Contents lists available at GrowingScience

International Journal of Industrial Engineering Computations

homepage: www.GrowingScience.com/ijiec

A robust approach for solving a vehicle routing problem with time windows with uncertain service and travel times

Mehdi Nasri[a*], Abdelmoutalib Metrane[a,b], Imad Hafidi[a] and Anouar Jamali[b]

[a]National School of Applied Sciences Khouribga, Sultan Moulay Slimane University, Beni-Mellal, Morocco
[b]Mohamed 6 Polytechnic University, Ben Guérir, Morocco

C H R O N I C L E

Article history:
Received March 23 2019
Received in Revised Format June 26 2019
Accepted July 6 2019
Available online July 9 2019

Keywords:
Robust approach
ALNS
Uncertainty
Measures of robustness
Monte-Carlo simulation

A B S T R A C T

The main purpose of this paper is to study the vehicle routing problem with hard time windows where the main challenges is to include both sources of uncertainties, namely the travel and the service time that can arise due to multiple causes. We propose a new resolution approach for the robust problem based on the implementation of an adaptive large neighborhood search algorithm and the use of efficient mechanisms to derive the best robust solution that responds to all uncertainties with reduced running times. The computational experiments are performed and improve the objective function of a set of instances with different levels of the uncertainty polytope to obtain the best robust solutions that protect from the violation of time windows for different scenarios.

1. Introduction

Since the pioneer paper of Dantzig and Ramser (1959) on the truck dispatching problem appeared at the end of the fifties of the last century, work in the field of the vehicle routing problem (VRP) has increased exponentially. Using a method based on a linear programming formulation, the authors of this work produced by hand calculations a near optimal-solution with four routes of a fleet of gasoline delivery trucks between a bulk terminal and twelve service stations supplied by a terminal. Nowadays, vehicle routing problem is considered as one of the most outstanding research achievements in the story of operations research and particularly in practice. There are important advances and new challenges that have been raised during the last few years such as radio frequency identification, and parallel computing (e.g. Pillac et al., 2013; Montoya-Torres, 2015). The class of VRP problems involves minimizing a travel distance of vehicles, starting and ending from a depot, to serve some customers. Typically the solution has to obey several other constraints, such as the consideration of travel, service, and waiting times together with time-window restrictions. This variant is called in the literature vehicle routing problem with time windows (Bräysy & Gendreau, 2005; Kallehauge et al., 2005; Rincon-Garcia et al., 2015).

* Corresponding author
E-mail: nasri.mathh@gmail.com (M. Nasri)

2020 Growing Science Ltd.
doi: 10.5267/j.ijiec.2019.7.002

For instance, three types of solution approaches can be used to solve these types of problems. First, the exact methods assert that the optimal is found if the method is given sufficiently in time and space. We cannot expect to construct exact algorithms which solve NP-hard problems. Second, the heuristics are solution methods that can quickly achieve a feasible solution in a reasonable quality. A special class called metaheuristics provides a high quality solution (Labadie et al., 2016). The third class of solutions is also a special class of heuristic which provides a guarantee on the error combined with near-optimal solution. An interesting topic on solving VRP consists in considering parameters affected by uncertainty, making the problem more realistic.

Different approaches have been proposed to deal with uncertainties in a VRP problem either in demand, travel time and/or service time. Among them, the stochastic approaches of vehicle routing problem SVRP have been treated in series of papers (Dror et al., 1993; Dror & Trudeau, 1986; Gendreau et al., 1996). The aim of the SVRP methodology is to find a near-best solution of the objective function responding to all possible data uncertainty. An alternative approach to handle the uncertain parameters is the robust optimization in which one can optimize against the worst scenario that can be generated from the source of uncertainty by using bi-objective function (Yousefi et al., 2017) and is immunized against this uncertainty (Sungur et al., 2008). In this context, the literature coats a large number of applications such as scheduling (Goren & Sabuncuoglu, 2008; Hazir et al., 2010), facility location (Minoux, 2010; Baron et al., 2011 ; Alumur et al., 2012 ; Gülpinar et al., 2013), inventory (Bienstock & Özbay, 2008; Ben-Tal et al.,2009), finance (Fabozzi et al., 2007; Gülpinar & Rustem, 2007; Pinar, 2007). In particular, the authors proposed a mathematical model for the robust optimization with uncertain demands (Moghaddam et al., 2012), heterogeneous fleet (Noorizadegan et al., 2012), routing with capacity (Sungur et al., 2008; Gounaris et al., 2013) and rail shuttle Routing problem (Rouky et al., 2018). For instance, this is equivalent to the deterministic case studied by Miller-Tucker-Zemlin formulation of the used VRP. We refer the reader to an excellent survey and tutorial of the robust vehicle routing proposed by Ordóñez (2010). We note that uncertainty in travel cost could be handled using the robust combinatorial optimization approach. Wu et al. (2017) proposed a linear model evaluated on a set of random instances for the vehicle routing problem with uncertain travel time to improve the robustness of the solution which enhance its quality compared with the worst case on a majority of scenarios. In the same spirit, Toklu et al. (2013) treated the VRP problem with capacity and uncertain travel costs based on a variant of the ant colony algorithm to generate sets of solutions of uncertainty levels and to analyze their effects on the problem.

The stochastic approach is also applicable for the vehicle routing problem with time window constraints (VRPTW). Errico et al. (2016) formulated the VRPTW with stochastic service times as a set partitioning problem and solve it by exact branch-cut-and-price algorithms. More precisely, they elaborated efficient algorithms by choosing label components, developing lower and upper bounds on partial route reduced the cost to be used in the column generation step. Unlike this approach, robust optimization seeks to get good solutions for the VRPTW problem by only considering nominal values and deviations possible uncertain data. Many works tackled the vehicle routing problem with time windows and uncertain travel times (Sungur et al., 2008). Agra et al. (2012) presented a general approach to the robust VRPTW problem with uncertain travel times. Travel times belong to a demand uncertainty polytope, which makes the problem more complex to solve than its deterministic equivalent. The advantage of the addition in complexity is that the model from Agra et al. (2012) is more usefule than the one from Sungur et al. (2008) and leads to less conservative robust solutions. Toklu et al. (2013) adapted their approach to solve the problem of VRPTW with uncertain travel times, whose objective was to minimize window time violation penalties by providing the decision makers a group of solutions found over several degrees of uncertainties considered. Agra et al. (2013) studied the VRPTW with uncertain travel times and proposed two robust formulations of the problem. The first extends the formulation of inequalities of resources and the second generalizes the formulation of inequalities of way. Their results show that the solution times are similar for both formulations while being significantly faster than the solutions times of a layered formulation recently proposed for the problem. Rouky et al. (2018) considered the vehicle routing problem with time windows (VRPTW) by introducing both uncertainties: the travel times of locomotives and the transfer times of shuttle as a model of the Rail Shuttle Routing Problem (RSRP) in the Le Havre port. In order to solve this problem with uncertainties, the authors proposed the Robust Ant Colony Optimization (RACO). The aim of this paper is to study the similar robust VRPTW including both uncertainties in travel times and service times as in Rouky et al. (2018). Our contribution to all previous works lies, first on the choice of the Adaptive Large Neighborhood Search (ALNS) metaheuristic. Moreover, the numerical results were tested on a set of small instances based on the reference of Solomon benchmark, and large instances of Gehring & Homberger's benchmark. The studied problem generated different scenarios and each scenario is performed by using the best known sampling method, Metropolis version algorithm of Monte Carlo simulation. It is worth mentioning that the ALNS approach used three different removal operators to maintain the diversity during the searching process namely: proximity operator, route portion operator and longest detour operator, and one repair operator based on the greedy insertion. For a complete description of this ALNS approach including construction and destruction operators, we refer the reader to the third section.

