSPTW($\mathcal{T}_K$), respectively. Each row of these tables presents the averaged statistics of the runs performed for each set over all its instances.

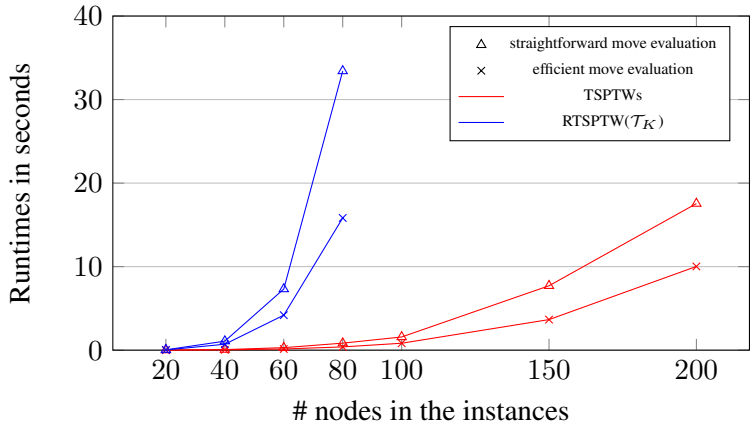


Figure 2: Runtimes (in seconds) for the two implementations.

In our experiments, GVNS-S and GVNS-Q were run 15 times for each instance, respectively. The name of the set is given in the first column. The second column represents the best-known solution (BKS), which corresponds to the best-known value from the literature (not necessarily optimal) or the best value from any run of our algorithms if it is better or if the respective instance has not be considered so far. The remaining columns are divided into two main blocks, the first reports the performance of the start-of-the-art heuristics, and the second reports the performance of GVNS-S and GVNS-Q. Because the best solutions reported by the algorithms from the literature were obtained for different numbers of runs, we base the comparison on the average solution quality and runtimes over the runs performed by the different algorithms. To ensure a fairer runtime comparison to the results from the literature, which were obtained by computers with different CPUs, we report in the tables the respective processor, its Passmark score for a single core (see `www.cpubenchmark.net`), and the resulting factor for the runtime correction. We use this factor to translate all runtimes into a common measure, and the runtimes reported in the tables based on different CPUs are already scaled. Note that values in bold in all tables indicate the best results in the comparison. For each problem variant, the detailed comparison on each individual set is available in Appendix B.

5.4.1 Results for the TSPTW-C

Our GVNS-S and GVNS-Q have been tested on all seven benchmark sets for the TSPTW-C. We compare the results obtained by our methods to those obtained by the GVNS heuristic of Mladenović, Todosijević, and Urošević (2013), denoted as MTU, and to those obtained by the VIG algorithm of Karabulut and Tasgetiren (2014), denoted as KT. MTU was run 15 times on each instance in the sets *AFG*, *Potvin* and *Pesant*, and 30 times on each one in the remaining sets, and 25 runs of KT were performed on each instance. As mentioned before, the comparison is therefore based on the average results over the runs of each algorithm.

Table 1 summarizes the comparison. Because no new BKS for this problem has been found by our methods, the BKS correspond to the best-known values from the literature as reported at `http://lopezibanez.eu/tsptw-instances`. Results of each set are presented using the following metrics: the gap metric $gap_a\%$ reports the percentage gap of the average solution found by the respective method to the BKS (computed as $100 \cdot \frac{avg-BKS}{BKS}\%$, where avg denotes the average solution found by the respective method); and the time metrics t_{avg} and t_{sd} represent, respectively, the average value and standard deviation of the runtimes in seconds over all runs.

Table 1: Aggregated results for all seven benchmark sets on the TSPTW-C.

		MTU			KT			GVNS-S			GVNS-Q		
Inst. set	BKS	$gap_a\%$	t_{avg}	t_{sd}	$gap_a\%$	t_{avg}	t_{sd}	$gap_a\%$	t_{avg}	t_{sd}	$gap_a\%$	t_{avg}	t_{sd}
Dumas	579.5	0.01	0.3	0.2	**0.00**	0.8	0.8	0.03	0.2	0.2	**0.00**	0.4	0.3
GDE	432.4	0.02	0.7	0.3	**0.00**	0.8	1.1	0.08	0.2	0.3	0.04	0.5	0.4
OT	737.7	**0.08**	10.9	3.1	0.20	27.5	30.7	0.28	5.2	1.5	0.13	9.3	2.7
DaSilva	12135.0	**0.07**	13.5	2.3	0.15	63.5	29.2	0.32	3.6	2.0	0.09	5.0	2.2
AFG	5550.8	**0.00**	3.4	3.3	-	-	-	0.06	0.4	0.2	0.03	1.2	0.4
Pesant	623.7	**0.00**	0.2	0.3	-	-	-	0.03	0.0	0.0	0.02	0.0	0.2
Potvin	634.7	**0.01**	0.4	0.4	-	-	-	0.06	0.0	0.0	0.06	0.0	0.1
Processor type		Core i3-380M			Core i5-540M			Xeon E5-2430v2					
Processor speed		2.53 GHz			2.53 GHz			2.50 GHz					
Passmark score		1016			1156			1483					
Time factor		0.7			0.8			1.0					

According to the obtained results, GVNS-Q achieves a better average solution quality and has a lower variability in most of the runs than GVNS-S, but it requires more runtimes. In terms of quality, MTU shows the best performance, being able to find good average solutions for all sets with a largest gap to the BKS within 0.08%. KT performs better than MTU on the sets *Dumas* and *GDE* but is dominated by MTU on the other two sets included in their experiments. GVNS-S performs worse than KT and MTU on all sets. GVNS-Q is able to find the BKS for the *Dumas* set in all runs, but for the other sets, it is dominated by MTU. However, the gaps between the average solutions found by GVNS-S and the BKS are always below 0.23% and those provided by GVNS-Q are always below 0.13%. Moreover, GVNS-Q finds better average solution values than KT on the *OT* and *DaSilva* instances. In terms of runtimes, both our methods outperform MTU and KT, especially on the large instances in the sets *OT* and *DaSilva*. In comparison to MTU (KT), the average runtime of GVNS-S is reduced by 67% (93%), and that of GVNS-Q is reduced by 43% (89%).

Because our solution quality is not competitive to MTU and KT on the *OT*, *DaSilva*, and *AFG* instances, we test GVNS-S and GVNS-Q on these three sets by allowing longer runtimes. Specifically, we repeat the whole GVNS procedure in the improvement phase (which includes calling the perturbation and the VND for all possible k values until k_{max}) until there has been no improvement for η_{rep} consecutive times. We compared the results obtained by our methods using increasing values of η_{rep} and decided to use $\eta_{rep} = 10$ to achieve a balanced performance regarding the solution quality and the runtimes. The results are shown in Table 2.

Table 2: Aggregated results for the *OT*, *DaSilva* and *AFG* instances on the TSPTW-C ($\eta_{rep} = 10$).

		MTU			KT			GVNS-S			GVNS-Q		
Inst. set	BKS	$gap_a\%$	t_{avg}	t_{sd}	$gap_a\%$	t_{avg}	t_{sd}	$gap_a\%$	t_{avg}	t_{sd}	$gap_a\%$	t_{avg}	t_{sd}
OT	737.7	0.08	10.9	3.1	0.20	27.5	30.7	0.13	17.0	2.8	**0.07**	24.3	3.2
DaSilva	12135.0	0.07	13.5	2.3	0.15	63.5	29.2	0.17	10.1	2.9	**0.05**	15.0	2.9
AFG	5550.8	**0.00**	3.4	3.3	-	-	-	0.03	1.7	0.5	0.01	4.6	0.9
Processor type		Core i3-380M			Core i5-540M			Xeon E5-2430v2					
Processor speed		2.53 GHz			2.53 GHz			2.50 GHz					
Passmark score		1016			1156			1483					
Time factor		0.7			0.8			1.0					

Both GVNS-S and GVNS-Q improve in terms of solution quality, finding better average solutions with smaller gaps to the BKS. Furthermore, GVNS-Q is now able to obtain better average values than not only KT but also MTU for the *OT* and *DaSilva* instances. Regarding the runtimes, both GVNS-S and GVNS-Q are still dominating KT. GVNS-S is still faster than MTU on the *DaSilva* and *AFG* instances, but GVNS-Q now requires more runtime than MTU on all three sets.

5.4.2 Results for the TSPTW-M

For the TSPTW-M, GVNS-S and GVNS-Q have also been tested on all seven sets, and the results are compared to those obtained by the GVNS heuristic of Amghar, Cordeau, and Gerdron (2019), denoted as ACG, and to those obtained by the ImaxLNS of Pralet (2023), denoted as P. Note that ACG was run 15 times and P was run 5 times on each instance of the six sets except the *DaSilva* set. The comparison is again based on the average results over all corresponding runs.

The summary of the comparison is shown in Table 3. As mentioned before, the BKS correspond to the best of the best-known values from the literature, which are reported in Pralet (2023), and the best values found by our methods. The results are presented using the gap metric RPD_a and the time metrics t_{avg} and t_{sd}. RPD_a denotes the average value of the relative percentage deviation (RPD) over all runs, where $RPD = 100 \cdot \frac{value-BKS}{BKS}\%$, and $value$ is the objective value of every single run. This measure was first used by López-Ibáñez and Blum (2010) to assess the average solution quality and also reported in Amghar et al. (2019) and Pralet (2023).

Table 3: Aggregated results for all seven benchmark sets on the TSPTW-M.

		ACG			P			GVNS-S			GVNS-Q		
Inst. set	BKS	RPD_a	t_{avg}	t_{sd}	RPD_a	t_{avg}	t_{sd}	RPD_a	t_{avg}	t_{sd}	RPD_a	t_{avg}	t_{sd}
Dumas	668.5	**0.00**	0.1	0.0	**0.00**	0.0	0.0	0.01	0.1	0.1	**0.00**	0.2	0.3
GDE	575.0	0.01	0.1	0.2	**0.00**	0.0	0.0	0.02	0.0	0.2	0.01	0.3	0.3
OT	996.2	0.02	2.8	2.0	**0.00**	0.0	1.2	**0.00**	0.9	0.5	**0.00**	2.5	0.5
AFG	8865.4	**0.00**	0.1	0.2	**0.00**	0.0	0.0	**0.00**	0.1	0.1	**0.00**	0.3	0.2
Pesant	794.9	0.11	0.3	0.4	**0.00**	0.1	0.0	0.07	0.0	0.0	0.05	0.0	0.2
Potvin	694.9	0.01	0.3	0.6	**0.00**	0.1	0.1	0.16	0.0	0.1	0.12	0.0	0.1
DaSilva	15869.8*	-	-	-	-	-	-	**0.00**	0.3	0.4	**0.00**	1.0	0.5
Processor type		Core i7-3770			Xeon E5-2660v3			Xeon E5-2430v2					
Processor speed		3.40 GHz			2.60 GHz			2.50 GHz					
Passmark score		2071			1816			1483					
Time factor		0.6 ($= 0.4^{\dagger} \times 1.4$)			1.2			-					

†: Runtimes reported in Amghar et al. (2019) were already scaled by multiplying 2.5, we retrieve the actual runtimes by first multiplying 0.4.

In terms of quality, P provides the best average solution values for all sets included in their experiments. GVNS-Q fails to find the BKS in all runs for the *GDE*, *Pesant* and *Potvin* sets with a largest average RPD of 0.12%. GVNS-S performs slightly worse than GVNS-Q with a higher largest average RPD value of 0.16%. ACG is on par with GVNS-Q on the sets *Dumas*, *GDE* and *AFG*, and is better than GVNS-Q on the *Potvin* instances. Both GVNS-S and GVNS-Q outperform ACG with respect to the average solution quality obtained over all runs for the sets *OT* and *Pesant*. In terms of runtimes, P is the fastest method on average, but it is slightly slower than our methods on the *Pesant* and *Potvin* sets. GVNS-S is faster than ACG on all

sets included in their experiments, especially on the *OT* set, reducing the runtime by approximately 67%.

We also applied our methods to solve the TSPTW-M on the *DaSilva* instances, and we provide in Table 3 the solutions for these instances (indicated by $*$) for the first time. The details of the results for these instances are available in Appendix B.

5.4.3 Results for the TSPTW-D

For the TSPTW-D, both GVNS-S and GVNS-Q were again applied to all seven sets. For the *AFG*, *GDE*, *OT*, and *Potvin* instances, the results are shown in comparison with the VND heuristic of Tilk and Irnich (2017), denoted as TI. Note that TI executes one run of their VND comprising Balas-Simonetti neighborhoods with different k values. To have a fair comparison, we compare their results with our average results over 15 runs.

Table 4 summaries the comparison on the four sets. Note that the BKS used in the comparison are those reported in Lera-Romero et al. (2022). Apart from the metrics used in Table 1, we provide in column $\#improved$ the number of instances with new BKS found by our methods, out of the total number of instances (shown in parenthesis) that were included in the experiments in Lera-Romero et al. (2022). A "-" in this column indicates that no new BKS has been found by our methods.

Table 4: Aggregated results for all seven benchmark sets on the TSPTW-D.

		TI		GVNS-S				GVNS-Q			
Inst. set	BKS	$gap_a\%$	t_{avg}	$\#improved$	$gap_a\%$	t_{avg}	t_{sd}	$\#improved$	$gap_a\%$	t_{avg}	t_{sd}
GDE	503.6	0.52	7.7	-	0.20	0.1	0.2	-	**0.08**	0.4	0.4
OT	946.2	0.42	42.0	10(25)	0.05	1.3	0.5	13(25)	**0.04**	5.2	1.4
AFG	8418.1	0.01	1.8	-	0.02	0.1	0.1	-	**0.00**	0.4	0.2
Potvin	648.4	1.21	1.3	-	0.72	0.0	0.0	-	**0.52**	0.0	0.1
Dumas	659.8*	-	-	-	0.04	0.1	0.1	-	**0.01**	0.2	0.3
DaSilva	15825.5*	-	-	-	**0.00**	0.4	0.4	-	**0.00**	1.1	0.5
Pesant	709.8*	-	-	-	0.36	0.0	0.0	-	**0.21**	0.0	0.2
Processor type		Core i7-2600			Xeon E5-2430v2						
Processor speed		3.40 GHz			2.50 GHz						
Passmark score		1741			1483						
Time factor		1.2			1.0						

According to the obtained results, we observe that compared to TI, GVNS-S already significantly narrows the average gaps to the BKS, and GVNS-Q further expands this advantage. More specifically, the gaps between the average solutions found by GVNS-S and the BKS are always within 0.72%, and those provided by GVNS-Q are further improved to be always below 0.52%. Moreover, for the *OT* set, GVNS-S provides 10 new BKS and GVNS-Q finds additional 3 new BKS. Note that these instances could not be solved to optimality by Lera-Romero et al. (2022). Regarding the runtime, GVNS-Q is slower than GVNS-S by approximately 73%. In comparison to TI, the average runtime of GVNS-S is reduced by about 98%, and that of GVNS-Q is reduced by 92%.

Additionally, we provide new solutions for the 10 *GDE* instances that were not included in both Tilk and Irnich (2017) and Lera-Romero et al. (2022). We also report for the first time the solutions for the *Dumas*, *Pesant*, and *DaSilva* instances for the TSPTW-D (indicated by $*$ in Table 4). Detailed results of these instances can be found in Appendix B.

5.4.4 Results for the TSPTW-S

For the TSPTW-S, we also report results for both GVNS-S and GVNS-Q on all seven sets. Because the TSPTW-S is investigated for the first time, no comparison methods are available. Therefore, we solved all instances of the TSPTW-S using ILOG CPLEX CP optimizer (constraint programming) with a time limit of 1800 seconds, and we compare the results of GVNS-S and GVNS-Q to those found by the CP optimizer.

Table 5 presents the results. We do not compare the results of GVNS-S and GVNS-Q to those of the CP optimizer on the *DaSilva* instances because the CP optimizer is only able to solve less than 20 (out of 125) instances feasibly within the given time limit. For other sets, the comparison only considers the instances in which a feasible solution has been found by the CP optimizer. This information is reported in column $\#feas$. Additionally, we report in column $\#opt$ the number of instances for which an optimal solution has been found, out of the total number of instances with feasible solutions. Besides the metrics used in the comparison for the TSPTW-D, we report also in column $gap_b\%$ the percentage gap of the best solution found by the method to the BKS (computed as $100 \cdot \frac{best-BKS}{BKS}\%$, where $best$ denotes the best solution found by the respective method).

Table 5: Aggregated results for all seven benchmark sets on the TSPTW-S.

		CP optimizer				GVNS-S					GVNS-Q				
Inst. set	BKS	$\#feas$	$\#opt$	$gap_b\%$	t_{avg}	$\#improved$	$gap_b\%$	$gap_a\%$	t_{avg}	t_{sd}	$\#improved$	$gap_b\%$	$gap_a\%$	t_{avg}	t_{sd}
Dumas	7.0	104(135)	100(104)	0.93	179.4	2(104)	**0.00**	0.08	0.2	0.2	2(104)	**0.00**	**0.07**	0.3	0.3
GDE	56.2	108(130)	102(108)	0.12	243.8	2(108)	**0.00**	0.03	0.3	0.3	2(108)	**0.00**	**0.00**	0.5	0.4
OT	55.3	18(25)	13(18)	0.20	681.5	2(18)	**0.00**	**0.00**	7.9	1.2	2(18)	**0.00**	**0.00**	8.9	1.3
AFG	433.8	50(50)	23(50)	0.08	1090.7	1(50)	**0.00**	0.03	1.0	0.2	1(50)	**0.00**	**0.00**	1.5	0.3
Pesant	59.9	25(27)	14(25)	2.49	942.5	3(25)	**0.00**	2.56	0.0	0.1	3(25)	**0.00**	**2.36**	0.1	0.2
Potvin	74.4	27(30)	15(27)	0.54	937.5	4(27)	**0.00**	0.46	0.0	0.0	4(27)	**0.00**	**0.25**	0.0	0.1
DaSilva	14.7	-	-	-	-	-	1.60	6.06	1.6	0.7	-	**0.00**	**2.53**	2.8	0.9

We recall that the TSPTW-S is a maximization problem. To be consistent with the comparisons for other problem variants, the gap metric is computed as $gap_b\% = \frac{BKS-best}{BKS} \times 100\%$. Hence a gap greater than zero still represents a deterioration. According to the results, GVNS-Q succeeds in finding the BKS for all sets. GVNS-S only fails to provide the BKS for the *DaSilva* instances, however, the results are still within 1.60% to the BKS. Compared to the CP optimizer, both our methods are able to find new BKS for 1 instance in the set *AFG*, 2 in the *Dumas*, *GDE* and *OT* sets, 3 in the *Pesant* set, and 4 in the *Potvin* set. Regarding the runtime, both methods require only a small amount of time to find solutions with good quality in comparison with the CP optimizer.

5.4.5 Results for the RTSPTW($\mathcal{T}_K$)

We decided to use only GVNS-S to solve the RTSPTW($\mathcal{T}_K$). We refrain from applying the GVNS-Q to the RTSPTW($\mathcal{T}_K$) because first, adapting the Balas-Simonetti neighborhood to the robust case with respect to $\mathcal{T}_K$ is non-trivial, and second, we observe a non-negligible runtime increasing when solving the deterministic TSPTW variants using GVNS-Q, and last but most important, GVNS-S already achieves very good performance on the RTSPTW($\mathcal{T}_K$) in the preliminary experiments. Based on the experiments for the TSPTW-C with different η_{rep} values, we decided to apply GVNS-S with $\eta_{rep} = 10$ on both *GDE-D* and *GDE-I* instances aiming for improved solution quality.

The results obtained by GVNS-S are compared to the BKS provided by the exact algorithm of Bartolini et al. (2021). Table 6 presents a summary of the comparison. The table is divided into two blocks, one reporting

the results on the set GDE-D, the other on the GDE-I. Considering the fact that some instances are proven infeasible or cannot be solved feasibly within a given time limit by the exact algorithm, we only report in each row aggregated information on the instances whose optimal solutions are known. Besides the metrics that have been already introduced, we report in column $\#opt$ the number of instances for which an optimal solution value has been found by GVNS-S, out of the instances that were solved to optimality by the exact algorithm of Bartolini et al. (2021).

Table 6: Aggregated results for GDE instances with up to 80 customers on the RTSPTW($\mathcal{T}_K$)-T ($\eta_{rep} = 10$).

Set GDE-D

	GVNS-S, $\Delta = 20$					GVNS-S, $\Delta = 40$					GVNS-S, $\Delta = 60$				
Inst.	$\#opt$	$\#improved$	$gap_b\%$	$gap_a\%$	t_{avg}	$\#opt$	$\#improved$	$gap_b\%$	$gap_a\%$	t_{avg}	$\#opt$	$\#improved$	$gap_b\%$	$gap_a\%$	t_{avg}
n20	25(25)	0(25)	**0.00**	**0.00**	0.1	25(25)	0(25)	**0.00**	**0.00**	0.1	25(25)	0(25)	**0.00**	0.04	0.1
n40	25(25)	0(25)	**0.00**	**0.00**	0.9	25(25)	0(25)	**0.00**	**0.00**	0.9	25(25)	0(25)	**0.00**	0.03	0.9
n60	24(24)	0(24)	**0.00**	**0.00**	4.7	24(24)	0(24)	**0.00**	0.03	4.7	22(22)	0(23)	**0.00**	0.02	4.9
n80	28(28)	0(28)	**0.00**	0.02	16.1	23(24)	0(25)	0.01	0.03	17.4	22(22)	1(25)	0.04	0.07	19.8

Set GDE-I

	GVNS-S, $\Delta = 20$					GVNS-S, $\Delta = 40$					GVNS-S, $\Delta = 60$				
Inst.	$\#opt$	$\#improved$	$gap_b\%$	$gap_a\%$	t_{avg}	$\#opt$	$\#improved$	$gap_b\%$	$gap_a\%$	t_{avg}	$\#opt$	$\#improved$	$gap_b\%$	$gap_a\%$	t_{avg}
n20	25(25)	0(25)	**0.00**	**0.00**	0.1	25(25)	0(25)	**0.00**	0.01	0.1	21(21)	0(21)	**0.00**	**0.00**	0.1
n40	25(25)	0(25)	**0.00**	**0.00**	0.9	25(25)	0(25)	**0.00**	**0.00**	0.8	20(20)	0(20)	**0.00**	0.03	0.8
n60	24(24)	0(24)	**0.00**	0.01	4.8	24(24)	0(24)	**0.00**	**0.00**	4.5	15(15)	0(15)	**0.00**	**0.00**	4.8
n80	25(25)	0(25)	**0.00**	0.01	17.6	25(25)	0(25)	**0.00**	0.01	16.8	23(23)	0(23)	**0.00**	0.02	19.6

The obtained results show that GVNS-S is highly successful in solving the RTSPTW($\mathcal{T}_K$) on instances with up to 80 customers. Looking at the instances of both sets GDE-D and GDE-I with a value of Δ up to 60, GVNS-S is able to find 570 optimal solution values out of the 571 found by the exact algorithm of Bartolini et al. (2021). Moreover, GVNS-S is able to find a new BKS for one instance "n80w140.002" with $\Delta = 60$. For those instances on which GVNS-S failed to find the BKS, the average gap between the best solutions obtained by GVNS-S and the BKS is always below 0.04%, and the average runtime of GVNS-S is always below 20 seconds for each instance class.

6 Conclusions

We study three variants of the deterministic TSPTW in which the objective functions are travel cost minimization, makespan minimization, and tour duration minimization. We introduce a new TSPTW variant with tour slack maximization, to provide solutions that are robust against delay up to a certain degree. We also address the robust TSPTW under travel time uncertainty which is defined by the knapsack-constrained uncertainty set $\mathcal{T}_K$ of Bartolini et al. (2021).

We develop a two-phase GVNS heuristic for solving all described variants. We show how to efficiently evaluate feasibility and profitability of a local search move by adapting the move evaluation approach that was originally introduced by Vidal et al. (2013), and we also extend the approach to efficiently check the robust feasibility in the RTSPTW($\mathcal{T}_K$). The effectiveness of the adapted approach is proven in the numerical experiments.

We compare the proposed heuristic to the state-of-the-art algorithms from the literature. We report the experimental results on different benchmark sets for all problem variants. Overall, our heuristic is successful in finding high-quality solutions within a reasonable amount of runtime for each problem variant, and is able to provide for the TSPTW-D, TSPTW-S, and the RTSPTW($\mathcal{T}_K$) new BKS for several instances.

Acknowledgments We thank the authors of Mladenović et al. (2013) and those of Kara et al. (2013) for providing additional information on their computational experiments that helped us to make fairer comparisons.

References

A. Agra, M. Christiansen, R. Figueiredo, L. M. Hvattum, M. Poss, and C. Requejo. The robust vehicle routing problem with time windows. *Computers & Operations Research*, 40:856–866, 2013.

K. Amghar, J.-F. Cordeau, and B. Gerdron. A general variable neighborhood search heuristic for the traveling salesman problem with time windows under completion time minimization. Technical report, CIRRELT-2019-29, 2019.

N. Ascheuer. *Hamiltonian path problem in the on-line optimization of flexible manufacturing system*. PhD thesis, Technical University Berlin, Germany, 1996.

N. Ascheuer, M. Fischetti, and M. Grötschel. Solving the asymmetric traveling salesman problem with time windows by branch-and-cut. *Mathematical Programming*, 90(3):475–506, 2001.

E. K. Baker. Technical note–an exact algorithm for the time-constrained traveling salesman problem. *Operations Research*, 31(5):938–945, 1983.

E. Balas. New classes of efficiently solvable generalized traveling salesman problems. *Annals of Operations Research*, 86(0):529–558, 1999.

E. Balas and N. Simonetti. Linear time dynamic-programming algorithms for new classes of restricted TSPs: A computational study. *INFORMS Journal on Computing*, 13(1):56–75, 2001.

E. Balas, N. Simonetti, and A. Vazacopoulos. Job shop scheduling with setup times, deadlines and precedence constraints. *Journal of Scheduling*, 11(4):253–262, 2008.

R. Baldacci, A. Mingozzi, and R. Roberti. New state-space relaxations for solving the traveling salesman problem with time windows. *INFORMS Journal on Computing*, 24(3):356–371, 2012.

