An Experimental Study of Large-scale Capacitated Vehicle Routing Problems

Er Zhuo[#], Yunjie Deng[#], Zhewei Su, Peng Yang, Bo Yuan and Xin Yao[*]

Shenzhen Key Laboratory of Computational Intelligence, University Key Laboratory of Evolving Intelligent Systems of Guangdong Province,

Department of Computer Science and Engineering, Southern University of Science and Technology, Shenzhen 518055, China.

Emails: {11611026, 11611006, 11611002}@mail.sustc.edu.cn; {yangp, yuanb, xiny}@sustc.edu.cn

***Abstract*—The recently proposed Scalable Approach Based on Hierarchical Decomposition (SAHiD) has shown its superiority on large-scale capacitated arc routing problems (CARP) in terms of both computational efficiency and solution quality. The main idea of SAHiD is that the underlying Hierarchical decomposition (HD) scheme is able to efficiently obtain a good permutation of tasks for CARP in a hierarchical divide-and-conquer way, where both the number and size of subproblems can be kept in tractable for large-scale problems with thousands of tasks. Motivated by the frequent observations of the similarity between CARP and Capacitated Vehicle Routing Problem (CVRP), the HD scheme and SAHiD algorithm are expected to work well on CVRPs. This paper applies SAHiD to large-scale CVRPs and discovers that SAHiD does not work as well as expected on large-scale CVRP. Possible reasons for this are given after extensive experimental studies. Two directions for improving SAHiD on large-scale CVRP are pointed out.**

Keywords—Capacitated vehicle routing problem (CVRP), hierarchical decomposition, comparative study.

I. INTRODUCTION

The efficient distribution of goods lies at the heart of business activities. With the rapid growing of the e-commerce logistics, routing algorithms for large-scale real-time scenarios are highly desired. For example, the number of express delivery requests in China is on average over 100 million per day and 70 thousand per minute [1], which necessitates efficient routing algorithms to compute economical routing plans in a large-scale transportation network within a few seconds.

The vehicle routing problem (VRP) [2] plays a crucial role in minimizing the transportation cost in distribution management. VRP is a set of problems that seeks an optimal set of routes to service a set of geographically-dispersed customers under different constraints. The capacitated vehicle routing problem (CVRP) constitutes the standard version of VRP respecting vehicle capacity constraints. CVRP is a challenging combinatorial optimization problem which has been proved to be NP-hard [3]. Numerous approaches [4]-[7] have been proposed to tackle CVRP problems [8]-[13]. Unfortunately, the majority of these works are either for small or median-scale problems (up to a few hundred customers), or not able to solve problems in real-time. Directly applying the same methods to solve problems of thousands or tens of thousands customers is extremely time-consuming and simply ineffective. For example, the iterated local search based metaheuristic algorithm (ILS-SP) [4] and the unified hybrid genetic search (UHGS) [5] are two state-of-the-art metaheuristics, while it takes more than 500 minutes on average for both of them to solve the largest CVRP instance with 1000 customers in a recent benchmark set [14]. Under these circumstances, the existing approaches are not able to satisfy the surging demands of e-commerce logistics.

The capacitated arc routing problem (CARP) [15] is a combinatorial optimization problem very similar to CVRP, except that the service constraints of CARP are on the edges/arcs rather than on the nodes. In recent years, several real-time algorithms focusing on large-scale instances have emerged in the study of CARP [16]-[19]. Among these algorithms, a scalable approach based on hierarchical decomposition (SAHiD) [16] is one of the most representative. Specifically, for large-scale CARP instances with over 3000 tasks, some existing state-of-the-art algorithms need to consume more than 30 minutes to reach the solution quality achieved by SAHiD within 30 seconds. The advantages of SAHiD are mainly based on the idea of hierarchical decomposition (HD), i.e., decomposing a large-scale problem hierarchically into subproblems and tackling the subproblems separately. First, with the divide-and-conquer manner, each subproblem becomes small-scaled and is easier to be solved, thereby making the whole problem easier to be solved. Meanwhile, based on the hierarchical decomposition, the total number of layers in the hierarchy increases logarithmically with

This work was supported by National Key R&D Program of China (Grant No. 2017YFC0804003), the Natural Science Foundation of China (61806090 and 61672478), Shenzhen Peacock Plan (Grant No. KQTD2016112514355531), the Program for University Key Laboratory of Guangdong Province (Grant No. 2017KSYS008), and the Science and Technology Innovation Committee Foundation of Shenzhen (Grant Nos. JCYJ20170307105521943, JCYJ20170817112421757, JCYJ20180504165652917).

[#] Er Zhuo and Yunjie Deng are co-first authors.

[*] Corresponding author.

the scale of CARP, which guarantees a better scalability than other decomposition-based CARP algorithms.

Unfortunately, SAHiD cannot be directly applied to most common e-commerce logistics scenarios, because the tasks required by e-commerce logistics are mostly in the form of nodes (customers) rather than arcs (roads). On the other hand, some researchers have studied the transformations between CVRP and CARP [15], [20]-[22]. Thus, considering the similarity between CVRP and CARP, the HD scheme lying in the core of SAHiD is expected to benefit the efficient solutions for CVRP, since the essence of both CVRP and CARP is seeking the optimal permutation and partition of a group of tasks, while the HD scheme basically serves as an effective search operator for permutation-based problems.

As a first attempt, this paper has conducted empirical studies to analyze whether it is applicable to directly employ SAHiD to tackle CVRP. Two possible approaches are considered in this paper, including applying SAHiD after transforming CVRP into CARP, as well as modifying SAHiD to make it compatible with the problem representation of CVRP. As it will be shown by the experimental results, the current implementation of HD (i.e., SAHiD), fails to tackle CVRP effectively. Related analysis reveals that the main cause of this failure is that the specific design of the merge-split operator and the HD scheme in current SAHiD do not fit into the characteristics of CVRP. Furthermore, based on the above analysis, we provide two possible directions for improving the performance of SAHiD on CVRP.

