Benchmarking Dynamic Capacitated Arc Routing Algorithms Using Real-World Traffic Simulation

Hao Tong*, Leandro L. Minku*, Stefan Menzel†, Bernhard Sendhoff† and Xin Yao*‡
*School of Computer Science, University of Birmingham, Birmingham, UK
†Honda Research Institute Europe GmbH, Offenbach, Germany
‡Department of Computer Science and Engineering, SUSTech, Shenzhen, China
Emails: hxt922@student.bham.ac.uk, L.L.Minku@bham.ac.uk,
stefan.menzel@honda-ri.de, bernhard.sendhoff@honda-ri.de, xiny@sustech.edu.cn

***Abstract*—The dynamic capacitated arc routing problem (DCARP) aims at re-scheduling the service plans of agents, such as vehicles in a city scenario, when dynamic events deteriorate the quality of the current schedule. Various algorithms have been proposed to solve DCARP instances in different dynamic scenarios. However, most existing work evaluated their algorithms' performance based on artificially constructed dynamic environments instead of using more realistic traffic simulations which are built on actual traffic data. In this paper, we constructed a novel DCARP benchmarking framework based on the Simulation of Urban MObility (SUMO) transportation simulation software, which allows to include real-world traffic environments for generating a set of DCARP instances from dynamic events, such as road congestion or task changes. The flexibility of the framework allows to develop DCARP optimization algorithms and evaluate their effectiveness more comprehensively. We use the benchmarking framework to generate 12 different dynamic instances using real-world traffic data of Dublin City. We then demonstrate the value of our framework by using these instances to compare our previously proposed hybrid local search algorithm (HyLS) with a state-of-the-art meta-heuristic optimization algorithm. The generated benchmark scenarios indicate that HyLS is a very effective optimizer on DCARP scenarios with real traffic data for reducing the total service cost. They also demonstrate the importance of our DCARP benchmarking framework for the development and benchmarking of optimization algorithms in more realistic scenarios.**

***Index Terms*—Dynamic capacitated arc routing problem, Online optimization, Real-world application, SUMO, Meta-heuristic algorithms**

I. Introduction

The dynamic capacitated arc routing problem (DCARP) aims at re-routing the service paths of vehicles for the capacitated arc routing problem when one or more dynamic events happen during vehicles' services and influence the current schedule [1], [2]. For example, a road may become congested or even not accessible anymore because of a traffic accident, which is likely to deteriorate the quality of the current schedule [3], [4]. Newly added tasks are not considered in the original service plan so that the vehicles are required to be re-scheduled for serving all available tasks [4], [5]. Previous work formulated the dynamic CARP (DCARP) to capture such dynamic scenarios and proposed a practical framework based on a virtual task strategy to handle the dynamic events [4], [6]. Uncertain CARP (UCARP) is another variation of CARP that also considers dynamic events. However, it aims at finding a *robust* solution to operate in uncertain environments by robust optimization [7], instead of rescheduling the service plan.

Even though both DCARP and UCARP have taken dynamic events into consideration when designing algorithms, a real dynamic environment for testing the algorithms is lacking. In UCARP, a benchmark was proposed based on the existing static CARP benchmark [8], [9]. A Gaussian distributed random variable was added into the static instance to simulate the uncertain factor of the environment. For DCARP, a simulation system that can simulate the vehicles' service process and randomly generate different dynamic events was designed [4]. However, the uncertain factors and dynamic events in the existing work are all artificially generated based on various distributions instead of any real scenarios. Effective algorithms evaluated only in such conditions are likely to fail in real traffic environments, especially for DCARP, in which the rescheduling highly depends on actual traffic conditions and the status of instances. Therefore, a dynamic environment containing real traffic data and conditions is required to test the proposed algorithms.

In this paper, we develop the first benchmarking framework in literature to be able to use real traffic scenarios, based on a traffic simulation software with a set of real traffic data. We focus in this paper on DCARP. Nevertheless, our proposed benchmarking framework is also suitable for evaluating UCARP algorithms. Our contributions are as follows:

- We propose the first DCARP scenario benchmarking framework containing real traffic data which can be used to evaluate the efficacy of algorithms in a real-world environment. The framework is able to integrate the solution produced by DCARP algorithms into a transportation environment that simulates real traffic conditions.
- To demonstrate the generalization ability and flexibility of the proposed benchmarking framework, we build 12 different dynamic scenarios with real traffic data in the designed simulation platform including different locations of depot, tasks and different service times of a day. These

Corresponding authors: Xin Yao (xiny@sustech.edu.cn) and Leandro L. Minku (L.L.Minku@bham.ac.uk). Xin Yao is also with [1]Research Institute of Trustworthy Autonomous Systems (RITAS), SUSTech, China. [2]Guangdong Key Laboratory of Brain-inspired Intelligent Computation, SUSTech, China. [3]School of Computer Science, University of Birmingham

dynamic scenarios can be used as a benchmark, containing the real traffic information in contrast to artificial datasets for evaluating DCARP algorithms in the future.
- To demonstrate the utility of our newly proposed framework, we evaluate and analyse the latest algorithm HyLS [6] using this framework, comparing with a state-of-the-art meta-heuristic algorithm [4], [10] in all constructed dynamic scenarios. The results show the platform can provide insightful information when comparing different algorithms. The platform allowed us to explain when and why HyLS outperforms the other algorithm.