E. Bartolini, D. Goeke, M. Schneider, and M. Ye. The robust TSPTW under knapsack-constrained travel time uncertainty. *Transportation Science*, 55(2):371–394, 2021.

D. Bertsimas and M. Sim. The price of robustness. *Operations Research*, 52:35–53, 2004.

N. Boland, M. Hewitt, D. M. Vu, and M. Savelsbergh. Solving the traveling salesman problem with time windows through dynamically generated time-expanded networks. In *International Conference on AI and OR Techniques in Constraint Programming for Combinatorial Optimization Problems*, pages 254–262. Springer, 2017.

S. Braaten, O. Gjønnes, L. M. Hvattum, and G. Tirado. Heuristics for the robust vehicle routing problem with time windows. *Expert Systems with Applications*, 77:136–147, 2017.

W. B. Carlton and J. W. Barnes. Solving the traveling salesman problem with time windows using tabu search. *IIE Transactions*, 28(8):617–629, 1996.

N. Christofides, A. Mingozzi, and P. Toth. State-space relaxation procedure for the computation of bounds to routing problems. *Networks*, 11(2):145–164, 1981.

R. F. Da Silva and S. Urrutia. A general VNS heuristic for the traveling salesman problem with time windows. *Discrete Optimization*, 7(4):203–211, 2010.

S. Dash, O. Günlük, A. Lodi, and A. Tramontani. A time bucket formulation for the traveling salesman problem with time windows. *INFORMS Journal of Computing*, 24(1):132–147, 2012.

Y. Dumas, J. Desrosiers, E. Gelinas, and M. M. Solomon. An optimal algorithm for the traveling salesman problem with time windows. *Operations Research*, 43(2):367–371, 1995.

D. Favarreto, E. Moretti, and P. Pellegrini. An ant colony system approach for variants of the traveling salesman problem with time windows. *Journal of Information and Operations Sciences*, 27(1):35–54, 2006.

F. Focacci, A. Lodi, and M. Milano. A hybrid exact algorithm for the TSPTW. *INFORMS Journal on Computing*, 14 (4):403–417, 2002.

M. Gendreau, A. Hertz, G. Laporte, and M. Stan. A generalized insertion heuristic for the traveling salesman problem with time windows. *Operations Research*, 46(3):330–335, 1998.

P. Hansen and N. Mladenović. Variable neighborhood search: Principles and applications. *European Journal of Operational Research*, 130(3):449–467, 2001.

C. Hu, J. Lu, X. Liu, and G. Zhang. Robust vehicle routing problem with hard time windows under demand and travel time uncertainty. *Computers & Operations Research*, 94:139–153, 2018.

S. Irnich. A unified modeling and solution framework for vehicle routing and local search-based metaheuristics. *INFORMS Journal on Computing*, 20(2):270–287, 2008.

I. Kara and T. Derya. Formulation for minimizing tour duration of the traveling salesman problem with time windows. *Procedia Economics and Finance*, 26:1026–1034, 2015.

I. Kara, O. N. Koc, F. Altiparmak, and B. Dengiz. New integer linear programming formulation for the traveling salesman problem with time windows: Minimizing tour duration with waiting times. *Optimization*, 62(10): 1309–1319, 2013.

K. Karabulut and M. F. Tasgetiren. A variable iterated greedy algorithm for the traveling salesman problem with time windows. *Information Sciences*, 279:383–395, 2014.

G. A. P. Kindervater and M. W. P. Savelsbergh. Vehicle routing: Handling edge exchanges. In *Local search in combinatorial optimization*, pages 337–360. Princeton University Press, 1997.

A. Langevin, M. Desrochers, J. Desrochers, S. Gélinas, and F. Soumis. A two-commodity flow formulation for the traveling salesman and the makespan problem with time windows. *Networks*, 23(7):631–640, 1993.

C. Lee, K. Lee, and S. Park. Robust vehicle routing problem with deadlines and travel time/demand uncertainty. *Journal of the Operational Research Society*, 63(9):1294–1306, 2012.

G. Lera-Romero, J. J. Miranda-Bront, and F. J. Soulignac. Dynamic programming for the time-dependent traveling salesman problem with time windows. *INFORMS Journal on Computing*, 34(6):3292–3308, 2022.

M. López-Ibáňez and C. Blum. Beam-ACO for the traveling salesman problem with time windows. *Computers & Operations Research*, 37(9):1570–1583, 2010.

C. B. Manuel López-Ibáňez, J. W. Ohlmann, and B. W. Thomas. The traveling salesman problems with time windows: Adapting algorithms from travel-time to makespan optimization. *Applied Soft Computing*, 13(9):3806–3815, 2013.

N. Mladenović and P. Hansen. Variable neighborhood search. *Computers & Operations Research*, 24(11):1097–1100, 1997.

N. Mladenović, R. Todosijević, and D. Urošević. An efficient general variable neighborhood search for large travelling salesman problem with time windows. *Yugoslav Journal of Operations Research*, 23(1):19–30, 2013.

P. Munari, A. Moreno, J. D. L. Vega, D. Alem, J. Gondzio, and R. Morabito. The robust vehicle routing problem with time windows: Compact formulation and branch-price-and-cut method. *Transportation Science*, 53(4): 1043–1066, 2019.

J. W. Ohlmann and B. W. Thomas. A compressed-annealing heuristic for the traveling salesman problem with time windows. *INFORMS Journal on Computing*, 19(1):80–90, 2007.

G. Pesant, M. Gendreau, J.-Y. Potvin, and J.-M. Rousseau. An exact constraint logic programming algorithm for the traveling salesman problem with time windows. *Transportation Science*, 32(1):12–29, 1998.

J.-Y. Potvin and S. Bengio. The vehicle routing problem with time windows – Part II: Genetic search. *INFORMS Journal on Computing*, 8(2):165–172, 1996.

C. Pralet. Iterated maximum large neighborhood search for the traveling salesman problem with time windows and its time-dependent version. *Computers & Operations Research*, 150:106078, 2023.

M. W. Savelsbergh. The vehicle routing problem with time windows: Minimizing route duration. *ORSA Journal on Computing*, 4(2):146–154, 1992.

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

C. Tilk and S. Irnich. Dynamic programming for the minimum tour duration problem. *Transportation Science*, 51(2): 549–565, 2017.

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

Z. Zhang, Y. Zhang, and R. Baldacci. Effective exact solution framework for routing optimization with time windows and travel time uncertainty. https://optimization-online.org/?p=18939, 2022.

Appendix A Algorithm for testing the robust feasibility with respect to $\mathcal{T}_K$

This section briefly discusses the $O(n^2)$ algorithm proposed by Bartolini et al. (2021) for efficiently testing the robust feasibility of a path $P = (i_0, ..., i_p)$ with respect to the uncertainty set $\mathcal{T}_K$. The algorithm can be described as follows. Let τ_h^k be the latest departure time from a node i_h of P with respect to a travel time vector $\mathbf{t} \in \mathcal{T}_K$ modeling a scenario where P starts to accumulate delays from a node i_k and each arc (i, j) on the subpath $(i_k, ..., i_h)$ (with $0 \leq k < h$) incurs its maximum allowed delay, i.e., $t_{ij} = t_{ij}^0 + \min\{\delta_{ij}, \Delta - \delta(P, k, i)\}$, where $\delta(P, k, i) = \min\{\Delta, \sum_{r=k}^{i-1} \delta_{i_r\, i_{r+1}}\}$ denotes the maximum cumulated delay starting from node i_k until i under the control of Δ. More briefly, τ_h^k is the departure time from i_h of P when P starts to accumulate as much delay as possible from a node i_k with $0 \leq k < h$. To simplify the notation, we use $r(P, k, i)$, called the maximum residual delay of P starting from node i onward, to represent the value of the term $\Delta - \delta(P, k, i)$. The entire set of departure times τ_h^k can be calculated using the following recursion:

$$\tau_h^k = \left\{ e_{i_h}, \tau_{h-1}^k + t_{i_{h-1}\, i_p}^0 + \min\left\{ r(P, k, h-1), \delta_{i_{h-1}\, i_h} \right\} \right\}, \quad k = h-1, \ldots, 0, \tag{13}$$

with $\tau_k^k = \tau_k^D$ for all $0 \leq k < h$ as the initialization. The worst-case departure time from i_h of P, denoted as $\hat{\tau}_h$, is then defined as:

$$\hat{\tau}_h = \max_{k=0,\ldots,h-1} \tau_h^k. \tag{14}$$

A tour P is robust feasible if and only if $\hat{\tau}_h \leq l_{i_h}$ for each $1 \leq h \leq p$. In this case, to test the robust feasibility of a tour P with n nodes, it is enough only to consider n scenarios identified by all possible nodes from which P starts to accumulate the maximum delay.

Appendix B Detailed computational results

In this section, we report for each variant detailed results obtained by our GVNS-S and GVNS-Q on the respective benchmark instance sets. In the tables, besides the metrics that were already introduced in the main text, $best$ denotes the best solution value obtained by the heuristic, avg and avg_{sd} denote the average value and the standard deviation of the solutions over all runs. Note that the results for *Dumas*, *GDE*, *OT*, and *DaSilva* instances are aggregated. Each row corresponds to the average results over five instances in the same class with the same number of nodes and the same maximum time window width.

B.1 TSPTW-C

Tables 7-16 and tables 10-14 report, for all instance sets individually, the results obtained by GVNS-S and GVNS-Q with $\eta_{rep} = 1$ and $\eta_{rep} = 10$ for the TSPTW-C, respectively. Note that the results of MTU and KT are directly taken from the papers of Mladenović et al. (2013) and Karabulut and Tasgetiren (2014). For single-thread computations, their CPUs are approximately 0.7 and 0.8 times faster than ours (see processor information in the main text). The runtimes in the tables are scaled.

Table 7: TSPTW-C: Results for the *Dumas* instances.

		MTU					KT					GVNS-S					GVNS-Q				
Inst.	BKS	$best$	avg	avg_{sd}	t_{avg}	t_{sd}	$best$	avg	avg_{sd}	t_{avg}	t_{sd}	$best$	avg	avg_{sd}	t_{avg}	t_{sd}	$best$	avg	avg_{sd}	t_{avg}	t_{sd}
n20w20	361.2	361.2	361.2	0.0	0.0	0.0	361.2	361.2	0.0	0.0	0.0	361.2	361.2	0.0	0.0	0.0	361.2	361.2	0.0	0.0	0.0
n20w40	316.0	316.0	316.0	0.0	0.0	0.0	316.0	316.0	0.0	0.0	0.0	316.0	316.1	0.5	0.0	0.0	316.0	316.0	0.0	0.0	0.0
n20w60	309.8	309.8	309.8	0.0	0.0	0.0	309.8	309.8	0.0	0.0	0.0	309.8	309.9	0.5	0.0	0.0	309.8	309.8	0.0	0.0	0.1
n20w80	311.0	311.0	311.0	0.0	0.0	0.0	311.0	311.0	0.0	0.0	0.0	311.0	311.0	0.0	0.0	0.0	311.0	311.0	0.0	0.0	0.1
n20w100	275.2	275.2	275.2	0.0	0.0	0.0	275.2	275.2	0.0	0.0	0.0	275.2	277.0	3.7	0.0	0.0	275.2	275.2	0.0	0.0	0.0
n40w20	486.6	486.6	486.6	0.0	0.0	0.0	486.6	486.6	0.0	0.0	0.0	486.6	486.6	0.0	0.0	0.0	486.6	486.6	0.0	0.0	0.1
n40w40	461.0	461.0	461.0	0.0	0.0	0.0	461.0	461.0	0.0	0.0	0.0	461.0	461.0	0.0	0.0	0.0	461.0	461.0	0.0	0.0	0.1
n40w60	416.4	416.4	416.4	0.0	0.0	0.0	416.4	416.4	0.0	0.0	0.0	416.4	416.4	0.0	0.0	0.0	416.4	416.4	0.0	0.1	0.2
n40w80	399.8	399.8	399.8	0.0	0.8	0.4	399.8	399.8	0.0	0.0	0.0	399.8	400.0	0.3	0.0	0.1	399.8	400.0	0.3	0.1	0.3
n40w100	377.0	377.0	377.0	0.0	0.0	0.0	377.0	377.0	0.0	0.0	0.0	377.0	377.1	0.2	0.0	0.0	377.0	377.1	0.2	0.1	0.3
n60w20	581.6	581.6	581.6	0.0	0.0	0.0	581.6	581.6	0.0	0.0	0.0	581.6	581.6	0.0	0.0	0.0	581.6	581.6	0.0	0.1	0.3
n60w40	590.2	590.2	590.6	1.9	0.1	0.0	590.2	590.2	0.0	0.0	0.0	590.2	590.2	0.0	0.0	0.1	590.2	590.2	0.0	0.1	0.3
n60w60	560.0	560.0	560.0	0.0	0.0	0.0	560.0	560.0	0.0	0.0	0.0	560.0	560.0	0.0	0.0	0.0	560.0	560.0	0.0	0.2	0.4
n60w80	508.0	508.0	508.0	0.0	0.1	0.1	508.0	508.0	0.0	0.0	0.0	508.0	508.1	0.1	0.1	0.2	508.0	508.1	0.1	0.2	0.4
n60w100	514.8	514.8	514.8	0.0	0.1	0.1	514.8	514.8	0.0	0.1	0.2	514.8	514.8	0.0	0.1	0.2	514.8	514.8	0.0	0.2	0.4
n80w20	676.6	676.6	676.6	0.0	0.0	0.0	676.6	676.6	0.0	0.0	0.0	676.6	676.6	0.0	0.0	0.1	676.6	676.6	0.0	0.1	0.3
n80w40	630.0	630.0	630.0	0.0	0.1	0.0	630.0	630.0	0.0	0.2	0.3	630.0	630.5	0.6	0.1	0.2	630.0	630.0	0.0	0.3	0.4
n80w60	606.4	606.4	606.7	1.2	0.7	0.4	606.4	606.4	0.0	0.1	0.2	606.4	606.4	0.1	0.1	0.3	606.4	606.5	0.1	0.3	0.5
n80w80	593.8	593.8	593.9	0.2	0.4	0.5	593.8	593.8	0.0	1.2	1.6	593.8	594.0	0.5	0.1	0.3	593.8	594.0	0.4	0.5	0.5
n100w20	757.6	757.6	757.6	0.0	0.0	0.0	757.6	757.6	0.0	0.2	0.5	757.6	757.6	0.0	0.1	0.3	757.6	757.6	0.0	0.2	0.4
n100w40	701.8	701.8	701.8	0.0	0.1	0.0	701.8	701.8	0.0	0.0	0.1	701.8	701.8	0.0	0.1	0.4	701.8	701.8	0.0	0.4	0.5
n100w60	696.6	696.6	696.6	0.0	0.1	0.1	696.6	696.6	0.0	0.1	0.2	696.6	696.6	0.0	0.2	0.4	696.6	696.6	0.0	0.6	0.5
n150w20	868.4	868.4	868.4	0.0	0.1	0.1	868.4	868.4	0.0	0.5	0.8	868.4	868.4	0.0	0.3	0.4	868.4	868.4	0.0	0.7	0.5
n150w40	834.8	834.8	834.8	0.0	0.3	0.3	834.8	834.8	0.0	1.3	1.6	834.8	834.8	0.0	0.5	0.5	834.8	834.8	0.0	1.3	0.4
n150w60	818.6	818.6	818.6	0.0	1.3	0.5	818.6	818.6	0.0	4.8	4.8	818.6	818.7	0.1	1.1	0.6	818.6	818.6	0.0	1.8	0.5
n200w20	1009.0	1009.0	1009.1	0.2	1.4	0.8	1009.0	1009.0	0.1	4.7	4.4	1009.0	1009.0	0.0	0.7	0.5	1009.0	1009.0	0.0	1.6	0.5
n200w40	984.2	984.2	984.2	0.2	3.6	1.0	984.2	984.3	0.3	7.4	7.6	984.2	984.3	0.1	1.8	0.6	984.2	984.2	0.0	3.2	0.6

Table 8: TSPTW-C: Results for the *GDE* instances.

		MTU					KT					GVNS-S					GVNS-Q				
Inst.	BKS	$best$	avg	avg_{sd}	t_{avg}	t_{sd}	$best$	avg	avg_{sd}	t_{avg}	t_{sd}	$best$	avg	avg_{sd}	t_{avg}	t_{sd}	$best$	avg	avg_{sd}	t_{avg}	t_{sd}
n20w120	265.6	265.6	265.6	0.0	0.0	0.0	265.6	265.6	0.0	0.0	0.0	265.6	265.7	0.2	0.0	0.0	265.6	265.6	0.0	0.0	0.0
n20w140	232.8	232.8	232.8	0.0	0.0	0.0	232.8	232.8	0.0	0.0	0.0	232.8	233.5	0.9	0.0	0.0	232.8	232.8	0.0	0.0	0.0
n20w160	218.2	218.2	218.2	0.0	0.0	0.0	218.2	218.2	0.0	0.0	0.0	218.2	218.3	0.1	0.0	0.0	218.2	218.2	0.0	0.0	0.1
n20w180	236.6	236.6	236.6	0.0	0.0	0.0	236.6	236.6	0.0	0.0	0.0	236.6	236.6	0.0	0.0	0.0	236.6	236.7	0.1	0.0	0.1
n20w200	241.0	241.0	241.0	0.0	0.0	0.0	241.0	241.0	0.0	0.0	0.0	241.0	241.2	0.6	0.0	0.0	241.0	241.3	1.2	0.0	0.1
n40w120	377.8	377.8	377.8	0.0	0.0	0.0	377.8	377.8	0.0	0.0	0.0	377.8	378.0	0.3	0.0	0.0	377.8	377.8	0.0	0.1	0.3
n40w140	364.4	364.4	364.4	0.0	0.0	0.0	364.4	364.4	0.0	0.0	0.0	364.4	364.4	0.0	0.0	0.1	364.4	364.4	0.0	0.1	0.3
n40w160	326.8	326.8	326.8	0.0	0.0	0.0	326.8	326.8	0.0	0.0	0.0	326.8	326.9	0.1	0.0	0.1	326.8	326.9	0.1	0.1	0.4
n40w180	330.4	330.4	330.5	0.9	1.5	0.8	330.4	330.4	0.0	0.0	0.0	330.4	330.5	0.3	0.0	0.1	330.4	330.5	0.3	0.2	0.4
n40w200	313.8	313.8	313.8	0.3	2.5	0.8	313.8	313.8	0.0	0.0	0.0	313.8	314.3	0.9	0.0	0.0	313.8	314.3	0.8	0.1	0.3
n60w120	451.0	451.0	451.0	0.0	0.2	0.1	451.0	451.0	0.0	0.1	0.2	451.0	451.1	0.2	0.1	0.3	451.0	451.0	0.1	0.3	0.4
n60w140	452.0	452.0	452.0	0.0	0.1	0.0	452.0	452.0	0.0	0.2	0.4	452.0	452.1	0.3	0.1	0.2	452.0	452.0	0.0	0.3	0.5
n60w160	464.0	464.0	464.6	0.2	0.0	0.0	464.0	464.0	0.0	0.5	0.8	464.0	464.0	0.1	0.1	0.3	464.0	464.0	0.1	0.3	0.5
n60w180	421.2	421.2	421.2	0.0	0.3	0.1	421.2	421.2	0.0	0.1	0.2	421.2	421.4	0.5	0.1	0.3	421.2	421.4	0.5	0.4	0.5
n60w200	427.4	427.4	427.4	0.0	0.2	0.1	427.4	427.4	0.0	0.0	0.2	427.4	429.1	2.2	0.1	0.3	427.4	427.6	0.7	0.4	0.5
n80w100	578.6	578.6	578.6	0.0	0.5	0.3	578.6	578.6	0.0	0.6	0.8	578.6	578.9	0.9	0.1	0.3	578.6	578.8	0.6	0.4	0.5
n80w120	541.4	541.4	541.4	0.1	0.9	0.5	541.4	541.4	0.0	2.2	2.9	541.4	541.9	0.7	0.2	0.4	541.4	541.6	0.5	0.8	0.5
n80w140	506.0	506.0	506.3	0.6	1.0	0.3	506.0	506.0	0.1	1.9	2.7	506.0	506.3	0.6	0.3	0.4	506.0	506.3	0.8	0.8	0.4
n80w160	504.8	504.8	505.1	1.2	1.0	0.8	504.8	504.8	0.0	0.4	0.7	504.8	505.0	0.4	0.2	0.4	504.8	504.9	0.2	0.7	0.5
n80w180	500.6	500.6	500.9	2.3	2.3	0.7	500.6	500.6	0.0	2.9	3.7	500.6	502.4	1.1	0.3	0.4	500.6	501.4	1.6	0.8	0.5
n80w200	481.4	481.4	481.4	0.0	0.3	0.1	481.4	481.7	0.2	2.5	4.4	481.8	481.8	0.0	0.3	0.5	481.8	481.8	0.0	0.9	0.5
n100w80	666.4	666.4	666.4	0.0	0.3	0.1	666.4	666.4	0.0	0.2	0.3	666.4	666.5	0.2	0.3	0.4	666.4	666.4	0.0	1.0	0.5
n100w100	640.6	640.6	641.0	1.5	1.9	0.8	640.6	640.6	0.1	1.3	2.7	640.6	640.7	0.3	0.4	0.5	640.6	641.0	0.2	1.2	0.4
n100w120	597.2	597.2	597.5	0.5	3.8	1.2	597.2	597.2	0.1	6.3	6.2	597.2	598.1	1.1	0.5	0.5	597.2	598.1	0.6	1.7	0.7
n100w140	548.4	548.4	548.4	0.0	0.1	0.0	548.4	548.4	0.0	1.5	2.3	548.4	548.5	0.1	0.5	0.5	548.4	548.4	0.1	1.6	0.6
n100w160	555.0	555.0	555.0	0.0	0.8	0.2	555.0	555.2	0.1	1.0	1.1	555.0	555.6	0.6	0.5	0.5	555.0	555.1	0.3	1.4	0.6

Table 9: TSPTW-C: Results for the *OT* instances.

		MTU					KT					GVNS-S					GVNS-Q				
Inst.	BKS	$best$	avg	avg_{sd}	t_{avg}	t_{sd}	$best$	avg	avg_{sd}	t_{avg}	t_{sd}	$best$	avg	avg_{sd}	t_{avg}	t_{sd}	$best$	avg	avg_{sd}	t_{avg}	t_{sd}
n150w120	722.0	722.0	722.1	0.6	7.7	2.5	722.0	722.5	0.8	22.8	27.5	722.8	723.3	0.6	2.6	1.0	722.0	723.0	1.1	5.1	1.5
n150w140	693.8	693.8	693.9	0.4	12.8	3.0	693.8	694.6	1.0	21.2	27.0	695.4	697.1	1.7	2.5	0.9	694.2	694.6	0.8	5.6	1.8
n150w160	671.0	671.0	672.6	2.9	9.2	3.4	671.0	672.0	1.5	17.8	22.9	671.8	672.6	1.4	3.1	1.0	671.2	671.9	1.4	5.9	1.6
n200w120	803.6	803.6	803.9	0.3	7.9	2.5	803.6	804.8	1.8	37.9	37.9	804.0	804.5	0.9	8.6	1.9	803.8	803.9	0.2	13.9	3.8
n200w140	798.0	798.0	798.7	1.6	16.9	4.0	798.0	802.0	4.6	38.0	38.4	799.8	801.0	1.8	9.1	2.6	798.4	799.8	2.3	15.8	5.0

Table 10: TSPTW-C: Results for the *OT* instances ($\eta_{rep} = 10$).

		MTU					KT					GVNS-S					GVNS-Q				
Inst.	BKS	$best$	avg	avg_{sd}	t_{avg}	t_{sd}	$best$	avg	avg_{sd}	t_{avg}	t_{sd}	$best$	avg	avg_{sd}	t_{avg}	t_{sd}	$best$	avg	avg_{sd}	t_{avg}	t_{sd}
n150w120	722.0	722.0	722.1	0.6	7.7	2.5	722.0	722.5	0.8	22.8	27.5	722.0	722.9	0.9	9.4	2.2	722.0	722.2	0.4	19.2	5.4
n150w140	693.8	693.8	693.9	0.4	12.8	3.0	693.8	694.6	1.0	21.2	27.0	694.4	695.3	0.8	9.3	2.4	694.0	694.3	0.5	20.5	4.5
n150w160	671.0	671.0	672.6	2.9	9.2	3.4	671.0	672.0	1.5	17.8	22.9	671.6	671.8	0.6	10.2	2.4	671.0	671.4	0.5	19.5	4.4
n200w120	803.6	803.6	803.9	0.3	7.9	2.5	803.6	804.8	1.8	37.9	37.9	803.6	803.8	0.4	28.2	3.1	803.6	803.8	0.3	31.0	0.8
n200w140	798.0	798.0	798.7	1.6	16.9	4.0	798.0	802.0	4.6	38.0	38.4	798.2	799.5	1.9	27.9	3.8	798.0	799.4	2.3	31.4	0.8

Table 11: TSPTW-C: Results for the *DaSilva* instances.

