

Evolving Robust Solutions to Uncertain Capacitated Arc Routing Problem with Time Window

1st Shuo Huang^{*†}, 1st Yinxi Tian^{*†} 2nd Xintong Duan ^{*}

^{*} Department of Computer Science and Engineering
Southern University of Science and Technology

[†] These authors contributed equally to this work and should be considered co-first authors

Abstract—The capacitated arc routing problem with time window has been studied for years, while uncertainties in real-world scenarios, such as uncertain demands and uncertain routing conditions, were rarely considered. In this paper, we formulate an uncertain capacitated arc routing problem with time window that takes into account the uncertain demand of tasks, the uncertain deadheading costs and the uncertain presence of tasks and paths. A number of problem instances are generated based on the benchmark instances of static capacitated arc routing problem with time window considering the aforementioned four uncertain factors. To tackle this new challenging problem, we adapt a state-of-the-art algorithm for solving uncertain capacitated arc routing problem, an estimation of distribution algorithm (EDA) with a stochastic local search, to find robust solutions. Another algorithm, a memetic algorithm with extended neighborhood search, is also adapted as a baseline solution algorithm to this challenging problems. Our experimental results indicate that our EDA-based algorithm is effective in finding robust solutions to uncertain capacitated arc routing problems with time window.

Index Terms—Uncertain capacitated arc routing problem with time window (UCARP-TW), uncertain capacitated arc routing problem (UCARP), capacitated arc routing problem with time window (CARP-TW), robust optimisation, estimation of distribution.

I. INTRODUCTION

Capacitated arc routing problem with time window (CARP-TW) is a well-known combinatorial optimization problem defined on a graph. Let $G(E, V)$ be a connected graph with an edge set E and a vertex set V . Each edge $e \in E$ has a positive cost $c(e)$ and a nonnegative demand $d(e)$. Each edge $e \in E$ with positive demand, called a task, is added with a time window. All the tasks make up a task set T . A group of vehicles in the depot $v_0 \in V$ arranged to serve all the tasks in T . Each of them has a capacity Q . A route is a sequence of tasks. Its cost can be calculated by adding up each task's cost and the time penalty caused by serving each task outside the time window. The solution of CARP-TW is a set of routes. The total cost $TC(s)$ of a solution s is the sum of all the routes' cost. The target of this problem is to find a solution s with as small $TC(s)$ as possible with the following constraints: The sum of demand of tasks served in one route should not exceed

Q ; Each task in the task set is served once and only once; Both of each route's endpoints should be the depot v_0 .

As an NP-hard problem [1], CARP-TW is addressed by many methods. Exact methods such as [2] and heuristic methods like [3] have been constructed to solve CARP-TW. Both of them are effective.

While the uncertainty of CARP-TW is vital in the real world, only a few works considered some random factors involved in CARP-TW [4]–[6]. Specifically, demand [5], [6] and time window [4] are considered uncertain in those works about CARP-TW. However, there is much more uncertainty in real life, like the change of graph, which might be caused by road maintenance.

CARP-TW is an extension of capacitated arc routing problem (CARP). To get a problem closer to real life, a problem called uncertain capacitated arc routing problem (UCARP) was proposed [7] by adding uncertainty to CARP. In UCARP, the uncertainty of demand of tasks, cost of edges presence of edges, and presence of tasks are added [8]. Considering the similarity between CARP and CARP-TW, these uncertainties can be combined with CARP-TW to form a new problem, uncertain capacitated arc routing problem with time window (UCARP-TW), which is more realistic compared with just considering the demand of edge or the time window.

Estimation of the distribution algorithm with a stochastic local search (EDASLS) is a state-of-art algorithm of UCARP [9]. It can be adjusted to get a new algorithm to solve the UCARP-TW. The new algorithm shows a premise in finding a good solution because of the excellent performance of EDASLS in UCARP and the similarity of UCARP and UCARP-TW.

In this work, UCARP-TW is first formalized mathematically. We then adapt the EDASLS to solve this problem¹. In UCARP-TW, four uncertainties are considered. In addition, a soft time window rather than a hard time is added to each task. An extra cost would be added to the total cost as a penalty for violation of time windows, which is more flexible and closer to real-life than forcing each task to be served in its time window. The time penalty is determined by the speed of the

This work was supported by the SUSTech Undergraduate Teaching Quality and Reform Project (Grant No. SJZLGC202101).

¹Our dataset, instance generator and program are available at <https://github.com/tian-y-x/UCARPTW>.

vehicle, the penalty coefficient and the hours it violates. Some coefficients can be adjusted according to the real situations. Moreover, we also focus on the robustness of the solution and try to find a solution with an acceptable cost when the graph and task set are changed.