The remainder of this paper is organized as follows. Section II discusses the related work and motivations. Section III introduces the proposed benchmarking framework with online optimization in detail. Section IV presents the constructed scenarios based on the designed platform and the results of using DCARP algorithms in all generated dynamic scenarios. Finally, section V presents the conclusions and future work.

II. Related work and Motivation

In vehicle scheduling, dynamic events such as road congestion or the occurrence of new tasks typically affect the currently deployed schedule, which can deteriorate or become infeasible. Thus, it is necessary to use dynamic optimization algorithms to improve the quality of a solution when dynamic events happen in DCARP scenarios [4]. In the literature, many algorithms have been proposed to tackle different dynamic events. For example, Monroy et al. [11] considered the failure of vehicles, and Padungwech et al. [5] focused on new tasks. In [4], a very general DCARP was considered, in which most common dynamic events such as road congestion, road closure, new tasks, etc., can be processed simultaneously. A virtual task-based optimization framework to optimize the DCARP instance benefiting from the existing powerful meta-heuristic algorithms for static CARP was also proposed [4]. Since the re-scheduling should be performed online during the vehicles' service process, the efficiency of the algorithm is much more important in the real scenario. Therefore, existing work proposed a more efficient dynamic optimization algorithm, i.e. HyLS [6], which is capable of providing a better solution within short time.

All above algorithms have been successfully evaluated in DCARP scenarios with artificial traffic environments. For example, Liu et al. [12] proposed a benchmark generator which is able to generate the basic CARP graph and DCARP instances based on a set of pre-defined parameters. In [4]–[6], the DCARP instances were generated based on the static CARP instances and adding dynamic features. However, it is unclear whether such artificial generators produce problem instances that are similar to real world ones. There is still no benchmarking framework with real traffic conditions for DCARP so that it is therefore necessary to design one for evaluating the efficacy of DCARP algorithms.

Simulation of Urban MObility (SUMO) is an open-source[1], highly portable, and continuous traffic simulation software [13]. There are three main parts in SUMO, including *Network building*, *Demand modeling* and *Simulation*. The *Networking building* and *Demand modeling* are defined by users, which provide the road networks and the simulated vehicles (vehicle types, and vehicles' routes), respectively. Then, the *Simulation* can simulate the traffic conditions according to the provided network and traffic demands (i.e., vehicles). A traffic control interface (TraCI)[2] is provided by SUMO which gives access to retrieve values from the simulated objects and manipulate their behavior online.

SUMO is able to simulate all traffic objects, such as vehicles and traffic lights, according to the provided real road network and real traffic, which constructs a traffic environment reflecting the real-world traffic [1]. Thus, it is possible to use it to build a virtual environment with real traffic conditions for DCARP. Since Traci in SUMO can directly obtain the online data and control the simulated object in an online manner, it is possible to obtain the DCARP instance from the simulation process and provide it to DCARP optimization algorithms for an improved new schedule.

As the existing benchmark for evaluating DCARP optimization algorithms' efficacy are all artificially designed, it is likely to miss partial problem characteristics of real-world applications. Therefore, this motivated us to design a benchmarking framework based on the SUMO for DCARP which can be a standard simulation platform for evaluating DCARP algorithms in real traffic conditions besides the artificial benchmarks once we have real road maps and traffic data[3]. The proposed benchmarking framework builds a bridge between DCARP optimization algorithms and SUMO such that DCARP optimization algorithms can benefit from the real traffic environment simulation in SUMO and be evaluated with real traffic data. The generated test DCARP scenarios with real traffic data contain characteristics that might not be captured by artificially generated data so that researchers can benefit from such insights and improve algorithms for the DCARP optimization.

III. Our SUMO-based Benchmarking Framework

SUMO can retrieve the real road network and the online traffic data and provide these information to (D)CARP optimization algorithms for high-quality executable solutions. The solutions obtained by optimization algorithms are required to be assigned and updated to the transportation environment in SUMO. Therefore, we proposed a SUMO-based benchmarking framework comprising SUMO, optimization algorithms, and the intermediate component linking the SUMO and (D)CARP optimization algorithms. The structure of the proposed SUMO-based benchmarking framework is presented in Figure 1.

Three different components form the framework. First, the data of the real environment for the simulation is required to

[1]https://sumo.dlr.de/docs/index.html

[2]https://sumo.dlr.de/docs/TraCI.html

[3]SUMO provides approaches for getting road networks and traffic data.