The remainder of this paper is structured in the following way. First of all, we define the problem, and we introduce its mathematical formulation. Next, we present our robust approach that is meant to solve the problem including a presentation of the ALNS preprocessing, destruction and insertion heuristics. Moreover, we evaluate the new approach using both the exact algorithm and the best-known heuristics. A detailed computational and comparative study is presented in Section 4 in order to provide perfectly robust conclusions. Finally, some concluding remarks are discussed.

2. Problem statement

This section is devoted to the statement of the vehicle routing problem with time windows under travel time and service time uncertainty. First, we introduce the deterministic model of the VRP problem which consists of an optimization of the total distance traveled by all vehicles under four constraints. Next, the service at any customer starts within a given time interval and it is not allowed to arrive late. Furthermore, if the vehicle arrives too early at a customer, it must wait until the time window opens. Taking into consideration these two constraints on time windows we transform the VRP problem to its VRPTW variant. To complete our statment of the problem, we introduce the source of uncertainties namely travel times and service times which makes the problem harder to solve than its deterministic counterpart. We suggest a new formulation of the uncertainty that was inspired by the recent work of Rouky et al. (2018) where the travel times and service times uncertainties belongs to a convex polytope in the travel time interval $[t_{ij}, t_{ij} + \Delta_{ij}]$ and service time interval $[P_i, P_i + \delta_i]$, where t_{ij} and P_i denote the nominal values, Δ_{ij} and δ_i are the maximum positive perturbations. The complexity of this problem leads us to look for robust solutions and therefore to min-max the objective function, this is the last part of the state of the art of our problem. Now, in order to describe our problem, let us denote the set of nodes by *N*, using *i* and *j* to denote general nodes, the depot will be denoted by *o*. The set of arcs is denoted as *A* and contains pairs of nodes, *(i, j)*. The set of vehicles is called *V* with elements *k*. Now we can assign to each edge *(i, j)* a cost t_{ij}, and to each node *i* a time window $[a_i, b_i]$. Then x_{ij}^k are binary decision variables that take the value 1 if vehicle *k* uses the edge *(i, j)* and 0, otherwise. The deterministic model of the VRP can be stated as follows:

$$min \sum_{k \in V} \sum_{(i,j) \in A} x_{ij}^k \mathrm{t_{ij}}$$

subject to

$$\sum_{k \in V} \sum_{j \in N} x_{ij}^k = 1 \quad \forall \, (i \in N) \tag{1}$$

$$\sum_{j\in N} x_{0j}^k = 1 \ \forall\, (k \in V) \tag{2}$$

$$\sum_{i\in N} x_{ih}^k - \sum_{j\in N} x_{hj}^k = 0 \ \forall (h \in N)\, \forall (k \in V) \tag{3}$$

$$\sum_{i\in N} x_{i0}^k = 1 \ \forall\, (k \in V) \tag{4}$$

Each customer must be visited once, which is ensured by the first constraint. The second constraint ensures that each tour passes through the depot. The constraint (3) is a flow conservation constraint. Finally, the last constraint guarantees that each tour ends at the depot. Since the service time P_i^k at any client i by vehicle k begins inside a given time interval $[a_i, b_i]$, we require an additional constraint.

$$a_i \le P_i^k \le b_i \ \forall (i \in N)\, \forall (k \in V). \tag{5}$$

The time windows considered here is hard, i.e. they cannot be violated, if the vehicle arrives earlier than required at a client i, it must hold up until the time window $[a_i, b_i]$ opens and moreover it is not permitted to arrive late.

$$P_i^k + t_{ij} - P_j^k \le M\big(1 - x_{ij}^k\big) \forall (i \in N)\, \forall (j \in N\{0\})\, \forall (k \in V), \tag{6}$$

where M is a great value. To model the uncertainty in travel times and service times in the presence of time windows, a step-wise (layered) formulation is used. As mentioned above , the sets of uncertainties related to these times are described as follows:

$$U_t = \left\{ \tilde{t} \in \mathbb{R}^{|A|} \ / \ \ \tilde{t}_{ij} = t_{ij} + \Delta_{ij}\varepsilon_{ij}\,, \sum_{(i,j)\in A} \varepsilon_{ij} \le \Gamma, 0 \le \varepsilon_{ij} \le 1, \forall (i,j) \in A \right\},$$

and

$$U_P = \left\{ \tilde{P} \in \mathbb{R}^{|N|} \ / \ \tilde{P}_i = P_i + \delta_i\omega_i\,, \sum_{i\in N} \omega_i \le \Lambda, 0 \le \omega_i \le 1, \forall (i \in N) \right\},$$

The uncertainty parameters Γ and Λ which vary respectively between 0 and $|N| + |V|$, and 0 and $|N|$, are called budgets of uncertainty. They are controling the maximum number of travel times and service times.

- If Γ=0 and Λ=0, the robust case is equivalent to the deterministic case.
- If $\Gamma = |N| + |V|$ and $\Lambda = |N|$ then this is the worst case where all travel times and service times are supposed uncertain and take at once their maximum values.