To dive deeper into this problem, we generate a UCARP-TW dataset from the classic CARP-TW dataset [10] by adding uncertainties to the demand and cost of the edges but also randomly deleting edges and tasks. This dataset can be used to test other algorithms as well in solving UCARP-TW in the future. We test our algorithm on this new dataset, and the experimental results show that our new approach for UCARP-TW performs very well.

The following are our contributions: (i) we formulate a new model for UCARP-TW, which considers more uncertain factors than other existing models and has more flexible time constraints; (ii) we create a UCARP-TW data set based on gdb [10] through an uncertainty generator [9]; (iii) we adapt the EDASLS to this new problem.

The remainder of this paper is organized as follows. Section II reviews the related work. We describe the UCARP-TW problem we defined in section III-A, formulate its mathematical model in section III-B and discuss how we create the data set in section III-C. In section IV, we explain how we use EDASLS to solve the UCARP-TW. The performance of our experiment is shown in section V. Section VI concludes our work and points out some future work.

II. RELATED WORK

This section reviews related work in routing problems with time windows considered uncertain factors.

A. Uncertain arc routing problems with time window

Tirkolaee et al. [11] considered uncertain factors on PCARP with time window. PCARP is an extended version of CARP that the required edge can change weekly. In this context, the uncertain demand is modelled as a symmetric and bounded random variable and using improved simulated annealing (SA) to find a robust solution that minimizes travel and waiting costs. Sun et al. [5] proposed an uncertain model of CARP-TW. In their model, the demand is stochastic but follows a predefined distribution. They developed a fuzzy expected value model and used credibility measures to eliminate the uncertain factors. In [6], Kahfi et al. presented an application of bank cash supply. They considered that the demand has uncertainty with unknown probability distributions. In this case, the time constraint is unable to violate but waiting time is allowed for each required arc at the start of service. Moreover, the modified Multi-Objective Dragonfly Algorithm (MODA) and Non-dominated Sorting Genetic Algorithm (NSGA-II) can provide a robust solution for indeterminate scenarios. Two numeric representations are applied on UCARP and compared with the tree representation in [12].

B. Uncertain vehicle routing problems with time window

Notably, uncertainty is also extensively studied in vehicle routing problem with time window (VRPTW). [13] focused

on the uncertainty of travel time and solved it by column generation. [9] focused on the uncertainty of service time and solved it by the branch-cut-price algorithm. Both [13] and [9] studied the circumstance that the uncertain factor followed a known distribution and solved it by an exact method. The solution gained by them is only appropriate for a particular case. While [14] considered the travel time uncertain with unknown probability distributions. [14] tackled uncertainties with range estimates instead of point estimates. It finds the worst-case within the given ranges, aiming for a robust solution with the minimum expected cost under different scenarios. In the work of [15], the travel time and demand are uncertain. [16]–[18] assumed both the travel time between any two customers and service time are uncertain. All of them proposed a new approach to solving it. [7] proposed a new variant that the travel time and travel cost are uncertain. [5] adapted a model that the demands of the customers are uncertain.

Most of the research assumes a priori known distribution, though an unknown distribution is more realistic. Besides, no one has considered the time window constraint, the uncertainty of the value of cost and demand, and the existence of tasks and edges simultaneously.

III. UNCERTAIN CAPACITATED ARC ROUTING PROBLEM WITH TIME WINDOW

This part shows some basic definitions and the mathematical model of UCARP-TW. The way we generate new dataset is also introduced here.

A. Problem Description

A UCARP-TW instance is defined on a complete and undirected graph $G = (V, E)$, in which V represents a set of vertices and E represents a set of edges. Each $e \in E$ has a cost $c(e)$. The value of an edge's cost is either a positive number or -1. An edge's cost is -1 if and only if it is missing. There is a *depot* $\in V$, and all vehicles must start and end at the *depot*. Subset $T \in E$ is the set of all the edges required to be served, called task set. Each required edge $e \in T$ is associated with a positive demand $d(e)$, which has to be served only once. A soft time window $[a, b]$ is also added to a task, and a cost would be added as a penalty if a task can not be served in the time window. Each vehicle would take a cost $c(e)$ for traversing each edge $e \in E$. In this work, the number of vehicles is assumed to be infinite, which means the initial time for each route is zero.

Defined in [8], the uncertainties of UCARP is reflected in four aspects:

- 1) The existence of the demand of an edge
- 2) The demand of an required edge
- 3) The existence of an edge
- 4) The cost of traversing an edge

A set of CARP-TW instances can be used to approximate a UCARP-TW instance. Let θ be a UCARP-TW instance where $\theta = \{I_1, I_2, \dots, I_N\}$. For any $j \in \{1, \dots, N\}$, I_j is a CARP-TW instance. All $I_j \in \theta$ are related but different from each other.

UT is the union set of task set of each CARP-TW instances in θ , i.e., $UT = \cup_{j=1}^N T_j, \forall j \in \{1, \dots, N\}$ and T_j is the task set of I_j [9]

The objective of this problem is to find a set of routes with minimum cost in a UCARP-TW instance. The total cost $TC(s)$ of a solution s in a UCARP-TW instance is defined as the maximum cost of each CARP-TW instance in this UCARP-TW instance.