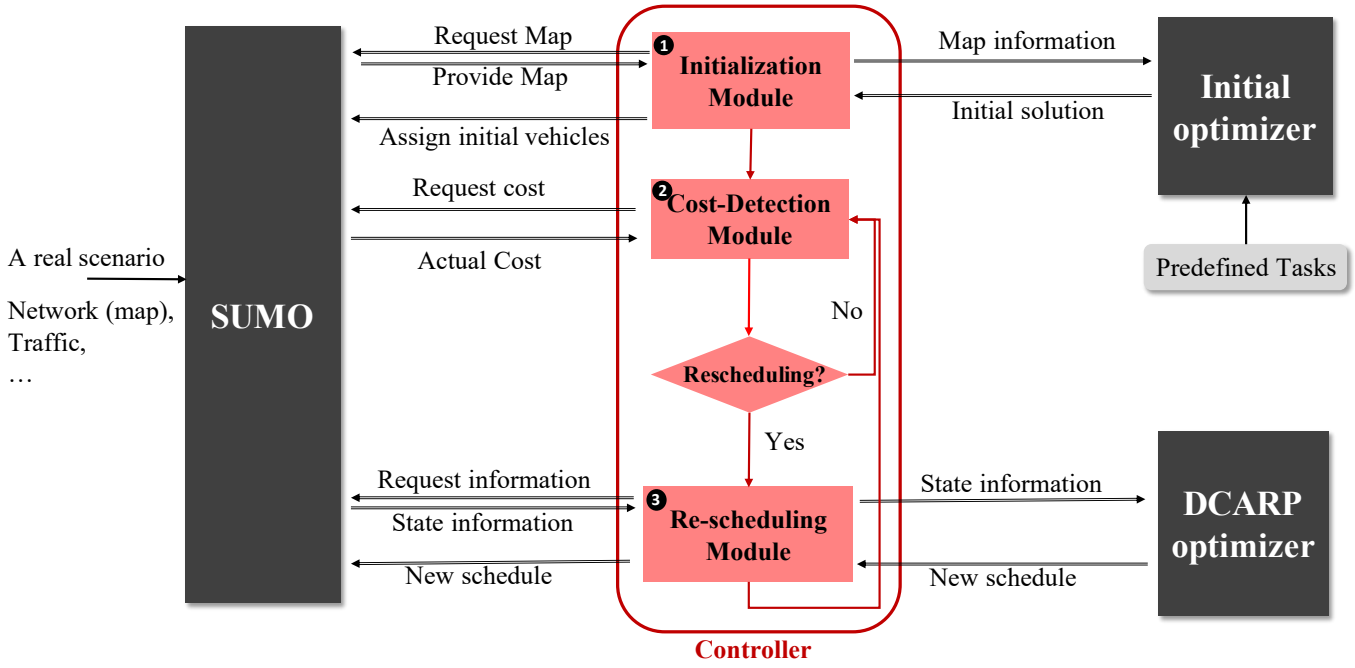


Fig. 1: The structure of SUMO-based benchmarking framework. Our main contribution is the "*Controller*" part.

be provided to SUMO, such as the road map and real traffic data. Then, the middle component, named as "*Controller*" in Figure 1, is the core component of this platform that transmits information between SUMO and the optimization algorithms. The final part of this benchmarking framework is the optimizer. The initial optimizer can be any suitable practical static CARP optimization algorithm for optimizing the initial solution for the static CARP instance. The DCARP optimizer is the algorithm whose performance is required to be evaluated by the platform. The three components are independent of each other. The communication between SUMO and the controller is based on TraCI, and between the controller and the optimizers is currently based on shared files. It is worthy to mention the communication based on shared files is not ideal especially when the optimizer takes a long time to solve DCARP, as the controller would have to wait the new schedule for a long time instead of deciding an acceptable solution on its own during the optimization. Therefore, it is valuable to employ message queue as the communication method in the future to make it more flexible.

The intermediate component *Controller* is the main component for driving the entire system. It mainly consists of three modules including *Initialization Module*, *Cost-Detection Module*, and *Re-schedule Module*:

a) Initialization Module: The initialization module is first activated before the start of the simulation. It is mainly responsible for requesting the map's road network after SUMO successfully loads the map, and then providing the network information to a predefined static CARP optimization algorithm. The algorithm feedbacks an initial solution to the initialization module. Then, the module activates the simulation process in SUMO with starting the initial vehicles and setting their service paths according to the solution.

b) Cost-Detection Module: After initial vehicles are assigned to serve tasks, and the simulation process starts running, the cost-detection module activates. A selection of dynamic events are likely to occur during the vehicles' service process, potentially affecting the cost of the current service plan. Therefore, the cost-detection module is used to collect the costs of the roads online during the simulation. Then, the cost-detection module calculates the total costs of the current service plan, which are transmitted to the next re-scheduling module to determine whether the dynamic events are too severe and deteriorate the current service plan a lot.

c) Re-scheduling Module: The re-scheduling module is activated if the difference between the current cost provided by the cost-detection module and the last recorded cost exceeds a pre-defined threshold, such as 10% increment. This indicates that some dynamic events have happened and have severely deteriorated the current service plan. The re-scheduling module requests the necessary state information from SUMO, including the current road map, the cost of each road, the positions of in-service vehicles, the remaining capacities of in-service vehicles and the remaining tasks, to construct a new DCARP instance. The module then passes the new DCARP instance to the DCARP optimization algorithm for an updated service plan. Finally, this module updates the paths of the vehicles or assign new vehicles in SUMO based on the new schedule obtained from the DCARP optimizer to continue serving the remaining tasks.

The controller continues to run the cost-detection module until the re-scheduling process is activated again. The whole

process terminates after all tasks have been served and all vehicles return to the depot.

