A Robust Pickup and Delivery Problem with Uncertain Travel Time

1st Zaher Al Chami

Univ. Bourgogne Franche-Comté FEMTO-ST Institute/CNRS,
Rue Thierry-Mieg (UTBM),
90 010 Belfort Cedex, France
zaher.al-chami@utbm.fr

2nd Bechara Bechara

Univ. Bourgogne Franche-Comté FEMTO-ST Institute/CNRS,
Rue Thierry-Mieg (UTBM),
90 010 Belfort Cedex, France
bechara.bechara@utbm.fr

3rd Hervé Manier

Univ. Bourgogne Franche-Comté FEMTO-ST Institute/CNRS,
Rue Thierry-Mieg (UTBM),
90 010 Belfort Cedex, France
herve.manier@utbm.fr

4th Marie-Ange Manier

Univ. Bourgogne Franche-Comté FEMTO-ST Institute/CNRS,
Rue Thierry-Mieg (UTBM),
90 010 Belfort Cedex, France
marie-ange.manier@utbm.fr

Abstract—This paper discusses a robust variant of the well known Pickup and Delivery Problem (PDP) which we called the Robust Selective PDP (RSPDP). In this problem, a fleet composed of several vehicles with a given capacity should satisfy a set of transportation requests while respecting various constraints in relation with the precedence between suppliers and customers, the capacity of vehicles, the opening and closing times of each site. Because of all these constraints, the honoring of all requests over a given period (for example, one day) is sometimes impossible so the selective aspect consists in choosing the sites to be served. This robust variant is characterized by the presence of different scenarios used to handle the uncertain traveling time for each arc. Subject to all scenarios and constraints, the goal is to find the best solution (with the higher profit) which validates all scenarios. This study introduces the mathematical formulation for this new variant. Then, an exact and a metaheuristic approaches will be used to solve it. Several tests have been done, on new generated instances for the considered problem, in order to show the efficiency of developed methods.

Index Terms—Robust optimization, Routing problems, Transportation, Exact method, Metaheuristic approach

I. Introduction

The Pickup and Delivery Problem (PDP) is one of the most studied problem in the literature. It aims at finding routes for a fleet of vehicles based on a depot. Those routes must start and end at the depot. Each vehicle should serve a set of transportation requests subject to a set of constraints.

A request consists in transporting goods from a supplier (pickup location) to a customer (delivery location). In the considered variant, paired demands are included so the precedence constraints must be respected. In other words, each supplier must be visited before its specific customer and by the same vehicle. Moreover, the depot, in the considered problem, can be the supplier and/or the customer for one or several transportation demands. So, three types of flow are considered: the first one between depot and customers; the second one is from suppliers to depot and the last one links suppliers to customers.

This study deals with the variant in an urban context where the planning of vehicle routes is often more critical because of the applied laws in many cities in relation with the street access and the restricted hours of operations. To take into consideration the limited hours of operations, time windows are included in the considered problem. Hence, each node is associated with a time interval and the vehicle cannot visit this node after the end of its interval. If it arrives earlier than the beginning of the time window, it has to wait until the opening of this node. The street access limitations are taken into account by considering that all vehicles having a limited capacity which cannot be exceeded.

Taking all the above mentioned constraints into consideration makes sometimes the visit of all nodes impossible (selective aspect). So the choice of the subset to be served is based on the profit associated to each site.

In real life contexts, traveling between two different sites can be done through many paths. The ideal path is the one with the shortest traveling time. This one is generally related to the traveled distance or the fuel consumption. But in some situations (scenarios) the chosen path may suffer from heavy traffic or accidents. Therefore carriers should use another path or take more time to reach their destinations.

This paper studies the Robust Selective Pickup and Delivery Problem (RSPDP) in which uncertain traveling times are modeled as a set of discrete scenarios and the optimization relies on maximizing the total collected profit while offering a robust solution which can be realized in all scenarios. In this study, the traveling cost is not considered as an objective because of the selective aspect. Indeed, the use of such an objective generates an optimal solution which consists in visiting no site.

Figure 1 shows a small example with a fleet composed of 1 vehicle, 4 suppliers (represented by triangles) and 4 customers (represented by circles). Each paired demand is illustrated by a couple of sites with the same number and the same color.