Robust optimization seeks to obtain good solutions for all the possible realizations of the uncertainties without it being necessary to define the laws of probability and considering only the nominal values and the possible deviations of the uncertain data. The uncertainties are introduced in the cost function as follows:

$$min \left(\sum_{k\in V} \sum_{(i,j)\in A} x_{ij}^k \mathrm{t_{ij}} + \max_{\{\Psi\, /\Psi\subset A, |\Psi| = \Gamma\}} \sum_{k\in V} \sum_{(i,j)\in \Psi} x_{ij}^k \Delta_{\mathrm{ij}} \right).$$

And the constraint (6) treating the time windows by this:

$$P_i^k + t_{ij} + \delta_i \nu_i^\theta + \Delta_{ij}\mu_{ij}^\Psi - P_j^k \le M\big(1 - x_{ij}^k\big)$$
$$\forall (i \in N)\, \forall (j \in N\backslash\{0\})\, \forall (k \in V), \forall (\theta \subset N)\ |\theta| = \Lambda, \forall (\Psi \subset A)\ |\Psi| = \Gamma$$

where ν_i^{θ} and μ_{ij}^{Ψ} are binary variables which only takes respectively two values: 1 if $i \in \theta$ and *(i, j)* $\in \Psi$ and 0 otherwise.

3. Robust approach for the VRPTW

Along these lines, we propose an adaptive large neighborhood search (ALNS) heuristic to integrate into our approach in order to deal with robust VRPTW. The proposed metaheuristic (ALNS) is an extension of the Large Neighborhood Search (LNS) heuristic, which was first introduced by Shaw (1998), ALNS is a metaheuristic proposed by Ropke and Pisinger (2006). It is a common technique used to enhance a locally optimal solution and can prevent getting stuck in premature convergence to local optima within tightly constrained search space. Given an initial solution obtained by a construction method, it is based on the idea of improving the initial solution by applying various destroy and repair operators to generate large neighborhoods through which the search space is explored (Palomo-Martínez Pamela et al., 2017). The ALNS has already been adapted to several transportation problems including vehicle routing (Ropke & Pisinger, 2006), arc routing (Angélica Salazar-Aguilar et al., 2012), inventory-routing (Coelho et al., 2012), and the reliable multiple allocation hub location problem (Chaharsooghi et al., 2017). The ALNS seems well-suited for the VRPTW, its power is manifested in the fact that each new solution is obtained by first removing a number of vertices, then re-inserting these vertices into the solution. ALNS was chosen because it outperforms other mono-objective algorithms applied to the same problem while keeping the simplicity and high performance that characterize local search algorithms.

3.1. Adaptive Large Neighborhood Search

We will now describe the ALNS that we have used in the present paper. We believe that ALNS can be applied to a large class of difficult optimization problems. In order to design an ALNS algorithm for a given optimization problem we need to:

- Choose a number of fast construction operators which are able to construct a full solution.
- Select a number of destruction operators. It might be sufficiently important to choose the destruction operators that are expected to work well with the construction operator, but it is unnecessary.

Here is the detailed algorithm:

Algorithm 1 Adaptive Large Neighborhood Search

```
Construct a feasible solution x; set x* = x
Repeat
        Choose a destroy neighborhood N⁻ and a repair neighborhood N⁺ using roulette wheel
        selection based on previously obtained scores π_J
        Generate a new solution x' from x using the heuristics corresponding to the chosen destroy
        and repair neighborhoods
        If x' can be accepted then
                x = x'
        End if
        If f(x') < f(x) then
                x* = x'
        End if
        Update scores π_j of N⁻ and N⁺
Until Stop criteria is met
Return x*
```

3.1.1 Initial solution generation

In order to deal with the initial solution, we apply a greedy algorithm, which will be used in the reconstruction phase of the ALNS. This operator aims to insert the non-inserted nodes by testing the different possible configurations and then giving a feasible solution. It is not necessary that the completion time of the initial solution be minimal, as this solution will be further enhanced using the ALNS method.

3.1.2 Solution destruction

During the destruction phase, we put forward three different removal methods to maintain the diversity during the searching process and to define the neighborhood to explore at each iteration. Each removal method aims to remove a predefined number of nodes. The first operator is known as proximity operator. Its objective is to select close clients according to a spatio-temporal measure (Shaw, 1998), and then remove clients engendering the higher value of this measure. Using the same technique, the route portion operator comes to give more flexibility than the proximity operator to change the routes. The principle consists in choosing a pivot client owned to a road and remove it as well as its adjacent clients. Then, we calculate the spatio-temporal measure, with the objective to select a second client belonging to another route but close to the initial pivot. The second pivot is removed from the solution as well as its adjacent clients and so on until all clients will be removed. The third operator is referred to as longest detour operator. The interest of this operator is to remove the customers that lead to the largest cost increases for servicing them. For more details, we refer the reader to (PrescottGagnon et al., 2009). The algorithms of the used destroy operators can be found in Annex.

3.1.3 Solution reconstruction

Solomon's insertion heuristic (1987) presented a technique for choosing the new customer to be inserted into a route using two criteria $c_1(i,u,j)$ and $c_2(i,u,j)$ to select customer u for insertion between adjacent clients i and j in the current partial route. The primary criterion c_1 calculate the best feasible insertion place in the current route for each unrouted client u as

$$c_1\big(i(u), u\ , j(u)\big) = \min_{\rho=1,\dots,m} c_1\big(i_{\rho-1}, u, i_\rho\big).$$

The second criterion c_2 selects the new inserted customer:

$$c_2\big(i(u^*), u^*\ , j(u^*)\big) = \max_u \big\{c_2\big(i(u), u\ , j(u)\big) \mid u\ is\ unrouted\ and\ route\ is\ feasibe\big\}\,.$$

Customer u^* is then inserted into the route between $i(u^*)$ and $j(u^*)$. The measurement of an insertion place c_1 depends on factors: the increase in total distance of the current route after the insertion, and the delay of service start time of the customer following the new inserted customer. To be more precise, $c_1(i,u,j)$ is calculated as:

$$c_1(i,u,j) = \alpha_1\left(d_{iu} + d_{uj} - \mu d_{ij}\right) + \alpha_2\big(b_{ju} - b_j\big),$$
$$\alpha_1 + \alpha_2 = 1, \alpha_1 \geq 0, \alpha_2 \geq 0, \mu \geq 0$$

where $d_{iu} + d_{uj}$ is the new distance between two nodes i and j after the insertion, b_j is the previous service start time, d_{ij} is the old distance between i and j and b_{ju} is the new service start time of customer j after the insertion of customer u. The criterion $c_2(i,u,j)$ is calculated as following

$$c_2(i,u,j) = \lambda d_{0u} - c_1(i,u,j), \lambda \geq 0$$

where the parameter λ is used to define how much the best insertion place for an unrouted customer depends on its distance from the depot and extra time required to visit the customer by the current vehicle.