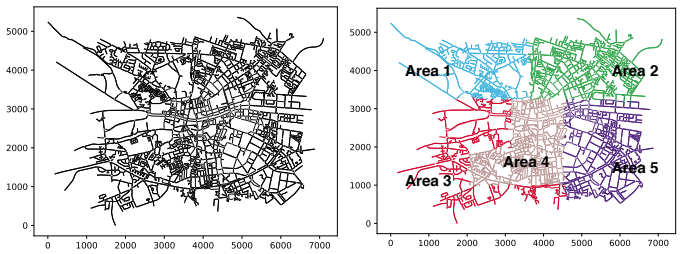
As the traffic data provided to SUMO only contains the traffic conditions, the tasks of CARP need to be defined and constructed by ourselves. For simplicity, in our benchmarking framework, the initial tasks are required to be defined in a separate file which is then provided to the initial optimizer for an initial solution. The simulation for the new tasks event is embedded in the re-scheduling module. For simplicity, in this version, we added some new tasks into the simulated DCARP scenario to simulate the new-task events in the real world, once the re-scheduling module is activated. In the future, we will update our simulator so that it could enable adding new tasks separately from the increase in traffic events. The source code of the proposed benchmarking framework is available in Github[4].

IV. Experimental Studies

In this section, in order to demonstrate the generalization ability and flexibility of our proposed benchmarking framework, we generate a series of DCARP scenarios using it based on real open-source traffic data. Then, a DCARP optimization algorithm, HyLS [6], is evaluated in all generated scenarios. The empirical results and further analysis are then presented.

A. Generate different DCARP scenarios

The actual traffic data is the source of our DCARP simulation. In our experiment, we use a set of open-source 24-hour traffic data from Dublin[5]. The traffic data comes from the Dublin SCATS dataset4, which consists of vehicle counts at every 6 minutes at 480 locations in the city center [14]. The road network of Dublin city is presented in Figure 2a, which has 6633 edges and 2895 nodes in total. We divide the whole map into five different areas including both city center and suburb areas for generating different DCARP scenarios according to the road's distribution, as shown in Figure 2b.



(a) (b)

Fig. 2: (a) The road network of Dublin. (b) Area segmentation of Dublin.

Based on the above road network, we generated 12 different DCARP scenarios with different settings for the location of the depots, the distribution of tasks and service starting time. The settings for each factor are as follows:

- Location of depots: busy area, uncrowded area.
- Distribution of tasks: whole city, busy area, uncrowded area.
- Service starting time: off-peak hours, near-peak hours.

To determine the busy degree of each area in Figure 2b and the peak/off-peak hours in provided traffic data, we define the "*crowded degree* (α)" for each road and the "*busy degree* (β)" for an area, as formulated by Eq. (1) and Eq. (2):

$$\alpha_i = 1 - \frac{\bar{v}_i}{v_i^{max}} \tag{1}$$

$$\beta = \frac{\sum_{i=1}^{N_{roads}} I(\alpha_i > 0.5)}{N_{roads}} \tag{2}$$

where $\bar{v}_i$ and v_i^{max} represent the average speed in the last 60 seconds and the limited maximum speed of one road i, respectively, N_{roads} denotes the total number of roads in an area and $I(\cdot)$ is the indicator function. Therefore, we calculated the *crowded degree* of each road in the whole network for every 60 seconds. Then, the *busy degree* of each area of Figure 2b can be estimated by Eq. (2). The dynamic *busy degree* of each area for every 60 seconds is presented in Figure 3.

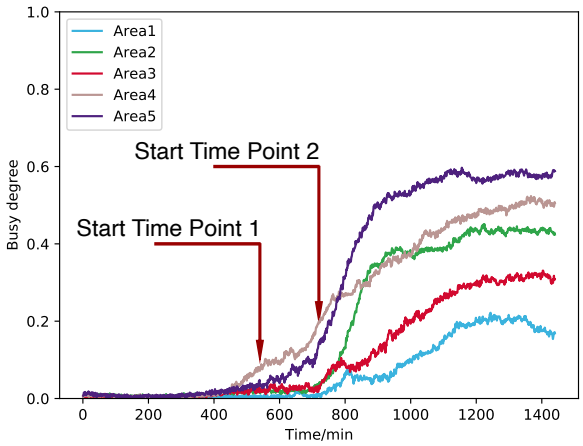


Fig. 3: The dynamic *busy degree* of each area for every 60 seconds.

According to Figure 3, *Area 1* and *Area 3* are determined as the uncrowded areas, and the remaining three areas are regarded as busy areas (Figure 4). Therefore, we can select depots and tasks in these divided areas. In Figure 4, we marked two points as locations of the depot in our DCARP scenarios, where one is in the uncrowded area, and another is in the busy area. The peak and off-peak hours can also be obtained from Figure 3, in which a small busy degree indicates the off-peak hours. Therefore, we select two time-points, i.e., 540 and 720, to be the starting time of the simulation, as pointed in Figure 3. Both starting time points belong to the off-peak hours because the service in the real-world application is usually provided

[4]https://github.com/HawkTom/DCARP-SUMO-Simulation

[5]https://github.com/maxime-gueriau/ITSC2020_CAV_impact/tree/master/Urban/Simulations/Base

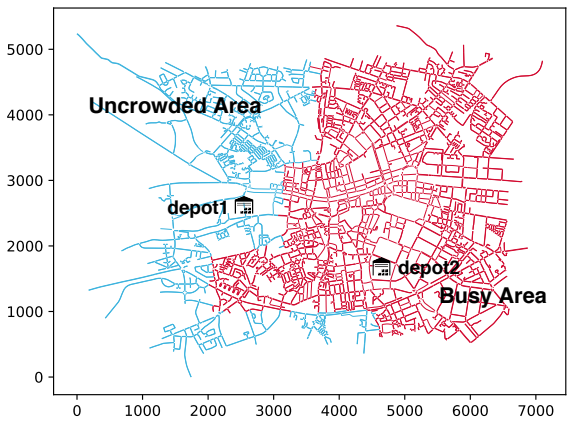


Fig. 4: The distribution of busy area and uncrowded area. The two positions of the depot.