Customer 4 is paired with the depot which is also the customer of supplier 5. In the presented solution, nodes numbered 2 were not visited. Let us consider that the solution of the
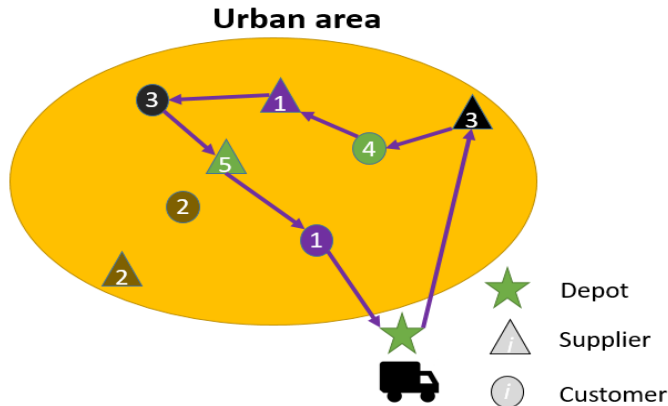

Fig. 1. Example of the RSPDP : ideal scenario

figure 1 is elaborated in the ideal scenario with a total profit of 100. Then, another scenario can be imagined where an accident may occur on the shortest path between supplier 1 and customer 3. So, another path is required to travel between those sites. In such a context, another solution can be obtained like the one in figure 2, but this time with a profit of 90. Here, the time delay on the arc linking supplier 1 and customer 3 has degraded the final obtained profit.
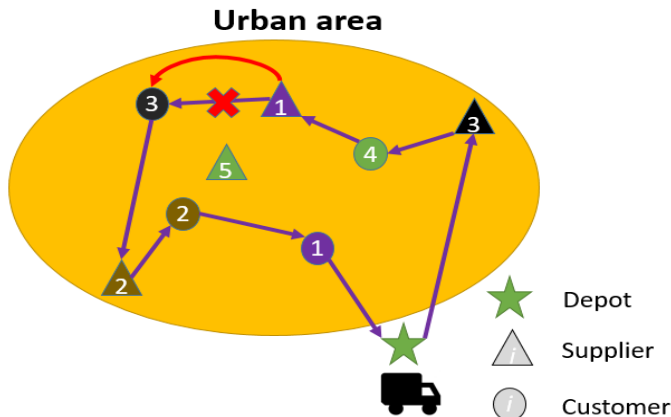

Fig. 2. Example of the RSPDP : a possible scenario

The remaining of this paper is organized as follows: a literature review is given in section II. Then, section III defines the RSPDP with a mathematical formulation. In section IV, the Greedy Randomized Adaptive Search Procedure (GRASP) is introduced to solve the considered variant. The experimental results are reported in section V. Finally, section VI concludes this study and gives direction for further research.

II. LITERATURE REVIEW

This section is divided into two parts. The first one provides a state of the art of various methods used to solve different variants of the PDP (subsection II-A). And the second one

deals with some studies on robust routing problems (subsection II-B).

A. Pickup and Delivery Problem

The PDP can be considered as a variant of the well known Vehicle Routing Problem (VRP). So it is a NP-hard problem [1]. Over the last decades, researchers have studied many variants of the PDP and they have used various algorithms to solve those variants. Two taxonomic reviews of the PDP literature have been published in [2] and [3].

Like other optimization problems, many resolution methods were introduced by researchers to solve the PDP. Those methods are divided into : exact and heuristic approaches. Among the exact ones, a Branch & Bound was developed in [4] to solve the PDP with Paired Demands (PDPPD). Moreover, researches in [5] have elaborated a Branch & Cut & Price approach to address the PDP with Time Windows and Paired Demands (PDPTWPD). In a previous work, an exact method based on linear programming was used to solve the Selective PDPTWPD (SPDPTWPD) in a mono-objective context [6]. And in another study, a lexicographic approach was developed to address the bi-objectives version of the SPDPTWPD [7].

As the exact approaches were limited to small instances, researchers have also used heuristic ones to solve bigger instances. For example, a tabu search combined with simulated annealing was introduced in [8] to solve the PDPTWPD. To address the same variant, researchers have used a hybrid Particle Swarm Optimization Algorithm [9]. Furthermore, two metaheuristic approaches were used to solve the SPDPTWPD: the first one is based on a hybrid genetic algorithm [10] and the second one combines a simulated annealing with a local search procedure [11].