3.1.3 Roulette wheel

For each iteration of the destruction phase, a roulette-wheel procedure is applied to select a method for generating the neighborhood (nodes to be removed). During the search process, the ALNS maintains a score φ_j which measures the best performance of an heuristic j in the past iterations. The roulette wheel selection consists in selecting an heuristic j with a probability $\frac{\varphi_j}{\sum_i \varphi_i}$. During M iterations, the score φ_i is reset and the probabilities of choosing an heuristic are recalculated (Pisinger and Ropke (2007)).

3.2 ALNS applied to the robust VRPTW

In this section, we apply the adaptive large neighborhood search (ALNS) to solve the VRPTW, assuming that the displacement and the service times are both objects of uncertainty. The robustness of this approach has been tested on several scenarios generated by the Monte Carlo tool of simulation. We will now describe how we have adapted the general ALNS to the robust VRPTW.
We consider that the uncertainties in travel times and service times are independent and are budgeted by a pair of degrees of robustness (Λ, Γ). In other words, Γ and Λ are interpreted as the maximum number of travel times and service times that can deviated from their nominal values, and are bounded respectively in the intervals $[0, |N| + |V|]$ and $[0, |N|]$. Therefore, our objective is to find for each (Λ, Γ) scenario a robust solution which minimize the total delays, or immunize against the violation of time windows.

Our algorithm is presented in detail in the subsections: (3.2.1), (3.2.2) and (3.2.3). Here are a few notations used in our Algorithm:

$R_N^{\Lambda,\Gamma}$: A possible realization.

S_{best}: Best robust solution.

S_N : Solutions achieved at the N^{th} realization.

TotalCost (.): A function used to calculate the total time of displacement of a solution

$\text{WorstEval}^{\Gamma}(.)$: A function used to calculate the worst evaluation of a solution

$Tr_k = (c_1 = o, c_2, \dots, c_n = o)$: The tour of the vehicle k

$\sigma_l = (c_1, c_2, \dots, c_l)$: A path of the tour Tr_k

$ArcSet^{\Gamma}$: The whole of the arcs which have the more large deviations of travel time

$NodeSet^{\Lambda}$: The whole of the nodes which have the Λ more large deviations of service time

$\xi(\sigma_l) = \{c_1, c_2, \dots, c_l\}$: All of the nodes which constitute σ_l

$Arc(\sigma_l) = \{\gamma_1 = (c_1, c_2), \gamma_2 = (c_2, c_3), \dots, \gamma_{h-1} = (c_{l-1}, c_l)\}$: The whole of the arcs which constitute the path σ_l

$({s_l}^{-k})$: The maximum date of arrival of the vehicle k at customer c_l

Here is the detailed algorithm of our approach:

Algorithm 2 The robust approach Algorithm

Parameters: Set $Solutions$, Set $realizations$
Outputs: Solution $solution$
$realizations \longleftarrow MonteCarlo()$
For each $realization \in realizations$ **do**
  $solution \longleftarrow ALNS(realization)$
  $solutions.add(solution)$
End for
For each $solution \in solutions$ **do**
  If $checkRobustness(solution) \neq True$ **then**
    $solutions.remove(solution)$
    Return NULL
  End if
  If $EvalWorstCase(solution) \neq True$ **then**
    $solutions.remove(solution)$
    Return NULL
  End if
End for
$solution \longleftarrow MinObjective(solutions)$
Return $solution$

3.2.1 Identification of scenarios

At this step, we performed the Metropolis Monte-Carlo sampling in order to generate a reduced set of possible realizations $R_N^{\Lambda,\Gamma}$ from uniform distribution. For the sake of clarity, a realization $R_N^{\Lambda,\Gamma}$ represents a possible scenario in which Γ displacement times take their maximum values $t_{ij} + \Delta_{ij}$, and Λ service times take their maximum values $P_i + \delta_i$. While the other arcs and the other nodes take respectively their nominal values t_{ij} and P_i. For more details about Metropolis-Monte-Carlo techniques, we refer the reader to the work of Mosegaard et al. (1995) and more recent version of Rubinstein et al. (2016).

3.2.2 Research of solution

For each realization $R_N^{\Lambda,\Gamma}$, we apply the Adaptive large neighborhood search in order to obtain a feasible solution noted S_N satisfies each scenario that we have already generalized by Monte-Carlo.

3.2.3 Robustness

We draw up different mechanisms for the study of the robustness of the solution obtained by using ALNS approach in the previous step. The first mechanism of robustness allowed to verify the feasibility of our solution by investigating the related time windows of each visited customer. It is not required at this level to test all possible realizations. The pseudo code of this method is shown in the algorithm 3. The second mechanism concerns the evaluation of robustness of the solution according to a defined robust criterion. The best case criterion is considered as a min-min strategy that tries to find the best solution overall scenarios. At the opposite, the worst case criterion provides a solution minimizing the worst deviation among all considered scenarios (e.g. Wu 2017). In between, an alternative robust criterion is related to as min-max deviation; more detail can be found in the paper of Aissi et al. (2009).

In the present paper, we are more interested in finding the best solution in the presence of the worst case scenario for VRPTW under uncertain travel and service times (see algorithm 4). For the full details of different steps of algorithm 3 and 4, we refer the reader to the work of Rouky et al., (2018).

Algorithm 3 Check of the robustness

$feasible \longleftarrow True$
For $k \longleftarrow$ 1 to $|V|$ **do**
    For $\longleftarrow$ to $|\xi(Tr_k)|$
        Calculate $ArcSet^{\Gamma}$ and $NodeSet^{\Lambda}$
        For $\lambda \longleftarrow$ to $l-1$ **do**
            If $l \leq \Gamma+1$ or $\gamma_\lambda \in ArcSet^{\Gamma}$ **then**
                $t_{\gamma_\lambda} \longleftarrow t_{\gamma_\lambda} + \Delta_{\gamma_\lambda}$
            End if
        End for
        For $i \longleftarrow$ to $l-1$ **do**
            If $1 \leq \Lambda$ or $c_i \in NodeSet^{\Lambda,h-1}$ **then**
                $P_{c_i} \longleftarrow P_{c_i} + \delta_{c_i}$
            End if
        End for
        $(s_1{}^{-k})^{\Gamma,\Lambda} \longleftarrow$
        For $i \longleftarrow$ to l **do**
            $(s_i{}^{-k})^{\Gamma,\Lambda} \longleftarrow \max((s_{i-1}{}^{-k})^{\Gamma,\Lambda} + t_{\gamma_{i-1}} + P_{c_{i-1}}, a_{c_i})$
        End for
        If $(s_l{}^{-k})^{\Gamma,\Lambda} > b_{c_l}$ **then**
            feasible takes false and the algorithm ends
        End if
    End for
End for