B. Solution representation

We set $s=(R_1, R_2, \dots, R_m)$ as a solution of the UCARP-TW, where $R_k=(t_{k,1}, \dots, t_{k,l_k})$. R_k is the k^{th} route. l_k is the length of the k^{th} route and $t_{k,b}$ is the b^{th} task served in the k^{th} route. Let head(t) and tail(t) be the two endpoints of task t, inv(t) is task t with inverse direction. Q is the capacity of a vehicle. $d(t, I_j)$ is the demand of task t in instance I_j .

C. Mathematical model

UCARP-TW's objective is to minimize the maximum cost of a solution among all the instances, called $R(s)$, which can be formulated as the following objective and constraints. Eq(2) guarantees that all the data set's tasks are served in a solution. Eq(3) guarantees the legality of a route, which means a vehicle can serve all the tasks in a route without returning to the depot before finishing the serving of this route. Eq(4) is the calculation of the route's cost in a particular instance. Eq(5) is the excess time. Eq(6) is the calculation of the served time of a task.

$$\text{minimize } \max_{I_j \in \theta} \sum_{k=1}^m C(R_k, I_j) \quad (1)$$

$$\text{s.t.} \quad \sum_{k=1}^m l_k = |UT|, \quad (2)$$

$$\sum_{i=1}^{l_k} d(t_{k,i}, I_j) \leq Q, \quad \forall k = 1, 2, \dots, m, \forall j = 1, 2, \dots, N \quad (3)$$

$$\begin{aligned} C(R_k, I_j) = & c(t_{k,1}, I_j) + \text{dis}(v_0, \text{head}(t_{k,1}), I_j) \\ & + \sum_{i=2}^{l_k} (c(t_{k,i}, I_j) + \text{dis}(\text{tail}(t_{k,i-1}), \text{head}(t_{k,i}), I_j) \\ & + \text{dis}(\text{tail}(t_{k,l_k}), v_0, I_j) \\ & + w * \phi(f_{k,i,j})^2 / (v * (b_{k,i} - a_{k,i}))) \end{aligned} \quad (4)$$

$$\phi(f_{k,i,j}) = \max(a_{k,i,j} - f_{k,i,j}, f_{k,i,j} - b_{k,i,j}, 0), \quad \forall k = 1, 2, \dots, m, \forall j = 1, 2, \dots, N \quad (5)$$

$$\begin{aligned} f_{k,i,j} = & cm(j) * \text{dis}(v_0, \text{head}(t_{k,1}), I_j) \\ & + dm(j) * c(t_{k,1}, I_j) \\ & + \sum_{m=1}^i (cm(j) * \text{dis}(\text{tail}(t_{k,m}), \text{head}(t_{k,m+1}), I_j) \end{aligned}$$

$$+ dm(j) * c(t_{k,m+1})) \quad (6)$$

D. Dataset for UCARP-TW

Considering that an edge's demand and cost can vary according to the environment in real-life scenarios, the costs and demands of edges can be characterized as a function of the environmental variables altogether denoted as ξ [19]. Mathematically, the presence of an edge (v_i, v_j) has a probability of q_{ij} . When this edge is present, the cost of traversing it should follow a certain distribution where the probability density function of this distribution is a function of the factors in ξ that are related to the cost of this edge. Otherwise, the cost of traversing it will be infinity. Similarly, the demand of an edge (v_i, v_j) has a probability of p_{ij} . When this edge is a task that needs to be served, the demand of this edge also follows a distribution whose probability density function is a function of certain factors in ξ related to the demand of the edge. Otherwise, the demand is 0.

We derive our UCARP-TW dataset from CARP-TW dataset by adding uncertainties to the existence of edges and tasks and the cost of demand of edges. The CARP-TW dataset proposed in [10] has four small subsets and each subset consists of eight instances of different sizes. The number of edges ranges from 15 to 90 and the number of vertices ranges from 10 to 60.

To generate the uncertain instances, we replace the demand with a Gamma distribution $G(k, \theta)$ as Gamma distribution is positive and similar to Gaussian distribution with sufficiently large k [19]. In this case, k is set to 20. Also, the demand of an edge (v_i, v_j) $d(v_i, v_j)$ has a probability of $1 - p_{ij}$ being zero and the cost of an edge (v_i, v_j) has a probability of $1 - q_{ij}$ of being a negative one. r is a random variable between 0 to 1. Therefore, the demand of edges of UCARP-TW should satisfy the following distribution:

$$D(v_i, v_j) \begin{cases} \sim G(k_{ij}^d, \theta_{ij}^d), & r < p_{ij}; \\ = 0, & \text{Otherwise.} \end{cases}$$

The cost of edges of UCARP-TW should satisfy the following distribution:

$$D(v_i, v_j) \begin{cases} \sim G(k_{ij}^c, \theta_{ij}^c), & r < q_{ij}; \\ = \infty, & \text{Otherwise.} \end{cases}$$

For more details, please refer to [19]. The UCARP generator can be downloaded from ².