B. Robust routing problems

To our best knowledge, there are no studies that deal with robust PDP in the literature. Therefore, this subsection presents some papers that address other routing problems from a robust point of view.

A literature review for the Robust Vehicle Routing Problem (RVRP) was presented by [12].

Among the exact method used to solve robust routing problems, a Branch & Bound was developed in [13] to deal with the RVRP. In this study, authors have considered uncertain demand on a set of fixed nodes and their method seeks to optimize the worst-case value over all scenarios. Such method does not discover all the real search space but it treat the problem as a normal VPP where each site demand is the worst over all scenarios. Another exact algorithm based on benders decomposition was introduced in [14] to solve the robust traveling salesman problem (RTSP) while considering uncertainty on travel time for each arc. The objective function was to find the tour which minimizes the distance in the worst case. Recently, a compact formulation and a Branch & Cut & Price method were elaborated in [15] to solve the RVRP. Authors of this paper took both customer demand and travel

time uncertainties into consideration. In addition, their method explores all the search space and gives a solution that remains feasible for all possible scenarios.

Heuristic approaches are also used to tackle robust routing problems. For example, the authors in [16] have elaborated a heuristic method to solve the RTSP where the distances between nodes are not known exactly. In their paper, they start with the minimal tour obtained in the ideal scenario and they use a recovery action once a scenario is realized. Moreover, a genetic algorithm was introduced in [17] to solve the RVRP with uncertain travel cost. Authors use the min-max optimization criterion. Then, they consider that the robust solution is the one which prevents against the worst case. Furthermore, a local search based metaheuristics were used by [18] to solve the RVRP with uncertain arc costs. They consider the min-max criterion against the worst scenario. But also the second worst scenario, the third one, etc. So they finally offer a Pareto front to help decision makers. Recently, a two-stage algorithm based on a modified variable neighborhood search was developed to solve RVRP with demand and travel time uncertainties in [19].

III. EXACT APPROACH FOR THE RSPDP

This section starts by giving some notations that will be used to formulate the new variant.

A. Notations :

1) Data:

- 0 : Id of the depot,
- B : Set of suppliers,
- C : Set of customers,
- Nodes = $B \cup C$: Set of all suppliers and customers,
- N = Nodes $\cup\, 0$: Set of all suppliers, customers and depot,
- $Supplier_i$: Supplier of the customer i (i $\in$ C),
- P_i : Profit associated with the node i (i $\in$ N),
- q_i : Goods quantity requested by node i (i $\in$ N)
  – If $q_i > 0$ then i $\in$ B else i $\in$ C,
- $[a_i, b_i]$: Time window of the node i (i $\in$ N),
- S_i : Service time needed to load or unload goods at node i (i $\in$ N),
- Scenarios : Set of possible scenarios,
- V : Set of available vehicles,
- Q^k : Maximal load of vehicle k (k $\in$ V),
- T_{ij}^{ks} : Time needed to travel directly from node i to node j using vehicle k in scenario s (i,j $\in$ N; k $\in$ V and s $\in$ Scenarios),
- M : Very big number.

2) Variables:

- A_i^{ks} : Starting service time of the vehicle k at node i in scenario s (k $\in$ V; i $\in$ N and s $\in$ Scenarios),
- D_i^{ks} : Departure time of the vehicle k from node i in scenario s (k $\in$ V; i $\in$ N and s $\in$ Scenarios),
- Y_i^k : Goods quantity in the vehicle k visiting node i (k $\in$ V and i $\in$ N),

- $X_{ij}^k = \begin{cases} 1 & \text{If the vehicle k moves from i to j (k} \in \text{V} \\ & \text{and i,j} \in \text{N)} \\ 0 & \text{Else} \end{cases}$

B. Mixed Linear Program for the RSPDP

After detailing all parameters, this subsection introduces the first mixed linear program for the Robust Selective Pickup and Delivery Problem (RSPDP) which can be as follows:

$$Maximize \sum_{i \in N} \sum_{j \in N} \sum_{k \in V} P_i X_{ij}^k \qquad (1)$$