Algorithm 4 Evaluation on the worst case

$WorstEval^{\Gamma}(S_N) \leftarrow$
Put in descending order all the arcs of $\gamma(S_N)$ according to their maximum deviations.
For $i \leftarrow$ 1 to **do**
    $WorstEval^{\Gamma}(S_N) \leftarrow WorstEval^{\Gamma}(S_N) + t_{\gamma_i} + \Delta_{\gamma_i}$
End for
For $i \leftarrow \Gamma+1$ to $|\gamma(S_N)|$ **do**
    $WorstEval^{\Gamma}(S_N) \leftarrow WorstEval^{\Gamma}(S_N) + t_{\gamma_i}$
End for
Return $WorstEval^{\Gamma}(S_N)$

4. Computational experiment

Since VRPTW and RVRPTW are both NP-Hard, so to provide perfect conclusions and comparative results, we considered several kinds of instances. The robust approach examined was tested on a set of small instances based on the reference of Solomon benchmark (1987) (Solomon, 1987), and large instances of Gehring & Homberger's benchmark. Since the uncertainty of RVRPTW is simulated by discrete scenarios using Monte-Carlo Simulation, the uncertain travel time of each arc and the uncertain service time at each node are generated randomly between 0 and 10, with (Γ, Λ) is the degree of robustness which represents the number of service times and the number of travel times assumed to be uncertain. The used instances are noted as follow $Gr\_\Gamma\_\Lambda\_i$, where $Gr = \{C1, C2, R1, R2, RC1, RC2\}$ corresponds to the class name of the benchmark of Solomon and Gehring & Homberger. and Λ represent the number of travel times and service times supposed uncertain. $i = \{100,200,400,600,800,1000\}$ is an index that represents the size of the instance.

Table 1 shows the results obtained for small instances (100 customers) by using Cplex for the deterministic VRPTW and the results obtained by our robust approach based on ALNS that deals with the VRPTW considering that travel times and service times are both uncertain. The column “Instance” displays the notation of the instance. The column “Initial solution” presents the initial solution with which the robust approach starts. The column “best” states the best values found by the robust approach with 10 runs while the column "mean" shows the average values found by the robust approach over 10 trials. The column “Optimal” displays the optimal solution for the deterministic VRPTW calculated by Cplex.

Table 1
Performance of our robust approach versus deterministic VRPTW (CPLEX)

Instance	Initial solution	Best solution	Mean solution	Optimal solution
R101_10_10_100	2176.59	1918.56	1981.45	1637.7
R106_10_10_100	1794.75	1570.49	1603.24	1234.6
R112_10_10_100	1292.24	1135.25	1165.39	978.7
R201_10_10_100	1751.13	1534.99	1544.19	1143.2
C101_10_10_100	917.76	870.46	881.17	827.3
C105_10_10_100	1016.49	872.35	906.45	827.3
C108_10_10_100	1172.46	982.18	1016.33	827.3
C201_10_10_100	782.39	606.44	645.1	589.1
C205_10_10_100	805.15	662.08	713.74	586.4
C208_10_10_100	831.3	727.28	779.43	585.8
RC101_10_10_100	2193.94	1882.76	1994.25	1619.8
RC105_10_10_100	1915.73	1589.8	1713.6	1513.7
RC108_10_10_100	1710.58	1474.84	1562.7	1114.2
RC201_10_10_100	1659.86	1463.38	1509.13	1261.8
RC202_10_10_100	1566.17	1398.14	1448.56	1092.3
RC205_10_10_100	1518.52	1418.02	1453.03	1154

Table 2 shows the results obtained for large instances by comparing the best-known results for the deterministic VRPTW to the results found by our robust approach based on ALNS that deals with the VRPTW considering that travel times and service times are both uncertain.

Table 2
Performance of our robust approach versus best known results

Instance	Initial solution	Best solution	Mean solution	Best known
R121_25_25_200	6610.35	5696.41	5911.84	4784.11
R141_25_25_400	12526.98	10939.12	11082.53	10372.31
R161_50_50_600	26176.96	24277.04	24910.6	21131.09
R181_50_50_800	41205.13	39011.14	39643.9	36852.06
R1101_100_100_1000	63493.03	60506.88	61702.03	53473.26
R125_25_25_200	5156.38	4667.38	4910.81	4107.86
R145_25_25_400	11714.54	11250	11665.48	9226.21
R165_50_50_600	22959.22	21231.41	21802.25	19588.89
R185_50_50_800	37521.62	36488.37	36983.03	33723.77
R1105_100_100_1000	58238.81	56015.9	56877.48	50876.21
R1210_25_25_200	3919.05	3676.6	3731.17	3301.18
R1410_25_25_400	10201.39	9847.68	10121.67	8094.1
R1610_50_50_600	21602.23	19874.36	20350.31	17748.83
R1810_50_50_800	36258.1	34792.04	35587.88	31086.85
R11010_100_100_1000	52326.08	50678.87	50961.69	47992.05
R221_25_25_200	5399.58	4785.14	4879.5	4483.86
R241_25_25_400	11836.97	10442.34	10975.86	9210.15
R261_50_50_600	24045.81	22070.83	22702.28	18206.8
R281_50_50_800	34167.08	32289.08	33162.16	28114.25
R2101_100_100_1000	51402.42	48077.63	49298.56	42188.86
R225_25_25_200	4226.4	3625.7	3871.07	3366.79
R245_25_25_400	10150.65	8340.94	8986.2	7128.93
R265_50_50_600	18968.26	17022.95	17790.63	15096.2
R285_50_50_800	29807.24	28110.59	28519.41	24285.89
R2105_100_100_1000	48290.25	46144.16	46894.91	36232.18
R2210_25_25_200	3312.17	2913.82	3066.73	2654.97
R2410_25_25_400	8482.81	6679.72	7785.34	5786.4
R2610_50_50_600	15874.92	14609.57	15165.41	12253.47
R2810_50_50_800	25210.16	23839.6	24127.11	20401.47

Table 2
Performance of our robust approach versus best known results (Continued)