One UCARP-TW instance contains 30 CARP-TW instances sharing the same graph structure in this study. Also, the random seed is set to zero and the shape parameter k of Gamma distribution is set to 20 while the scale parameter $\theta = \text{the static demand or cost of an edge}/k$. It is worth noting that only the cost and demand of edges are uncertain but there is no uncertainty on the time window of edges, which means the time window on the same edge is the same in one UCARP-TW instance.

²<http://home.ustc.edu.cn/~meiyi>

TABLE I
NOTATION TABLE

Name	Description
$t_{k,i}$	the i^{th} task of the k^{th} route
$d(t_{k,i}, I_j)$	demand of $t_{k,i}$ in the j^{th} instance
$c(t_{k,i}, I_j)$	cost of edge of $t_{k,i}$ in the j^{th} instance
Q	the capacity of a vehicle
I_j	the j^{th} CARP instance
θ	a UCARP instance ($\theta = \{I_1, I_2, \dots, I_N\}$)
UT	the union of the task sets in all the CARP-TW instances
$f_{k,i,j}$	the time when task $t_{k,i}$ of the j^{th} instance is served
$dis(a, b, I_j)$	the shortest distance between node a and b in the j^{th} instance
v	the velocity of each vehicle
w	the penalty coefficient for excess time
$\phi(f_{k,i,j})$	the time penalty of the k^{th} route's i^{th} task in the j^{th} instance
$dm(j)$	the multiplier for serving a task in the j^{th} instance
$cm(j)$	the multiplier for traversing a task in the j^{th} instance

IV. ADAPTING EDASLS FOR SOLVING UCARP-TW

Considering that the main difference between UCARP-TW and CARP-TW is the uncertainty in the graph and demand, it is possible to get better performance if we focus more on the uncertainty. EDASLA is an effective meta-heuristics for UCARP with good performance, which can be used to solve UCARP on a large scale [9]. MAENS is the earliest algorithm applied to solve UCARP and show a good performance on it. We also modify MAENS to solve UCARP-TW. Algorithm 1 shows the pseudocode of EDASLS and Algorithm 2 shows the pseudocode of MAENS.

In the adaption of EDASLS [9], we set the permutation of all the task as the chromosome of our problem, which can be evaluated and split into routes through Ulusoy's splitting procedure [20]. There are $|\theta|$ CARP-TW instances in a UCARP-TW instance and a chromosome can be turned into $|\theta|$ solutions. Set S_{I_j} as the solution got from the instance I_j and $F(c)$ as the fitness of chromosome c , we have: $F(c) = \max_{I_j \in \theta} TC(S_{I_j}, I_j)$. To get a robust solution from the $|\theta|$ solutions, we run each of them on $|\theta|$ CARP-TW instances. The one with minimum maximum in $|\theta|$ instances would be chosen as the solution we want. If its cost is smaller than the global solution s_r , it would replace s_r .

At first, we use path-scanning [21], best-insertion [9] and reshuffling to create the initial population. Then a famous algorithm called EHBSA [22] is applied to the population. During the process, stochastic local search is randomly adapted to improve the chromosome. After reaching the stopping criteria, the best solution is picked out as the robust solution this algorithm finds.

In the modification of MAENS, we refer to the way of [19]. Considering that the target of UCARP in [19] is to find a solution with good average performance while our work focus on a good performance in the worst case, we replace the fitness function of MAENS as the $R(s)$.

Best-insertion is applied to build the initial population. Then, new chromosomes would be produced through crossover operator and local search repeatedly until reaching the criteria of stopping. Finally, it would return the best solution it finds.

Algorithm 1: EDASLS

Data: a set of instances θ , a set of tasks UT , probability of local search pls , total times of fitness evaluation FEm

Result: the best solution s_r

```

1  $pop \leftarrow \text{initialization}()$ 
2 for each  $c_i \in pop$  do
3    $F(c_i) \leftarrow \text{CEvaluate}(c_i, \theta, UT)$ 
4   update  $s_r$ 
5    $index[i] \leftarrow i$ 
6 end
7 sort index in ascending order according to the fitness value.
8 while the evaluate times is smaller than  $FEm$  do
9   for  $i=0$  to  $popsize/2 - 1$  do
10    for  $j=0$  to  $UT-1$  do
11       $pop\_temp[i][j] \leftarrow pop[index[i]][j]$ 
12    end
13  end
14   $ehm \leftarrow$  an edge histogram matrix based on  $pop\_temp$ 
15  sample a random integer  $ra$  from the uniform distribution between 0 to  $popsize-1$ 
16   $child \leftarrow \text{EHBSA}(pop[ra], ehm, 2, \text{size of } UT)$ 
17  sample a random number  $rb$  from the uniform distribution between 0 to 1
18  if  $rb < pls$  then
19     $child \leftarrow \text{StochasticLS}(child, ehm, \theta)$ 
20  end
21  update  $s_r$  after evaluating child
22  if child is better than  $pop[ra]$  and is different from any other chromosomes in the population then
23    replace  $pop[ra]$  with child
24  end
25  sort index in ascending order according to the fitness value.
26 end
27 return  $s_r$ 