Subject to:

$$\sum_{i \in N} \sum_{k \in V} X_{ij}^k \leq 1 \qquad \forall j \in Nodes \qquad (2)$$

$$\sum_{j \in N} \sum_{k \in V} X_{ij}^k \leq 1 \qquad \forall i \in Nodes \qquad (3)$$

$$\sum_{i \in N} X_{iu}^k - \sum_{j \in N} X_{uj}^k = 0 \qquad \forall k \in V; \forall u \in Nodes \qquad (4)$$

$$\sum_{i=1}^{Nodes} X_{0i}^k = 1 \qquad \forall k \in V \qquad (5)$$

$$\sum_{j=1}^{Nodes} X_{j0}^k = 1 \qquad \forall k \in V \qquad (6)$$

$$Y_0^k = \sum_{i \in C/Supplier_i = 0} (-q_i \sum_{j \in N/j \neq i} X_{ji}^k) \qquad \forall k \in V \qquad (7)$$

$$Y_j^k \geq Y_i^k + q_j - M(1 - X_{ij}^k) \quad \forall i,j \in N; \forall k \in V \qquad (8)$$

$$Y_j^k \leq Y_i^k + q_j + M(1 - X_{ij}^k) \quad \forall i,j \in N; \forall k \in V \qquad (9)$$

$$Q^k \geq Y_i^k \geq 0 \qquad \forall i \in N; \forall k \in V \qquad (10)$$

$$A_j^{ks} \geq D_i^{ks} + T_{ij}^{ks} - M(1 - X_{ij}^k) \\ \forall i,j \in N; \forall k \in V; \forall s \in Scenarios \qquad (11)$$

$$D_i^{ks} \geq A_i^{ks} + S_i - M(1 - X_{ij}^k) \\ \forall i,j \in Nodes; \forall k \in V; \forall s \in Scenarios \qquad (12)$$

$$D_i^{ks} \leq A_i^{ks} + S_i + M(1 - X_{ij}^k) \\ \forall i,j \in Nodes; \forall k \in V; \forall s \in Scenarios \qquad (13)$$

$$a_i \sum_{j=1}^{N} X_{ij}^k \leq A_i^{ks} \leq b_i \sum_{j=1}^{N} X_{ij}^k \\ \forall i \in N; \forall k \in V; \forall s \in Scenarios \qquad (14)$$

$$D_0^{ks} = 0 \qquad \forall k \in V; \forall s \in Scenarios \qquad (15)$$

$$D_f^{ks} \leq A_c^{ks} \qquad \forall f \in B; \forall c \in C/Supplier_c = f; \\ \forall k \in V; \forall s \in Scenarios \qquad (16)$$

$$\sum_{i \in N/i != f} X_{if}^k - \sum_{j \in N/j != c} X_{cj}^k = 0 \\ \forall f \in B; \forall c \in C/Supplier_c = f; \forall k \in V \qquad (17)$$

Constraint (1) represent the objective function. The aim is to maximize the total collected profit. Constraints (2) and (3) verify that each node is visited at most once. The routing continuity is ensured by constraints (4). To confirm that each route begins and finishes at the depot, we used constraints (5) and (6). The respect of each vehicle capacity is guaranteed by (7),(8), (9) and (10). Constraints (11) update the starting service time at each node for each scenario while taking into account the associated traveling time. This starting time should be in the time windows of this site (constraints (14)) and it will be used to update the departure time from this site (constraints (12) and (13)). All vehicle routes should begin from the depot at time equal to zero (constraints (15)). Constraints (16) and (17) are used to verify that the supplier is visited before its customer and by the same vehicle.

This MILP is implemented using commercial solver CPLEX 12.7.1 in order to get optimal solutions for the considered variant. The main drawback of this exact approach is the long solving time. Therefore, a metaheuristic approach has been also implemented to solve the RSPDP.

IV. METAHEURISTIC APPROACH FOR THE RSPDP

the metaheuristic approach is based on a GRASP (Greedy Randomized Adaptive Search Procedure) which is largely used to tackle such combinatorial optimization problems. This method consists in mixing the guided aspect and the random one to generate a solution. It was first introduced in [20] as a multi-start metaheuristic in which the construction of the solution is partially randomized and coupled with a local search approach. For example: authors in [21] used GRASP to address the traveling salesman problem. Recently, a GRASP approach hybridized with a genetic algorithm was applied to solve a multi-period variant of PDP [22].

The GRASP approach is divided into two parts: the constructive phase (subsection IV-A) and the improvement one (subsection IV-B).