Instance	Initial solution	Best solution	Mean solution	Best known
R21010_100_100_1000	42490.62	40103.23	41335.65	30215.24
C121_25_25_200	3399.45	2915.86	3029.34	2704.57
C141_25_25_400	9205.81	8138.18	8584.48	7152.02
C161_50_50_600	17624.24	15511.06	16672.38	14095.64
C181_50_50_800	29445.55	28491.39	29075.07	25030.36
C1101_100_100_1000	52251.98	51066.63	51982.04	42478.95
C125_25_25_200	3116.86	2929.73	3058.28	2702.05
C145_25_25_400	9992.46	8485.01	8964.53	7152.02
C165_50_50_600	18388.46	17227.57	18035.76	14085.72
C185_50_50_800	30275.37	28646.79	29428.55	25166.28
C1105_100_100_1000	52938.75	52150.1	52467.93	42469.18
C1210_25_25_200	3148.39	2808.41	2990.42	2643.51
C1410_25_25_400	9195.54	8208.69	8632.71	6860.63
C1610_50_50_600	16565.12	15842.58	16210.46	13637.34
C1810_50_50_800	29738.8	27501.34	27112.31	24070.17
C11010_100_100_1000	50163.43	46321.77	47806.49	39858.64
C221_25_25_200	2316.76	2190.94	2244.38	1931.44
C241_25_25_400	5043.61	4632.06	4957.28	4116.05
C261_50_50_600	11601.77	10617.6	11308.25	7774.1
C281_50_50_800	15126.33	14213.43	14833.2	11654.81
C2101_100_100_1000	21742.75	20735.79	21212.43	16879.24
C225_25_25_200	2444.25	2164.43	2243.59	1878.85
C245_25_25_400	5243.39	4165.92	4405.26	3938.69
C265_50_50_600	11527.53	10346.3	10596.96	7575.2
C285_50_50_800	16054.91	14992.32	15575.45	11425.23
C2105_100_100_1000	25179.55	24255.43	24799.67	16561.29
C2210_25_25_200	2282.97	2008.49	2101.45	1806.58
C2410_25_25_400	5012.63	4399.61	4687.95	3827.15
C2610_50_50_600	11334.43	10294.91	10523.94	7255.69
C2810_50_50_800	14967.74	14043.06	14651.07	10977.36
C21010_100_100_1000	25907.63	24024.33	24624.61	15943.34
RC121_25_25_200	4360.55	3815.01	4173.33	3602.8
RC141_25_25_400	12220.71	10673.61	11061.17	8573.96
RC161_50_50_600	23620.95	21917.12	22561.48	17014.17
RC181_50_50_800	37849.35	35821.46	36257.21	31117.04
RC1101_100_100_1000	56325.86	53322.91	54165.08	46138.01
RC125_25_25_200	4182.22	3601.92	3762.9	3371
RC145_25_25_400	11602.21	10604.17	10930.51	8172.64
RC165_50_50_600	21047.61	20073.28	20400.21	16566.24
RC185_50_50_800	35558.2	34959.07	35397.32	29796.67
RC1105_100_100_1000	56705.85	53937.4	54407.58	45313.38
RC1210_25_25_200	3902.19	3349.59	3514.39	3000.3
RC1410_25_25_400	8908.94	8007.56	8397.93	7596.04
RC1610_50_50_600	18946.53	18149.09	18591.05	15675.99
RC1810_50_50_800	32010.09	31610.09	31838.24	28474.35
RC11010_100_100_1000	49491.71	47296.55	48010.76	43679.61
RC221_25_25_200	3865.62	3357.12	3460.46	3099.53
RC241_25_25_400	9326.5	7160.2	7667.79	6682.37
RC261_50_50_600	17021.84	16167.74	16660.9	13324.93
RC281_50_50_800	29056.9	28307.07	28849.06	20981.14
RC2101_100_100_1000	44947.06	43853.65	44307.34	30278.5
RC225_25_25_200	3511.92	3232.09	3457.93	2911.46
RC245_25_25_400	9809.92	7445.5	8708.31	6710.12
RC265_50_50_600	17329.58	16076.16	16844.96	13000.84
RC285_50_50_800	26883.72	26099.31	26607.32	19136.03
RC2105_100_100_1000	42576.12	40999.51	41311.28	27140.77
RC2210_25_25_200	2773.13	2397.68	2506.15	2015.6
RC2410_25_25_400	5213.86	4722.23	4911.98	4278.61
RC2610_50_50_600	12878.34	11991.88	12302.9	9069.41
RC2810_50_50_800	20326.42	18361.79	18830.3	14439.14
RC2101_100_100_1000	32588.02	30015.61	30598.91	21910.33

In order to visualize the impact of increasing degrees of uncertainty on objective function, we set the value of to 25 and we adjusted the value of Λ for multiple instances of size 100. Here is the curve obtained:

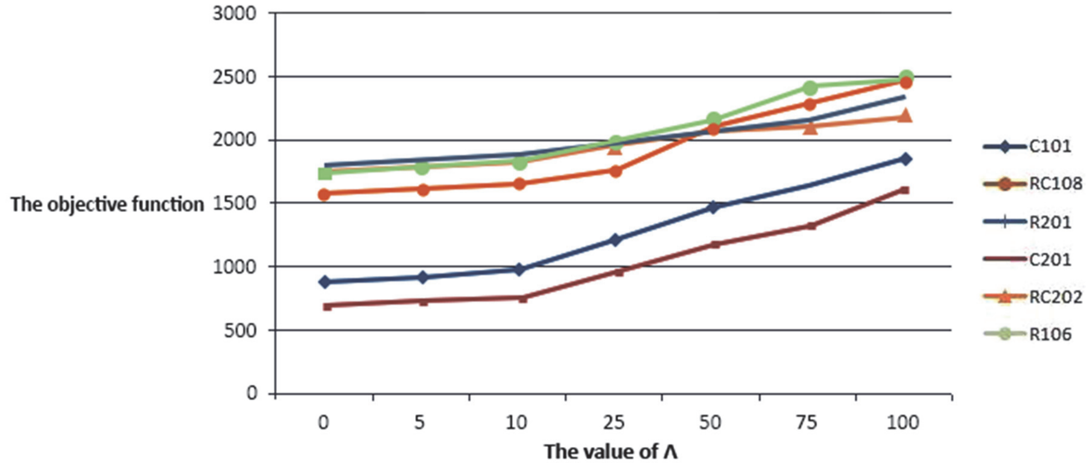

Fig. 1. Values of the objective function when adjusting the value of Λ

The graph shows clearly that the objective function increases according to degree of uncertainty. To the best of our knowledge, this contribution is the first work to be devoted to the study of VRPTW considering both the uncertainties on travel times and service times. Due to the lack of work in this direction, we compared our results with the deterministic VRPTW literature even if the two problems are different in the sense that a good solution found for the deterministic case could become worse in the presence of uncertainties or even unreachable.