		MTU					KT					GVNS-S					GVNS-Q				
Inst.	BKS	$best$	avg	avg_{sd}	t_{avg}	t_{sd}	$best$	avg	avg_{sd}	t_{avg}	t_{sd}	$best$	avg	avg_{sd}	t_{avg}	t_{sd}	$best$	avg	avg_{sd}	t_{avg}	t_{sd}
n200w100	10019.6	10019.6	10020.4	0.8	0.0	0.0	10019.6	10019.6	0.0	0.2	0.2	10019.6	10019.6	0.0	0.3	0.4	10019.6	10019.6	0.0	0.6	0.4
n200w200	9252.0	9252.0	9254.2	11.4	0.1	0.0	9252.0	9252.0	0.0	5.1	5.4	9252.0	9254.7	6.8	0.3	0.4	9252.0	9254.7	6.8	0.5	0.5
n200w300	8022.8	8022.8	8023.1	0.3	6.9	2.3	8022.8	8024.3	1.8	34.7	31.4	8031.4	8038.1	9.3	0.5	0.5	8025.8	8025.8	0.1	0.8	0.5
n200w400	7062.4	7062.4	7072.4	19.3	8.1	2.5	7062.4	7069.3	3.9	51.9	44.4	7079.8	7099.2	25.4	0.8	0.5	7066.6	7066.8	0.4	1.5	0.5
n200w500	6466.2	6466.2	6472.7	11.4	9.5	2.9	6466.2	6472.4	4.7	54.8	45.3	6474.6	6483.0	9.4	1.5	0.6	6466.2	6466.5	0.9	3.3	1.2
n250w100	12633.0	12633.0	12633.0	0.0	0.0	0.0	12633.0	12633.0	0.0	0.4	0.6	12633.0	12633.0	0.1	0.5	0.5	12633.0	12633.0	0.0	0.9	0.3
n250w200	11310.4	11310.4	11314.0	5.0	0.2	0.1	11310.4	11311.7	0.1	19.2	16.4	11310.8	11311.9	2.9	0.6	0.5	11310.8	11311.8	2.7	0.9	0.3
n250w300	10230.4	10230.4	10231.0	3.4	2.5	1.3	10230.4	10233.5	1.8	39.0	27.2	10237.0	10237.8	2.8	1.0	0.5	10233.6	10233.9	1.3	1.5	0.5
n250w400	8896.2	8896.2	8897.9	5.3	25.8	5.3	8896.2	8903.6	4.6	75.6	46.1	8911.6	8928.6	17.1	1.9	0.9	8897.6	8899.5	4.1	2.7	0.9
n250w500	8066.4	8069.8	8083.5	13.2	28.9	5.5	8066.4	8088.5	10.6	98.4	54.6	8121.2	8127.9	10.0	2.2	0.8	8085.4	8087.9	4.0	4.4	1.3
n300w100	15041.2	15041.2	15041.2	0.0	0.0	0.0	15041.2	15041.2	0.0	0.5	0.6	15041.2	15041.2	0.0	1.0	0.3	15041.2	15041.2	0.0	1.8	0.4
n300w200	13845.0	13851.4	13857.6	14.9	0.4	0.1	13851.4	13852.0	0.0	19.1	12.7	13852.8	13857.3	7.2	1.1	0.5	13852.8	13857.0	6.7	1.6	0.6
n300w300	11475.2	11477.2	11478.8	2.7	7.5	2.3	11477.2	11481.0	0.2	62.2	37.7	11488.2	11499.8	15.4	3.4	2.0	11487.0	11492.3	9.0	4.6	2.1
n300w400	10402.8	10402.8	10419.6	25.5	20.6	4.1	10402.8	10413.3	5.9	81.2	42.3	10450.0	10465.2	32.2	4.2	2.3	10420.6	10424.3	9.4	5.5	2.5
n300w500	9842.2	9842.2	9849.2	7.9	33.9	4.3	9842.2	9872.5	13.2	125.7	51.2	9922.6	9939.0	21.8	3.8	1.3	9854.6	9862.3	10.1	7.1	2.7
n350w100	17494.0	17494.0	17494.0	0.0	0.0	0.0	17494.0	17494.0	0.0	4.1	4.4	17494.0	17494.0	0.0	1.5	0.3	17494.0	17494.0	0.0	2.4	0.4
n350w200	15672.0	15672.0	15672.0	0.0	1.2	0.6	15672.0	15674.6	1.2	50.7	34.3	15672.0	15672.9	1.7	2.3	1.3	15672.0	15672.6	1.0	3.0	1.4
n350w300	13648.8	13648.8	13660.8	17.8	9.0	2.6	13648.8	13658.8	3.7	94.6	51.3	13661.8	13671.2	15.7	5.9	2.9	13656.6	13661.6	8.5	7.1	3.0
n350w400	12083.2	12083.2	12090.6	9.5	32.1	5.4	12083.2	12143.4	18.0	150.5	43.0	12161.6	12179.5	15.7	5.3	1.9	12115.6	12123.8	9.0	7.7	2.8
n350w500	11347.8	11347.8	11360.6	17.7	40.4	5.1	11348.0	11399.4	19.9	138.6	47.7	11399.2	11437.6	30.9	9.2	4.3	11360.4	11365.6	8.6	14.6	6.0
n400w100	19454.8	19454.8	19454.8	0.0	0.0	0.0	19454.8	19454.8	0.0	3.8	3.9	19454.8	19454.8	0.0	2.2	0.4	19454.8	19454.8	0.0	3.2	0.4
n400w200	18439.8	18439.8	18442.6	5.1	1.2	0.3	18439.8	18440.5	0.0	35.3	21.4	18439.8	18440.4	1.4	3.1	2.0	18439.8	18440.4	1.4	4.3	2.2
n400w300	15872.8	15872.8	15875.8	8.5	19.7	3.3	15872.8	15891.6	5.9	104.5	41.6	15902.2	15913.9	17.8	6.1	4.0	15896.2	15906.4	15.6	7.1	4.2
n400w400	14079.4	14079.4	14112.0	24.4	37.6	4.7	14086.6	14184.9	32.4	166.2	26.5	14180.4	14202.7	35.9	14.8	11.1	14091.4	14091.7	1.0	16.0	10.5
n400w500	12716.6	12716.6	12755.8	26.9	53.1	5.3	12724.6	12802.0	24.3	172.6	39.8	12809.4	12838.1	25.9	15.8	8.3	12745.2	12749.1	6.0	21.3	10.0

Table 12: TSPTW-C: Results for the *DaSilva* instances ($\eta_{rep} = 10$).

		MTU					KT					GVNS-S					GVNS-Q				
Inst.	BKS	$best$	avg	avg_{sd}	t_{avg}	t_{sd}	$best$	avg	avg_{sd}	t_{avg}	t_{sd}	$best$	avg	avg_{sd}	t_{avg}	t_{sd}	$best$	avg	avg_{sd}	t_{avg}	t_{sd}
n200w100	10019.6	10019.6	10020.4	0.8	0.0	0.0	10019.6	10019.6	0.0	0.2	0.2	10019.6	10019.6	0.0	1.3	0.3	10019.6	10019.6	0.0	2.9	0.4
n200w200	9252.0	9252.0	9254.2	11.4	0.1	0.0	9252.0	9252.0	0.0	5.1	5.4	9252.0	9252.2	0.4	1.1	0.4	9252.0	9252.2	0.4	2.3	0.4
n200w300	8022.8	8022.8	8023.1	0.3	6.9	2.3	8022.8	8024.3	1.8	34.7	31.4	8025.2	8028.2	5.2	2.1	0.8	8023.2	8023.2	0.0	4.5	0.6
n200w400	7062.4	7062.4	7072.4	19.3	8.1	2.5	7062.4	7069.3	3.9	51.9	44.4	7073.6	7081.5	10.5	3.1	1.0	7063.0	7064.3	2.4	7.6	1.2
n200w500	6466.2	6466.2	6472.7	11.4	9.5	2.9	6466.2	6472.4	4.7	54.8	45.3	6469.8	6473.6	5.4	6.1	1.8	6466.2	6466.2	0.1	12.9	2.0
n250w100	12633.0	12633.0	12633.0	0.0	0.0	0.0	12633.0	12633.0	0.0	0.4	0.6	12633.0	12633.0	0.0	2.4	0.4	12633.0	12633.0	0.0	4.9	0.4
n250w200	11310.4	11310.4	11314.0	5.0	0.2	0.1	11310.4	11311.7	0.1	19.2	16.4	11310.4	11311.2	1.5	2.4	0.6	11310.4	11311.2	1.5	4.2	0.6
n250w300	10230.4	10230.4	10231.0	3.4	2.5	1.3	10230.4	10233.5	1.8	39.0	27.2	10237.0	10237.7	2.7	3.5	1.1	10232.6	10232.9	1.1	6.1	0.7
n250w400	8896.2	8896.2	8897.9	5.3	25.8	5.3	8896.2	8903.6	4.6	75.6	46.1	8902.4	8910.5	8.9	6.1	2.0	8896.4	8897.4	1.4	9.1	2.0
n250w500	8066.4	8069.8	8083.5	13.2	28.9	5.5	8066.4	8088.5	10.6	98.4	54.6	8100.6	8108.9	9.6	9.0	2.5	8072.0	8072.7	2.0	17.6	3.7
n300w100	15041.2	15041.2	15041.2	0.0	0.0	0.0	15041.2	15041.2	0.0	0.5	0.6	15041.2	15041.2	0.0	4.7	0.4	15041.2	15041.2	0.0	8.9	0.5
n300w200	13845.0	13851.4	13857.6	14.9	0.4	0.1	13851.4	13852.0	0.0	19.1	12.7	13851.4	13852.9	1.6	4.4	0.9	13851.4	13853.5	2.3	7.0	1.1
n300w300	11475.2	11477.2	11478.8	2.7	7.5	2.3	11477.2	11481.0	0.2	62.2	37.7	11478.0	11482.5	6.3	8.9	3.1	11477.2	11479.7	3.9	13.0	3.4
n300w400	10402.8	10402.8	10419.6	25.5	20.6	4.1	10402.8	10413.3	5.9	81.2	42.3	10428.0	10436.5	11.3	11.0	4.2	10410.0	10416.9	7.5	18.8	5.0
n300w500	9842.2	9842.2	9849.2	7.9	33.9	4.3	9842.2	9872.5	13.2	125.7	51.2	9874.8	9896.3	22.2	14.8	3.7	9844.0	9851.2	7.5	27.5	3.1
n350w100	17494.0	17494.0	17494.0	0.0	0.0	0.0	17494.0	17494.0	0.0	4.1	4.4	17494.0	17494.0	0.0	7.2	0.5	17494.0	17494.0	0.0	11.5	0.5
n350w200	15672.0	15672.0	15672.0	0.0	1.2	0.6	15672.0	15674.6	1.2	50.7	34.3	15672.0	15672.1	0.2	6.7	1.9	15672.0	15672.1	0.2	11.4	2.5
n350w300	13648.8	13648.8	13660.8	17.8	9.0	2.6	13648.8	13658.8	3.7	94.6	51.3	13652.2	13655.3	4.9	14.6	4.7	13650.2	13652.9	4.3	20.5	5.2
n350w400	12083.2	12083.2	12090.6	9.5	32.1	5.4	12083.2	12143.4	18.0	150.5	43.0	12113.0	12134.3	25.5	17.3	4.9	12092.2	12106.9	12.4	23.4	4.8
n350w500	11347.8	11347.8	11360.6	17.7	40.4	5.1	11348.0	11399.4	19.9	138.6	47.7	11387.8	11399.0	19.6	24.2	5.8	11359.8	11362.5	5.8	31.0	4.2
n400w100	19454.8	19454.8	19454.8	0.0	0.0	0.0	19454.8	19454.8	0.0	3.8	3.9	19454.8	19454.8	0.0	11.0	0.4	19454.8	19454.8	0.0	16.1	0.5
n400w200	18439.8	18439.8	18442.6	5.1	1.2	0.3	18439.8	18440.5	0.0	35.3	21.4	18439.8	18440.3	1.0	10.3	3.5	18439.8	18440.3	1.0	14.5	3.9
n400w300	15872.8	15872.8	15875.8	8.5	19.7	3.3	15872.8	15891.6	5.9	104.5	41.6	15893.8	15902.1	16.4	15.8	5.9	15880.4	15891.6	12.5	22.6	6.8
n400w400	14079.4	14079.4	14112.0	24.4	37.6	4.7	14086.6	14184.9	32.4	166.2	26.5	14133.6	14149.7	18.2	29.3	11.4	14083.0	14083.6	1.7	36.9	10.8
n400w500	12716.6	12716.6	12755.8	26.9	53.1	5.3	12724.6	12802.0	24.3	172.6	39.8	12779.2	12787.9	14.7	34.2	9.2	12732.8	12740.9	8.2	39.2	8.3

Table 13: TSPTW-C: Results for the *AFG* instance.

		MTU					GVNS-S					GVNS-Q				
Inst.	BKS	$best$	RPD_a	RPD_{sd}	t_{avg}	t_{sd}	$best$	RPD_a	RPD_{sd}	t_{avg}	t_{sd}	$best$	RPD_a	RPD_{sd}	t_{avg}	t_{sd}
rbg010a	671	671	0.00	0.00	0.0	0.0	671	0.00	0.00	0.0	0.0	671	0.00	0.00	0.0	0.0
rbg016a	938	938	0.00	0.00	0.0	0.0	938	0.00	0.00	0.0	0.0	938	0.00	0.00	0.0	0.0
rbg016b	1304	1304	0.00	0.00	0.0	0.0	1304	0.01	0.03	0.0	0.0	1304	0.00	0.00	0.0	0.0
rbg017.2	852	852	0.00	0.00	0.0	0.0	852	0.00	0.00	0.0	0.0	852	0.00	0.00	0.0	0.0
rbg017	893	893	0.00	0.00	0.0	0.0	893	0.00	0.00	0.0	0.0	893	0.00	0.00	0.0	0.0
rbg017a	4296	4296	0.00	0.00	0.0	0.0	4296	0.00	0.00	0.1	0.2	4296	0.00	0.00	0.0	0.0
rbg019a	1262	1262	0.00	0.00	0.0	0.0	1262	0.00	0.00	0.0	0.0	1262	0.00	0.00	0.0	0.0
rbg019b	1866	1866	0.00	0.00	0.0	0.0	1866	0.00	0.00	0.0	0.0	1866	0.00	0.00	0.0	0.0
rbg019c	4536	4536	0.00	0.00	0.0	0.0	4536	0.00	0.00	0.0	0.0	4536	0.00	0.00	0.0	0.0
rbg019d	1356	1356	0.00	0.00	0.0	0.0	1356	0.00	0.00	0.0	0.0	1356	0.00	0.00	0.0	0.0
rbg020a	4689	4689	0.00	0.00	0.0	0.0	4689	0.00	0.00	0.0	0.0	4689	0.00	0.00	0.1	0.2
rbg021	4536	4536	0.00	0.00	0.0	0.0	4528	0.03	0.09	0.0	0.0	4528	0.02	0.06	0.1	0.2
rbg021.2	4528	4528	0.00	0.00	0.0	0.0	4528	0.00	0.00	0.0	0.0	4528	0.00	0.00	0.1	0.2
rbg021.3	4528	4528	0.00	0.00	0.0	0.0	4525	0.00	0.00	0.0	0.0	4525	0.00	0.00	0.1	0.2
rbg021.4	4525	4525	0.00	0.00	0.0	0.0	4515	0.02	0.05	0.0	0.0	4515	0.02	0.04	0.1	0.2
rbg021.5	4515	4515	0.00	0.00	0.0	0.0	4480	0.02	0.07	0.0	0.0	4480	0.03	0.08	0.1	0.2
rbg021.6	4480	4480	0.00	0.00	0.0	0.0	4479	0.00	0.00	0.1	0.2	4479	0.00	0.00	0.0	0.0
rbg021.7	4479	4479	0.00	0.00	0.0	0.0	4478	0.01	0.01	0.0	0.0	4478	0.00	0.01	0.0	0.0
rbg021.8	4478	4478	0.00	0.00	0.0	0.0	4478	0.00	0.00	0.0	0.0	4478	0.00	0.00	0.0	0.0
rbg021.9	4478	4478	0.00	0.00	0.0	0.0	4536	0.00	0.00	0.0	0.0	4536	0.00	0.00	0.0	0.0
rbg027a	5091	5091	0.00	0.00	0.0	0.0	5091	0.00	0.00	0.0	0.0	5091	0.00	0.00	0.0	0.0
rbg031a	1863	1863	0.00	0.00	0.0	0.0	1863	0.03	0.11	0.0	0.0	1863	0.00	0.00	0.1	0.2
rbg033a	2069	2069	0.00	0.00	0.0	0.0	2069	0.00	0.00	0.0	0.0	2069	0.00	0.00	0.1	0.2
rbg034a	2222	2222	0.00	0.00	0.0	0.0	2222	0.00	0.00	0.0	0.0	2222	0.00	0.00	0.1	0.2
rbg035a.2	2056	2056	0.00	0.00	0.0	0.0	2056	0.01	0.05	0.1	0.2	2056	0.00	0.00	0.1	0.3
rbg035a	2144	2144	0.00	0.00	0.0	0.0	2144	0.00	0.00	0.0	0.0	2144	0.00	0.00	0.1	0.2
rbg038a	2480	2480	0.00	0.00	0.0	0.0	2480	0.03	0.07	0.0	0.0	2480	0.01	0.04	0.1	0.2
rbg040a	2378	2378	0.00	0.00	0.0	0.0	2379	0.04	0.00	0.1	0.2	2378	0.00	0.00	0.2	0.4
rbg041a	2598	2598	0.00	0.00	6.2	7.5	2600	0.09	0.03	0.1	0.2	2599	0.04	0.01	0.2	0.4
rbg042a	2772	2772	0.03	0.04	9.6	14.4	2785	0.55	0.12	0.0	0.0	2778	0.34	0.21	0.1	0.3
rbg048a	9383	9383	0.00	0.00	0.7	0.7	9402	0.21	0.01	0.1	0.2	9399	0.20	0.08	0.3	0.4
rbg049a	10018	10018	0.00	0.01	4.1	10.3	10023	0.10	0.07	0.0	0.0	10018	0.02	0.02	0.2	0.4
rbg050a	2953	2953	0.01	0.03	7.5	11.6	2960	0.31	0.05	0.1	0.2	2958	0.22	0.05	0.4	0.5
rbg050b	9863	9863	0.00	0.00	4.1	5.5	9879	0.21	0.04	0.0	0.0	9872	0.09	0.01	0.3	0.5
rbg050c	10024	10024	0.00	0.00	1.4	2.1	10027	0.07	0.03	0.1	0.2	10024	0.02	0.02	0.3	0.5
rbg055a	3761	3761	0.00	0.00	0.0	0.0	3761	0.01	0.03	0.1	0.2	3761	0.00	0.00	0.3	0.4
rbg067a	4625	4625	0.00	0.00	0.0	0.0	4625	0.00	0.00	0.1	0.2	4625	0.00	0.00	0.3	0.5
rbg086a	8400	8400	0.00	0.00	0.7	0.7	8400	0.00	0.00	0.2	0.4	8400	0.00	0.00	1.0	0.4
rbg092a	7158	7158	0.00	0.00	3.4	4.1	7158	0.03	0.01	0.3	0.4	7158	0.00	0.00	1.5	0.6
rbg125a	7936	7936	0.00	0.00	0.0	0.0	7936	0.02	0.03	0.5	0.5	7936	0.00	0.00	2.1	0.8
rbg132.2	8191	8191	0.01	0.01	7.5	11.6	8195	0.09	0.04	1.0	0.4	8192	0.01	0.00	3.9	1.1
rbg132	8468	8468	0.00	0.00	4.1	4.1	8471	0.07	0.04	0.5	0.5	8468	0.00	0.00	2.5	0.9
rbg152.3	9788	9788	0.00	0.01	11.0	12.3	9793	0.09	0.06	1.9	0.6	9788	0.06	0.05	5.6	1.5
rbg152	10032	10032	0.00	0.00	7.5	11.0	10039	0.07	0.01	0.6	0.5	10032	0.02	0.01	2.9	0.7
rbg172a	10950	10950	0.01	0.01	13.7	11.6	10952	0.06	0.07	1.1	0.4	10950	0.01	0.02	3.5	0.8
rbg193.2	12138	12138	0.00	0.00	13.0	13.7	12160	0.20	0.03	3.0	0.7	12150	0.11	0.03	9.0	1.9
rbg193	12535	12535	0.04	0.03	13.0	10.3	12557	0.19	0.04	0.9	0.5	12545	0.09	0.01	2.3	0.7
rbg201a	12948	12948	0.02	0.02	15.8	11.0	12968	0.22	0.04	1.7	0.5	12949	0.02	0.02	5.1	1.2
rbg233.2	14494	14494	0.01	0.03	22.6	8.9	14515	0.15	0.00	6.2	1.7	14503	0.08	0.05	15.7	3.2
rbg233	14992	14993	0.06	0.03	22.6	11.6	15019	0.19	0.01	1.5	0.7	14999	0.08	0.02	3.5	1.1

Table 14: TSPTW-C: Results for the *AFG* instance ($\eta_{rep} = 10$).

		MTU					GVNS-S					GVNS-Q				
Inst.	BKS	$best$	RPD_a	RPD_{sd}	t_{avg}	t_{sd}	$best$	RPD_a	RPD_{sd}	t_{avg}	t_{sd}	$best$	RPD_a	RPD_{sd}	t_{avg}	t_{sd}
rbg010a	671	671	0.00	0.00	0.0	0.0	671	0.00	0.00	0.0	0.0	671	0.00	0.00	0.0	0.0
rbg016a	938	938	0.00	0.00	0.0	0.0	938	0.00	0.00	0.0	0.0	938	0.00	0.00	0.0	0.0
rbg016b	1304	1304	0.00	0.00	0.0	0.0	1304	0.00	0.00	0.0	0.0	1304	0.00	0.00	0.1	0.2
rbg017.2	852	852	0.00	0.00	0.0	0.0	852	0.00	0.00	0.0	0.0	852	0.00	0.00	0.1	0.2
rbg017	893	893	0.00	0.00	0.0	0.0	893	0.00	0.00	0.1	0.2	893	0.00	0.00	0.1	0.2
rbg017a	4296	4296	0.00	0.00	0.0	0.0	4296	0.00	0.00	0.0	0.0	4296	0.00	0.00	0.0	0.0
rbg019a	1262	1262	0.00	0.00	0.0	0.0	1262	0.00	0.00	0.0	0.0	1262	0.00	0.00	0.1	0.2
rbg019b	1866	1866	0.00	0.00	0.0	0.0	1866	0.00	0.00	0.0	0.0	1866	0.00	0.00	0.1	0.2
rbg019c	4536	4536	0.00	0.00	0.0	0.0	4536	0.00	0.00	0.1	0.2	4536	0.00	0.00	0.1	0.2
rbg019d	1356	1356	0.00	0.00	0.0	0.0	1356	0.00	0.00	0.0	0.0	1356	0.00	0.00	0.0	0.0
rbg020a	4689	4689	0.00	0.00	0.0	0.0	4689	0.00	0.00	0.0	0.0	4689	0.00	0.00	0.1	0.2
rbg021	4536	4536	0.00	0.00	0.0	0.0	4528	0.00	0.00	0.1	0.2	4528	0.00	0.00	0.1	0.2
rbg021.2	4528	4528	0.00	0.00	0.0	0.0	4528	0.00	0.00	0.0	0.0	4528	0.00	0.00	0.1	0.2
rbg021.3	4528	4528	0.00	0.00	0.0	0.0	4525	0.00	0.00	0.1	0.2	4525	0.00	0.00	0.1	0.3
rbg021.4	4525	4525	0.00	0.00	0.0	0.0	4515	0.00	0.00	0.0	0.0	4515	0.00	0.00	0.1	0.2
rbg021.5	4515	4515	0.00	0.00	0.0	0.0	4480	0.00	0.00	0.0	0.0	4480	0.00	0.00	0.1	0.3
rbg021.6	4480	4480	0.00	0.00	0.0	0.0	4479	0.00	0.00	0.0	0.0	4479	0.00	0.00	0.1	0.3
rbg021.7	4479	4479	0.00	0.00	0.0	0.0	4478	0.00	0.01	0.0	0.0	4478	0.00	0.01	0.1	0.2
rbg021.8	4478	4478	0.00	0.00	0.0	0.0	4478	0.00	0.00	0.0	0.0	4478	0.00	0.00	0.1	0.3
rbg021.9	4478	4478	0.00	0.00	0.0	0.0	4536	0.00	0.00	0.0	0.0	4536	0.00	0.00	0.1	0.2
rbg027a	5091	5091	0.00	0.00	0.0	0.0	5091	0.00	0.00	0.0	0.0	5091	0.00	0.00	0.1	0.3
rbg031a	1863	1863	0.00	0.00	0.0	0.0	1863	0.00	0.00	0.1	0.2	1863	0.00	0.00	0.3	0.5
rbg033a	2069	2069	0.00	0.00	0.0	0.0	2069	0.00	0.00	0.0	0.0	2069	0.00	0.00	0.3	0.5
rbg034a	2222	2222	0.00	0.00	0.0	0.0	2222	0.00	0.00	0.0	0.0	2222	0.00	0.00	0.3	0.4
rbg035a.2	2056	2056	0.00	0.00	0.0	0.0	2056	0.00	0.01	0.1	0.3	2056	0.00	0.00	0.5	0.5
rbg035a	2144	2144	0.00	0.00	0.0	0.0	2144	0.00	0.00	0.1	0.2	2144	0.00	0.00	0.5	0.5
rbg038a	2480	2480	0.00	0.00	0.0	0.0	2480	0.01	0.04	0.1	0.2	2480	0.00	0.00	0.5	0.5
rbg040a	2378	2378	0.00	0.00	0.0	0.0	2378	0.02	0.02	0.1	0.2	2378	0.00	0.00	0.6	0.5
rbg041a	2598	2598	0.00	0.00	6.2	7.5	2599	0.06	0.02	0.1	0.2	2599	0.04	0.00	0.9	0.3
rbg042a	2772	2772	0.03	0.04	9.6	14.4	2774	0.14	0.12	0.1	0.2	2774	0.14	0.06	0.5	0.5
rbg048a	9383	9383	0.00	0.00	0.7	0.7	9385	0.06	0.06	0.2	0.4	9383	0.01	0.04	1.2	0.5
rbg049a	10018	10018	0.00	0.01	4.1	10.3	10018	0.04	0.03	0.1	0.3	10018	0.00	0.01	1.0	0.4
rbg050a	2953	2953	0.01	0.03	7.5	11.6	2955	0.08	0.06	0.3	0.5	2955	0.08	0.06	1.5	0.6
rbg050b	9863	9863	0.00	0.00	4.1	5.5	9872	0.11	0.04	0.3	0.4	9865	0.03	0.01	1.5	0.7
rbg050c	10024	10024	0.00	0.00	1.4	2.1	10026	0.03	0.01	0.2	0.4	10024	0.00	0.01	1.5	0.6
rbg055a	3761	3761	0.00	0.00	0.0	0.0	3761	0.00	0.00	0.1	0.3	3761	0.00	0.00	1.1	0.2
rbg067a	4625	4625	0.00	0.00	0.0	0.0	4625	0.00	0.00	0.2	0.4	4625	0.00	0.00	1.5	0.5
rbg086a	8400	8400	0.00	0.00	0.7	0.7	8400	0.00	0.00	0.6	0.5	8400	0.00	0.00	4.3	0.4
rbg092a	7158	7158	0.00	0.00	3.4	4.1	7158	0.02	0.01	0.9	0.3	7158	0.00	0.00	5.9	1.3
rbg125a	7936	7936	0.00	0.00	0.0	0.0	7936	0.01	0.01	2.2	0.8	7936	0.00	0.00	8.9	0.5
rbg132.2	8191	8191	0.01	0.01	7.5	11.6	8193	0.03	0.01	4.5	1.3	8191	0.00	0.01	19.4	3.7
rbg132	8468	8468	0.00	0.00	4.1	4.1	8471	0.06	0.03	2.3	0.9	8468	0.00	0.00	8.7	1.2
rbg152.3	9788	9788	0.00	0.01	11.0	12.3	9789	0.04	0.03	9.4	4.0	9788	0.03	0.02	26.8	4.0
rbg152	10032	10032	0.00	0.00	7.5	11.0	10035	0.06	0.02	2.6	0.5	10032	0.00	0.01	13.8	2.9
rbg172a	10950	10950	0.01	0.01	13.7	11.6	10950	0.02	0.05	4.0	1.2	10950	0.00	0.00	15.9	4.5
rbg193.2	12138	12138	0.00	0.00	13.0	13.7	12153	0.15	0.04	13.5	2.4	12145	0.09	0.02	30.7	0.6
rbg193	12535	12535	0.04	0.03	13.0	10.3	12549	0.13	0.02	3.6	1.4	12538	0.05	0.02	10.7	3.0
rbg201a	12948	12948	0.02	0.02	15.8	11.0	12949	0.01	0.00	7.3	1.9	12948	0.01	0.01	23.8	6.5
rbg233.2	14494	14494	0.01	0.03	22.6	8.9	14515	0.15	0.00	25.3	3.9	14500	0.10	0.04	31.1	0.2
rbg233	14992	14993	0.06	0.03	22.6	11.6	15015	0.18	0.04	4.9	1.4	14999	0.06	0.02	15.6	4.1

Table 15: TSPTW-C: Results for the *Pesant* instances.