```

Algorithm 2: MAENS

Data: a set of instances θ , a set of tasks UT , probability of local search pls , total times of fitness evaluation FEm

Result: the best solution s_r

```
1 pop  $\leftarrow$  initialization with best insertion
2 for each  $c_i \in pop$  do
3    $F(c_i) \leftarrow CEvaluate(c_i, \theta, UT)$ 
4   update  $s_r$ 
5 end
6 while the evaluate times is smaller than  $FEm$  do
7   randomly pick  $c_i \in pop$  and  $c_j \in pop$ 
8   generating  $c_x$  by using the crossover operator to  $c_i$ 
   and  $c_j$ .
9   sample a random number  $rb$  from the uniform
   distribution between 0 to 1
10  if  $rb < pls$  then
11     $c_x \leftarrow ExtendedLS(c_x, \theta)$ 
12  end
13  update  $s_r$  after evaluating  $c_x$ 
14  if  $c_x$  is better than  $pop[ra]$  and is different from
   any other chromosomes in the population then
15    replace  $pop[ra]$  with  $c_x$ 
16  end
17 end
18 return  $s_r$ 
```

V. EXPERIMENTAL STUDIES

To evaluate the effectiveness of the adapt EDASLS, we run the two algorithms on the new dataset we create. In this work, the CARP-TW dataset used to develop UCARP-TW dataset contain problem sets A, B, C, and D proposed in [10]. The main difference among the four sets is the width of the time window: The time window is becoming larger and larger from A to D. Each of these sets contains eight CARP-TW instances with different number of tasks, edges and nodes. The parameters for EDASLS in UCARP-TW algorithm are shown in Table III and the parameters of MAENS are listed in Table IV. Both algorithms are assigned with the same times of fitness evaluation times. Fitness evaluation is the calculation of a chromosome's fitness. In this work, 20 independent runs are performed on every instance of UCARP-TW and the average execution time, average $R(s_{best})$ and standard deviation of fitness are calculated. s_{best} is the current best solution. All the implementation is by C++ and the processor is AMD Ryzen Threadripper PRO 3995WX 64-Cores, 2.7GHz, 256MB RAM with Ubuntu 20.04.2 LTS operating system.

Fig 1 shows the convergence curves of two algorithms in instances TW-A10A, TW-A13C, TW-B20B, TW-B40D, TW-C40D and TW-D60A. Moreover, Wilcoxon rank-sum test with 0.05 significance is applied to the 20 executions and the average value of EDASLS would be marked with '*' if it has a significantly better performance than MAENS in this instance.

From the convergence curves, we can see that Both two algorithms converge faster in smaller instances than in large instances. Both algorithms can get a solution with acceptable cost within 300000 fitness evaluations in instances with less than or equal to 20 vertices.

From the Average column of Table II, we can see that both algorithms find the same result in the four small instances TW-A10A, TW-B10A, TW-C10A and TW-D10A. In other 28 instances, EDASLS can find a better solution than MAENS in 11 cases. However, the Wilcoxon rank-sum test with 0.05 significance shows that EDASLS can find a better solution in 19 instances. In some instances, EDASLS performs worse than MAENS, but win in Wilcoxon rank-sum test. There is a common point in these instances: The distance in average value is small and the variance is large.

In most instances with less than 60 vertices, modified EDASLS can find a better solution. However, in all the instances with 60 vertices, EDASLS's performance is worse than MAENS. That's because MAENS's search step is larger than EDASLS, which means it is less likely to fall into a local minimum. However, in instances with small solution space, EDASLS can find the best solution while MAENS may miss it.

In all the instances except A60A and A13B of the problem set A, EDASLS performs better than MAENS, while MAENS performs better than EDASLS in problem set D except for D40D. It shows that EDASLS can tackle problems with strict time windows better than dealing with problems with wide time windows compared with MAENS. The reason may be that EDASLS use ehbm both in EHBSA and stochastic local search, which can increase the probability of two tasks with close time windows being served adjacently. Time penalty can be huge if the served time exceeds the limitation of a time window a lot and the use of ehbm can reduce the time penalty quickly.

In most instances, EDASLS can get a better initial value. That's because EDASLS uses CEvaluate to get the best partition way of a chromosome, while MAENS can not guarantee each solution is the best partition on the current executive order, which leads to the huge initial time penalty of MAENS.