A. Constructive phase

In this phase, a function of ordering is applied on a candidate list (CL) composed of all sites which can be inserted into the solution at each iteration. The function of ordering is based on the Waiting Time (WT) parameter calculated using equation (18). Using our GRASP, the sites having the bigger WT values are prioritized because they have a bigger possibility to stay feasible among different scenarios. The overall algorithm of our GRASP is illustrated by algorithm 1.

$$WT(i,j,k,s) = a_i - T_{ji}^{ks} - D_j^k \qquad (18)$$

The procedure begins by generating the first route. So, the depot is firstly chosen as the reference site. Then, a site is picked randomly from the restricted list of candidates (RCL) composed of the first α sites of CL. Let us note that the random selection of a candidate is equally probable. The procedure tries to insert the chosen site and its paired node (if different from the depot) into the route. After a feasible insertion, the improvement phase is applied and the last site is considered as

---

Algorithm 1: GRASP method

Input : α, β; **Output :** Solution Sol;
for *v=1 until* $|V|$ **do**
    Counter $\leftarrow$ 0;
    Ref-Node $\leftarrow$ Depot;
    Flag 1: Compute WT for all sites using
      WT(i,Ref-Node,v,ideal scenario);
    CL $\leftarrow$ Sort all non-assigned nodes by decreasing
      value of WT;
    RCL $\leftarrow$ First α sites of CL;
    Flag 2: Choose a site S randomly from RCL;
    Insert S and its paired site (if different than the
      depot) into route v;
    if *Route v is feasible and Route v is robust* **then**
      Apply improvement phase on route v;
      Ref-Node $\leftarrow$ Site at last position in route v;
      Counter $\leftarrow$ 0;
      Goto Flag 1;
    else
      Remove inserted S and its paired site;
      Counter++;
      if *Counter* $> \beta$ **then**
        | Leave the loop and goto the next route;
      else
        | Goto Flag 2;
      end
    end
end

---

the new reference node. The procedure is repeated until we did the maximum number of attempts without improvement fixed by the parameter β. Once the first route is constructed, the procedure moves on the next one and so on until the solution is built.

B. Improvement phase

Algorithm 2: Improvement phase

Input : σ, route r; **Output :** route r;
Counter=0;
while *Counter* $< \sigma$ **do**
    route r_temp $\leftarrow$ r;
    Exchange two randomly chosen sites in route
      r_temp (different from the depot);
    if *r_temp is feasible and r_temp is robust and*
    distance(r_temp) < distance(r) **then**
      r $\leftarrow$ r_temp;
      Counter $\leftarrow$ 0;
    else
      r_temp $\leftarrow$ r;
      Counter++;
    end
end

To improve the solution under construction, this phase consists in minimizing the distance of the solution which can help to insert more sites. So, it may increase the total collected profit. Indeed, two randomly chosen sites are exchanged in a route. If the route is still feasible (respects all the constraints), robust (works for all scenarios) and has a lower distance, then we keep it. This process is repeated until σ times without improvement. The algorithm 2 describes this phase.

V. Experimental results

As we mentioned earlier, the considered variant, to our best knowledge, is not treated yet in the literature. Hence, this paper presents the first instances dedicated to the RSPDP. Indeed, the instances introduced in [6] to solve the SPDP are used and adapted to the considered problem. 48 new instances are generated and solved by our exact and GRASP approaches. Each instance is labelled as RSPDP-i1-i2-i3-i4-i5 where :

- i1 is equal to "C" if the nodes are clustered; "R" if the nodes are distributed randomly and "RC" if the nodes are partially clustered and partially randomly distributed.
- i2 is the number of nodes.
- i3 is the number of scenarios.
- i4 is the ratio of the new path to the ideal time for each disturbed arc in a given scenario.
- i5 is the number of affected arcs per scenario.

The results were obtained using a commercial solver CPLEX 12.7.1 and a computer Intel core i7, 2.80 GHz CPU and 16 GB of RAM. The GRASP approach was coded in C++. The results are reported in table I. The GRASP parameters used to get those solutions are fixed after several tests as follows : $\alpha=\frac{i2}{2}$, $\beta=7$ and $\sigma=3$. In fact, if α decreases, the GRASP will get closer to the classic greedy method and if it increases, the GRASP will act like a random method. On the other hand, increasing β and σ will increase the solving time.