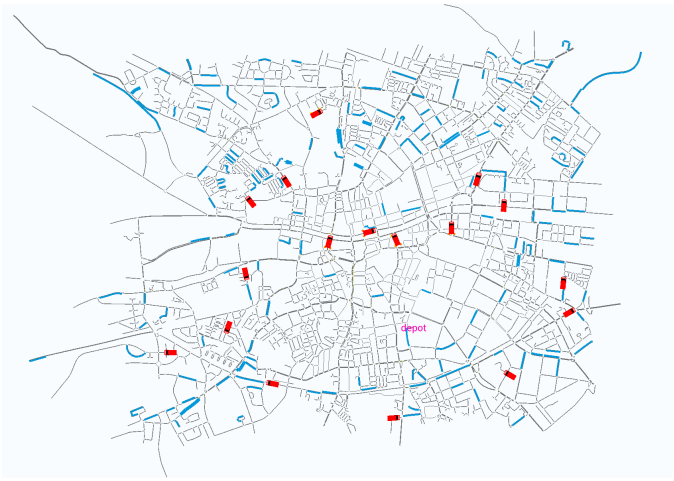


Fig. 5: One screenshot of DCARP scenario simulation in SUMO-based benchmarking framework in our experiment. It contains all cars and vehicles defined in the Dublin real traffic data, but only all unserved tasks (in blue) and in-service vehicles (in red) are highlighted.

in off-peak time. Nevertheless, the service can last until the peak hours, so we select a starting time point from which the vehicles are likely to meet peak hours, i.e., near-peak hours.

Besides, it is also important to determine the tasks and the service cost. As the whole map has been divided into busy and uncrowded areas, we randomly selected 200 tasks from one or multiple designated areas for each DCARP scenario. A screenshot of a DCARP scenario simulation is presented in Figure 5 for the scenario with a depot in the busy area, tasks distributed in the whole city and service starting from the off-peak time, containing all cars and vehicles defined in the traffic data. However, only all unserved tasks and in-service vehicles are highlighted. In the simulated traffic environment, the traffic congestion highly influences the service time. Therefore, the total time for completing all remaining tasks from the current time is defined as the cost in our DCARP scenarios, calculated by the cost-detection module every 30 seconds during the whole service process. Then, the re-scheduling module activates if the current estimated total cost is 10% more than the last recorded cost. The main dynamic event in our generated DCARP scenario is the road congestion due to the limitation of used traffic data. To simulate newly added tasks, we uniformly select 5 arcs which are not pre-defined tasks within the same area as initial tasks to be new tasks and add them into the simulation once the re-schedule module activates. All generated scenarios are available in Github[6].

B. Evaluating HyLS in DCARP scenarios

The DCARP optimization algorithms target to reduce the total cost of serving tasks in the scenario with dynamic events. We applied an efficient DCARP algorithm, i.e., HyLS [6], to the 12 different DCARP scenarios to investigate the influence of the dynamic optimization on DCARP scenarios such that insight can be obtained as to why and when HyLS works well. HyLS focuses on rescheduling the service plan within a very short time for a DCARP instance. It executes several neighborhood moves in parallel to produce synergies for rescheduling the current solution. One solution archive is maintained in HyLS to store solutions, which potentially guides the optimization to a better search area. We applied it to all generated DCARP scenarios having real traffic conditions to demonstrate that using optimization algorithms to tackle dynamic events helps a lot in saving the total cost of the whole service. The termination condition for HyLS in each round of optimization is 5 seconds, to satisfy the requirement of a quick response in the real scenario.

In all different generated DCARP scenarios, we recorded the total used time for completing all tasks and all vehicles returning to the depot, i.e., the total cost in our scenarios, of using HyLS to re-schedule the service plan and without using any re-scheduling strategy. The obtained results are presented in Table I, in which there are 12 scenarios in total with different settings of starting time, depot location, and distribution of tasks. The number of instances in each DCARP scenario is presented in column $\mathbf{N_I}$. These instances were produced as a result of the dynamic events that were severe enough to cause the rescheduling module to activate. $\mathbf{T_{DO}}$ and $\mathbf{T_{WDO}}$ denote the total time used after using dynamic optimization and without dynamic optimization, respectively. The values in Table I are all deterministic because the HyLS is a deterministic algorithm. We also calculated the time difference between using and not using dynamic optimization in column $\mathbf{T_{Diff}}$ as calculated by Eq. (3):

$$T_{Diff} = \frac{T_{WDO} - T_{DO}}{T_{WDO}} \times 100\% \tag{3}$$

The number of DCARP instances of a DCARP scenario reflects the degree of dynamic events that influence the service