		MTU					GVNS-S					GVNS-Q				
Inst.	BKS	$best$	RPD_a	RPD_{sd}	t_{avg}	t_{sd}	$best$	RPD_a	RPD_{sd}	t_{avg}	t_{sd}	$best$	RPD_a	RPD_{sd}	t_{avg}	t_{sd}
rc201.0	628.62	628.62	0.00	0.00	0.0	0.0	628.62	0.00	0.00	0.1	0.2	628.62	0.00	0.00	0.0	0.0
rc201.1	654.70	654.70	0.00	0.00	0.0	0.0	654.70	0.00	0.00	0.0	0.0	654.70	0.00	0.00	0.1	0.2
rc201.2	707.65	707.65	0.00	0.00	0.0	0.0	707.65	0.00	0.00	0.0	0.0	707.65	0.00	0.00	0.1	0.2
rc201.3	422.54	422.54	0.00	0.00	0.0	0.0	422.54	0.00	0.00	0.0	0.0	422.54	0.00	0.00	0.0	0.0
rc202.0	496.22	496.22	0.00	0.00	0.0	0.0	496.22	0.00	0.00	0.0	0.0	496.22	0.00	0.00	0.0	0.0
rc202.1	426.53	426.53	0.00	0.00	0.0	0.0	426.53	0.00	0.00	0.0	0.0	426.53	0.00	0.00	0.1	0.2
rc202.2	611.77	611.77	0.00	0.00	0.0	0.0	611.77	0.00	0.00	0.0	0.0	611.77	0.00	0.00	0.1	0.2
rc202.3	627.85	627.85	0.00	0.00	0.0	0.0	627.85	0.00	0.00	0.0	0.0	627.85	0.00	0.00	0.0	0.0
rc203.0	727.45	727.45	0.00	0.00	0.0	0.0	727.45	0.00	0.00	0.0	0.0	727.45	0.00	0.00	0.1	0.2
rc203.1	726.99	726.99	0.00	0.00	0.0	0.0	726.99	0.30	0.15	0.0	0.0	726.99	0.02	0.09	0.1	0.2
rc203.2	617.46	617.46	0.00	0.00	0.0	0.0	617.47	0.00	0.00	0.0	0.0	617.47	0.00	0.00	0.0	0.0
rc204.0	541.45	541.45	0.00	0.00	0.0	0.0	541.45	0.00	0.00	0.0	0.0	541.45	0.00	0.00	0.1	0.2
rc204.1	485.37	485.37	0.00	0.00	0.0	0.0	485.37	0.00	0.00	0.0	0.0	485.37	0.00	0.00	0.1	0.2
rc204.2	778.40	778.40	0.00	0.00	1.4	3.4	778.40	0.01	0.01	0.1	0.2	778.40	0.00	0.01	0.1	0.2
rc205.0	511.65	511.65	0.00	0.00	0.0	0.0	511.65	0.00	0.00	0.0	0.0	511.65	0.00	0.00	0.0	0.0
rc205.1	491.22	491.22	0.00	0.00	0.0	0.0	491.22	0.00	0.00	0.0	0.0	491.22	0.00	0.00	0.1	0.2
rc205.2	714.69	714.69	0.00	0.00	0.0	0.0	714.70	0.00	0.00	0.0	0.0	714.70	0.00	0.00	0.1	0.2
rc205.3	601.24	601.24	0.00	0.00	0.0	0.0	601.24	0.02	0.07	0.0	0.0	601.24	0.02	0.07	0.0	0.0
rc206.0	835.23	835.23	0.00	0.00	2.1	2.1	835.23	0.00	0.00	0.0	0.0	835.23	0.00	0.00	0.0	0.0
rc206.1	664.73	664.73	0.00	0.00	0.0	0.0	664.73	0.00	0.00	0.0	0.0	664.73	0.00	0.00	0.1	0.2
rc206.2	655.37	655.37	0.00	0.00	0.0	0.0	655.37	0.41	0.51	0.1	0.2	655.37	0.41	0.51	0.0	0.0
rc207.0	806.69	806.69	0.00	0.00	0.0	0.0	806.69	0.00	0.00	0.0	0.0	806.69	0.00	0.00	0.1	0.2
rc207.1	726.36	726.36	0.00	0.00	0.0	0.0	726.36	0.00	0.00	0.0	0.0	726.36	0.00	0.00	0.1	0.2
rc207.2	546.41	546.41	0.00	0.00	0.0	0.0	546.41	0.00	0.00	0.0	0.0	546.41	0.00	0.00	0.0	0.0
rc208.0	820.56	820.56	0.00	0.00	0.7	2.7	820.56	0.00	0.00	0.1	0.2	820.56	0.00	0.00	0.1	0.2
rc208.1	509.04	509.04	0.00	0.00	0.0	0.0	509.04	0.00	0.00	0.0	0.0	509.04	0.00	0.00	0.1	0.2
rc208.2	503.92	503.92	0.00	0.00	0.0	0.0	503.92	0.00	0.00	0.1	0.2	503.92	0.00	0.00	0.1	0.2

Table 16: TSPTW-C: Results for the *Potvin* instances.

		MTU					GVNS-S					GVNS-Q				
Inst.	BKS	$best$	RPD_a	RPD_{sd}	t_{avg}	t_{sd}	$best$	RPD_a	RPD_{sd}	t_{avg}	t_{sd}	$best$	RPD_a	RPD_{sd}	t_{avg}	t_{sd}
rc_201.1	444.54	444.54	0.00	0.00	0.0	0.0	444.54	0.00	0.00	0.0	0.0	444.54	0.00	0.00	0.0	0.0
rc_201.2	711.54	711.54	0.00	0.00	0.0	0.0	711.54	0.00	0.00	0.0	0.0	711.54	0.00	0.00	0.0	0.0
rc_201.3	790.61	790.61	0.00	0.00	0.0	0.0	790.61	0.29	0.30	0.0	0.0	790.61	0.29	0.30	0.1	0.2
rc_201.4	793.64	793.64	0.00	0.00	0.0	0.0	793.64	0.00	0.00	0.0	0.0	793.64	0.00	0.00	0.1	0.2
rc_202.1	771.78	771.78	0.00	0.00	9.6	8.2	771.78	0.01	0.02	0.0	0.0	771.78	0.01	0.02	0.1	0.2
rc_202.2	304.14	304.14	0.00	0.00	0.0	0.0	304.14	0.00	0.00	0.0	0.0	304.14	0.00	0.00	0.1	0.2
rc_202.3	837.72	837.72	0.00	0.00	0.0	0.0	837.72	0.00	0.00	0.0	0.0	837.72	0.00	0.00	0.0	0.0
rc_202.4	793.03	793.03	0.00	0.00	0.0	0.0	793.03	0.00	0.00	0.0	0.0	793.03	0.00	0.00	0.0	0.0
rc_203.1	453.48	453.48	0.00	0.00	0.0	0.0	453.48	0.03	0.09	0.0	0.0	453.48	0.03	0.09	0.0	0.0
rc_203.2	784.16	784.16	0.00	0.00	0.0	0.0	784.16	0.00	0.00	0.0	0.0	784.16	0.00	0.00	0.0	0.0
rc_203.3	817.53	817.53	0.00	0.00	0.0	0.0	817.53	0.00	0.00	0.0	0.0	817.53	0.00	0.00	0.1	0.2
rc_203.4	314.29	314.29	0.00	0.00	0.0	0.0	314.29	0.00	0.00	0.0	0.0	314.29	0.00	0.00	0.0	0.0
rc_204.1	878.64	878.64	0.00	0.00	0.0	0.0	878.64	0.16	0.18	0.1	0.2	878.64	0.16	0.18	0.1	0.3
rc_204.2	662.16	662.16	0.00	0.00	0.0	0.0	662.16	0.00	0.00	0.0	0.0	662.16	0.00	0.00	0.1	0.3
rc_204.3	455.03	455.03	0.00	0.00	0.0	0.0	455.03	0.00	0.00	0.0	0.0	455.03	0.00	0.00	0.0	0.0
rc_205.1	343.21	343.21	0.00	0.00	0.0	0.0	343.21	0.00	0.00	0.0	0.0	343.21	0.00	0.00	0.0	0.0
rc_205.2	755.93	755.93	0.00	0.00	0.0	0.0	755.93	0.00	0.00	0.0	0.0	755.93	0.00	0.00	0.1	0.2
rc_205.3	825.06	825.06	0.00	0.00	0.0	0.0	825.06	0.00	0.00	0.0	0.0	825.06	0.00	0.00	0.0	0.0
rc_205.4	760.47	760.47	0.00	0.00	0.0	0.0	760.47	0.25	0.14	0.0	0.0	760.47	0.25	0.14	0.0	0.0
rc_206.1	117.85	117.85	0.00	0.00	0.0	0.0	117.85	0.00	0.00	0.0	0.0	117.85	0.00	0.00	0.0	0.0
rc_206.2	828.06	828.06	0.00	0.00	0.0	0.0	828.06	0.35	0.66	0.0	0.0	828.06	0.30	0.68	0.1	0.2
rc_206.3	574.42	574.42	0.00	0.00	0.0	0.0	574.42	0.01	0.05	0.0	0.0	574.42	0.01	0.05	0.0	0.0
rc_206.4	831.67	831.67	0.05	0.18	0.7	2.7	831.67	0.05	0.18	0.0	0.0	831.67	0.05	0.18	0.1	0.2
rc_207.1	732.68	732.68	0.00	0.00	0.0	0.0	732.68	0.00	0.00	0.0	0.0	732.68	0.00	0.00	0.0	0.0
rc_207.2	701.25	701.25	0.00	0.00	0.0	0.0	701.25	0.20	0.24	0.0	0.0	701.25	0.20	0.24	0.0	0.0
rc_207.3	682.40	682.40	0.00	0.00	0.0	0.0	682.40	0.05	0.18	0.0	0.0	682.40	0.05	0.18	0.1	0.2
rc_207.4	119.64	119.64	0.00	0.00	0.0	0.0	119.64	0.00	0.00	0.0	0.0	119.64	0.00	0.00	0.0	0.0
rc_208.1	789.25	789.25	0.33	0.28	0.7	1.4	789.25	0.52	0.14	0.0	0.0	789.25	0.52	0.14	0.1	0.2
rc_208.2	533.78	533.78	0.00	0.00	0.0	0.0	533.78	0.00	0.00	0.0	0.0	533.78	0.00	0.00	0.1	0.2
rc_208.3	634.44	634.44	0.00	0.00	0.0	0.0	634.44	0.00	0.00	0.0	0.0	634.44	0.00	0.00	0.0	0.0

B.2 TSPTW-M

Tables 17-23 report, for all instance sets individually, the results obtained by GVNS-S and GVNS-Q for the TSPTW-M. Note that the results of ACG and P are directly taken from the papers of Amghar et al. (2019) and Pralet (2023). Their CPUs are approximately 1.4 and 1.2 times faster than ours (see processor information in the main text). The runtimes in the tables are scaled. A "-" in the ACG results means that Amghar et al. (2019) have found a new BKS when compared to Manuel López-Ibáňez et al. (2013).

Table 17: TSPTW-M: Results for the *Dumas* instances.

		ACG					P					GVNS-S					GVNS-Q				
Inst.	BKS	$best$	RPD_a	RPD_{sd}	t_{avg}	t_{sd}	$best$	RPD_a	RPD_{sd}	t_{avg}	t_{sd}	$best$	RPD_a	RPD_{sd}	t_{avg}	t_{sd}	$best$	RPD_a	RPD_{sd}	t_{avg}	t_{sd}
n20w20	370.4	370.4	0.00	0.00	0.0	0.0	370.4	0.00	0.00	0.0	0.0	370.4	0.00	0.00	0.0	0.0	370.4	0.00	0.00	0.0	0.1
n20w40	342.8	342.8	0.00	0.00	0.0	0.0	342.8	0.00	0.00	0.0	0.0	342.8	0.00	0.00	0.0	0.0	342.8	0.00	0.00	0.0	0.0
n20w60	362.0	362.0	0.00	0.00	0.0	0.0	362.0	0.00	0.00	0.0	0.0	362.0	0.00	0.00	0.0	0.0	362.0	0.00	0.00	0.0	0.0
n20w80	363.4	363.4	0.00	0.00	0.0	0.0	363.4	0.00	0.00	0.0	0.0	363.4	0.00	0.00	0.0	0.0	363.4	0.00	0.00	0.0	0.0
n20w100	331.6	331.6	0.00	0.00	0.0	0.0	331.6	0.00	0.00	0.0	0.0	331.6	0.32	0.63	0.0	0.0	331.6	0.00	0.00	0.0	0.0
n40w20	521.2	521.2	0.00	0.00	0.0	0.0	521.2	0.00	0.00	0.0	0.0	521.2	0.00	0.00	0.0	0.0	521.2	0.00	0.00	0.0	0.1
n40w40	512.2	512.2	0.00	0.00	0.0	0.0	512.2	0.00	0.00	0.0	0.0	512.2	0.00	0.00	0.0	0.0	512.2	0.00	0.00	0.0	0.1
n40w60	481.4	481.4	0.00	0.00	0.0	0.0	481.4	0.00	0.00	0.0	0.0	481.4	0.00	0.00	0.0	0.1	481.4	0.00	0.00	0.1	0.2
n40w80	486.6	486.6	0.00	0.00	0.0	0.0	486.6	0.00	0.00	0.0	0.0	486.6	0.00	0.00	0.0	0.0	486.6	0.00	0.00	0.1	0.2
n40w100	463.0	463.0	0.00	0.00	0.0	0.0	463.0	0.00	0.00	0.0	0.0	463.0	0.00	0.00	0.0	0.0	463.0	0.00	0.00	0.1	0.2
n60w20	626.8	626.8	0.00	0.00	0.0	0.0	626.8	0.00	0.00	0.0	0.0	626.8	0.00	0.00	0.0	0.0	626.8	0.00	0.00	0.0	0.1
n60w40	654.4	654.4	0.00	0.00	0.0	0.0	654.4	0.00	0.00	0.0	0.0	654.4	0.00	0.00	0.0	0.0	654.4	0.00	0.00	0.0	0.1
n60w60	672.8	672.8	0.00	0.00	0.0	0.0	672.8	0.00	0.00	0.0	0.0	672.8	0.00	0.00	0.0	0.0	672.8	0.00	0.00	0.1	0.2
n60w80	628.2	628.2	0.00	0.00	0.0	0.0	628.2	0.00	0.00	0.0	0.0	628.2	0.00	0.00	0.0	0.1	628.2	0.00	0.00	0.1	0.3
n60w100	620.2	620.6	0.06	0.00	0.0	0.0	620.2	0.00	0.00	0.0	0.0	620.6	0.06	0.00	0.0	0.0	620.2	0.00	0.00	0.1	0.3
n80w20	748.2	748.2	0.00	0.00	0.0	0.0	748.2	0.00	0.00	0.0	0.0	748.2	0.00	0.00	0.0	0.1	748.2	0.00	0.00	0.1	0.2
n80w40	725.6	725.6	-	-	0.0	0.0	725.6	0.00	0.00	0.0	0.0	725.6	0.00	0.00	0.0	0.1	725.6	0.00	0.00	0.1	0.3
n80w60	712.6	712.6	0.00	0.00	0.0	0.0	712.6	0.00	0.00	0.0	0.0	712.6	0.00	0.00	0.0	0.1	712.6	0.00	0.00	0.1	0.3
n80w80	714.6	714.6	-	-	0.0	0.0	714.6	0.00	0.00	0.0	0.0	714.6	0.00	0.00	0.1	0.2	714.6	0.00	0.00	0.2	0.4
n100w20	823.0	823.0	0.00	0.00	0.0	0.0	823.0	0.00	0.00	0.0	0.0	823.0	0.00	0.00	0.0	0.0	823.0	0.00	0.00	0.1	0.3
n100w40	821.0	821.0	0.00	0.00	0.0	0.0	821.0	0.00	0.00	0.0	0.0	821.0	0.00	0.00	0.0	0.1	821.0	0.00	0.00	0.2	0.4
n100w60	817.2	817.2	0.00	0.00	0.0	0.0	817.2	0.00	0.00	0.0	0.0	817.2	0.00	0.00	0.1	0.2	817.2	0.00	0.00	0.2	0.4
n150w20	978.4	978.4	0.00	0.00	0.0	0.0	978.4	0.00	0.00	0.0	0.0	978.4	0.00	0.00	0.1	0.2	978.4	0.00	0.00	0.2	0.4
n150w40	990.4	990.4	0.00	0.00	0.0	0.0	990.4	0.00	0.00	0.0	0.0	990.4	0.00	0.00	0.1	0.4	990.4	0.00	0.00	0.4	0.5
n150w60	988.6	988.6	0.00	0.00	0.6	0.0	988.6	0.00	0.00	0.0	0.0	988.6	0.00	0.00	0.3	0.4	988.6	0.00	0.00	0.7	0.5
n200w20	1137.8	1137.8	0.00	0.00	0.6	0.6	1137.8	0.00	0.00	0.0	0.0	1137.8	0.00	0.00	0.2	0.4	1137.8	0.00	0.00	0.5	0.5
n200w40	1156.0	1156.0	0.00	0.00	0.6	0.6	1156.0	0.00	0.00	0.0	0.0	1156.0	0.00	0.00	0.5	0.5	1156.0	0.00	0.00	1.0	0.4

Table 18: TSPTW-M: Results for the *GDE* instances.

		ACG					P					GVNS-S					GVNS-Q				
Inst.	BKS	$best$	RPD_a	RPD_{sd}	t_{avg}	t_{sd}	$best$	RPD_a	RPD_{sd}	t_{avg}	t_{sd}	$best$	RPD_a	RPD_{sd}	t_{avg}	t_{sd}	$best$	RPD_a	RPD_{sd}	t_{avg}	t_{sd}
n20w120	319.6	319.6	0.00	0.00	0.0	0.0	319.6	0.00	0.00	0.0	0.0	319.6	0.02	0.03	0.0	0.0	319.6	0.00	0.00	0.0	0.1
n20w140	286.2	286.2	0.00	0.00	0.0	0.0	286.2	0.00	0.00	0.0	0.0	286.2	0.00	0.00	0.0	0.0	286.2	0.00	0.00	0.0	0.0
n20w160	311.4	311.4	0.00	0.00	0.0	0.0	311.4	0.00	0.00	0.0	0.0	311.4	0.00	0.00	0.0	0.0	311.4	0.00	0.00	0.0	0.0
n20w180	311.2	311.2	0.00	0.00	0.0	0.0	311.2	0.00	0.00	0.0	0.0	311.2	0.06	0.04	0.0	0.0	311.2	0.00	0.00	0.0	0.0
n20w200	281.8	281.8	0.00	0.00	0.0	0.0	281.8	0.00	0.00	0.0	0.0	281.8	0.09	0.16	0.0	0.0	281.8	0.02	0.05	0.0	0.1
n40w120	470.6	470.6	0.00	0.00	0.0	0.0	470.6	0.00	0.00	0.0	0.0	470.6	0.03	0.13	0.0	0.1	470.6	0.01	0.04	0.1	0.3
n40w140	458.2	458.2	0.00	0.00	0.0	0.0	458.2	0.00	0.00	0.0	0.0	458.2	0.00	0.00	0.0	0.0	458.2	0.00	0.00	0.1	0.3
n40w160	426.8	426.8	0.00	0.00	0.0	0.0	426.8	0.00	0.00	0.0	0.0	426.8	0.00	0.00	0.0	0.1	426.8	0.00	0.00	0.1	0.3
n40w180	427.4	427.4	0.00	0.00	0.0	0.0	427.4	0.00	0.00	0.0	0.0	427.4	0.05	0.14	0.0	0.0	427.4	0.00	0.00	0.1	0.3
n40w200	412.0	412.0	0.00	0.00	0.0	0.0	412.0	0.00	0.00	0.0	0.0	412.0	0.00	0.00	0.0	0.0	412.0	0.00	0.00	0.1	0.3
n60w120	573.8	573.8	0.05	0.08	0.6	0.6	573.8	0.00	0.00	0.0	0.0	573.8	0.00	0.00	0.0	0.1	573.8	0.00	0.00	0.2	0.4
n60w140	600.0	600.0	0.00	0.00	0.0	0.0	600.0	0.00	0.00	0.0	0.0	600.0	0.00	0.00	0.0	0.1	600.0	0.00	0.00	0.2	0.4
n60w160	619.6	619.6	0.00	0.00	0.0	0.0	619.6	0.00	0.00	0.0	0.0	619.6	0.00	0.00	0.1	0.2	619.6	0.00	0.00	0.2	0.4
n60w180	576.0	576.0	0.00	0.00	0.0	0.0	576.0	0.00	0.00	0.0	0.0	576.0	0.00	0.00	0.0	0.1	576.0	0.00	0.00	0.2	0.4
n60w200	570.2	570.2	0.02	0.02	0.0	0.6	570.2	0.00	0.00	0.0	0.0	570.2	0.14	0.50	0.0	0.1	570.2	0.13	0.50	0.2	0.4
n80w100	711.2	711.2	0.00	0.00	0.0	0.0	711.2	0.00	0.00	0.0	0.0	711.2	0.00	0.00	0.1	0.2	711.2	0.00	0.00	0.2	0.4
n80w120	697.4	697.4	0.01	0.01	0.0	0.0	697.4	0.00	0.00	0.0	0.0	697.4	0.00	0.00	0.1	0.3	697.4	0.00	0.00	0.4	0.5
n80w140	672.8	672.8	0.00	0.00	0.0	0.0	672.8	0.00	0.00	0.0	0.0	672.8	0.00	0.00	0.1	0.2	672.8	0.00	0.00	0.4	0.5
n80w160	653.6	653.6	0.23	0.24	1.2	1.2	653.6	0.00	0.00	0.0	0.0	653.6	0.12	0.16	0.1	0.3	653.6	0.00	0.00	0.5	0.5
n80w180	656.4	656.4	0.05	0.09	0.6	0.6	656.4	0.00	0.00	0.0	0.0	656.4	0.00	0.00	0.1	0.3	656.4	0.00	0.00	0.4	0.5
n80w200	646.2	646.2	0.00	0.00	1.2	1.8	646.2	0.00	0.00	0.0	0.0	646.2	0.00	0.00	0.1	0.3	646.2	0.00	0.00	0.5	0.5
n100w80	805.8	805.8	0.00	0.00	0.0	0.0	805.8	0.00	0.00	0.0	0.0	805.8	0.00	0.00	0.1	0.3	805.8	0.00	0.00	0.3	0.5
n100w100	795.8	795.8	0.00	0.00	0.0	0.0	795.8	0.00	0.00	0.0	0.0	795.8	0.00	0.00	0.1	0.3	795.8	0.00	0.00	0.6	0.5
n100w120	895.4	895.4	0.00	0.00	0.0	0.0	895.4	0.00	0.00	0.0	0.0	895.4	0.00	0.00	0.1	0.3	895.4	0.00	0.00	0.5	0.5
n100w140	906.4	906.4	0.00	0.00	0.0	0.0	906.4	0.00	0.00	0.0	0.0	906.4	0.00	0.00	0.1	0.3	906.4	0.00	0.00	0.5	0.5
n100w160	865.0	865.0	0.00	0.00	0.0	0.0	865.0	0.00	0.00	0.0	0.0	865.0	0.00	0.00	0.1	0.3	865.0	0.00	0.00	0.6	0.5

Table 19: TSPTW-M: Results for the *OT* instances.