Table I shows the obtained profit for each approach as well as solving time in seconds. In addition, it gives the profit optimal obtained while considering the ideal scenario. The gaps between exact and metaheuristic methods in terms of profit and solving time are also reported. The obtained profits in a robust context either using CPLEX or GRASP are sometimes equal to the optimal profit in the ideal scenario which means that another solution having the same profit can be obtained despite all traveling time perturbations (for example: instance RSPDP-C20-50-1-38). In other cases, the perturbations have remarkably degraded the profit (for example: instance RSPDP-RC50-50-2-245). In addition, the exact approach has failed to solve all instances to the optimality while limiting the solving time to two hours (generally used in the literature). Solutions with "*" represent the best feasible solutions obtained using CPLEX after the fixed time limit. The GRASP has successfully solved all instances in a very short time regarding to the exact approach. The average gap in terms of solving time is 86.33% with a standard deviation of 4.32%.

Moreover, using the metaheuristic approach, the optimal solution is obtained in 14 cases (when the gap is equal to 0). For instance RSPDP-C50-100-1-490, although the gap is equal

to zero but the solution obtained is not optimal. Indeed, it is equal to the best solution obtained by CPLEX. Furthermore, a better profit than the one found by CPLEX after two hours of running was obtained in 16 cases (when we have a negative gap). On the other hand, profits obtained by the exact approach are better than the ones getting using the GRASP method in 17 cases (where a positive gap is reported). The average gap in terms of profit is 7.28% with a standard deviation of 8.26% if we took into consideration only the cases where CPLEX has obtained the optimal solution within the fixed time limit. All gap values are computed using this equation:

$$GAP(CPLEX|GRASP) = \frac{100 * (Profit_{CPLEX} - Profit_{GRASP})}{Profit_{GRASP}} \quad (19)$$

The new generated instances can be downloaded using this link: https://drive.google.com/drive/folders/1vFi5IgWH6S8LIrxZ0iwKhp7Ba41PqWy9?usp=sharing.

To summarize this part, two main points can be remarked. Firstly, dealing with a robust variant degrades the total profit regarding to the one obtained in the ideal case but the advantage is that the obtained profit is luckily to be achieved in real life. Secondly, adding scenarios can increase remarkably the complexity of the variant. Therefore, exact methods will be unable to provide the optimal solution in a reasonable time and the use of an approximate method becomes a necessity.

VI. Conclusion

In this paper, a new variant of the Pickup and Delivery Problem (PDP) was introduced. We called it the Robust Selective PDP (RSPDP) when travel times are subject to uncertainty. The aim of this variant is to find a good solution which remains feasible over a set of different scenarios. Two methods were then elaborated to solve this new variant: the first one used the well-known solver CPLEX and the other one is a Greedy Randomized Adaptive Search Procedure (GRASP).

In order to validate the two approaches, 48 instances dedicated to the RSPDP have been generated. On the one hand, the obtained results show the efficiency of our metaheuristic approach. And on the other hand, the difference between the profit in ideal scenario and the ones founded by robust approaches shows that the majority of ideal profit cannot be achieved when taking uncertain travel times into account.

Future works will be focused on improving the metaheuristic approach. In addition, trying to generate scenarios more appropriate to real life situations will be studied. Extending our variant by dealing with new objective functions or taking more real-life constraints into consideration can be also the subject of our next studies.