[6]https://github.com/HawkTom/DCARP-SUMO-Simulation

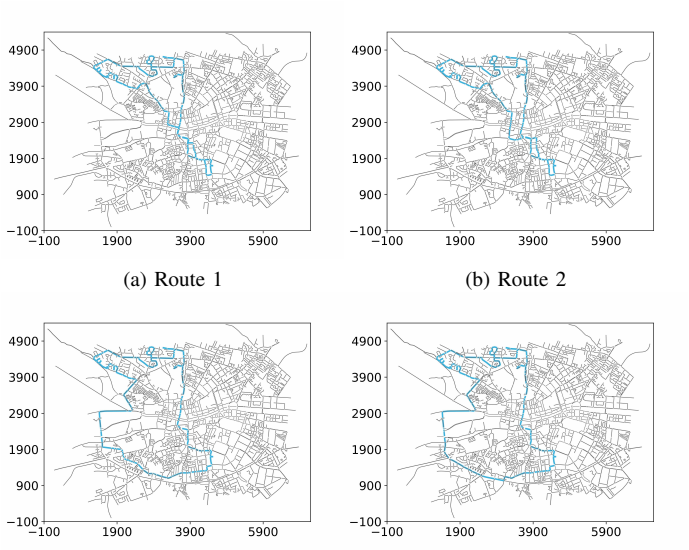
TABLE I: The cost results of using dynamic optimization (HyLS algorithm) and without dynamic optimization in each generated DCARP scenario.

Starting Time	Depot Location	Tasks' Distribution	Index	$\mathbf{N_I}$	$\mathbf{T_{DO}}(s)$	$\mathbf{T_{WDO}}(s)$	$\mathbf{T_{Diff}}(s)$
Off-peak hours	center	whole city	1	7	5522	7209	23.40%
		busy area	2	10	5841	6498	10.11%
		uncrowded area	3	9	5345	5397	0.96%
	suburb	whole city	4	2	2649	4102	35.42%
		busy area	5	10	5853	6560	10.78%
		uncrowded area	6	1	1677	1981	15.35%
Near-peak hours	center	whole city	7	10	4903	8257	40.62%
		busy area	8	10	6576	9178	28.35%
		uncrowded area	9	3	4052	4908	17.44%
	suburb	whole city	10	10	4020	6258	35.76%
		busy area	11	10	7027	8530	17.62%
		uncrowded area	12	0	1710	1710	0

$\mathbf{N_I}$: The number of instances in this DCARP scenario.
$\mathbf{T_{DO}}$: Total time used after dynamic optimization. $\mathbf{T_{WDO}}$: Total time used without dynamic optimization.
$\mathbf{T_{Diff}}$: The time difference between using and not using dynamic optimization.

plan. If one scenario has more DCARP instances, this means that the dynamic events were more severe and easier to deteriorate the currently deployed service plan. In contrast, a scenario with fewer DCARP instances indicates that fewer dynamic events significantly affected the current deployed service plan. As presented in Table I, scenarios with tasks distributed in the busy area contain the most DCARP instances compared with scenarios with other tasks' distributions. For scenarios with starting times at off-peak hours, dynamic events are more likely to affect the service plan in the scenario with the depot located in the city center (i.e., busy area). Nevertheless, for scenarios with starting times at near-peak hours, the depot location has no significant influence on the number of DCARP instances. Therefore, we can conclude that if tasks are distributed in busy areas or the service is likely to last until peak hours, the service is more likely to be affected by dynamic events.

Furthermore, the time difference between using dynamic optimization and without dynamic optimization reflects the effectiveness of dynamic optimization algorithms. A big time difference indicates that the corresponding DCRAP scenario is highly affected by the dynamic events, and such scenarios can significantly benefit from dynamic optimization algorithms. As presented in Table I, scenarios with tasks distributed over the whole city have the largest time difference compared with scenarios with tasks distributed just in one specific area. It is because the vehicles need to drive much farther to serve tasks over the whole city. On the other hand, vehicles' service are also more likely to be affected by dynamic events occurring over the whole city. The starting service time also has a significant influence on the time difference, as shown in Table I. The service starting from a time point that is possible to last until the peak hours is more likely to be affected by the severe peak-hour traffic. Finally, the influence of depot location highly depends on the starting time because the traffic difference between busy areas and uncrowded areas in off-peak hours is smaller than that of peak hours. Therefore, we can conclude that the scenarios with tasks distributed in a large area, including both busy and uncrowded areas, are much more suitable to be tackled by dynamic optimization, and the total cost can be greatly saved.



(a) Route 1 (b) Route 2

(c) Route 3 (d) Route 4

Fig. 6: An example of one vehicle's route's updates in the scenario 1 of Table I.

Besides, we also present an example of one vehicle's route's updates by dynamic optimization in a DCARP scenario in Figure 6. Route 1 is its initial route, and three updates are applied to this vehicle. After the simulation begins, this vehicle starts its service according to its initially assigned route. However, the cost-detection module detects that the current

service plan is affected by dynamic events, so that the re-schedule module activates. As a result, this vehicle is assigned with a new service path to save the total cost. The further observation reveals that its service route is mainly affected by road congestion.