The execution time of EDASLS is larger than MAENS, that's because Ulusoy's splitting is a time-costing activity. What's more, EDASLS needs to compare $|\theta|$ solution's $R(s)$ and choose the best one in updating the best solution, while MAENS's chromosome is a split solution, which means it just needs one calculation of $R(s)$ to update the best solution.

VI. CONCLUSION AND FUTURE WORK

In this paper, we formulate a new problem UCARP-TW taking the time window constraint and uncertain factors into account, whose objective is to find a robust solution. Besides, two famous algorithms are modified to solve it, and a new dataset for UCARP-TW has been generated to test our algorithms. four uncertainty to CARP-TW, more uncertainties like the variation of time windows can be considered. Moreover, the modified EDASLS and modified MAENS should be used

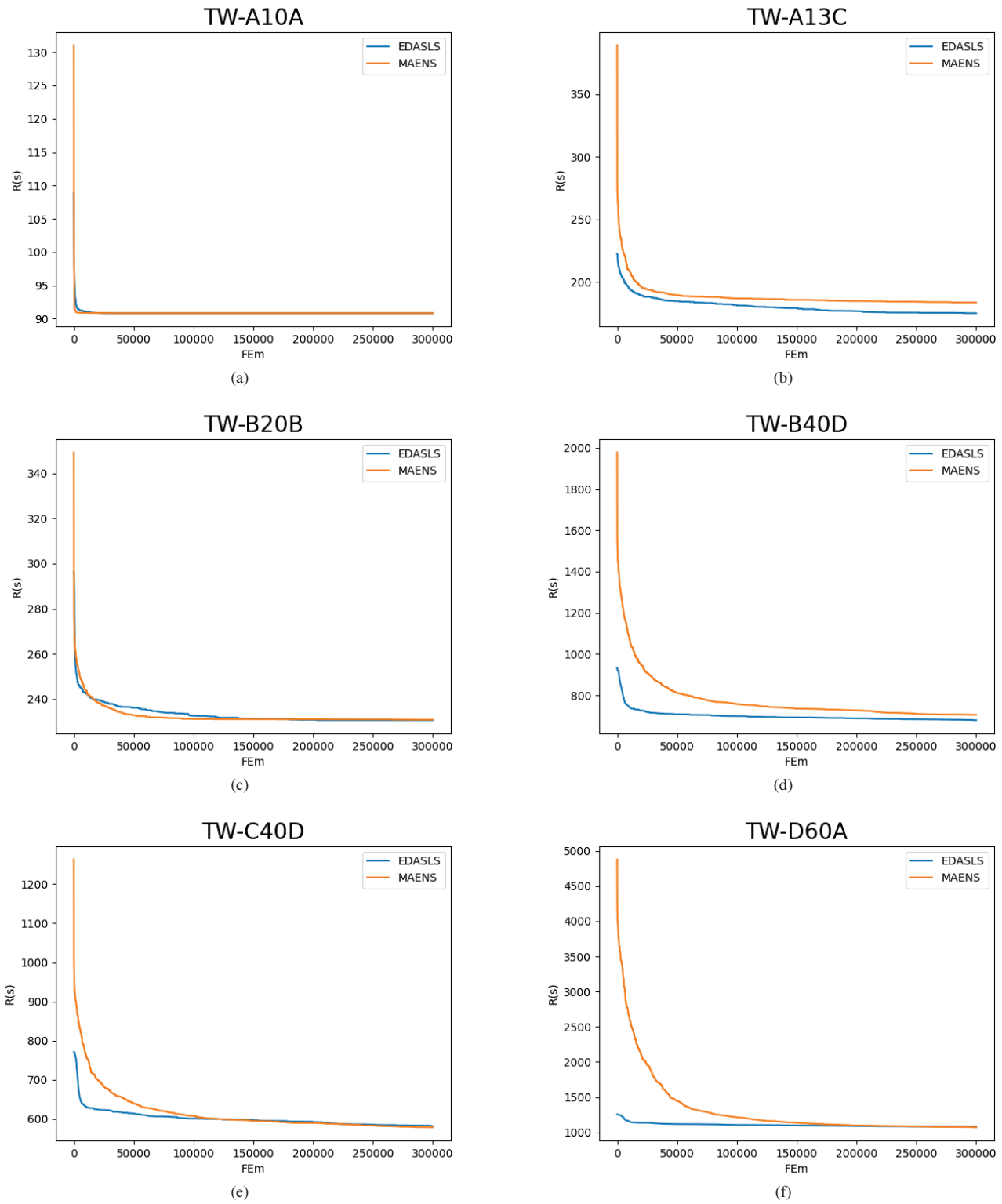


Fig. 1. The convergence curves of our algorithms on instances TW-A10A, TW-A13C, TW-B20B, TW-B40D, TW-C40D and TW-D60A