From Table 1, the computation of the mean absolute percentage error (MAPE) of the data set indicates how much error in predicting the robust mean solution compared with the deterministic optimal solution. By including the uncertainties in travel times and service times, the minimization problem becomes robust and MAPE is about 22.68%. We need then extra cost to the objective function but the related counterpart risk gained is the protection against the delays and the solution is of good quality. In the meanwhile, for large instances, we concluded from table 2 that MAPE is about 22.57% where in this case the observed solution is the robust mean one compared to the deterministic case. The fact that the MAPE value for large and small instance around 23% is beyond our expectation and we believed that the robustness of the solution against the uncertain data can be achieved in the most cases by making the solution feasible for any travel times and service times in the uncertainly sets.

It is then clear that our approach is very powerful in terms of the robustness since it included several algorithms (Robustness verification, worst-case evaluation ...) which leads to near best solutions for all the possible realizations of the uncertainties without any further considerations but only nominal values and deviations possible uncertain data are sufficient.

5. Conclusion

Our main goal in this paper was to consider the robust vehicle routing problem with time windows under both travel times and service times uncertainties. For this purpose, a new resolution robust approach has been suggested to minimize the total distance of the travel time in the presence of the maximum deviations of possible uncertain data. In this contrast, we generated all possible scenarios by using Metropolis-Monte Carlo simulation and we opt for the adaptive large neighborhood search ALNS algorithm to solve each sub-problem related to each scenario. In this context, several destroy/repair method has been combined to explore multiple neighborhoods within the same search and defined implicitly the large neighborhood. In order to study the feasibility of the resulting solution, following the previous works, efficient mechanisms have been developed; the first concerns the verification of the robustness, while the second takes into consideration the evaluation of the solution on the worst case.

The introduction of an effective way of modeling and handling several uncertainty data levels defined by pairs of uncertainty (Λ, Γ), which represent respectively the number of service times and the number of travel times assumed uncertain, has been tested on several sets of problems and showed improved robustness results for benchmark instances. Furthermore, the resolved method using the ALNS approach confirmed as in the previous work a great protection against uncertainties compared to what would have been found if a deterministic solution had been applied.

In this work, the computational experiments were performed to examine our proposed new approach compared to the deterministic VRPTW literature on a set of small instances based on the Solomon VRPTW benchmark and large instances of Gehring & Homberger benchmark. The results have shown the robustness of our solutions against delays and offer decision-making tool that allows choosing the level of protection, as well as the deterministic solution, is applied.

Future work will focus on the extension of the robust routing vehicle problem with time windows, in which both unexpected delays in travel time and service time may occur, to the application of parallel adaptive large neighborhood search in order to develop fast optimization procedures able to react in real time to changes in the problem information.

Acknowledgments

The first author is indebted to the Moroccan CNRST National Centre for Scientific and Technical Research for the partial funding of this research project.

References

Agra, A., Christiansen, M., Figueiredo, R., Hvattum, L., Poss, M., & Requejo, C. (2012). Layered Formulation for the robust vehicle routing problem with time windows. *Lecture Notes in Computer Science*, 7422, 249-260.

Agra, A., Christiansen, M., Figueiredo, R., Hvattum, L., Poss, M., & Requejo, C. (2013). The robust vehicle routing problem with time windows. *Computers and Operations Research, 40*(3), 856-866.

Aissi, H., Bazgan, C., & Vanderpooten, D. (2009). Min-max and min-max regret versions of combinatorial optimization problems: A survey. *European Journal of Operational Research, 197*(2), 427–438.

Alumur, S., Nickel, S., & Saldanha-da-Gama, F. (2012). Hub location under uncertainty. *Transportation Research Part B: Methodological, 46*(4), 529-543.

Angélica Salazar-Aguilar, M., Langevin, A., & Laporte, G. (2012). Synchronized arc routing for snow plowing operations. *Computers & Operations Research, 39*(7), 1432-1440.

Baron, O., Milner, J., & Naseraldin, H. (2011). Facility location: A robust optimization approach. *Production and Operations Management, 20*(5), 772-785.

Ben-Tal, A., Boaz, G., & Shimrit, S. (2009). Robust multi-echelon multi-period inventory control. *European Journal of Operational Research, 199*(3), 922-935.

Bertsimas, D., & Sim, M. (2003). Robust discrete optimization and network flows. *Mathematical Programming, 98*(1-3), 49-71.

Bienstock, D., & Özbay, N. (2008). Computing robust basestock levels. *Discrete Optimization, 5*(2), 389-414.

Bräysy, O., & Gendreau, M. (2005). Vehicle Routing Problem with time windows. *Part II: Metaheuristics. Journal Transportation Science, 39*(1), 119-139.

Chaharsooghi, S., Momayezi, F., & Ghaffarinasab, N. (2017). An adaptive large neighborhood search heuristic for solving the reliable multiple allocation hub location problem under hub disruptions. *International Journal of Industrial Engineering Computations, 8*(2), 191-202.

Coelho, L. C., Cordeau, J.-F., & Laporte, G. (2012). Consistency in multi-vehicle inventory routing. *Transportation Research Part C: Emerging technologies, 24*:270-287.

Dantzig, G. B., & Ramser, J. H. (1959). The truck dispatching problem. *Management Science, 6*(1), 80-91.

Dror, M., Laporte, G., & Louveaux, F.V. (1993). Vehicle Routing with Stochastic Demands and Restricted Failures. *ZOR – Methods and Models of Operations Research, 37*(3), 273-183.

Dror, M., & Trudeau, P. (1986). Stochastic vehicle routing with modified savings algorithm. *European Journal of Operational Research, 23*(2), 228-235.

Errico, F., Desaulniers, G., Gendreau, M., Rei, W., & Rousseau, L.-M. (2016). A priori optimization with recourse for the vehicle routing problem with hard time windows and stochastic service times. *European Journal of Operational Research, 249*(1), 55-66, 2016.

Fabozzi, F. J., Kolm, P. N., Pachamanova, D., & Focardi, S. M. (2007). Robust portfolio optimization and management. *Wiley*.

Gendreau, M., Hertz, A., Laporte, G., & Stan, M. (1998). A Generalized Insertion Heuristic for the Traveling Salesman Problem with Time Windows. *Operations Research, 46*(3), 330-335.

Gendreau, M., Laporte, G., & Séguin, R. (1996). Stochastic vehicle routing. *European Journal of Operational Research, 88*(1), 3-12.

Goren, S., & Sabuncuoglu, I. (2008). Robustness and stability measures for scheduling: single-machine environment. *IIE Transactions, 40*(1):66-83.

Gounaris, C.E., Wiesemann, W., & Floudas, C.A. (2013). The robust capacitated vehicle routing problem under demand uncertainty. *Operations Research, 61*(3), 677-693.