C. Comparing algorithms with real DCARP instances

The effectiveness of using DCARP optimization algorithm (HyLS) has been evaluated in the DCARP scenario with real traffic conditions in the previous subsection. Now, we compare different DCARP optimization algorithms using all generated DCARP instances. The meta-heuristic algorithm, Memetic Algorithm with Extended Neighborhood Search (MAENS) [10] with the virtual-task framework has been demonstrated as the most powerful algorithm for DCARP optimization if there is no requirement for a quick response in previous work [4]. However, if the time limitation is very strict, the HyLS was shown to be more suitable than MAENS in previous work [6]. To further evaluate their performance in the real DCARP instances, we applied them to optimize all generated DCARP instances (82 instances) in our experiment. The results are presented in Table II, which presents the mean cost of MAENS over 25 independent runs and the unique obtained cost of HyLS as it is a deterministic algorithm. The better results for each DCARP instance are highlighted in Table II. We only present the obtained results on all instances generated in the first DCARP scenario of Table I due to the page limitation.

TABLE II: The obtained cost of HyLS and mean cost of MAENS over 25 independent runs on DCARP instances generated in the first scenario of Table I. The better results are highlighted for each DCARP instance.

Index	N_T	HyLS	MAENS
1	184	***21961***	22584
2	162	***21484***	22120
3	80	***16036***	16098
4	11	***7362***	***7362***
5	12	6069	***5610***
6	16	6211	***6196***
7	18	6323	***6270***

As shown in Table II, HyLS outperforms MAENS in the first three DCARP instances while MAENS obtains better results in the last three DCARP instances. Both algorithms obtain the same final cost in the 4^{th} instance. The main factor influencing the algorithm's performance is the number of tasks. We have listed the number of tasks of each DCARP instance in Table II, i.e., N_T. From the table, the instances in which HyLS outperforms MAENS have a large number of tasks. In contrast, the number of tasks of instances where MAENS outperforms or is similar to HyLS is very small.

To demonstrate the above observation, we divided all DCARP instances into three categories according to the two algorithms' relative performance and then calculated the number of tasks of each DCARP instance in each category. The results are shown in Figure 7. We performed a Kruskal-Wallis test with a 0.05 significance level to compare the number of tasks in the three categories shown in the figure. The p-value is equal to 5.20e-11, indicating the three categories are significantly different from each other. The number of tasks in the category where HyLS outperforms MAENS is significantly larger than that of the other two categories. Therefore, we can conclude that the HyLS is much better than MAENS in the real DCARP instances which have a large number of tasks and require a quick response.

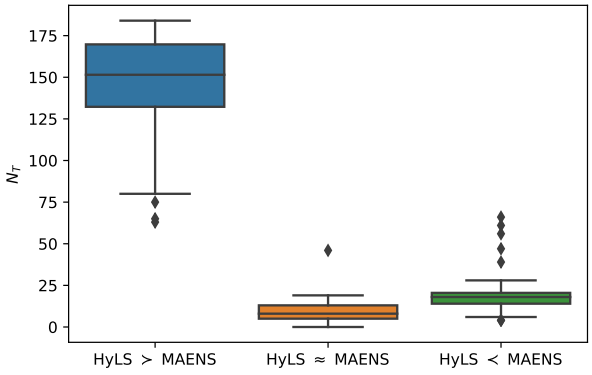


Fig. 7: The distribution of number tasks of DCARP instances in three categories: HyLS wins over ($\succ$) MAENS, HyLS draws against ($\approx$) MAENS and HyLS loses to ($\prec$) MAENS.

The reason behind the conclusion is the size of the search space and the effectiveness of search operators. The instances with a large number of tasks have a huge search space such that it is very hard to find a high-quality solution. Therefore, the effectiveness of the search operators plays an important role for the short time limitation. Even though MAENS is very powerful for the DCARP instances, its effective operators are very time-consuming and are hard to find a good solution within such a short time. The HyLS applies an exhaustive local search and executes multiple local search operators in parallel to boost its performance. Therefore, HyLS is significantly better than MAENS on DCARP instances with a large number of tasks. If an instance only has a few tasks, the local search operators in MAENS are capable of handling a small number of tasks within a short time so that MAENS can reach or even outperform HyLS.

V. Conclusion

In this paper, we designed a dynamic CARP benchmarking framework based on a traffic simulation software, i.e., SUMO, providing a more realistic environment to evaluate DCARP optimization algorithms. The core component in this platform is a *Controller* linking the SUMO and (D)CARP optimization algorithms. The actual traffic data is fed to the SUMO for transportation simulation. The *Controller* retrieves the traffic data from SUMO to construct a DCARP instance which is then transmitted to a dynamic optimization algorithm, and it is also

responsible for updating the current service plan according to the optimization result.