		ACG					P					GVNS-S					GVNS-Q				
Inst.	BKS	$best$	RPD_a	RPD_{sd}	t_{avg}	t_{sd}	$best$	RPD_a	RPD_{sd}	t_{avg}	t_{sd}	$best$	RPD_a	RPD_{sd}	t_{avg}	t_{sd}	$best$	RPD_a	RPD_{sd}	t_{avg}	t_{sd}
n150w120.001	972	972	0.00	0.00	1.8	1.8	972	0.00	0.00	0.0	1.1	972	0.00	0.00	0.5	0.5	972	0.00	0.00	1.7	0.5
n150w120.002	917	917	0.00	0.00	0.6	0.0	917	0.00	0.00	0.0	0.0	917	0.00	0.00	0.4	0.5	917	0.00	0.00	1.5	0.5
n150w120.003	909	910	0.29	0.28	6.6	6.6	909	0.00	0.00	0.0	3.9	909	0.00	0.00	0.6	0.5	909	0.00	0.00	3.0	0.9
n150w120.004	925	925	0.00	0.00	1.2	1.2	925	0.00	0.00	0.0	0.7	925	0.00	0.00	0.6	0.5	925	0.00	0.00	1.8	0.4
n150w120.005	907	907	0.00	0.00	1.2	0.6	907	0.00	0.00	0.0	0.4	907	0.00	0.00	0.5	0.5	907	0.00	0.00	1.7	0.4
n150w140.001	1008	1008	0.00	0.00	0.6	0.0	1008	0.00	0.00	0.0	0.0	1008	0.00	0.00	0.5	0.5	1008	0.00	0.00	1.3	0.5
n150w140.002	1020	1020	0.00	0.00	1.8	1.8	1020	0.00	0.00	0.0	1.1	1020	0.00	0.00	0.5	0.5	1020	0.00	0.00	2.3	0.4
n150w140.003	844	844	0.00	0.00	0.6	0.0	844	0.00	0.00	0.0	0.0	844	0.00	0.00	0.4	0.5	844	0.00	0.00	1.3	0.6
n150w140.004	898	898	0.00	0.00	0.6	0.6	898	0.00	0.00	0.0	0.4	898	0.00	0.00	0.6	0.5	898	0.00	0.00	1.7	0.6
n150w140.005	926	926	0.00	0.00	0.6	0.0	926	0.00	0.00	0.0	0.0	926	0.00	0.00	0.5	0.5	926	0.00	0.00	1.3	0.6
n150w160.001	959	959	0.00	0.00	0.6	0.0	959	0.00	0.00	0.0	0.0	959	0.00	0.00	0.6	0.5	959	0.00	0.00	1.5	0.5
n150w160.002	890	890	0.03	0.05	10.8	9.0	890	0.00	0.00	0.0	5.4	890	0.00	0.00	0.7	0.6	890	0.00	0.00	2.1	0.7
n150w160.003	934	934	0.00	0.00	0.6	0.0	934	0.00	0.00	0.0	0.0	934	0.00	0.00	0.5	0.5	934	0.00	0.00	1.5	0.5
n150w160.004	912	912	0.00	0.00	0.6	0.0	912	0.00	0.00	0.0	0.0	912	0.00	0.00	0.6	0.5	912	0.00	0.00	1.3	0.5
n150w160.005	920	920	0.00	0.00	0.6	0.0	920	0.00	0.00	0.0	0.0	920	0.00	0.00	0.5	0.5	920	0.00	0.00	1.2	0.4
n200w120.001	1089	1089	0.28	1.04	7.8	6.6	1089	0.00	0.00	0.0	3.9	1089	0.00	0.00	1.5	0.5	1089	0.00	0.00	3.9	0.9
n200w120.002	1072	1072	0.00	0.00	1.2	0.6	1072	0.00	0.00	0.0	0.4	1072	0.00	0.00	1.1	0.3	1072	0.00	0.00	3.1	0.4
n200w120.003	1128	1128	0.00	0.00	4.2	3.6	1128	0.00	0.00	0.0	2.1	1128	0.01	0.02	1.3	0.5	1128	0.00	0.00	3.0	0.4
n200w120.004	1072	1072	0.00	0.00	1.8	0.6	1072	0.00	0.00	0.0	0.4	1072	0.00	0.00	1.4	0.5	1072	0.00	0.00	4.1	0.4
n200w120.005	1073	1073	0.00	0.00	1.2	0.0	1073	0.00	0.00	0.0	0.0	1073	0.00	0.00	1.4	0.5	1073	0.00	0.00	3.9	0.2
n200w140.001	1138	1138	0.00	0.00	11.4	10.2	1138	0.00	0.00	0.0	6.1	1138	0.00	0.00	1.5	0.6	1138	0.00	0.00	4.7	0.6
n200w140.002	1087	1087	0.00	0.00	1.8	0.0	1087	0.00	0.00	0.0	0.0	1087	0.00	0.00	1.1	0.3	1087	0.00	0.00	3.1	0.3
n200w140.003	1083	1083	0.00	0.00	3.0	1.8	1083	0.00	0.00	0.0	1.1	1083	0.00	0.00	1.5	0.5	1083	0.00	0.00	4.5	0.5
n200w140.004	1100	1100	0.00	0.00	6.0	4.8	1100	0.00	0.00	0.0	2.9	1100	0.00	0.00	1.7	0.6	1100	0.00	0.00	3.7	0.7
n200w140.005	1121	1121	0.00	0.00	3.0	1.2	1121	0.00	0.00	0.0	0.7	1121	0.00	0.00	1.4	0.6	1121	0.00	0.00	2.7	0.4

Table 20: TSPTW-M: Results for the *Pesant* instances.

		ACG					P					GVNS-S					GVNS-Q				
Inst.	BKS	$best$	RPD_a	RPD_{sd}	t_{avg}	t_{sd}	$best$	RPD_a	RPD_{sd}	t_{avg}	t_{sd}	$best$	RPD_a	RPD_{sd}	t_{avg}	t_{sd}	$best$	RPD_a	RPD_{sd}	t_{avg}	t_{sd}
rc201.0	853.71	853.71	0.00	0.00	0.00	0.00	853.71	0.00	0.00	0.0	0.0	853.71	0.00	0.00	0.0	0.0	853.71	0.00	0.00	0.0	0.0
rc201.1	850.48	850.48	0.00	0.00	0.00	0.00	850.48	0.00	0.00	0.0	0.0	850.48	0.00	0.00	0.0	0.0	850.48	0.00	0.00	0.0	0.0
rc201.2	883.97	883.97	0.00	0.00	0.00	0.00	883.97	0.00	0.00	0.0	0.0	883.97	0.00	0.00	0.0	0.0	883.97	0.00	0.00	0.0	0.0
rc201.3	722.43	722.43	0.00	0.00	0.00	0.00	722.43	0.00	0.00	0.0	0.0	722.43	0.00	0.00	0.0	0.0	722.43	0.00	0.00	0.0	0.0
rc202.0	850.48	850.48	0.00	0.00	0.00	0.00	850.48	0.00	0.00	0.0	0.0	850.48	0.00	0.00	0.0	0.0	850.48	0.00	0.00	0.1	0.2
rc202.1	702.28	702.28	0.00	0.00	0.00	0.00	702.28	0.00	0.00	0.0	0.0	702.28	0.00	0.00	0.0	0.0	702.28	0.00	0.00	0.0	0.0
rc202.2	853.71	853.71	0.00	0.00	0.00	0.00	853.71	0.00	0.00	0.0	0.0	853.71	0.00	0.00	0.0	0.0	853.71	0.00	0.00	0.0	0.0
rc202.3	883.97	883.97	0.00	0.00	0.00	0.00	883.97	0.00	0.00	0.0	0.0	883.97	0.00	0.00	0.0	0.0	883.97	0.00	0.00	0.1	0.2
rc203.0	870.52	870.52	0.00	0.00	0.00	0.00	870.52	0.00	0.00	0.0	0.0	870.52	0.00	0.00	0.0	0.0	870.52	0.00	0.00	0.1	0.2
rc203.1	850.48	850.48	0.00	0.00	0.00	0.00	850.48	0.00	0.00	0.0	0.0	850.48	0.00	0.00	0.0	0.0	850.48	0.00	0.00	0.1	0.2
rc203.2	853.71	853.71	0.00	0.00	0.00	0.00	853.71	0.00	0.00	0.0	0.0	853.71	0.00	0.00	0.0	0.0	853.71	0.00	0.00	0.1	0.2
rc204.0	839.24	839.24	0.00	0.00	0.00	0.00	839.24	0.00	0.00	0.0	0.0	839.24	0.00	0.00	0.0	0.0	839.24	0.00	0.00	0.1	0.2
rc204.1	492.60	492.60	0.00	0.00	0.00	0.00	492.60	0.00	0.00	0.0	0.0	492.61	0.03	0.12	0.0	0.0	492.61	0.03	0.12	0.1	0.3
rc204.2	870.52	870.52	2.65	2.24	0.00	0.60	870.52	0.00	0.00	0.1	0.1	870.52	0.14	0.20	0.0	0.0	870.52	0.04	0.14	0.1	0.2
rc205.0	834.62	834.62	0.00	0.00	0.00	0.00	834.62	0.00	0.00	0.0	0.0	834.62	0.00	0.00	0.0	0.0	834.62	0.00	0.00	0.0	0.0
rc205.1	899.24	899.24	0.00	0.00	0.00	0.00	899.24	0.00	0.00	0.0	0.0	899.24	0.00	0.00	0.0	0.0	899.24	0.00	0.00	0.0	0.0
rc205.2	908.79	908.79	0.00	0.00	0.00	0.00	908.79	0.00	0.00	0.0	0.0	908.79	0.00	0.00	0.0	0.0	908.79	0.00	0.00	0.1	0.2
rc205.3	684.21	684.21	0.31	0.81	0.00	0.00	684.21	0.00	0.00	0.0	0.0	684.21	0.00	0.00	0.0	0.0	684.21	0.00	0.00	0.1	0.2
rc206.0	893.21	893.21	0.00	0.00	0.00	0.00	893.21	0.00	0.00	0.0	0.0	893.22	0.00	0.00	0.1	0.2	893.22	0.15	0.56	0.1	0.2
rc206.1	756.45	756.45	0.00	0.00	0.60	1.20	756.45	0.00	0.00	0.0	0.0	756.45	0.86	0.55	0.0	0.0	756.45	0.36	0.44	0.1	0.2
rc206.2	776.19	776.19	0.00	0.00	1.80	4.80	776.19	0.00	0.00	0.0	0.0	776.19	0.09	0.32	0.0	0.0	776.19	0.03	0.06	0.1	0.2
rc207.0	847.63	847.63	0.00	0.00	0.00	0.00	847.63	0.00	0.00	0.0	0.0	847.64	0.00	0.00	0.0	0.0	847.64	0.00	0.00	0.0	0.0
rc207.1	785.37	785.37	0.00	0.00	0.00	0.00	785.37	0.00	0.00	0.1	0.1	785.37	0.00	0.00	0.1	0.2	785.37	0.04	0.16	0.1	0.2
rc207.2	650.80	650.80	0.00	0.00	0.00	0.00	650.80	0.00	0.00	0.1	0.0	650.80	0.07	0.08	0.0	0.0	650.80	0.00	0.00	0.1	0.2
rc208.0	836.04	836.04	0.04	0.09	6.60	4.80	836.04	0.00	0.00	1.5	0.6	836.04	0.56	1.44	0.1	0.3	836.04	0.56	1.44	0.1	0.3
rc208.1	615.51	615.51	0.00	0.00	0.00	0.00	615.51	0.00	0.00	0.0	0.0	615.51	0.00	0.00	0.1	0.2	615.51	0.00	0.00	0.1	0.2
rc208.2	596.21	596.21	0.00	0.00	0.00	0.00	596.21	0.00	0.00	0.0	0.0	596.21	0.12	0.32	0.0	0.0	596.21	0.16	0.33	0.1	0.2

Table 21: TSPTW-M: Results for the *AFG* instances.

		ACG					P					GVNS-S					GVNS-Q				
Inst.	BKS	$best$	RPD_a	RPD_{sd}	t_{avg}	t_{sd}	$best$	RPD_a	RPD_{sd}	t_{avg}	t_{sd}	$best$	RPD_a	RPD_{sd}	t_{avg}	t_{sd}	$best$	RPD_a	RPD_{sd}	t_{avg}	t_{sd}
rbg010a	3840	3840	0.00	0.00	0.0	0.0	3840	0.00	0.00	0.0	0.0	3840	0.00	0.00	0.0	0.0	3840	0.00	0.00	0.0	0.0
rbg016a	2596	2596	0.00	0.00	0.0	0.0	2596	0.00	0.00	0.0	0.0	2596	0.00	0.00	0.0	0.0	2596	0.00	0.00	0.0	0.0
rbg016b	2094	2094	0.00	0.00	0.0	0.0	2094	0.00	0.00	0.0	0.0	2094	0.00	0.00	0.0	0.0	2094	0.00	0.00	0.0	0.0
rbg017.2	2351	2351	0.00	0.00	0.0	0.0	2351	0.00	0.00	0.0	0.0	2351	0.00	0.00	0.0	0.0	2351	0.00	0.00	0.0	0.0
rbg017	2351	2351	0.00	0.00	0.0	0.0	2351	0.00	0.00	0.0	0.0	2351	0.00	0.00	0.0	0.0	2351	0.00	0.00	0.0	0.0
rbg017a	4296	4296	0.00	0.00	0.0	0.0	4296	0.00	0.00	0.0	0.0	4296	0.00	0.00	0.0	0.0	4296	0.00	0.00	0.0	0.0
rbg019a	2694	2694	0.00	0.00	0.0	0.0	2694	0.00	0.00	0.0	0.0	2694	0.00	0.00	0.0	0.0	2694	0.00	0.00	0.1	0.2
rbg019b	3840	3840	0.00	0.00	0.0	0.0	3840	0.00	0.00	0.0	0.0	3840	0.00	0.00	0.0	0.0	3840	0.00	0.00	0.0	0.0
rbg019c	4536	4536	0.00	0.00	0.0	0.0	4536	0.00	0.00	0.0	0.0	4536	0.00	0.00	0.0	0.0	4536	0.00	0.00	0.0	0.0
rbg019d	3479	3479	0.00	0.00	0.0	0.0	3479	0.00	0.00	0.0	0.0	3479	0.00	0.00	0.0	0.0	3479	0.00	0.00	0.1	0.2
rbg020a	4689	4689	0.00	0.00	0.0	0.0	4689	0.00	0.00	0.0	0.0	4689	0.00	0.00	0.0	0.0	4689	0.00	0.00	0.1	0.2
rbg021.2	4528	4528	0.00	0.00	0.0	0.0	4528	0.00	0.00	0.0	0.0	4528	0.00	0.00	0.1	0.2	4528	0.01	0.04	0.0	0.0
rbg021.3	4528	4528	0.00	0.00	0.0	0.0	4528	0.00	0.00	0.0	0.0	4528	0.00	0.00	0.0	0.0	4528	0.00	0.00	0.0	0.0
rbg021.4	4525	4525	0.00	0.00	0.0	0.0	4525	0.00	0.00	0.0	0.0	4525	0.00	0.00	0.0	0.0	4525	0.00	0.00	0.0	0.0
rbg021.5	4516	4516	0.00	0.00	0.0	0.0	4516	0.00	0.00	0.0	0.0	4516	0.00	0.00	0.1	0.2	4516	0.02	0.04	0.0	0.0
rbg021.6	4492	4492	0.00	0.00	0.0	0.0	4492	0.00	0.00	0.0	0.0	4492	0.04	0.05	0.1	0.2	4492	0.02	0.02	0.1	0.2
rbg021.7	4481	4481	0.00	0.00	0.0	0.0	4481	0.00	0.00	0.0	0.0	4481	0.02	0.04	0.0	0.0	4481	0.02	0.04	0.1	0.2
rbg021.8	4481	4481	0.00	0.00	0.0	0.0	4481	0.00	0.00	0.0	0.0	4481	0.03	0.11	0.0	0.0	4481	0.00	0.00	0.1	0.2
rbg021.9	4481	4481	0.00	0.00	0.0	0.0	4481	0.00	0.00	0.0	0.0	4481	0.00	0.00	0.0	0.0	4481	0.02	0.02	0.1	0.2
rbg021	4536	4536	0.00	0.00	0.0	0.0	4536	0.00	0.00	0.0	0.0	4536	0.00	0.00	0.1	0.2	4536	0.00	0.00	0.1	0.2
rbg027a	5093	5093	0.00	0.00	0.0	0.0	5093	0.00	0.00	0.0	0.0	5093	0.01	0.02	0.0	0.0	5093	0.00	0.00	0.1	0.2
rbg031a	3498	3498	0.00	0.00	0.0	0.0	3498	0.00	0.00	0.0	0.0	3498	0.00	0.00	0.0	0.0	3498	0.00	0.00	0.1	0.2
rbg033a	3757	3757	0.00	0.00	0.0	0.0	3757	0.00	0.00	0.0	0.0	3757	0.00	0.00	0.0	0.0	3757	0.00	0.00	0.1	0.2
rbg034a	3314	3314	0.00	0.00	0.0	0.0	3314	0.00	0.00	0.0	0.0	3314	0.00	0.00	0.0	0.0	3314	0.00	0.00	0.1	0.2
rbg035a.2	3325	3325	0.00	0.00	0.0	0.0	3325	0.00	0.00	0.0	0.0	3325	0.00	0.00	0.0	0.0	3325	0.00	0.00	0.1	0.2
rbg035a	3388	3388	0.00	0.00	0.0	0.0	3388	0.00	0.00	0.0	0.0	3388	0.00	0.00	0.0	0.0	3388	0.00	0.00	0.1	0.2
rbg038a	5699	5699	0.00	0.00	0.0	0.0	5699	0.00	0.00	0.0	0.0	5699	0.00	0.00	0.0	0.0	5699	0.00	0.00	0.1	0.2
rbg040a	5679	5679	0.00	0.00	0.0	0.0	5679	0.00	0.00	0.0	0.0	5679	0.00	0.00	0.0	0.0	5679	0.00	0.00	0.1	0.3
rbg041a	3793	3793	0.00	0.00	0.0	0.0	3793	0.00	0.00	0.0	0.0	3793	0.00	0.00	0.0	0.0	3793	0.00	0.00	0.1	0.2
rbg042a	3260	3260	0.00	0.00	0.6	0.6	3260	0.00	0.00	0.0	0.0	3261	0.03	0.00	0.0	0.0	3260	0.00	0.01	0.1	0.2
rbg048a	9799	9799	0.00	0.00	0.0	0.0	9799	0.00	0.00	0.0	0.0	9799	0.00	0.00	0.1	0.2	9799	0.00	0.00	0.1	0.3
rbg049a	13257	13257	0.00	0.00	0.0	0.0	13257	0.00	0.00	0.0	0.0	13257	0.00	0.00	0.0	0.0	13257	0.00	0.00	0.1	0.2
rbg050a	12050	12050	0.00	0.00	0.0	0.0	12050	0.00	0.00	0.0	0.0	12050	0.00	0.00	0.0	0.0	12050	0.00	0.00	0.1	0.3
rbg050b	11957	11957	0.00	0.00	0.0	0.0	11957	0.00	0.00	0.0	0.0	11957	0.00	0.00	0.0	0.0	11957	0.00	0.00	0.1	0.3
rbg050c	10985	10985	0.00	0.00	0.0	0.0	10985	0.00	0.00	0.0	0.0	10985	0.00	0.00	0.0	0.0	10985	0.00	0.00	0.1	0.3
rbg055a	6929	6929	0.00	0.00	0.0	0.0	6929	0.00	0.00	0.0	0.0	6929	0.00	0.00	0.0	0.0	6929	0.00	0.00	0.1	0.3
rbg067a	10331	10331	0.00	0.00	0.0	0.0	10331	0.00	0.00	0.0	0.0	10331	0.00	0.00	0.0	0.0	10331	0.00	0.00	0.2	0.4
rbg086a	16899	16899	0.00	0.00	0.0	0.0	16899	0.00	0.00	0.0	0.0	16899	0.00	0.00	0.1	0.2	16899	0.00	0.00	0.4	0.5
rbg092a	12501	12501	0.00	0.00	0.0	0.0	12501	0.00	0.00	0.0	0.0	12501	0.00	0.00	0.1	0.2	12501	0.00	0.00	0.3	0.5
rbg125a	14214	14214	0.00	0.00	0.0	0.0	14214	0.00	0.00	0.0	0.0	14214	0.00	0.00	0.1	0.2	14214	0.00	0.00	0.9	0.3
rbg132.2	18524	18524	0.00	0.00	0.0	0.0	18524	0.00	0.00	0.0	0.0	18524	0.00	0.00	0.1	0.3	18524	0.00	0.00	1.0	0.0
rbg132	18524	18524	0.00	0.00	0.0	0.0	18524	0.00	0.00	0.0	0.0	18524	0.00	0.00	0.1	0.2	18524	0.00	0.00	0.6	0.5
rbg152.3	17455	17455	0.00	0.00	0.0	0.0	17455	0.00	0.00	0.0	0.0	17455	0.00	0.00	0.3	0.4	17455	0.00	0.00	1.6	0.5
rbg152	17455	17455	0.00	0.00	0.0	0.0	17455	0.00	0.00	0.0	0.0	17455	0.00	0.00	0.1	0.2	17455	0.00	0.00	0.4	0.5
rbg172a	17783	17783	0.00	0.00	6.6	7.8	17783	0.00	0.00	0.1	0.1	17783	0.00	0.00	0.1	0.3	17783	0.00	0.00	1.4	0.5
rbg193.2	21401	21401	0.00	0.00	0.0	0.0	21401	0.00	0.00	0.0	0.0	21401	0.00	0.00	0.5	0.5	21401	0.00	0.00	2.3	0.4
rbg193	21401	21401	0.00	0.00	0.0	0.0	21401	0.00	0.00	0.0	0.0	21401	0.00	0.00	0.1	0.3	21401	0.00	0.00	0.5	0.5
rbg201a	21380	21380	0.00	0.00	0.0	0.0	21380	0.00	0.00	0.0	0.0	21380	0.00	0.00	0.2	0.4	21380	0.00	0.00	1.3	0.4
rbg233.2	26143	26143	0.00	0.00	0.0	0.0	26143	0.00	0.00	0.0	0.0	26143	0.00	0.00	0.6	0.5	26143	0.00	0.00	3.2	0.7
rbg233	26143	26143	0.00	0.00	0.0	0.0	26143	0.00	0.00	0.0	0.0	26143	0.00	0.00	0.3	0.4	26143	0.00	0.00	1.0	0.0

Table 22: TSPTW-M: Results for the *Potvin* instances.

		ACG					P					GVNS-S					GVNS-Q				
Inst.	BKS	$best$	RPD_a	RPD_{sd}	t_{avg}	t_{sd}	$best$	RPD_a	RPD_{sd}	t_{avg}	t_{sd}	$best$	RPD_a	RPD_{sd}	t_{avg}	t_{sd}	$best$	RPD_a	RPD_{sd}	t_{avg}	t_{sd}
rc_201.1	592.06	592.06	0.00	0.00	0.0	0.0	592.06	0.00	0.00	0.0	0.0	592.06	0.00	0.00	0.0	0.0	592.06	0.00	0.00	0.0	0.0
rc_201.2	860.17	860.17	0.00	0.00	0.0	0.0	860.17	0.00	0.00	0.0	0.0	860.18	0.00	0.00	0.0	0.0	860.18	0.00	0.00	0.0	0.0
rc_201.3	853.71	853.71	0.00	0.00	0.0	0.0	853.71	0.00	0.00	0.0	0.0	853.71	0.00	0.00	0.0	0.0	853.71	0.00	0.00	0.0	0.0
rc_201.4	889.18	889.18	0.00	0.00	0.0	0.0	889.18	0.00	0.00	0.0	0.0	889.18	0.00	0.00	0.0	0.0	889.18	0.00	0.00	0.0	0.0
rc_202.1	850.48	850.48	0.00	0.00	0.0	0.0	850.48	0.00	0.00	0.0	0.0	850.48	0.00	0.00	0.0	0.0	850.48	0.00	0.00	0.0	0.0
rc_202.2	338.52	338.52	0.00	0.00	0.0	0.0	338.52	0.00	0.00	0.0	0.0	338.52	0.00	0.00	0.0	0.0	338.52	0.00	0.00	0.0	0.0
rc_202.3	894.10	894.10	0.00	0.00	0.0	0.0	894.10	0.00	0.00	0.0	0.0	894.10	0.00	0.00	0.0	0.0	894.10	0.00	0.00	0.0	0.0
rc_202.4	853.71	853.71	-	-	0.0	0.0	853.71	0.00	0.00	0.0	0.0	853.71	0.17	0.65	0.0	0.0	853.71	0.00	0.00	0.1	0.2
rc_203.1	488.42	488.42	0.00	0.00	0.0	0.0	488.42	0.00	0.00	0.0	0.0	488.42	0.00	0.00	0.0	0.0	488.42	0.00	0.00	0.0	0.0
rc_203.2	853.71	853.71	0.00	0.00	0.0	0.0	853.71	0.00	0.00	0.0	0.0	853.71	0.00	0.00	0.0	0.0	853.71	0.00	0.00	0.1	0.2
rc_203.3	921.44	921.44	0.00	0.00	0.0	0.0	921.44	0.00	0.00	0.0	0.0	921.44	0.00	0.00	0.1	0.2	921.44	0.00	0.00	0.1	0.2
rc_203.4	338.52	338.52	-	-	0.0	0.0	338.52	0.00	0.00	0.0	0.0	338.52	0.00	0.00	0.0	0.0	338.52	0.00	0.00	0.1	0.2
rc_204.1	917.83	917.83	0.11	0.28	4.8	8.4	917.83	0.00	0.00	0.4	0.3	925.23	0.81	0.00	0.1	0.2	917.83	0.11	0.27	0.1	0.2
rc_204.2	690.06	690.06	-	-	0.0	0.0	690.06	0.00	0.00	0.1	0.1	690.06	0.13	0.50	0.0	0.0	690.06	0.13	0.50	0.1	0.2
rc_204.3	455.03	455.03	0.00	0.00	0.0	0.0	455.03	0.00	0.00	0.0	0.0	455.03	0.12	0.44	0.1	0.2	455.03	0.00	0.00	0.1	0.2
rc_205.1	417.81	417.81	0.00	0.00	0.0	0.0	417.81	0.00	0.00	0.0	0.0	417.81	0.00	0.00	0.0	0.0	417.81	0.00	0.00	0.1	0.2
rc_205.2	820.19	820.19	0.00	0.00	0.0	0.0	820.19	0.00	0.00	0.0	0.0	820.19	0.04	0.15	0.0	0.0	820.19	0.04	0.15	0.0	0.0
rc_205.3	950.05	950.05	0.00	0.00	0.0	0.0	950.05	0.00	0.00	0.0	0.0	950.05	0.00	0.00	0.0	0.0	950.05	0.00	0.00	0.0	0.0
rc_205.4	837.71	837.71	0.00	0.00	0.0	0.0	837.71	0.00	0.00	0.0	0.0	837.71	0.26	0.66	0.0	0.0	837.71	0.26	0.66	0.0	0.0
rc_206.1	117.85	117.85	0.00	0.00	0.0	0.0	117.85	0.00	0.00	0.0	0.0	117.85	0.00	0.00	0.0	0.0	117.85	0.00	0.00	0.0	0.0
rc_206.2	870.49	870.49	0.00	0.00	2.4	5.4	870.49	0.00	0.00	0.1	0.0	870.49	0.53	0.76	0.0	0.0	870.49	0.49	0.70	0.1	0.2
rc_206.3	650.59	650.59	0.00	0.00	0.0	0.0	650.59	0.00	0.00	0.0	0.0	650.59	0.00	0.00	0.0	0.0	650.59	0.00	0.00	0.0	0.0
rc_206.4	911.98	911.98	-	-	2.4	4.8	911.98	0.00	0.00	0.1	0.0	911.98	0.34	1.26	0.0	0.0	911.98	0.08	0.30	0.1	0.2
rc_207.1	804.67	804.67	0.00	0.00	0.6	0.6	804.67	0.00	0.00	0.1	0.0	810.01	1.79	0.46	0.1	0.2	808.65	0.94	0.59	0.1	0.2
rc_207.2	713.90	713.90	0.00	0.00	0.0	0.0	713.90	0.00	0.00	0.0	0.0	713.90	0.00	0.00	0.0	0.0	713.90	0.00	0.00	0.0	0.0
rc_207.3	745.77	745.77	0.00	0.00	0.0	0.0	745.77	0.00	0.00	0.2	0.2	745.77	0.18	0.30	0.0	0.0	745.77	1.16	1.85	0.1	0.2
rc_207.4	133.14	133.14	0.00	0.00	0.0	0.0	133.14	0.00	0.00	0.0	0.0	133.14	0.00	0.00	0.0	0.0	133.14	0.00	0.00	0.0	0.0
rc_208.1	810.70	810.70	0.04	0.01	0.0	0.0	810.70	0.00	0.00	0.7	0.4	810.70	0.48	1.33	0.1	0.2	810.70	0.45	1.29	0.1	0.2
rc_208.2	579.51	579.51	0.00	0.00	0.0	0.0	579.51	0.00	0.00	0.0	0.0	579.51	0.04	0.11	0.0	0.0	579.51	0.04	0.11	0.1	0.3
rc_208.3	686.80	686.80	0.00	0.00	0.0	0.0	686.80	0.00	0.00	0.5	0.5	686.80	0.00	0.00	0.1	0.3	686.80	0.00	0.00	0.2	0.4

Table 23: TSPTW-M: Results for the *DaSilva* instances.