TABLE II
RESULTS ON UCARP-TW INSTANCES

Name	$\ E\ $	$\ V\ $	EDASLS			MAENS		
			Time	Average	Std	Time	Average	Std
<i>TW-A10A</i>	15	10	70.641	90.842	0	7.783	90.842	0
<i>TW-A13A</i>	23	13	160.033	185.636*	2.813	18.941	193.284	5.979
<i>TW-A13B</i>	23	13	158.854	179.139	2.785	18.588	176.933	3.954
<i>TW-A13C</i>	23	13	157.168	175.928*	2.789	18.969	183.200	5.159
<i>TW-A20B</i>	31	20	212.887	247.942*	1.683	37.789	249.237	3.357
<i>TW-A40C</i>	69	40	578.084	809.623*	21.222	128.912	871.779	56.644
<i>TW-A40D</i>	69	40	583.794	848.272*	17.593	130.958	1027.870	92.512
<i>TW-A60A</i>	90	60	674.334	1655.830	9.151	257.859	1622.120	25.070
<i>TW-B10A</i>	15	10	70.593	82.579	0	7.604	82.579	0
<i>TW-B13A</i>	23	13	160.007	149.652	0.633	18.443	147.943	0
<i>TW-B13B</i>	23	13	161.979	140.942*	1.402	18.446	141.065	1.002
<i>TW-B13C</i>	23	13	159.265	150.801*	0.870	18.120	150.247	1.636
<i>TW-B20B</i>	31	20	213.763	230.416*	0.995	37.637	231.638	2.106
<i>TW-B40C</i>	69	40	571.281	639.606*	9.402	121.363	641.109	23.801
<i>TW-B40D</i>	69	40	579.946	680.500*	6.026	127.788	691.244	31.884
<i>TW-B60A</i>	90	60	683.095	1619.740	11.745	259.33	1589.450	26.381
<i>TW-C10A</i>	15	10	66.043	81.825	0	7.334	81.825	0
<i>TW-C13A</i>	23	13	156.627	132.535*	0.272	17.822	132.505	0.471
<i>TW-C13B</i>	23	13	153.734	127.652*	0.214	17.843	128.303	0.640
<i>TW-C13C</i>	23	13	153.987	130.692	0.317	18.091	130.164	0.438
<i>TW-C20B</i>	31	20	195.448	223.820*	1.293	37.625	223.025	2.159
<i>TW-C40C</i>	69	40	573.546	542.597	6.012	118.332	525.987	7.148
<i>TW-C40D</i>	69	40	571.863	580.169*	6.639	123.160	575.086	18.435
<i>TW-C60A</i>	90	60	667.590	1599.950	12.612	257.117	1558.630	21.442
<i>TW-D10A</i>	15	10	70.758	71.446	0	6.170	71.446	0
<i>TW-D13A</i>	23	13	169.597	150.090	1.330	17.880	148.622	2.725
<i>TW-D13B</i>	23	13	168.055	141.363*	1.673	17.744	141.291	1.181
<i>TW-D13C</i>	23	13	167.389	137.920*	0.431	14.999	137.898	0.393
<i>TW-D20B</i>	31	20	244.711	176.745	1.557	27.597	171.136	1.181
<i>TW-D40C</i>	69	40	689.533	624.665*	8.233	121.677	620.248	22.350
<i>TW-D40D</i>	69	40	691.836	681.678*	9.909	127.346	729.017	47.701
<i>TW-D60A</i>	90	60	916.391	1084.470*	9.056	210.576	1079.51	31.822

TABLE III
THE PARAMETERS OF EDASLS

Name	Description	Value
v	Velocity of Vehicle	30
w	Penalty Coefficient For Excess Time	1
$popsiz$	Population Size	120
n	Number Of Sub Segments In EHBSA	2
p_{ls}	Probability of executing stochastic local search	0.1
FE_m	Maximum number of fitness evaluation	300000

TABLE IV
THE PARAMETERS OF MAENS

Name	Description	Value
v	Velocity of Vehicle	30
w	Penalty Coefficient For Excess Time	1
$popsiz$	Population Size	30
$opsize$	Size of offspring generated in each generation	180
p	Number Of Routes Picked in Merge-Split operator	2
p_{ls}	Probability of executing stochastic local search	0.2
FE_m	Maximum number of fitness evaluation	300000

to solve large-scale UCARP-TW instances, especially real-world problems.

There are some potential directions to be explored. For example, the CARP-TW dataset gdb is a small-scale dataset, so other datasets with larger scales could be added to test the two algorithms. Besides, other state-of-the-art algorithms can

also be adapted to solve this problem.