Based on such benchmarking framework, we constructed different DCARP scenarios with the road network and the traffic data from the real transportation environment. In this paper, we embedded a set of real 24-hour traffic data of Dublin city into our benchmarking framework and provided 12 different DCARP scenarios according to the different settings of the service starting time, depot location, and tasks' distribution. Therefore, the efficacy of DCARP optimization algorithm also can be evaluated in our benchmarking framework on different DCARP scenarios. We evaluated a previously proposed DCARP optimization algorithm, i.e., HyLS, in all constructed scenarios in our experiment. The empirical results demonstrated that HyLS is very effective on DCARP scenarios with the real traffic data for reducing the total service cost, especially on scenarios with tasks distributed in a large area, including busy and uncrowded areas. The experiments on the benchmarking framework also demonstrated the necessity of using dynamic optimization for DCARP scenarios.

Since the platform is able to simulate a complete service until all tasks are served, the benchmarking framework can also facilitate DCARP optimization algorithm comparisons and help to analyze the characteristics of different algorithms by comparing them on DCARP instances with different features. In our experiment, we compared the algorithm HyLS with the state-of-the-art meta-heuristic algorithm, i.e., MAENS [10], in our platform. The results showed that HyLS significantly outperformed MAENS in the DCARP instances with a large number of tasks due to the quick response but its performance became worse than that of MAENS for instances containing only a few tasks, which was not found in previous work.

It is worth noting that it is essential to evaluate algorithms on a diverse set of dynamic scenarios in order to avoid drawing biased or wrong conclusions from only a limited set of dynamic scenarios [15]. Our proposed benchmarking framework in this paper will help us to increase the diverse range of dynamic scenarios.

In the future, we are going to build a set of user-friendly interfaces for this benchmarking framework so that more researchers can use it to evaluate their algorithms. Furthermore, it is also valuable to construct more DCARP scenarios including small, medium and large numbers of cities for analyzing the effectiveness of each optimization algorithm in different DCARP scenarios.

ACKNOWLEDGMENT

Hao Tong gratefully acknowledges the financial support from Honda Research Institute Europe (HRI-EU). This work was also support by Research Institute of Trustworthy Autonomous Systems (RITAS), the Guangdong Provincial Key Laboratory (Grant No. 2020B121201001), the Program for Guangdong Introducing Innovative and Enterpreneurial Teams (Grant No. 2017ZT07X386), the Shenzhen Science and Technology Program (Grant No. KQTD2016112514355531).

REFERENCES

[1] M. C. Mourão and L. S. Pinto, "An updated annotated bibliography on arc routing problems," *Networks*, vol. 70, no. 3, pp. 144–194, Aug. 2017.

[2] Á. Corberán, R. Eglese, G. Hasle, I. Plana, and J. M. Sanchis, "Arc routing problems: A review of the past, present, and future," *Networks*, Jun. 2020.

[3] M. Liu, H. K. Singh, and T. Ray, "A memetic algorithm with a new split scheme for solving dynamic capacitated arc routing problems," in *2014 IEEE Congress on Evolutionary Computation (CEC)*. IEEE, 2014, pp. 595–602.

[4] H. Tong, L. L. Minku, S. Menzel, B. Sendhoff, and X. Yao, "A novel generalised meta-heuristic framework for dynamic capacitated arc routing problems," *IEEE Transactions on Evolutionary Computation*, pp. 1–15, 2022, doi:10.1109/tevc.2022.3147509.

[5] W. Padungwech, J. Thompson, and R. Lewis, "Effects of update frequencies in a dynamic capacitated arc routing problem," *Networks*, vol. 76, no. 4, pp. 522–538, 2020.

[6] H. Tong, L. L. Minku, S. Menzel, B. Sendhoff, and X. Yao, "A hybrid local search framework for the dynamic capacitated arc routing problem," in *Proceedings of the Genetic and Evolutionary Computation Conference Companion*, 2021, pp. 139–140.

[7] 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.

[8] 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.

[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] K. Tang, Y. Mei, and X. Yao, "Memetic algorithm with extended neighborhood search for capacitated arc routing problems," *IEEE Transactions on Evolutionary Computation*, vol. 13, no. 5, pp. 1151–1166, 2009.

[11] M. Monroy-Licht, C. A. Amaya, A. Langevin, and L.-M. Rousseau, "The rescheduling arc routing problem," *International Transactions in Operational Research*, vol. 24, no. 6, pp. 1325–1346, 2017.

[12] M. Liu, H. K. Singh, and T. Ray, "A benchmark generator for dynamic capacitated arc routing problems," in *2014 IEEE Congress on Evolutionary Computation (CEC)*. IEEE, 2014, pp. 579–586.

[13] P. A. Lopez, M. Behrisch, L. Bieker-Walz, J. Erdmann, Y.-P. Flötteröd, R. Hilbrich, L. Lücken, J. Rummel, P. Wagner, and E. Wießner, "Microscopic traffic simulation using sumo," in *2018 21st International Conference on Intelligent Transportation Systems (ITSC)*. IEEE, 2018, pp. 2575–2582.

[14] M. Guériau and I. Dusparic, "Quantifying the impact of connected and autonomous vehicles on traffic efficiency and safety in mixed traffic," in *2020 IEEE 23rd International Conference on Intelligent Transportation Systems (ITSC)*. IEEE, 2020, pp. 1–8.

[15] D. Herring, M. Kirley, and X. Yao, "Reproducibility and baseline reporting for dynamic multi-objective benchmark problems," *Proceedings of the Genetic and Evolutionary Computation Conference (GECCO)*, 2022, doi: 10.1145/3512290.3528791.