		GVNS-S							GVNS-Q						
Inst.	BKS*	$best$	avg	avg_{sd}	RPD_a	RPD_{sd}	t_{avg}	t_{sd}	$best$	avg	avg_{sd}	RPD_a	RPD_{sd}	t_{avg}	t_{sd}
n200w100	10577.0	10577.0	10577.0	0.0	0.00	0.00	0.1	0.2	10577.0	10577.0	0.0	0.00	0.00	0.3	0.5
n200w200	10697.6	10697.6	10697.7	0.5	0.00	0.00	0.1	0.3	10697.6	10697.6	0.0	0.00	0.00	0.3	0.4
n200w300	10527.6	10527.6	10527.6	0.0	0.00	0.00	0.1	0.3	10527.6	10527.6	0.0	0.00	0.00	0.4	0.5
n200w400	10477.6	10477.6	10477.6	0.0	0.00	0.00	0.1	0.3	10477.6	10477.6	0.0	0.00	0.00	0.5	0.5
n200w500	10431.6	10431.6	10431.6	0.0	0.00	0.00	0.2	0.4	10431.6	10431.6	0.0	0.00	0.00	0.7	0.4
n250w100	13390.6	13390.6	13390.6	0.0	0.00	0.00	0.1	0.3	13390.6	13390.6	0.0	0.00	0.00	0.5	0.5
n250w200	13041.6	13041.6	13041.6	0.0	0.00	0.00	0.1	0.3	13041.6	13041.6	0.0	0.00	0.00	0.4	0.5
n250w300	13542.4	13542.4	13542.4	0.0	0.00	0.00	0.1	0.3	13542.4	13542.4	0.0	0.00	0.00	0.4	0.5
n250w400	13248.6	13248.6	13248.6	0.0	0.00	0.00	0.2	0.4	13248.6	13248.6	0.0	0.00	0.00	0.7	0.4
n250w500	13220.0	13220.0	13220.0	0.0	0.00	0.00	0.3	0.4	13220.0	13220.0	0.0	0.00	0.00	1.0	0.4
n300w100	15918.8	15918.8	15918.8	0.0	0.00	0.00	0.2	0.4	15918.8	15918.8	0.0	0.00	0.00	0.7	0.5
n300w200	16038.0	16038.0	16038.0	0.0	0.00	0.00	0.2	0.4	16038.0	16038.0	0.0	0.00	0.00	0.5	0.5
n300w300	15579.8	15579.8	15580.1	1.3	0.00	0.01	0.3	0.4	15579.8	15579.8	0.0	0.00	0.00	0.8	0.5
n300w400	15726.8	15726.8	15726.8	0.0	0.00	0.00	0.3	0.5	15726.8	15726.8	0.0	0.00	0.00	1.0	0.5
n300w500	16047.8	16047.8	16047.8	0.0	0.00	0.00	0.5	0.5	16047.8	16047.8	0.0	0.00	0.00	1.6	0.5
n350w100	18585.0	18585.0	18585.0	0.0	0.00	0.00	0.3	0.4	18585.0	18585.0	0.0	0.00	0.00	0.9	0.5
n350w200	18353.8	18353.8	18353.8	0.0	0.00	0.00	0.3	0.4	18353.8	18353.8	0.0	0.00	0.00	0.7	0.4
n350w300	18362.8	18362.8	18362.8	0.0	0.00	0.00	0.4	0.5	18362.8	18362.8	0.0	0.00	0.00	1.0	0.5
n350w400	18335.2	18335.2	18335.2	0.0	0.00	0.00	0.5	0.5	18335.2	18335.2	0.0	0.00	0.00	1.5	0.5
n350w500	19024.2	19024.2	19024.2	0.0	0.00	0.00	0.7	0.5	19024.2	19024.2	0.0	0.00	0.00	2.1	0.6
n400w100	20742.6	20742.6	20742.6	0.0	0.00	0.00	0.5	0.5	20742.6	20742.6	0.0	0.00	0.00	1.4	0.5
n400w200	21480.2	21480.2	21480.4	0.7	0.00	0.00	0.5	0.5	21480.2	21480.2	0.0	0.00	0.00	1.2	0.6
n400w300	21325.2	21325.2	21325.2	0.0	0.00	0.00	0.6	0.5	21325.2	21325.2	0.0	0.00	0.00	1.2	0.5
n400w400	21035.2	21035.2	21035.2	0.0	0.00	0.00	0.9	0.5	21035.2	21035.2	0.0	0.00	0.00	1.7	0.7
n400w500	21035.6	21035.6	21035.7	0.2	0.00	0.00	1.0	0.4	21035.6	21035.7	0.2	0.00	0.00	2.6	0.7

*: Best-known solutions correspond to the best solution values found by any run of GVNS-S and GVNS-Q for all the instances.

B.3 TSPTW-D

Tables 24-30 report, for all instance sets individually, the results obtained by GVNS-S and GVNS-Q for the TSPTW-D. Note that the optimal solutions are taken from the paper of Lera-Romero et al. (2022), and the results of TI are taken directly from the papers of Tilk and Irnich (2017). The CPU of TI is approximately 1.2 times faster than ours (see processor information in the main text), and the runtimes in the tables are scaled. Because the authors did not report the runtimes of TI on the level of each instance, we only present the best solution values in $best$ for TI results.

Table 24: TSPTW-D: Results for the *GDE* instances.

		TI	GVNS-S					GVNS-Q				
Inst.	BKS	$best$	$best$	avg	avg_{sd}	t_{avg}	t_{sd}	$best$	avg	avg_{sd}	t_{avg}	t_{sd}
n20w120	281.6	281.6	281.8	281.9	0.3	0.0	0.0	281.6	281.6	0.0	0.0	0.1
n20w140	246.2	246.2	246.6	247.1	0.4	0.0	0.0	246.4	246.5	0.1	0.1	0.2
n20w160	254.4	254.4	254.4	254.5	0.1	0.0	0.0	254.4	254.8	0.4	0.1	0.2
n20w180	263.6	267.2	263.6	264.3	1.2	0.0	0.0	263.6	263.6	0.0	0.1	0.2
n20w200	246.8	252.8	247.2	247.8	1.3	0.0	0.0	246.8	246.8	0.0	0.0	0.1
n40w120	428.4	429.0	429.8	430.0	0.6	0.0	0.0	429.0	429.1	0.2	0.1	0.3
n40w140	404.4	404.4	405.4	405.5	0.3	0.0	0.0	404.4	404.5	0.1	0.2	0.4
n40w160	369.4	370.2	370.0	370.4	1.3	0.0	0.0	370.0	370.1	0.2	0.2	0.4
n40w180	370.4	373.0	372.2	374.0	1.9	0.0	0.1	370.6	371.9	1.4	0.1	0.3
n40w200	349.2	361.2	349.4	349.8	0.6	0.0	0.1	349.2	349.7	0.4	0.2	0.4
n60w120	526.0	526.0	526.6	526.7	0.1	0.1	0.2	526.0	526.0	0.0	0.3	0.5
n60w140	557.6	558.4	557.6	558.0	0.7	0.1	0.2	557.6	557.9	0.5	0.4	0.5
n60w160	560.2	560.2	560.2	560.2	0.0	0.0	0.1	560.2	560.2	0.0	0.3	0.5
n60w180	512.0	516.0	512.0	512.1	0.3	0.0	0.1	512.0	512.1	0.4	0.4	0.5
n60w200	504.4	510.2	504.8	505.7	1.6	0.0	0.1	504.4	504.5	0.1	0.5	0.5
n80w100	696.0	701.0	696.0	696.0	0.0	0.1	0.2	696.0	696.0	0.0	0.3	0.5
n80w120	643.6	643.6	643.6	644.0	1.1	0.1	0.3	643.6	644.0	1.1	0.7	0.4
n80w140	615.6	618.2	615.6	615.9	0.7	0.1	0.3	615.6	615.8	0.1	0.8	0.5
n80w160	596.8	598.2	598.8	599.5	1.0	0.1	0.3	596.8	597.9	1.2	0.9	0.4
n80w180	593.8	594.6	593.8	594.2	0.5	0.1	0.3	593.8	594.1	0.3	0.8	0.5
n80w200	581.0	585.2	583.0	583.3	0.6	0.1	0.4	583.0	583.0	0.0	0.9	0.5
n100w80	792.8	-	792.8	793.0	0.3	0.1	0.3	792.8	793.2	0.3	0.4	0.5
n100w100	772.8	-	772.8	773.0	0.3	0.1	0.3	772.8	773.0	0.3	0.8	0.4
n100w120	838.8	839.4	838.8	838.8	0.0	0.1	0.3	838.8	838.8	0.0	0.9	0.4
n100w140	842.4	842.4	842.4	842.4	0.0	0.1	0.3	842.4	842.4	0.0	0.9	0.4
n100w160	804.2	804.2	804.2	804.2	0.0	0.1	0.4	804.2	804.2	0.0	0.9	0.4

Results of two *GDE* classes "n100w80" and "n100w100" were not reported in both Tilk and Irnich (2017) and Lera-Romero et al. (2022).

Table 25: TSPTW-D: Results for the *OT* instances.

		TI	GVNS-S					GVNS-Q				
Inst.	BKS	$best$	$best$	avg	avg_{sd}	t_{avg}	t_{sd}	$best$	avg	avg_{sd}	t_{avg}	t_{sd}
n150w120.001	920.0	920.0	920.4	920.8	0.6	0.7	0.6	920.0	920.1	0.3	4.0	1.0
n150w120.002	869.2	869.2	869.2	869.3	0.1	0.5	0.5	869.2	869.2	0.1	2.9	0.9
n150w120.003	861.1	861.1	861.1	861.1	0.0	0.9	0.6	861.1	861.1	0.0	3.3	0.9
n150w120.004	875.9	875.9	877.9	878.6	1.2	1.0	0.5	877.0	877.0	0.0	4.5	1.3
n150w120.005	855.5	855.5	855.5	855.5	0.0	0.8	0.5	855.5	855.5	0.0	3.2	0.9
n150w140.001	956.0	956.0	956.0	956.1	0.2	0.7	0.4	956.0	956.0	0.0	3.5	0.9
n150w140.002	972.2	981.1	972.2	972.8	0.7	0.7	0.4	972.2	972.4	0.5	3.9	1.0
n150w140.003	790.4	791.5	790.4	790.5	0.2	0.9	0.3	790.4	790.4	0.0	4.3	0.9
n150w140.004	849.2	849.4	849.5	849.5	0.0	1.1	0.2	849.5	849.5	0.1	4.3	1.3
n150w140.005	871.2	871.2	871.2	871.2	0.0	0.7	0.5	871.2	871.2	0.1	3.3	0.5
n150w160.001	899.6	906.2	899.8	899.8	0.0	1.0	0.5	899.6	899.8	0.1	5.2	1.4
n150w160.002	841.4	890.5	841.4	844.7	3.5	1.3	0.4	841.4	844.7	3.5	5.1	1.5
n150w160.003	882.7	888.6	882.7	882.9	0.6	0.6	0.5	882.7	882.7	0.0	3.1	1.1
n150w160.004	862.4	863.2	862.4	862.5	0.2	0.9	0.5	862.4	862.4	0.0	3.7	0.5
n150w160.005	864.3	866.9	865.2	866.2	1.3	0.6	0.5	864.3	864.3	0.0	2.7	0.9
n200w120.001	1045.4	1045.4	1045.4	1045.4	0.0	1.8	0.7	1045.4	1045.4	0.1	6.1	1.7
n200w120.002	1022.5	1022.5	1022.5	1022.5	0.0	1.5	0.5	1022.5	1022.5	0.0	5.4	0.9
n200w120.003	1081.0	1081.0	1081.0	1081.1	0.2	1.9	0.6	1081.0	1081.0	0.0	6.1	1.5
n200w120.004	1017.7	1017.7	1017.8	1017.8	0.0	2.0	0.5	1017.7	1017.8	0.0	8.7	2.4
n200w120.005	1025.2	1038.5	1025.2	1025.2	0.0	2.5	0.5	1025.2	1025.2	0.0	8.9	1.7
n200w140.001	1089.3	1089.3	1089.6	1089.9	0.8	1.9	0.4	1089.3	1092.1	10.0	7.9	2.6
n200w140.002	1039.8	1039.8	1039.8	1039.8	0.0	1.7	0.5	1039.8	1039.8	0.0	6.8	1.9
n200w140.003	1040.4	1040.4	1040.4	1040.4	0.0	2.2	0.5	1040.4	1040.4	0.0	7.5	2.1
n200w140.004	1051.8	1051.8	1051.8	1051.9	0.1	3.2	1.0	1051.8	1051.9	0.1	8.1	2.4
n200w140.005	1070.0	1074.3	1070.0	1070.0	0.0	1.9	0.6	1070.0	1070.0	0.0	7.4	2.1

Table 26: TSPTW-D: Results for the *Potvin* instances.

		TI	GVNS-S					GVNS-Q				
Inst.	BKS	$best$	$best$	avg	avg_{sd}	t_{avg}	t_{sd}	$best$	avg	avg_{sd}	t_{avg}	t_{sd}
rc_201.1	503.52	503.52	503.52	503.52	0.00	0.0	0.0	503.52	503.52	0.00	0.0	0.0
rc_201.2	756.33	756.33	756.33	756.33	0.00	0.0	0.0	756.33	756.33	0.00	0.0	0.0
rc_201.3	816.05	816.05	816.05	816.05	0.00	0.0	0.0	816.05	816.05	0.00	0.0	0.0
rc_201.4	812.07	812.07	812.07	812.07	0.00	0.0	0.0	812.07	812.07	0.00	0.0	0.0
rc_202.1	772.15	788.20	772.16	772.17	0.02	0.0	0.0	772.16	772.17	0.02	0.1	0.2
rc_202.2	315.04	315.04	315.04	315.04	0.00	0.0	0.0	315.04	315.04	0.00	0.0	0.0
rc_202.3	870.69	892.03	870.69	871.66	2.23	0.0	0.0	870.69	870.69	0.00	0.1	0.2
rc_202.4	794.46	794.46	794.46	803.59	10.77	0.0	0.0	794.46	797.87	9.89	0.0	0.0
rc_203.1	453.40	453.40	453.40	462.23	32.08	0.0	0.0	453.40	461.99	32.14	0.0	0.0
rc_203.2	807.98	809.42	807.99	823.17	36.98	0.1	0.2	807.99	818.04	37.62	0.1	0.2
rc_203.3	874.03	887.33	875.29	882.58	2.87	0.0	0.0	874.25	877.42	1.24	0.1	0.2
rc_203.4	317.41	318.41	318.41	318.94	0.75	0.1	0.2	318.41	318.94	0.75	0.0	0.0
rc_204.1	880.69	898.89	880.81	882.22	1.15	0.1	0.2	880.69	880.69	0.00	0.2	0.4
rc_204.2	671.20	674.62	672.32	672.88	2.08	0.1	0.2	671.20	671.80	0.56	0.1	0.3
rc_204.3	454.95	454.95	454.95	455.41	1.72	0.1	0.2	454.95	454.95	0.00	0.1	0.2
rc_205.1	375.49	375.49	375.49	375.95	1.73	0.0	0.0	375.49	376.41	2.35	0.0	0.0
rc_205.2	788.69	794.95	794.95	794.95	0.00	0.0	0.0	794.95	794.95	0.00	0.0	0.0
rc_205.3	827.64	827.64	827.64	833.59	19.31	0.0	0.0	827.64	832.83	19.43	0.1	0.2
rc_205.4	789.37	789.37	789.37	798.29	15.78	0.0	0.0	789.37	798.29	15.78	0.0	0.0
rc_206.1	117.84	117.84	117.84	117.84	0.00	0.0	0.0	117.84	117.84	0.00	0.0	0.0
rc_206.2	843.49	843.49	843.49	845.80	4.21	0.0	0.0	843.49	845.80	4.21	0.0	0.0
rc_206.3	577.23	577.23	577.71	578.46	2.80	0.0	0.0	577.23	577.83	2.23	0.1	0.2
rc_206.4	838.30	847.70	838.30	848.29	28.81	0.0	0.0	838.30	848.29	28.81	0.1	0.2
rc_207.1	732.60	770.56	732.60	733.12	1.04	0.0	0.0	732.60	733.02	1.06	0.1	0.2
rc_207.2	701.16	703.47	701.16	712.20	20.94	0.0	0.0	701.16	712.20	20.94	0.0	0.0
rc_207.3	682.30	732.86	692.11	720.36	53.18	0.0	0.0	685.30	702.80	52.89	0.1	0.2
rc_207.4	119.61	119.61	119.61	119.61	0.00	0.0	0.0	119.61	119.61	0.00	0.0	0.0
rc_208.1	789.12	799.04	793.48	794.10	1.57	0.0	0.0	793.48	794.41	1.85	0.1	0.2
rc_208.2	533.69	554.78	543.34	544.46	2.81	0.1	0.2	535.78	540.07	5.34	0.1	0.3
rc_208.3	634.36	679.02	634.79	634.79	0.00	0.0	0.0	634.79	634.79	0.00	0.1	0.2

Table 27: TSPTW-D: Results for the *AFG* instances.

		TI	GVNS-S					GVNS-Q				
Inst.	BKS	$best$	$best$	avg	avg_{sd}	t_{avg}	t_{sd}	$best$	avg	avg_{sd}	t_{avg}	t_{sd}
rbg010a	2975	2975	2975	2975.0	0.0	0.0	0.0	2975	2975.0	0.0	0.0	0.0
rbg016a	2465	2465	2465	2468.5	5.7	0.0	0.0	2465	2465.6	0.5	0.0	0.0
rbg016b	1304	1304	1304	1304.0	0.0	0.0	0.0	1304	1304.0	0.0	0.0	0.0
rbg017.2	1351	1351	1351	1351.0	0.0	0.0	0.0	1351	1351.0	0.0	0.0	0.0
rbg017	1756	1756	1756	1756.0	0.0	0.0	0.0	1756	1756.0	0.0	0.0	0.0
rbg017a	4296	4296	4296	4296.0	0.0	0.0	0.0	4296	4296.0	0.0	0.0	0.0
rbg019a	2448	2448	2448	2448.0	0.0	0.0	0.0	2448	2448.0	0.0	0.0	0.0
rbg019b	2975	2975	2975	2975.0	0.0	0.0	0.0	2975	2975.0	0.0	0.0	0.0
rbg019c	4536	4536	4536	4536.0	0.0	0.0	0.0	4536	4536.0	0.0	0.0	0.0
rbg019d	2917	2917	2917	2917.0	0.0	0.0	0.0	2917	2917.0	0.0	0.0	0.0
rbg020a	4689	4689	4689	4689.0	0.0	0.0	0.0	4689	4689.0	0.0	0.1	0.2
rbg021.2	4528	4528	4528	4528.0	0.0	0.1	0.2	4528	4528.5	2.0	0.0	0.0
rbg021.3	4528	4528	4528	4528.0	0.0	0.0	0.0	4528	4528.0	0.0	0.1	0.2
rbg021.4	4525	4525	4525	4525.0	0.0	0.0	0.0	4525	4525.0	0.0	0.1	0.2
rbg021.5	4516	4516	4516	4516.0	0.0	0.0	0.0	4516	4516.0	0.0	0.1	0.2
rbg021.6	4484	4489	4484	4487.9	1.7	0.0	0.0	4484	4484.4	1.5	0.1	0.2
rbg021.7	4479	4481	4479	4481.9	3.4	0.0	0.0	4479	4479.1	0.2	0.1	0.2
rbg021.8	4478	4481	4478	4478.0	0.0	0.0	0.0	4478	4478.0	0.0	0.1	0.2
rbg021.9	4478	4481	4478	4478.0	0.0	0.1	0.2	4478	4478.1	0.3	0.0	0.0
rbg021	4536	4536	4536	4536.0	0.0	0.0	0.0	4536	4536.0	0.0	0.1	0.2
rbg027a	5093	5093	5093	5093.3	1.0	0.0	0.0	5093	5093.0	0.0	0.0	0.0
rbg031a	2953	2953	2953	2953.0	0.0	0.0	0.0	2953	2953.0	0.0	0.1	0.2
rbg033a	3157	3157	3157	3157.0	0.0	0.0	0.0	3157	3157.0	0.0	0.1	0.2
rbg034a	2714	2714	2714	2714.0	0.0	0.0	0.0	2714	2714.0	0.0	0.1	0.2
rbg035a.2	2715	2715	2715	2715.0	0.0	0.0	0.0	2715	2715.0	0.0	0.1	0.2
rbg035a	2874	2874	2874	2874.0	0.0	0.0	0.0	2874	2874.0	0.0	0.1	0.2
rbg038a	5115	5115	5115	5115.0	0.0	0.0	0.0	5115	5115.0	0.0	0.1	0.2
rbg040a	5079	5079	5079	5079.0	0.0	0.0	0.0	5079	5079.0	0.0	0.1	0.2
rbg041a	3245	3245	3245	3245.0	0.0	0.0	0.0	3245	3245.0	0.0	0.1	0.3
rbg042a	2962	2962	2972	2979.9	7.4	0.0	0.0	2963	2964.1	2.7	0.1	0.3
rbg048a	9793	9793	9793	9793.0	0.0	0.1	0.2	9793	9793.0	0.0	0.1	0.2
rbg049a	12657	12657	12657	12657.0	0.0	0.1	0.2	12657	12657.0	0.0	0.1	0.2
rbg050a	11450	11450	11450	11450.0	0.0	0.0	0.0	11450	11450.0	0.0	0.1	0.3
rbg050b	11357	11357	11357	11357.0	0.0	0.0	0.0	11357	11357.0	0.0	0.1	0.3
rbg050c	10431	10431	10431	10431.0	0.0	0.0	0.0	10431	10431.0	0.0	0.1	0.3
rbg055a	6367	6367	6367	6367.0	0.0	0.0	0.0	6367	6367.0	0.0	0.2	0.4
rbg067a	9736	9736	9736	9736.0	0.0	0.1	0.2	9736	9736.0	0.0	0.2	0.4
rbg086a	16299	16299	16299	16299.0	0.0	0.0	0.0	16299	16299.0	0.0	0.3	0.5
rbg092a	11924	11924	11924	11924.0	0.0	0.1	0.2	11924	11924.0	0.0	0.3	0.5
rbg125a	13652	13652	13652	13652.0	0.0	0.0	0.0	13652	13652.0	0.0	0.8	0.4
rbg132.2	17524	17524	17524	17524.0	0.0	0.1	0.3	17524	17524.0	0.0	1.1	0.2
rbg132	17929	17929	17929	17929.0	0.0	0.1	0.2	17929	17929.0	0.0	0.6	0.5
rbg152.3	16455	16455	16455	16455.0	0.0	0.3	0.5	16455	16455.0	0.0	1.6	0.5
rbg152	17019	17019	17019	17019.0	0.0	0.1	0.2	17019	17019.0	0.0	0.9	0.3
rbg172a	17221	17221	17221	17221.4	0.5	0.2	0.4	17221	17221.7	0.4	1.1	0.5
rbg193.2	20401	20401	20401	20401.0	0.0	0.5	0.5	20401	20401.0	0.0	2.3	0.5
rbg193	20869	20869	20869	20869.0	0.0	0.2	0.4	20869	20869.0	0.0	0.8	0.4
rbg201a	20818	20818	20818	20818.0	0.0	0.2	0.4	20818	20818.0	0.0	1.3	0.5
rbg233.2	25143	25143	25143	25143.0	0.0	0.7	0.5	25143	25143.0	0.0	3.4	0.5
rbg233	25691	25691	25691	25691.0	0.0	0.3	0.4	25691	25691.0	0.0	1.1	0.3

Table 28: TSPTW-D: Results for the *Dumas* instances.