REFERENCES

- [1] B. L. Golden and R. T. Wong, "Capacitated arc routing problems," *Networks*, vol. 11, no. 3, pp. 305–315, 1981.
- [2] E. L. Johnson, S. Wöhlk *et al.*, "Solving the capacitated arc routing problem with time windows using column generation," *Dept. Bus. Stud., Univ. Aarhus, Aarhus, Denmark, Tech. Rep. L-2008-09*, 2009.
- [3] M. Reghioui, C. Prins, and N. Labadi, "Grasp with path relinking for the capacitated arc routing problem with time windows," in *Workshops on Applications of Evolutionary Computation*. Springer, 2007, pp. 722–731.
- [4] E. B. Tirkolaee, I. Mahdavi, and M. M. S. Esfahani, "A robust periodic capacitated arc routing problem for urban waste collection considering drivers and crew's working time," *Waste Management*, vol. 76, pp. 138–146, 2018.
- [5] Y. Sun, X. Liang, X. Li, and C. Zhang, "A fuzzy programming method for modeling demand uncertainty in the capacitated road–rail multimodal routing problem with time windows," *Symmetry*, vol. 11, no. 1, p. 91, 2019.
- [6] A. Kahfi, R. Tavakkoli-Moghaddam, and S. M. Seyed Hosseini, "Robust bi-objective location-arc routing problem with time windows: A case study of an iranian bank," *International Journal of Supply and Operations Management*, vol. 8, no. 1, pp. 1–17, 2021.
- [7] M. A. Ardeh, Y. Mei, M. Zhang, and X. Yao, "Knowledge transfer genetic programming with auxiliary population for solving uncertain capacitated arc routing problem," *IEEE Transactions on Evolutionary Computation*, 2022.
- [8] J. Liu, K. Tang, and X. Yao, "Robust optimization in uncertain capacitated arc routing problems: Progresses and perspectives," *IEEE Computational Intelligence Magazine*, vol. 16, no. 1, pp. 63–82, 2021.

- [9] J. Wang, K. Tang, J. A. Lozano, and X. Yao, "Estimation of the distribution algorithm with a stochastic local search for uncertain capacitated arc routing problems," *IEEE Transactions on Evolutionary Computation*, vol. 20, no. 1, pp. 96–109, 2015.
- [10] S. Wøhlk, *Contributions to arc routing*. Ph.D. thesis, Faculty of Social Sciences, University of Southern Denmark(2005).
- [11] E. B. Tirkolaee, A. A. R. Hosseinabadi, M. Soltani, A. K. Sangaiah, and J. Wang, "A hybrid genetic algorithm for multi-trip green capacitated arc routing problem in the scope of urban services," *Sustainability*, vol. 10, no. 5, p. 1366, 2018.
- [12] Y.-H. Jia, Y. Mei, and M. Zhang, "Learning heuristics with different representations for stochastic routing," *IEEE Transactions on Cybernetics*, 2022.
- [13] D. Taş, N. Dellaert, T. van Woensel, and T. De Kok, "The time-dependent vehicle routing problem with soft time windows and stochastic travel times," *Transportation Research Part C: Emerging Technologies*, vol. 48, pp. 66–83, 2014.
- [14] J. Han, C. Lee, and S. Park, "A robust scenario approach for the vehicle routing problem with uncertain travel times," *Transportation Science*, vol. 48, no. 3, pp. 373–390, 2014.
- [15] C. Lee, K. Lee, and S. Park, "Robust vehicle routing problem with deadlines and travel time/demand uncertainty," *Journal of the Operational Research Society*, vol. 63, no. 9, pp. 1294–1306, 2012.
- [16] M. Nasri, A. Metrane, I. Hafidi, and A. Jamali, "A robust approach for solving a vehicle routing problem with time windows with uncertain service and travel times," *International Journal of Industrial Engineering Computations*, vol. 11, no. 1, pp. 1–16, 2020.
- [17] J. Zhang, W. H. Lam, and B. Y. Chen, "A stochastic vehicle routing problem with travel time uncertainty: trade-off between cost and customer service," *Networks and Spatial Economics*, vol. 13, no. 4, pp. 471–496, 2013.
- [18] 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*, vol. 40, no. 3, pp. 856–866, 2013.
- [19] Y. Mei, K. Tang, and X. Yao, "Capacitated arc routing problem in uncertain environments," in *IEEE Congress on Evolutionary Computation*. IEEE, 2010, pp. 1–8.
- [20] G. Ulusoy, "The fleet size and mix problem for capacitated arc routing," *European Journal of Operational Research*, vol. 22, no. 3, pp. 329–337, 1985.
- [21] B. L. Golden, J. S. DeArmon, and E. K. Baker, "Computational experiments with algorithms for a class of routing problems," *Computers & Operations Research*, vol. 10, no. 1, pp. 47–59, 1983.
- [22] S. Tsutsui, M. Pelikan, and D. E. Goldberg, "Using edge histogram models to solve permutation problems with probabilistic model-building genetic algorithms," *IlligAL Report*, vol. 2003022, 2003.