VII. acknowledgments

Instance	Profit in ideal scenario	CPLEX		GRASP		GAP (CPLEX|GRASP)	
		Profit	CPU time (s)	Profit	CPU time (s)	Profit	CPU time (s)
RSPDP-C20-50-1-38	180	180	56.15	180	6	0.00	89.31
RSPDP-C20-50-1-76	180	180	58.69	150	6	16.67	89.78
RSPDP-C20-50-2-38	180	180	58.94	160	8	11.11	86.43
RSPDP-C20-50-2-76	180	180	110.76	180	9	0.00	91.87
RSPDP-C20-100-1-38	180	180	119.26	160	20	11.11	83.23
RSPDP-C20-100-1-76	180	180	122.35	180	20	0.00	83.65
RSPDP-C20-100-2-38	180	180	124.08	160	21	11.11	83.08
RSPDP-C20-100-2-76	180	130	129.07	130	21	0.00	83.73
RSPDP-R20-50-1-38	190	190	57.86	180	12	5.26	79.26
RSPDP-R20-50-1-76	190	160	59.12	160	13	0.00	78.01
RSPDP-R20-50-2-38	190	170	110.02	160	13	5.88	88.18
RSPDP-R20-50-2-76	190	80	112.15	80	14	0.00	87.52
RSPDP-R20-100-1-38	190	140	223.76	140	22	0.00	90.17
RSPDP-R20-100-1-76	190	190	226.88	190	23	0.00	89.86
RSPDP-R20-100-2-38	190	180	228.76	180	24	0.00	89.51
RSPDP-R20-100-2-76	190	180	231.54	150	25	16.67	89.20
RSPDP-RC20-50-1-38	160	150	57.04	140	9	6.67	84.22
RSPDP-RC20-50-1-76	160	140	58.97	120	9	14.29	84.74
RSPDP-RC20-50-2-38	160	80	59.07	80	10	0.00	83.07
RSPDP-RC20-50-2-76	160	130	111.55	100	12	23.08	89.24
RSPDP-RC20-100-1-38	160	120	222.86	90	22	25.00	90.13
RSPDP-RC20-100-1-76	160	140	225.72	140	23	0.00	89.81
RSPDP-RC20-100-2-38	160	90	227.69	80	23	11.11	89.90
RSPDP-RC20-100-2-76	160	120	230.77	120	25	0.00	89.17
RSPDP-C50-50-1-245	480	220*	7200	360	527	-63.64	92.68
RSPDP-C50-50-1-490	480	280*	7200	380	568	-35.71	92.11
RSPDP-C50-50-2-245	480	270*	7200	340	573	-25.93	92.04
RSPDP-C50-50-2-245	480	380*	7200	330	578	13.16	91.97
RSPDP-C50-100-1-245	480	250*	7200	340	1089	-36.00	84.88
RSPDP-C50-100-1-490	480	360*	7200	360	1096	0.00	84.78
RSPDP-C50-100-2-245	480	250*	7200	330	1103	-32.00	84.68
RSPDP-C50-100-2-490	480	280*	7200	340	1109	-21.43	84.60
RSPDP-R50-50-1-245	490	300	5732.59	280	767	6.67	86.62
RSPDP-R50-50-1-490	490	310*	7200	400	774	-29.03	89.25
RSPDP-R50-50-2-245	490	330*	7200	400	778	-21.21	89.19
RSPDP-R50-50-2-245	490	280	5823.46	280	783	0.00	86.55
RSPDP-R50-100-1-245	490	340*	7200	390	1390	-14.71	80.69
RSPDP-R50-100-1-490	490	360	6492.13	310	1405	13.89	78.36
RSPDP-R50-100-2-245	490	290*	7200	360	1410	-24.14	80.42
RSPDP-R50-100-2-490	490	290*	7200	340	1425	-17.24	80.21
RSPDP-RC50-50-1-245	320	140*	7200	200	632	-42.86	91.22
RSPDP-RC50-50-1-490	320	160*	7200	220	657	-37.50	90.88
RSPDP-RC50-50-2-245	320	120	5681.15	100	679	16.67	88.05
RSPDP-RC50-50-2-245	320	210*	7200	290	712	-38.10	90.11
RSPDP-RC50-100-1-245	320	160*	7200	220	1296	-37.50	82.00
RSPDP-RC50-100-1-490	320	180*	7200	210	1305	-16.67	81.88
RSPDP-RC50-100-2-245	320	130	6235.96	100	1312	23.08	78.96
RSPDP-RC50-100-2-490	320	160	6318.12	160	1350	0.00	78.63

TABLE I
EXPERIMENTAL RESULTS ON NEW GENERATED INSTANCES

REFERENCES

[1] J. Desrosiers, Y. Dumas, M. M. Solomon, and F. Soumis, "Time constrained routing and scheduling," *Handbooks in operations research and management science*, vol. 8, pp. 35–139, 1995.

[2] S. N. Parragh, K. F. Doerner, and R. F. Hartl, "A survey on pickup and delivery problems. part i: Transportation between customers and depot," *Journal für Betriebswirtschaft*, vol. 58, no. 1, pp. 21–51, 2008.