		GVNS-S					GVNS-Q				
Inst.	BKS*	$best$	avg	avg_{sd}	t_{avg}	t_{sd}	$best$	avg	avg_{sd}	t_{avg}	t_{sd}
n20w20	368.2	368.2	368.2	0.1	0.0	0.0	368.2	368.2	0.0	0.0	0.0
n20w40	338.8	338.8	338.8	0.1	0.0	0.0	338.8	338.9	0.1	0.0	0.0
n20w60	351.8	351.8	351.8	0.0	0.0	0.0	351.8	351.8	0.0	0.0	0.0
n20w80	348.4	348.8	348.8	0.0	0.0	0.0	348.4	348.4	0.0	0.0	0.0
n20w100	306.0	306.0	307.9	3.8	0.0	0.0	306.0	306.1	0.1	0.0	0.1
n40w20	519.4	519.4	519.4	0.0	0.0	0.0	519.4	519.4	0.0	0.0	0.1
n40w40	504.6	504.6	504.7	0.1	0.0	0.0	504.6	504.7	0.1	0.0	0.1
n40w60	470.4	470.4	470.4	0.1	0.0	0.0	470.4	470.4	0.0	0.0	0.1
n40w80	471.0	471.0	471.1	0.2	0.0	0.0	471.0	471.0	0.0	0.1	0.2
n40w100	449.6	449.6	449.7	0.2	0.0	0.0	449.6	449.6	0.0	0.1	0.2
n60w20	624.0	624.0	624.0	0.0	0.0	0.0	624.0	624.0	0.0	0.1	0.2
n60w40	646.6	646.6	646.6	0.0	0.0	0.0	646.6	646.6	0.0	0.1	0.3
n60w60	663.4	663.4	663.4	0.1	0.0	0.1	663.4	663.5	0.1	0.1	0.3
n60w80	614.2	614.2	614.2	0.0	0.0	0.0	614.2	614.2	0.0	0.1	0.3
n60w100	603.6	604.6	604.7	0.2	0.0	0.1	603.6	603.8	0.4	0.2	0.4
n80w20	744.4	744.4	744.4	0.0	0.0	0.0	744.4	744.4	0.0	0.1	0.2
n80w40	718.8	718.8	718.8	0.0	0.0	0.1	718.8	718.9	0.2	0.2	0.4
n80w60	705.0	705.0	705.0	0.0	0.0	0.1	705.0	705.0	0.0	0.2	0.4
n80w80	700.8	700.8	700.9	0.3	0.1	0.2	700.8	700.8	0.0	0.3	0.5
n100w20	819.6	819.6	819.6	0.0	0.0	0.1	819.6	819.6	0.0	0.1	0.3
n100w40	813.4	813.4	813.5	0.1	0.0	0.1	813.4	813.4	0.0	0.3	0.4
n100w60	807.8	807.8	807.8	0.0	0.1	0.2	807.8	807.8	0.0	0.3	0.5
n150w20	974.2	974.2	974.2	0.0	0.1	0.2	974.2	974.2	0.0	0.3	0.4
n150w40	985.2	985.2	985.2	0.0	0.2	0.4	985.2	985.2	0.0	0.6	0.5
n150w60	979.2	979.4	979.5	0.2	0.3	0.4	979.2	979.3	0.2	0.8	0.5
n200w20	1134.6	1134.6	1134.6	0.0	0.2	0.4	1134.6	1134.6	0.0	0.7	0.4
n200w40	1152.8	1152.8	1152.8	0.0	0.5	0.5	1152.8	1152.8	0.0	1.2	0.5

*: Best-known solutions correspond to best solution values found by any run of GVNS-S and GVNS-Q for all instances.

Table 29: TSPTW-D: Results for the *DaSilva* instances.

		GVNS-S					GVNS-Q				
Inst.	BKS*	$best$	avg	avg_{sd}	t_{avg}	t_{sd}	$best$	avg	avg_{sd}	t_{avg}	t_{sd}
n200w100	10566.6	10566.6	10566.6	0.0	0.1	0.3	10566.6	10566.6	0.0	0.4	0.5
n200w200	10673.0	10673.0	10673.1	0.5	0.1	0.3	10673.0	10673.0	0.0	0.3	0.5
n200w300	10485.0	10485.0	10485.0	0.0	0.1	0.3	10485.0	10485.0	0.0	0.4	0.5
n200w400	10406.2	10406.2	10406.2	0.0	0.1	0.3	10406.2	10406.2	0.0	0.6	0.5
n200w500	10339.8	10339.8	10339.8	0.0	0.1	0.3	10339.8	10339.8	0.0	0.8	0.5
n250w100	13379.2	13379.2	13379.2	0.0	0.1	0.4	13379.2	13379.2	0.0	0.6	0.5
n250w200	13010.2	13010.2	13010.2	0.0	0.1	0.3	13010.2	13010.2	0.0	0.4	0.5
n250w300	13509.2	13509.2	13509.2	0.0	0.1	0.3	13509.2	13509.2	0.0	0.4	0.5
n250w400	13176.0	13176.0	13176.0	0.0	0.2	0.4	13176.0	13176.0	0.0	0.8	0.4
n250w500	13165.6	13165.6	13165.7	0.2	0.3	0.4	13165.6	13165.6	0.1	1.1	0.5
n300w100	15904.0	15904.0	15904.0	0.0	0.2	0.4	15904.0	15904.0	0.0	0.8	0.4
n300w200	16010.0	16010.0	16010.0	0.0	0.2	0.4	16010.0	16010.0	0.0	0.5	0.5
n300w300	15522.6	15522.6	15523.4	2.5	0.3	0.4	15522.6	15522.8	0.7	0.9	0.4
n300w400	15674.6	15674.6	15674.6	0.0	0.3	0.5	15674.6	15674.6	0.0	1.2	0.5
n300w500	16002.0	16002.0	16002.0	0.0	0.4	0.5	16002.0	16002.0	0.0	1.7	0.6
n350w100	18576.8	18576.8	18576.8	0.0	0.3	0.5	18576.8	18576.8	0.0	1.0	0.4
n350w200	18340.2	18340.2	18340.2	0.0	0.3	0.5	18340.2	18340.2	0.0	0.8	0.5
n350w300	18321.0	18321.0	18321.0	0.0	0.4	0.5	18321.0	18321.0	0.0	1.1	0.4
n350w400	18300.0	18300.0	18300.0	0.0	0.6	0.5	18300.0	18300.0	0.0	1.6	0.5
n350w500	18965.0	18965.0	18965.0	0.0	0.7	0.5	18965.0	18965.0	0.0	2.3	0.7
n400w100	20727.2	20727.2	20727.2	0.0	0.5	0.5	20727.2	20727.2	0.0	1.5	0.4
n400w200	21449.4	21449.4	21449.6	0.7	0.5	0.5	21449.4	21449.6	0.7	1.3	0.5
n400w300	21260.6	21260.6	21260.8	0.5	0.7	0.5	21260.6	21260.8	0.5	1.4	0.5
n400w400	20945.2	20945.2	20945.2	0.0	0.9	0.4	20945.2	20946.2	1.8	1.9	0.7
n400w500	20928.8	20928.8	20928.9	0.3	1.1	0.3	20928.8	20928.8	0.0	3.1	1.0

*: Best-known solutions correspond to best solution values found by any run of GVNS-S and GVNS-Q for all instances.

Table 30: TSPTW-D: Results for the *Pesant* instances.

		GVNS-S					GVNS-Q				
Inst.	BKS*	$best$	avg	avg_{sd}	t_{avg}	t_{sd}	$best$	avg	avg_{sd}	t_{avg}	t_{sd}
rc201.0	758.60	758.94	758.94	0.00	0.0	0.0	758.60	758.60	0.00	0.0	0.0
rc201.1	784.17	784.17	786.68	3.66	0.0	0.0	784.17	786.68	3.66	0.0	0.0
rc201.2	798.26	798.26	799.77	1.85	0.0	0.0	798.26	798.26	0.00	0.1	0.2
rc201.3	635.07	635.07	635.07	0.00	0.0	0.0	635.07	635.07	0.00	0.0	0.0
rc202.0	742.19	742.19	745.36	4.78	0.0	0.0	742.19	742.19	0.00	0.0	0.0
rc202.1	598.88	598.88	598.88	0.00	0.0	0.0	598.88	598.88	0.00	0.0	0.0
rc202.2	794.68	794.68	794.68	0.00	0.0	0.0	794.68	794.68	0.00	0.0	0.0
rc202.3	679.04	679.04	679.82	0.39	0.0	0.0	679.04	679.82	0.39	0.0	0.0
rc203.0	761.05	761.05	761.51	1.74	0.0	0.0	761.05	761.05	0.00	0.1	0.3
rc203.1	800.32	803.20	804.32	3.20	0.0	0.0	800.32	801.47	1.82	0.1	0.3
rc203.2	794.68	794.68	794.68	0.00	0.0	0.0	794.68	794.68	0.00	0.0	0.0
rc204.0	606.40	608.21	614.52	4.07	0.0	0.0	606.40	609.51	5.22	0.1	0.3
rc204.1	485.37	485.37	485.37	0.00	0.0	0.0	485.37	485.37	0.00	0.1	0.2
rc204.2	865.08	867.35	880.71	26.74	0.1	0.2	865.08	878.45	25.87	0.1	0.3
rc205.0	631.95	631.95	631.95	0.00	0.0	0.0	631.95	633.26	3.34	0.1	0.2
rc205.1	770.42	770.42	770.42	0.00	0.0	0.0	770.42	770.42	0.00	0.1	0.2
rc205.2	849.55	849.55	849.55	0.00	0.0	0.0	849.55	849.55	0.00	0.0	0.0
rc205.3	651.16	651.16	652.44	3.26	0.1	0.2	651.16	651.16	0.00	0.0	0.0
rc206.0	835.23	836.22	840.15	10.89	0.0	0.0	835.23	836.22	2.95	0.0	0.0
rc206.1	696.51	696.51	698.59	2.23	0.0	0.0	696.51	698.09	2.24	0.1	0.2
rc206.2	713.01	717.91	732.23	7.94	0.0	0.0	713.01	721.90	10.70	0.0	0.0
rc207.0	806.69	806.69	806.69	0.00	0.0	0.0	806.69	806.69	0.00	0.1	0.2
rc207.1	726.36	726.36	726.88	1.34	0.0	0.0	726.36	726.83	1.21	0.1	0.2
rc207.2	546.41	546.41	546.41	0.00	0.1	0.2	546.41	546.41	0.00	0.1	0.2
rc208.0	820.56	820.56	830.37	28.65	0.1	0.2	820.56	830.37	28.65	0.1	0.2
rc208.1	509.04	509.04	509.04	0.00	0.0	0.0	509.04	509.04	0.00	0.1	0.2
rc208.2	503.92	503.92	503.92	0.00	0.1	0.2	503.92	503.92	0.00	0.1	0.2

*: Best-known solutions correspond to best solution values found by any run of GVNS-S and GVNS-Q for all instances.

B.4 TSPTW-S

Tables 31-36 report, for all instance sets individually, the results obtained by GVNS-S and GVNS-Q for the TSPTW-S. Note that the BKS correspond to the best of the best solution values (not necessarily optimal) returned by the CP optimizer and the best solution values found by any run of our heuristics. In column *opt* for the results of CP optimizer, a "✓" indicates that an optimal solution has been found, a "-" indicates that no feasible solution is available, and an empty entry means that a feasible solution has been found but its optimality has not been proven.

Table 31: TSPTW-S: Results for the *Dumas* instances.

		CP optimizer				GVNS-S						GVNS-Q					
Inst.	BKS	$\#feas$	$\#opt$	$best$	t_{avg}	$\#improved$	$best$	avg	avg_{sd}	t_{avg}	t_{sd}	$\#improved$	$best$	avg	avg_{sd}	t_{avg}	t_{sd}
n20w20	2.6	5(5)	5(5)	2.6	0.2	0(5)	2.6	2.6	0.0	0.0	0.0	0(5)	2.6	2.6	0.0	0.0	0.0
n20w40	4.0	5(5)	5(5)	4.0	2.2	0(5)	4.0	4.0	0.0	0.0	0.0	0(5)	4.0	4.0	0.0	0.0	0.0
n20w60	10.2	5(5)	5(5)	10.2	2.1	0(5)	10.2	10.2	0.0	0.0	0.0	0(5)	10.2	10.2	0.0	0.0	0.0
n20w80	14.6	5(5)	5(5)	14.6	13.8	0(5)	14.6	14.6	0.0	0.0	0.0	0(5)	14.6	14.6	0.0	0.0	0.0
n20w100	26.4	5(5)	5(5)	26.4	19.1	0(5)	26.4	26.2	0.4	0.0	0.0	0(5)	26.4	26.3	0.3	0.1	0.2
n40w20	1.6	5(5)	5(5)	1.6	1.3	0(5)	1.6	1.6	0.0	0.0	0.0	0(5)	1.6	1.6	0.0	0.0	0.1
n40w40	6.2	5(5)	5(5)	6.2	14.3	0(5)	6.2	6.2	0.0	0.0	0.0	0(5)	6.2	6.2	0.0	0.1	0.2
n40w60	8.6	5(5)	5(5)	8.6	71.8	0(5)	8.6	8.6	0.0	0.0	0.0	0(5)	8.6	8.6	0.0	0.1	0.2
n40w80	13.8	5(5)	4(5)	13.2	366.6	1(5)	13.8	13.8	0.0	0.0	0.1	1(5)	13.8	13.8	0.0	0.1	0.3
n40w100	13.4	5(5)	5(5)	13.4	13.2	0(5)	13.4	13.4	0.0	0.0	0.0	0(5)	13.4	13.4	0.0	0.1	0.2
n60w20	1.8	5(5)	5(5)	1.8	20.1	0(5)	1.8	1.8	0.0	0.0	0.0	0(5)	1.8	1.8	0.0	0.1	0.3
n60w40	6.0	5(5)	5(5)	6.0	148.7	0(5)	6.0	6.0	0.0	0.0	0.1	0(5)	6.0	6.0	0.0	0.1	0.2
n60w60	3.3	5(5)	5(5)	5.2	126.8	0(5)	3.3	3.3	0.0	0.0	0.1	0(5)	3.3	3.3	0.0	0.1	0.3
n60w80	7.0	3(5)	2(3)	5.0	685.3	1(3)	7.3	7.3	0.0	0.0	0.2	1(3)	7.3	7.3	0.0	0.1	0.3
n60w100	14.0	4(5)	4(4)	13.5	333.7	0(4)	13.5	13.5	0.0	0.1	0.3	0(4)	13.5	13.5	0.0	0.2	0.4
n80w20	1.7	5(5)	5(5)	1.6	101.2	0(5)	1.7	1.7	0.0	0.0	0.2	0(5)	1.7	1.7	0.0	0.2	0.4
n80w40	3.8	5(5)	4(5)	3.8	483.2	0(5)	3.8	3.8	0.0	0.3	0.4	0(5)	3.8	3.8	0.0	0.7	0.4
n80w60	6.2	4(5)	3(4)	5.8	500.2	0(4)	5.8	5.8	0.0	0.1	0.3	0(4)	5.8	5.8	0.0	0.2	0.4
n80w80	10.0	2(5)	2(2)	13.0	381.1	0(2)	13.0	13.0	0.0	0.2	0.4	0(2)	13.0	13.0	0.0	0.3	0.5
n100w20	1.8	4(5)	4(4)	1.8	1.9	0(5)	1.8	1.8	0.0	0.1	0.3	0(5)	1.8	1.8	0.0	0.1	0.3
n100w40	4.5	2(5)	2(2)	5.5	227.0	0(2)	5.5	5.4	0.4	0.2	0.4	0(2)	5.5	5.4	0.4	0.3	0.4
n100w60	8.5	3(5)	3(3)	7.0	273.3	0(3)	7.0	7.0	0.0	0.3	0.4	0(3)	7.0	7.0	0.0	0.4	0.5
n150w20	2.0	2(5)	2(2)	1.5	478.6	0(2)	1.5	1.5	0.0	0.2	0.4	0(2)	1.5	1.5	0.0	0.4	0.5
n150w40	3.2	0(5)	0(0)	-	1800.0	0(0)	3.2	3.2	0.0	0.5	0.5	0(0)	3.2	3.2	0.0	0.8	0.4
n150w60	4.3	3(5)	3(3)	4.3	52.6	0(3)	4.3	4.3	0.0	0.8	0.4	0(3)	4.3	4.3	0.0	1.0	0.4
n200w20	3.0	1(5)	1(1)	3.0	38.7	0(1)	3.0	3.0	0.0	1.1	0.7	0(1)	3.0	3.0	0.0	2.0	0.6
n200w40	2.0	1(5)	1(1)	2.0	307.4	0(1)	2.0	2.0	0.0	1.4	0.5	0(1)	2.0	2.0	0.0	1.7	0.4

Table 32: TSPTW-S: Results for the *GDE* instances.

		CP optimizer				GVNS-S						GVNS-Q					
Inst.	BKS	$\#feas$	$\#opt$	$best$	t_{avg}	$\#improved$	$best$	avg	avg_{sd}	t_{avg}	t_{sd}	$\#improved$	$best$	avg	avg_{sd}	t_{avg}	t_{sd}
n20w120	52.6	5(5)	5(5)	52.6	3.3	0(5)	52.6	52.4	0.2	0.0	0.0	0(5)	52.6	52.6	0.1	0.0	0.0
n20w140	56.6	5(5)	5(5)	56.6	2.7	0(5)	56.6	56.6	0.0	0.0	0.0	0(5)	56.6	56.6	0.0	0.0	0.1
n20w160	60.6	5(5)	5(5)	60.6	6.0	0(5)	60.6	60.6	0.0	0.0	0.0	0(5)	60.6	60.6	0.0	0.1	0.2
n20w180	68.2	5(5)	5(5)	68.2	18.5	0(5)	68.2	68.2	0.0	0.0	0.0	0(5)	68.2	68.2	0.0	0.1	0.2
n20w200	80.0	5(5)	5(5)	80.0	3.9	0(5)	80.0	79.9	0.4	0.0	0.0	0(5)	80.0	80.0	0.0	0.1	0.2
n40w120	53.2	5(5)	5(5)	53.2	162.3	0(5)	53.2	53.2	0.0	0.0	0.1	0(5)	53.2	53.2	0.0	0.1	0.3
n40w140	57.6	5(5)	5(5)	57.6	78.3	0(5)	57.6	57.6	0.0	0.1	0.2	0(5)	57.6	57.6	0.0	0.1	0.3
n40w160	60.0	5(5)	5(5)	61.0	89.1	0(5)	60.0	60.0	0.0	0.1	0.2	0(5)	60.0	60.0	0.0	0.1	0.3
n40w180	65.6	5(5)	5(5)	65.6	3.7	0(5)	65.6	65.6	0.0	0.1	0.2	0(5)	65.6	65.6	0.0	0.1	0.3
n40w200	63.6	5(5)	5(5)	63.6	26.9	0(5)	63.6	63.6	0.0	0.1	0.2	0(5)	63.6	63.6	0.0	0.1	0.4
n60w120	52.8	5(5)	5(5)	52.8	128.1	0(5)	52.8	52.8	0.0	0.1	0.3	0(5)	52.8	52.8	0.0	0.4	0.5
n60w140	40.5	5(5)	5(5)	42.0	474.6	0(5)	40.5	40.5	0.0	0.2	0.4	0(5)	40.5	40.5	0.0	0.4	0.5
n60w160	59.8	4(5)	3(4)	59.8	531.5	0(5)	59.8	59.8	0.0	0.1	0.3	0(5)	59.8	59.8	0.0	0.3	0.5
n60w180	60.0	5(5)	3(5)	64.0	1000.3	0(5)	60.0	60.0	0.0	0.2	0.4	0(5)	60.0	60.0	0.0	0.4	0.5
n60w200	66.6	5(5)	4(5)	66.2	448.6	1(5)	66.6	66.6	0.0	0.2	0.4	1(5)	66.6	66.6	0.0	0.4	0.5
n80w100	11.0	0(5)	0(0)	-	1800.0	0(0)	11.0	11.0	0.0	0.2	0.4	0(0)	11.0	11.0	0.0	0.3	0.4
n80w120	54.0	4(5)	4(4)	54.0	72.2	0(4)	54.0	54.0	0.0	0.4	0.5	0(4)	54.0	54.0	0.0	0.9	0.4
n80w140	57.5	4(5)	4(4)	57.5	159.4	0(4)	57.5	57.5	0.0	0.4	0.5	0(4)	57.5	57.5	0.0	0.8	0.2
n80w160	57.3	4(5)	4(4)	57.3	216.9	0(4)	57.3	57.3	0.0	0.4	0.5	0(4)	57.3	57.3	0.0	0.7	0.5
n80w180	63.0	2(5)	0(2)	61.5	1800.2	1(2)	63.0	63.0	0.0	0.7	0.5	1(2)	63.0	63.0	0.0	1.2	0.4
n80w200	56.5	2(5)	2(2)	56.5	12.7	0(2)	56.5	56.5	0.0	0.4	0.5	0(2)	56.5	56.5	0.0	0.7	0.4
n100w80	13.7	3(5)	3(3)	13.7	179.4	0(3)	13.7	13.7	0.0	0.4	0.5	0(3)	13.7	13.7	0.0	0.6	0.5
n100w100	20.7	3(5)	3(3)	20.7	476.7	0(3)	20.7	20.7	0.0	0.5	0.5	0(3)	20.7	20.7	0.0	0.7	0.5
n100w120	57.0	4(5)	4(4)	57.0	130.6	0(4)	57.0	57.0	0.0	0.7	0.5	0(4)	57.0	57.0	0.0	1.4	0.5
n100w140	64.8	4(5)	4(4)	64.8	12.9	0(4)	64.8	64.8	0.0	0.8	0.4	0(4)	64.8	64.8	0.0	1.4	0.5
n100w160	62.8	4(5)	4(4)	62.8	55.2	0(4)	62.8	62.8	0.0	0.9	0.4	0(4)	62.8	62.8	0.0	1.5	0.5

Table 33: TSPTW-S: Results for the *OT* instances.

		CP optimizer			GVNS-S					GVNS-Q				
Inst.	BKS	opt	$best$	t_{avg}	$best$	avg	avg_{sd}	t_{avg}	t_{sd}	$best$	avg	avg_{sd}	t_{avg}	t_{sd}
n150w120.001	55	✓	55	239.8	55	55	0.0	4.9	1.0	55	55	0.0	6.4	1.1
n150w120.002	50	-	-	1800.0	50	50	0.0	4.5	1.0	50	50	0.0	5.6	0.8
n150w120.003	57	✓	57	9.1	57	57	0.0	3.5	1.0	57	57	0.0	4.7	1.1
n150w120.004	55	-	-	1800.0	55	55	0.0	3.2	0.4	55	55	0.0	4.6	0.5
n150w120.005	54	✓	54	192.5	54	54	0.0	3.7	0.5	54	54	0.0	4.7	0.8
n150w140.001	52	✓	52	16.6	52	52	0.0	3.7	0.6	52	52	0.0	4.4	0.6
n150w140.002	50	✓	50	214.6	50	50	0.0	3.9	0.6	50	50	0.0	4.7	0.6
n150w140.003	56	✓	56	195.1	56	56	0.0	5.7	1.1	56	56	0.0	5.9	1.1
n150w140.004	56		55	1800.2	56	56	0.0	4.1	0.9	56	56	0.0	4.9	1.0
n150w140.005	62	✓	62	157.7	62	62	0.0	4.4	0.5	62	62	0.0	5.5	0.7
n150w160.001	66	✓	66	348.2	66	66	0.0	5.3	0.8	66	66	0.0	6.2	1.0
n150w160.002	60	-	-	1800.0	60	60	0.0	5.9	1.1	60	60	0.0	6.6	1.2
n150w160.003	53	-	-	1800.0	53	53	0.0	4.7	0.9	53	53	0.0	5.3	1.1
n150w160.004	59	-	-	1800.0	59	59	0.0	6.0	0.7	59	59	0.0	7.1	1.0
n150w160.005	59	✓	59	442.2	59	59	0.0	4.8	0.8	59	59	0.0	5.6	0.8
n200w120.001	55		55	1800.3	55	55	0.0	13.0	2.2	55	55	0.0	14.9	2.5
n200w120.002	51	✓	51	526.1	51	51	0.0	10.8	1.3	51	51	0.0	12.1	1.4
n200w120.003	56	-	-	1800.0	56	56	0.0	13.8	1.5	56	56	0.0	15.9	1.7
n200w120.004	58	✓	58	35.7	58	58	0.0	14.8	1.6	58	58	0.0	16.9	1.8
n200w120.005	50	✓	50	582.2	50	50	0.0	11.3	1.9	50	50	0.0	12.6	2.2
n200w140.001	56		55	1800.2	56	56	0.0	13.5	2.0	56	56	0.0	14.5	2.4
n200w140.002	52	-	-	1800.0	52	52	0.0	13.9	2.5	52	52	0.0	14.8	2.2
n200w140.003	53	✓	53	305.6	53	53	0.0	11.1	1.5	53	53	0.0	11.7	1.7
n200w140.004	52		52	1800.3	52	52	0.0	10.5	0.9	52	52	0.0	11.2	1.2
n200w140.005	53		53	1800.2	53	53	0.0	14.1	1.8	53	53	0.0	12.4	1.1

Table 34: TSPTW-S: Results for the *AFG* instances.