Gülpinar, N., Pachamanova, D., Çanakoglu, E. (2013). Robust strategies for facility location under uncertainty. *European Journal of Operational Research, 255*(1), 21-35.

Gülpinar, N., & Rustem, B. (2007). Worst-case robust decisions for multi-period mean-variance portfolio optimization. *European Journal of Operational Research, 183*(3), 981-1000.

Hazir, Ö., Haouari, M., & Erel, E. (2010). Robust scheduling and robustness measures for the discrete time/cost trade-off problem. *European Journal of Operational Research, 207*(2), 633-643.

Kallehauge, B., Larsen, J., Madsen, O., & Solomon, M. (2005). Vehicle Routing Problem with Time Windows. *(eds) Column Generation*. Springer, 67-98.

Mosegaard, K., & Tarantola, A. (1995). Monte Carlo sampling of solutions to inverse problems. *Journal of Geophysical Research: Solid Earth, 100*(B7), 12431-12447.

Minoux, M. (2010). Robust network optimization under polyhedral demand uncertainty is NP-hard. *Discrete Applied Mathematics, 158*(5), 597-603.

Moghaddam, B. F., Ruiz, R., & Sadjadi, S. J. (2012). Vehicle routing problem with uncertain demands: An advanced particle swarm algorithm. *Computers & Industrial Engineering, 62*(1), 306-317.

Montoya-Torres, J. R., Franco, J. L., Isaza, S. N., Jiménez, H. F., & Herazo-Padilla, N. (2015). A literature review on the vehicle routing problem with multiple depots. *Computers & Industrial Engineering, 79*, 115-129.

Labadie, N., Prins, C., & Prodhon, C. (2016). Metaheuristics for Vehicle Routing Problems. *WILEY, 3*.

Noorizadegan, M., Galli L. & Chen B. (2012). On the heterogeneous vehicle routing problem under demand uncertainty. In: *21st International Symposium on Mathematical Programming, Berlin, Germany, 1-25*.

Ordóñez, F. (2010). Robust Vehicle Routing. *Tutorials in Operations Research, Institue for Operations Research ans the Management Sciences (INFORMS)*, 153-178, Maryland, USA.

Palomo-Martínez Pamela, J., Angélica Salazar-Aguilar, M. & Laporte, G. (2017). Planning a selective delivery schedule through Adaptive Large Neighborhood Search. *Computers & Industrial Engineering, 112*, 368-378.

Pillac, V., Gendreau, M., Guéret, C. & Medaglia, A. L. (2013). A review of dynamic vehicle routing problems. *European Journal of Operational Research, 255*(1), 1-11.

Pinar, Ç. (2007). Robust scenario optimization based on downside-risk measure for multi-period portfolio selection. *OR Spectrum, 29*(2), 295-309.

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

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

Rincon-Garcia, N., Waterson, B., & Cherrett, T. (2017). A hybrid metaheuristic for the time-dependent vehicle routing problem with hard time windows. *International Journal of Industrial Engineering Computations, 8*(1), 141-160.

Ropke, S., & Pisinger, D. (2006). An adaptive large neighborhood search heuristic for the pickup and delivery problem with time windows. *Transportation Science, 40*(4), 455-472.

Rouky, N., Boukachour, J., Boudebous, D., & Alaoui, A. (2018). A Robust Metaheuristic for the Rail Shuttle Routing Problem with Uncertainty: A Real Case Study in the Le Havre Port. *The Asian Journal of Shipping and Logistics, 34*, 171-187.

Rubinstein, R. Y., Kroese, D. P. (2016). Simulation and the Monte Carlo Method. *Wiley Publishing, 3rd*.

Shaw, P. (1998). Using constraint programming and local search methods to solve vehicle routing problems. *In International conference on principles and practice of constraint programming*. Springer Berlin Heidelberg, 1520, 417-431.

Solomon, M. (1987). Algorithms for the Vehicle Routing and Scheduling Problem with time Window Constraints. *Operations Research, 35*(2), 254-265.

Sungur, I., Ordóñez, F., & Dessouky, M. (2008). A robust optimization approach for the capacitated vehicle routing problem with demand uncertainty. *IIE Transactions, 40*(5), 509-523.

Toklu, N. E., Montemanni, R., & Gambardella, L. M. (2013). An ant colony system for the capacitated vehicle routing problem with uncertain travel costs. Swarm Intelligence (SIS), *Proceeding IEEE Symposium on*, 32-39.

Wu, L., Hifi, M., & Bederina, H. (2017). A new robust criterion for the vehicle routing problem with uncertain travel time. *Computers & Industrial Engineering, 112*, 607-615.

Yousefi, H., Tavakkoli-Moghaddam, R., Oliaei, N., Mohammadi, M., & Mozaffari, A. (2017). Solving a bi-objective vehicle routing problem under uncertainty by a revised multi-choice goal programming approach. *International Journal of Industrial Engineering Computations, 8*(3), 283-302.

Appendices

We introduce some algorithms used for the three different destruction operators of our Adaptive Large Neighborhood Search (ALNS) to ensure the diversity of the searching process. In contrast of the LNS heuristic which uses only one destroy procedure and one repair procedure, the ALNS uses an adaptive layer with a set of removal and insertion heuristics and applies them by preference using a selection mechanism that considers the statistics obtained during the search based on their performance and past success. The detail of each algorithm is given as follows:

Appendix A

Algorithm 5 Proximity operator

Select randomly a node $firstToRemove$ from the list of clients that have been removed, and add it to the list $removedNodes$
Add the other nodes to $remainingNodes$
For $i = 1$ to $numToRelax$ **do**
    Choose randomly a node $removedNodeId$ from $removedNodes$
    Choose by Rank and relatedness a node $nodeId$
    Add $nodeId$ to $removedNodes$
    Remove $nodeId$ from $remainingNodes$
End For
Return $removedNodes$

Appendix B

Algorithm 6 Route portion operator

Choose the First node to delete and its adjacent and add them to $adjacents$ list
Add the nodes of $adjacents$ to $removedNodes$ list and remove them from $remainingNodes$
Add the other nodes to $remainingNodes$
For $i = 1$ to $numToRelax$ **do**
    Choose randomly a node $removedNodeId$ from $removedNodes$
    Choose by Rank and relatedness a node $nodeId$
    Add $nodeId$ to $removedNodes$
    Remove $nodeId$ from $remainingNodes$
End For
Return $removedNodes$

Appendix C

Algorithm 7 Longest detour operator

Inputs: feasible solution x, Array $maxTab$
Outputs: z max detour
For $route \in routes(x)$ **do**
    $m = MaxDetour$
    Add m to $maxTab$
End for
$z = max(maxTab)$
Return z