[3] S. N. Parragh, K. F. Doerner, and R. F. Hartl, "A survey on pickup and delivery problems. part ii: Transportation between pickup and delivery locations," *Journal für Betriebswirtschaft*, vol. 58, no. 2, pp. 81–117, 2008.

[4] F. Carrabs, R. Cerulli, and J.-F. Cordeau, "An additive branch-and-bound algorithm for the pickup and delivery traveling salesman problem with lifo or fifo loading," *INFOR: Information Systems and Operational Research*, vol. 45, no. 4, pp. 223–238, 2007.

[5] S. Ropke and J.-F. Cordeau, "Branch and cut and price for the pickup and delivery problem with time windows," *Transportation Science*, vol. 43, no. 3, pp. 267–286, 2009.

[6] Z. Al Chami, H. Manier, and M.-A. Manier, "New model for a variant of pick up and delivery problem," in *2016 IEEE International Conference on Systems, Man, and Cybernetics (SMC)*, Oct 2016, pp. 001 708–001 713.

[7] Z. Al Chami, H. Manier, and M.-A. Manier, "A lexicographic approach for the bi-objective selective pickup and delivery problem with time windows and paired demands," *Annals of Operations Research*, Apr

2017. [Online]. Available: https://doi.org/10.1007/s10479-017-2500-9

[8] H. Li and A. Lim, "A metaheuristic for the pickup and delivery problem with time windows," *International Journal on Artificial Intelligence Tools*, vol. 12, no. 02, pp. 173–186, 2003.

[9] S. Zou, J. Li, and X. Li, "A hybrid particle swarm optimization algorithm for multi-objective pickup and delivery problem with time windows." *JCP*, vol. 8, no. 10, pp. 2583–2589, 2013.

[10] Z. Al Chami, H. Manier, M.-A. Manier, and C. Fitouri, "A hybrid genetic algorithm to solve a multi-objective pickup and delivery problem," *IFAC-PapersOnLine*, vol. 50, no. 1, pp. 14 656 – 14 661, 2017, 20th IFAC World Congress.

[11] Z. Al Chami, H. El Flity, H. Manier, and M.-A. Manier, "A new metaheuristic to solve a selective pickup and delivery problem," in *2018 4th International Conference on Logistics Operations Management (GOL)*, April 2018, pp. 1–5.

[12] F. Ordóñez, "Robust vehicle routing," *TUTORIALS in Operations Research*, pp. 153–178, 2010.

[13] I. Sungur, F. Ordóñez, and M. Dessouky, "A robust optimization approach for the capacitated vehicle routing problem with demand uncertainty," *IIE Transactions*, vol. 40, no. 5, pp. 509–523, 2008.

[14] R. Montemanni, J. Barta, and L. Gambardella, "An exact algorithm for the robust travelling salesman problem with interval data," *Proceedings of ODYSSEUS 2006*, pp. 256–258, 2006.

[15] P. Munari, A. Moreno, J. De La Vega, D. Alem, J. Gondzio, and R. Morabito, "The robust vehicle routing problem with time windows: compact formulation and branch-price-and-cut method."

[16] A. Chassein and M. Goerigk, "On the recoverable robust traveling salesman problem," *Optimization Letters*, vol. 10, no. 7, pp. 1479–1492, 2016.

[17] E. L. Solano-Charris, C. Prins, and A. C. Santos, "A robust optimization approach for the vehicle routing problem with uncertain travel cost," in *Control, Decision and Information Technologies (CoDIT), 2014 International Conference on*. IEEE, 2014, pp. 098–103.

[18] E. Solano-Charris, C. Prins, and A. C. Santos, "Local search based metaheuristics for the robust vehicle routing problem with discrete scenarios," *Applied Soft Computing*, vol. 32, pp. 518–531, 2015.

[19] 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*, vol. 94, pp. 139–153, 2018.

[20] T. A. Feo and M. G. Resende, "Greedy randomized adaptive search procedures," *Journal of global optimization*, vol. 6, no. 2, pp. 109–133, 1995.

[21] S. H. Lee, "Greedy randomized adaptive search procedure for traveling salesman problem," Ph.D. dissertation, Texas A&M University, 2006.

[22] Z. Al Chami, H. Manier, M.-A. Manier, and E. Chebib, "An advanced grasp-hga combination to solve a multi-period pickup and delivery problem," *Expert Systems with Applications*, vol. 105, pp. 262 – 272, 2018.