		CP optimizer			GVNS-S					GVNS-Q				
Inst.	BKS	*opt*	*best*	t_{avg}	*best*	*avg*	avg_{sd}	t_{avg}	t_{sd}	*best*	*avg*	avg_{sd}	t_{avg}	t_{sd}
rbg010a	843	✓	843	0.1	843	843.0	0.0	0.0	0.0	843	843.0	0.0	0.0	0.0
rbg016a	116	✓	116	49.8	116	116.0	0.0	0.0	0.0	116	116.0	0.0	0.0	0.0
rbg016b	243	✓	243	0.5	243	243.0	0.0	0.0	0.0	243	243.0	0.0	0.0	0.0
rbg017.2	1074	✓	1074	79.2	1074	1074.0	0.0	0.0	0.0	1074	1074.0	0.0	0.0	0.0
rbg017	474	✓	474	65.9	474	474.0	0.0	0.0	0.0	474	474.0	0.0	0.0	0.0
rbg017a	6	✓	6	0.0	6	6.0	0.0	0.0	0.0	6	6.0	0.0	0.0	0.0
rbg019a	228	✓	228	9.4	228	228.0	0.0	0.0	0.0	228	228.0	0.0	0.0	0.0
rbg019b	635	✓	635	4.8	635	635.0	0.0	0.0	0.0	635	635.0	0.0	0.0	0.0
rbg019c	6	✓	6	0.0	6	6.0	0.0	0.0	0.0	6	6.0	0.0	0.0	0.0
rbg019d	518	✓	518	0.3	518	518.0	0.0	0.0	0.0	518	518.0	0.0	0.1	0.2
rbg020a	6	✓	6	0.1	6	6.0	0.0	0.0	0.0	6	6.0	0.0	0.0	0.0
rbg021.2	6	✓	6	0.0	6	6.0	0.0	0.0	0.0	6	6.0	0.0	0.0	0.0
rbg021.3	6	✓	6	0.1	6	6.0	0.0	0.0	0.0	6	6.0	0.0	0.0	0.0
rbg021.4	6	✓	6	0.1	6	6.0	0.0	0.0	0.0	6	6.0	0.0	0.0	0.0
rbg021.5	6	✓	6	0.1	6	6.0	0.0	0.0	0.0	6	6.0	0.0	0.0	0.0
rbg021.6	731		731	1800.1	731	731.0	0.0	0.0	0.0	731	731.0	0.0	0.0	0.0
rbg021.7	1331		1331	1800.0	1331	1327.1	14.7	0.0	0.0	1331	1331.0	0.0	0.0	0.0
rbg021.8	1931		1931	1800.1	1931	1931.0	0.0	0.0	0.0	1931	1931.0	0.0	0.0	0.0
rbg021.9	2531		2531	1800.0	2531	2531.0	0.0	0.0	0.0	2531	2531.0	0.0	0.0	0.0
rbg021	6	✓	6	0.1	6	6.0	0.0	0.0	0.0	6	6.0	0.0	0.0	0.0
rbg027a	6	✓	6	0.1	6	6.0	0.0	0.0	0.0	6	6.0	0.0	0.0	0.0
rbg031a	485	✓	485	157.3	485	485.0	0.0	0.0	0.0	485	485.0	0.0	0.1	0.2
rbg033a	484	✓	484	173.3	484	484.0	0.0	0.0	0.0	484	484.0	0.0	0.1	0.2
rbg034a	442	✓	442	431.6	442	442.0	0.0	0.0	0.0	442	442.0	0.0	0.1	0.2
rbg035a.2	505		505	1800.0	505	505.0	0.0	0.1	0.2	505	505.0	0.0	0.1	0.2
rbg035a	504	✓	504	3.3	504	504.0	0.0	0.1	0.2	504	504.0	0.0	0.1	0.2
rbg038a	431		431	1800.1	431	431.0	0.0	0.1	0.2	431	431.0	0.0	0.1	0.2
rbg040a	440		440	1800.1	440	440.0	0.0	0.0	0.0	440	440.0	0.0	0.1	0.3
rbg041a	359	✓	359	4958.4	359	359.0	0.0	0.1	0.2	359	359.0	0.0	0.1	0.2
rbg042a	236		227	1800.0	236	235.8	0.7	0.1	0.2	236	236.0	0.0	0.3	0.4
rbg048a	6	✓	6	0.1	6	6.0	0.0	0.1	0.2	6	6.0	0.0	0.1	0.2
rbg049a	32		32	1800.0	32	32.0	0.0	0.0	0.0	32	32.0	0.0	0.1	0.2
rbg050a	574		574	1800.1	574	574.0	0.0	0.1	0.2	574	574.0	0.0	0.2	0.4
rbg050b	39		39	1800.1	39	39.0	0.0	0.0	0.0	39	39.0	0.0	0.1	0.3
rbg050c	23		23	1800.0	23	23.0	0.0	0.1	0.2	23	23.0	0.0	0.1	0.3
rbg055a	442		442	1800.1	442	442.0	0.0	0.1	0.2	442	442.0	0.0	0.2	0.4
rbg067a	442		442	1800.1	442	442.0	0.0	0.1	0.2	442	442.0	0.0	0.3	0.5
rbg086a	38		38	1800.1	38	38.0	0.0	0.1	0.2	38	38.0	0.0	0.4	0.5
rbg092a	38		38	1800.0	38	38.0	0.0	0.1	0.2	38	38.0	0.0	0.5	0.5
rbg125a	442		442	1800.1	442	441.4	2.2	1.2	0.4	442	442.0	0.0	1.9	0.2
rbg132.2	823		823	1800.0	823	823.0	0.0	1.8	0.4	823	823.0	0.0	2.3	0.4
rbg132	223		223	1800.1	223	223.0	0.0	0.5	0.5	223	223.0	0.0	1.4	0.5
rbg152.3	1423		1423	1800.0	1423	1423.0	0.0	5.3	0.9	1423	1423.0	0.0	5.1	0.5
rbg152	223		223	1800.1	223	223.0	0.0	1.2	0.4	223	223.0	0.0	2.6	0.5
rbg172a	322		322	1800.0	322	320.3	0.7	3.5	0.5	322	321.7	0.7	5.9	1.6
rbg193.2	695		695	1800.1	695	695.0	0.0	7.5	0.9	695	695.0	0.0	9.6	0.8
rbg193	146		146	1800.1	146	146.0	0.0	1.9	0.2	146	146.0	0.0	4.7	0.6
rbg201a	322		322	1800.1	322	320.9	1.0	6.0	1.3	322	321.6	0.8	10.5	2.2
rbg233.2	695		695	1800.2	695	693.9	2.7	17.5	2.5	695	695.0	0.0	22.3	0.9
rbg233	146		146	1800.1	146	146.0	0.0	2.9	0.6	146	146.0	0.0	8.0	0.6

Table 35: TSPTW-S: Results for the *Pesant* instances.

		CP optimizer			GVNS-S					GVNS-Q				
Inst.	BKS	*opt*	*best*	t_{avg}	*best*	*avg*	avg_{sd}	t_{avg}	t_{sd}	*best*	*avg*	avg_{sd}	t_{avg}	t_{sd}
rc201.0	15	✓	15	29.1	15	15.0	0.0	0.0	0.0	15	15.0	0.0	0.1	0.2
rc201.1	10		10	1800.0	10	10.0	0.0	0.0	0.0	10	10.0	0.0	0.1	0.2
rc201.2	34	✓	34	23.6	34	34.0	0.0	0.0	0.0	34	34.0	0.0	0.1	0.2
rc201.3	63	✓	63	12.6	63	63.0	0.0	0.0	0.0	63	63.0	0.0	0.1	0.2
rc202.0	70	✓	70	3.4	70	70.0	0.0	0.0	0.0	70	70.0	0.0	0.1	0.2
rc202.1	50	✓	50	709.6	50	50.0	0.0	0.0	0.0	50	50.0	0.0	0.0	0.0
rc202.2	16		16	1800.0	16	16.0	0.0	0.0	0.0	16	16.0	0.0	0.0	0.0
rc202.3	76	✓	76	32.5	76	72.3	9.5	0.0	0.0	76	76.0	0.0	0.1	0.2
rc203.0	43		43	1800.1	43	43.0	0.0	0.0	0.0	43	43.0	0.0	0.1	0.2
rc203.1	58		36	1800.1	58	58.0	0.0	0.1	0.2	58	58.0	0.0	0.1	0.2
rc203.2	32	✓	32	92.3	32	31.7	0.7	0.0	0.0	32	31.7	0.7	0.0	0.0
rc204.0	78	✓	78	2.7	78	78.0	0.0	0.0	0.0	78	78.0	0.0	0.1	0.2
rc204.1	82		82	1800.1	82	82.0	0.0	0.0	0.0	82	82.0	0.0	0.1	0.2
rc204.2	67	-	-	1800.0	67	66.6	1.0	0.1	0.2	67	66.6	1.0	0.1	0.3
rc205.0	36	✓	36	98.4	36	36.0	0.0	0.0	0.0	36	36.0	0.0	0.1	0.2
rc205.1	48	✓	48	0.6	48	48.0	0.0	0.0	0.0	48	48.0	0.0	0.0	0.0
rc205.2	20	✓	20	757.7	20	18.7	4.7	0.0	0.0	20	18.7	4.7	0.0	0.0
rc205.3	19	✓	19	192.9	19	19.0	0.0	0.1	0.2	19	19.0	0.0	0.1	0.2
rc206.0	27		21	1800.0	27	16.6	4.1	0.0	0.0	27	16.6	4.1	0.1	0.2
rc206.1	62	✓	62	595.2	62	62.0	0.0	0.1	0.2	62	62.0	0.0	0.1	0.3
rc206.2	51	✓	51	1211.8	51	50.2	1.3	0.0	0.0	51	50.2	1.3	0.1	0.2
rc207.0	22		22	1800.1	22	19.5	3.9	0.1	0.2	22	19.5	3.9	0.1	0.2
rc207.1	66		66	1800.1	66	65.6	0.5	0.1	0.2	66	65.6	0.5	0.1	0.3
rc207.2	95		95	1800.0	95	95.0	0.0	0.1	0.2	95	95.0	0.0	0.1	0.2
rc208.0	66	-	-	1800.0	66	62.7	8.5	0.1	0.2	66	62.7	8.5	0.1	0.3
rc208.1	224		224	1800.1	224	224.0	0.0	0.0	0.0	224	224.0	0.0	0.1	0.2
rc208.2	200		196	1800.1	200	200.0	0.0	0.1	0.2	200	200.0	0.0	0.1	0.3

Table 36: TSPTW-S: Results for the *Potvin* instances.

		CP optimizer			GVNS-S					GVNS-Q				
Inst.	BKS	*opt*	*best*	t_{avg}	*best*	*avg*	avg_{sd}	t_{avg}	t_{sd}	*best*	*avg*	avg_{sd}	t_{avg}	t_{sd}
rc_201.1	41	✓	41	4.3	41	41.0	0.0	0.0	0.0	41	41.0	0.0	0.0	0.0
rc_201.2	21	✓	21	46.4	21	21.0	0.0	0.1	0.2	21	21.0	0.0	0.1	0.2
rc_201.3	31	✓	31	36.5	31	30.3	1.4	0.0	0.0	31	30.3	1.4	0.0	0.0
rc_201.4	8	✓	8	1478.7	8	8.0	0.0	0.0	0.0	8	8.0	0.0	0.0	0.0
rc_202.1	14		14	1800.0	14	13.2	3.0	0.0	0.0	14	14.0	0.0	0.0	0.0
rc_202.2	120	✓	120	0.4	120	120.0	0.0	0.0	0.0	120	120.0	0.0	0.0	0.0
rc_202.3	26	✓	26	37.3	26	25.6	1.5	0.0	0.0	26	25.6	1.5	0.0	0.0
rc_202.4	28		28	1800.1	28	28.0	0.0	0.0	0.0	28	28.0	0.0	0.0	0.0
rc_203.1	41		41	1800.0	41	41.0	0.0	0.0	0.0	41	41.0	0.0	0.1	0.2
rc_203.2	56	✓	56	574.5	56	56.0	0.0	0.0	0.0	56	56.0	0.0	0.1	0.2
rc_203.3	31		31	1800.0	31	31.0	0.0	0.0	0.0	31	31.0	0.0	0.1	0.3
rc_203.4	120	✓	120	0.3	120	120.0	0.0	0.0	0.0	120	120.0	0.0	0.0	0.0
rc_204.1	36	-	-	1800.0	36	35.4	0.5	0.1	0.2	36	35.4	0.5	0.1	0.3
rc_204.2	70		70	1800.1	70	70.0	0.0	0.0	0.0	70	70.0	0.0	0.1	0.2
rc_204.3	80		80	1800.0	80	80.0	0.0	0.0	0.0	80	80.0	0.0	0.0	0.0
rc_205.1	55	✓	55	14.6	55	55.0	0.0	0.0	0.0	55	55.0	0.0	0.0	0.0
rc_205.2	3	✓	3	54.6	3	3.0	0.0	0.0	0.0	3	3.0	0.0	0.0	0.0
rc_205.3	10	✓	10	472.7	10	10.0	0.0	0.0	0.0	10	10.0	0.0	0.0	0.0
rc_205.4	17	✓	17	907.7	17	17.0	0.0	0.0	0.0	17	17.0	0.0	0.1	0.2
rc_206.1	218	✓	218	0.1	218	215.5	6.5	0.0	0.0	218	215.5	6.5	0.0	0.0
rc_206.2	46		42	1800.1	46	46.0	0.0	0.0	0.0	46	46.0	0.0	0.1	0.2
rc_206.3	99	✓	99	84.9	99	99.0	0.0	0.0	0.0	99	99.0	0.0	0.1	0.2
rc_206.4	41	-	-	1800.0	41	41.0	0.0	0.0	0.0	41	41.0	0.0	0.1	0.2
rc_207.1	61		61	1800.1	61	61.0	0.0	0.0	0.0	61	61.0	0.0	0.1	0.3
rc_207.2	63		63	1800.1	63	63.0	0.0	0.0	0.0	63	63.0	0.0	0.1	0.2
rc_207.3	94		92	1800.0	94	92.5	1.1	0.1	0.2	94	92.5	1.1	0.1	0.2
rc_207.4	250	✓	250	0.1	250	250.0	0.0	0.0	0.0	250	250.0	0.0	0.0	0.0
rc_208.1	97	-	-	1800.0	97	91.9	3.6	0.1	0.2	97	91.9	3.6	0.1	0.3
rc_208.2	199		197	1800.0	199	199.0	0.0	0.1	0.2	199	199.0	0.0	0.1	0.2
rc_208.3	181		176	1800.0	181	180.5	1.4	0.1	0.3	181	180.5	1.4	0.3	0.4

Table 37: TSPTW-S: Results for the *DaSilva* instances.

		GVNS-S					GVNS-Q				
Inst.	BKS*	$best$	avg	avg_{sd}	t_{avg}	t_{sd}	$best$	avg	avg_{sd}	t_{avg}	t_{sd}
n200w100	3.4	3.4	3.4	0.0	0.2	0.4	3.4	3.4	0.0	0.6	0.5
n200w200	11.4	11.4	11.3	0.4	0.3	0.5	11.4	11.4	0.0	0.7	0.4
n200w300	13.2	13.2	13.1	0.3	0.5	0.5	13.2	13.1	0.3	1.0	0.3
n200w400	25.2	25.2	23.5	1.9	0.7	0.4	25.2	25.2	0.0	1.4	0.5
n200w500	32.4	32.4	32.4	0.0	1.1	0.5	32.4	32.4	0.0	2.3	0.6
n250w100	2.8	2.8	2.8	0.0	0.4	0.5	2.8	2.8	0.0	0.8	0.4
n250w200	8.2	7.8	7.5	0.6	0.5	0.5	8.2	8.0	0.6	0.9	0.5
n250w300	14.0	14.0	14.0	0.0	0.8	0.5	14.0	14.0	0.1	1.5	0.7
n250w400	20.6	20.6	19.6	2.0	1.1	0.5	20.6	20.6	0.1	1.9	0.7
n250w500	25.4	25.4	25.4	0.0	1.7	0.5	25.4	25.4	0.0	2.8	0.7
n300w100	4.0	4.0	4.0	0.0	0.9	0.4	4.0	4.0	0.0	2.0	0.5
n300w200	7.4	7.4	7.3	0.3	0.8	0.5	7.4	7.3	0.3	1.3	0.7
n300w300	12.8	12.8	11.8	1.6	1.1	0.6	12.8	12.4	0.8	2.0	0.9
n300w400	14.2	14.2	13.0	1.9	1.7	0.6	14.2	14.1	0.4	2.7	0.6
n300w500	25.2	24.8	24.6	0.5	3.2	0.8	25.2	24.9	0.3	4.7	0.8
n350w100	3.0	3.0	3.0	0.0	1.2	0.4	3.0	3.0	0.0	2.2	0.5
n350w200	6.8	6.8	6.3	0.7	1.0	0.5	6.8	6.4	0.7	1.6	0.6
n350w300	15.6	15.4	13.7	2.6	1.7	0.9	15.6	14.6	1.7	2.9	1.2
n350w400	17.8	17.8	15.8	2.7	2.9	1.0	17.8	16.1	2.5	4.7	1.8
n350w500	31.6	30.2	28.6	2.1	4.4	1.0	31.6	31.5	0.3	8.0	2.0
n400w100	4.0	4.0	4.0	0.0	2.3	0.5	4.0	4.0	0.0	4.0	0.7
n400w200	8.6	8.6	7.7	1.0	1.7	0.8	8.6	7.5	1.2	2.5	1.0
n400w300	9.6	9.0	7.8	1.6	2.0	1.0	9.6	8.4	1.6	3.4	1.3
n400w400	20.8	16.8	14.3	3.3	3.1	1.4	20.8	19.1	3.3	5.2	2.1
n400w500	28.8	24.4	22.0	2.6	5.3	1.8	28.8	28.2	1.0	8.1	2.2

*: Best-known solutions correspond to the best values found by any run of GVNS-S and GVNS-Q for all instances.

B.5 RTSPTW($\mathcal{T}_K$)

Tables 38 and 39 report the results obtained by solving the RTSPTW($\mathcal{T}_K$) using GVNS-S with $\eta_{rep} = 10$. Note that the averages were computed only considering the instances with a feasible solution found by Bartolini et al. (2021) with a time limit of 1200 seconds.

Table 38: RTSPTW($\mathcal{T}_K$)-T: Results for the GDE-D instances with up to 80 customers.

	$\Delta = 20$						$\Delta = 40$						$\Delta = 60$					
		GVNS-S						GVNS-S						GVNS-S				
Inst.	BKS	*best*	*avg*	avg_{sd}	t_{avg}	t_{sd}	BKS	*best*	*avg*	avg_{sd}	t_{avg}	t_{sd}	BKS	*best*	*avg*	avg_{sd}	t_{avg}	t_{sd}
n20w120	281.0	281.0	281.0	0.0	0.1	0.3	299.0	299.0	299.0	0.0	0.1	0.2	300.8	300.8	300.8	0.0	0.1	0.2
n20w140	236.2	236.2	236.2	0.0	0.1	0.3	242.2	242.2	242.2	0.0	0.1	0.2	243.0	243.0	243.0	0.0	0.1	0.3
n20w160	232.6	232.6	232.6	0.0	0.1	0.3	233.2	233.2	233.2	0.0	0.1	0.2	233.6	233.6	234.1	0.6	0.1	0.2
n20w180	240.2	240.2	240.2	0.0	0.1	0.3	250.6	250.6	250.6	0.0	0.1	0.3	252.8	252.8	252.8	0.0	0.1	0.3
n20w200	243.2	243.2	243.2	0.0	0.1	0.3	248.8	248.8	248.8	0.1	0.1	0.2	249.8	249.8	249.8	0.0	0.1	0.3
n40w120	389.0	389.0	389.0	0.0	0.7	0.5	403.6	403.6	403.6	0.0	0.6	0.5	404.6	404.6	404.6	0.0	0.7	0.5
n40w140	378.0	378.0	378.0	0.0	0.8	0.4	381.2	381.2	381.2	0.0	0.7	0.5	382.0	382.0	382.0	0.0	0.8	0.4
n40w160	335.4	335.4	335.4	0.0	1.0	0.3	337.0	337.0	337.0	0.0	0.9	0.4	337.0	337.0	337.1	0.5	0.9	0.4
n40w180	337.2	337.2	337.2	0.0	1.0	0.3	340.8	340.8	340.8	0.1	0.9	0.5	349.8	349.8	350.0	0.7	0.9	0.5
n40w200	319.4	319.4	319.4	0.0	1.1	0.4	323.2	323.2	323.2	0.0	1.1	0.2	323.4	323.4	323.5	0.3	1.2	0.6
n60w120	474.2	474.2	474.3	0.1	4.0	0.8	476.0	476.0	476.0	0.0	4.0	1.1	478.2	478.2	478.2	0.0	4.1	1.5
n60w140	452.0	452.0	452.0	0.0	4.1	0.8	455.8	455.8	456.0	0.4	4.0	1.1	453.0	453.0	453.1	0.4	4.8	1.9
n60w160	480.4	480.4	480.4	0.0	4.9	0.9	481.6	481.6	481.6	0.0	4.3	0.8	481.8	481.8	481.8	0.0	4.2	1.2
n60w180	427.2	427.2	427.2	0.0	5.0	0.9	432.4	432.4	432.9	0.8	5.1	1.5	434.4	434.4	434.4	0.0	5.3	2.0
n60w200	431.0	431.0	431.0	0.1	5.8	1.2	431.2	431.2	431.2	0.1	5.8	1.9	431.2	431.2	431.4	0.2	6.0	2.4
n80w100	577.3	577.3	577.6	0.5	12.2	4.5	-	-	-	-	-	-	-	-	-	-	-	-
n80w120	553.6	553.6	553.6	0.0	13.7	2.7	562.8	562.8	562.8	0.0	13.7	4.7	564.6	564.6	564.8	0.3	16.2	5.9
n80w140	524.2	524.2	524.2	0.0	14.3	2.4	528.4	528.6	528.7	0.4	13.9	3.9	537.8	537.8	538.3	0.6	16.3	6.9
n80w160	514.6	514.6	514.7	0.3	19.3	4.4	520.6	520.6	520.9	0.8	19.1	6.7	522.6	522.8	523.0	0.5	20.6	10.4
n80w180	513.6	513.6	514.0	0.3	17.1	3.4	517.0	517.0	517.0	0.0	19.3	6.8	518.8	518.8	518.8	0.1	23.5	11.1
n80w200	486.8	486.8	486.8	0.1	20.2	3.7	490.4	490.4	490.5	0.2	20.9	5.6	492.0	492.8	492.8	0.0	22.4	8.3

Table 39: RTSPTW($\mathcal{T}_K$)-T: Results for the GDE-I instances with up to 80 customers.

	$\Delta = 20$						$\Delta = 40$						$\Delta = 60$					
		GVNS-S						GVNS-S						GVNS-S				
Inst.	BKS	*best*	*avg*	avg_{sd}	t_{avg}	t_{sd}	BKS	*best*	*avg*	avg_{sd}	t_{avg}	t_{sd}	BKS	*best*	*avg*	avg_{sd}	t_{avg}	t_{sd}
n20w120	291.0	291.0	291.0	0.0	0.1	0.2	307.6	307.6	307.6	0.0	0.1	0.2	320.7	320.7	320.7	0.0	0.1	0.2
n20w140	236.2	236.2	236.2	0.0	0.1	0.2	253.4	253.4	253.4	0.0	0.1	0.2	246.7	246.7	246.7	0.0	0.0	0.1
n20w160	233.4	233.4	233.4	0.0	0.1	0.2	235.0	235.0	235.0	0.0	0.1	0.2	243.4	243.4	243.4	0.0	0.1	0.2
n20w180	242.4	242.4	242.4	0.0	0.1	0.3	254.0	254.0	254.1	0.1	0.1	0.3	256.0	256.0	256.0	0.0	0.1	0.2
n20w200	243.2	243.2	243.2	0.0	0.1	0.3	252.8	252.8	252.9	0.3	0.1	0.3	257.8	257.8	257.9	0.2	0.1	0.2
n40w120	391.6	391.6	391.6	0.0	0.7	0.4	412.8	412.8	412.8	0.0	0.6	0.5	437.0	437.0	437.0	0.0	0.6	0.5
n40w140	378.4	378.4	378.4	0.0	0.8	0.5	382.0	382.0	382.0	0.0	0.7	0.5	384.3	384.3	384.3	0.0	0.7	0.5
n40w160	338.2	338.2	338.3	0.2	1.0	0.3	342.4	342.4	342.4	0.0	0.9	0.5	349.2	349.2	349.2	0.0	0.9	0.5
n40w180	340.6	340.6	340.6	0.0	0.9	0.3	349.6	349.6	349.6	0.0	0.8	0.5	366.6	366.6	367.1	1.4	0.9	0.5
n40w200	321.2	321.2	321.2	0.0	1.2	0.4	331.4	331.4	331.4	0.0	1.2	0.5	340.5	340.5	340.5	0.0	1.1	0.5
n60w120	478.8	478.8	478.9	0.1	4.2	0.9	488.8	488.8	488.8	0.0	3.5	0.9	-	-	-	-	-	-
n60w140	451.0	451.0	451.0	0.1	4.0	0.7	469.8	469.8	469.9	0.3	4.4	1.3	-	-	-	-	-	-
n60w160	485.8	485.8	485.8	0.0	4.5	0.8	492.8	492.8	492.8	0.0	4.1	0.9	502.6	502.6	502.6	0.0	3.8	1.1
n60w180	428.4	428.4	428.4	0.0	5.3	1.1	436.6	436.6	436.6	0.0	5.1	1.2	447.6	447.6	447.6	0.0	5.1	1.5
n60w200	431.2	431.2	431.2	0.1	6.1	1.6	441.6	441.6	441.6	0.0	5.6	1.7	447.2	447.2	447.2	0.0	5.5	2.0
n80w100	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-
n80w120	554.0	554.0	554.0	0.0	14.2	3.2	571.6	571.6	571.6	0.0	12.4	3.4	595.0	595.0	595.0	0.0	14.0	4.4
n80w140	524.8	524.8	524.8	0.0	15.5	2.9	536.8	536.8	536.9	0.1	13.2	3.1	551.0	551.0	551.0	0.0	15.7	5.5
n80w160	517.8	517.8	517.9	0.3	19.5	5.9	529.2	529.2	529.3	0.3	15.7	4.8	544.8	544.8	544.8	0.0	18.2	6.5
n80w180	513.8	513.8	514.0	0.4	18.0	3.8	524.2	524.2	524.2	0.1	20.2	7.2	537.4	537.4	537.9	1.0	25.0	10.0
n80w200	488.0	488.0	488.0	0.0	20.9	5.0	497.8	497.8	497.9	0.1	22.5	6.7	510.4	510.4	510.4	0.0	25.1	10.7