The rest of this paper is organized as follows. Section II states the problems, reviews the related work and summaries the key steps of SAHiD. Empirical studies are presented in Section III to analyze the performance of SAHiD on CVRP. Section IV concludes this paper and suggests directions for future work.

II. BACKGROUND

In this section, the preliminary background of this paper is presented. We start from the problem definition of CVRP and CARP, and then summarize the key subroutines of SAHiD. For more details of SAHiD, we refer the readers to [16].

A. CVRP and CARP

The CVRP is defined on a connected undirected graph $G = (V, E)$. The set $V = \{0, 1, \dots, n\}$ is a vertex set. Vertex 0 corresponds to the depot, and each vertex $i \in V\backslash\{0\}$ represents a customer having demand $q_i \geq 0$. Each edge $e \in E = \{(i,j): i,j \in V, i < j\}$ is associated with a travel cost $c_{ij} \geq 0$, which represents the distance from i to j. A fleet of m identical vehicles is available at the depot, each of them is associated with a capacity Q. The objective of CVRP is to determine a set of vehicle routes with minimal total costs, subject to the following constraints.

1) Each route starts and ends at the depot.
2) Each customer is visited exactly once by one vehicle.
3) The total demand of the customers served by any vehicle does not exceed the vehicle capacity Q.

The CVRP has been the subject of intensive research for 60 years. Since CVRP belongs to the class of NP-hard problems, exact algorithms are only capable of solving relatively small CVRPs to optimality within reasonable computing time. For more information on exact solution methods, we refer to the survey of Toth and Vigo [23]. For larger-scale CVRPs, designing efficient heuristics is a more popular approach. Heuristics for CVRP generally fall into three categories: constructive heuristics, improvement heuristics and metaheuristics.

Constructive heuristics generates the initial solution for further improvement. Typically, the construction is carried out by either merging routes based on a *savings* criterion, e.g., the savings heuristic of Clarke and Wright [24], or by sequentially inserting vertices into the current partial solution, e.g., the insertion heuristics of Mole and Jameson [25], Christofides *et al.* [11]. Improvement heuristics for CVRP can be classified as intra-route improvements and inter-route improvements. For intra-route improvements, any improvement heuristic designed for TSP can be applied, such as 2-opt, 3-opt and Or-opt. Inter-route improvements explore the neighborhood more extensively by doing multi-route edge exchanges. Classical operators seeking for inter-route improvements include relocate, swap and 2-opt*. Metaheuristics for CVRP can be roughly divided into two groups: local search methods and population-based methods. Local search methods proceed by iteratively exploring the solution space and evaluating the neighborhoods [26]-[29]. Population-based methods are often inspired by natural law, aiming at effectively combining and generating solutions in the pool of solutions [30]-[32].

CARP plays the same role in arc routing as CVRP in node routing. Instead of serving a set of customers, CARP seeks a minimum cost set of routes to cover a set of required edges and/or arcs, where each edge can be considered as a road in the real world [33]-[34]. It can be intuitively observed that CVRP and CARP are quite similar to each other. Furthermore, it has been shown that CVRP can be transformed into CARP [15], and CARP can be transformed into CVRP [20]-[22]. All of the three transformations of CARP into CVRP requires additional constraints of the resulting CVRP instance, either setting edge costs to infinity or fixing the visiting sequence of vertices. Besides, these three transformations also entail an increase of the problem size in the resulting CVRP. The idea of [15] is to split each customer vertex in CVRP into two vertices joined by a zero-length arc with a demand equal to the original vertex demand. In this way, CVRP is transformed into a particular case of CARP with all the required arcs having length equal to zero.

The similarity within the definitions of CVRP and CARP along with their transformations into each other serve as a motivation for us to directly employ SAHiD to tackle CVRP. The solving strategy of SAHiD is briefly summarized below.

B. Brief Descriptions of SAHiD

SAHiD [16] can be characterized as an iterated local search embedded with the hierarchical decomposition (HD) scheme.

The importance of the HD scheme is twofold. First, based on the idea of divide-and-conquer, the HD scheme divides the tasks into subgroups and solves the induced subproblems recursively. Solving each subproblem requires finding the optimal permutation of tasks in each subgroup, which is a partial solution

Algorithm 1 HD(VT)

Input: virtual task set VT
Output: a permutation of tasks PT

```
repeat
    randomly choose the cluster number K ∈ [1, β · |VT|];
    divide VT into groups by using k-means;
    order the virtual tasks within each group;
    VT ← {permutation of tasks in each group};
until |VT| = 1;
return the permutation of tasks in VT;
```

Algorithm 2 HDU(VT)

Input: virtual task set VT
Output: a feasible solution s

```
apply HD(VT) to generate a permutation of tasks PT;
apply Ulusoy's splitting procedure to partition PT into a solution s;
return s;
```

Algorithm 3 LS(s)

Input: solution s
Output: potentially improved solution s

```
apply the reverse operator to improve s;
if s is not updated then
    apply the MS operator to improve s;
    if s is updated then
        apply the reverse operator to improve s;
    end if
end if
return s;
```

Algorithm 4 SAHiD(T)

Input: task set T
Output: a feasible solution s^*

```
generate an initial solution s using HDU(T);
apply LS(s) to improve s;
s* ← s;
while stopping criteria are not met do
    generate a virtual task set VT by splitting the routes of s;
    generate a solution s' using HDU(VT);
    apply LS(s') to improve s';
    if s' is acceptable then
        s ← s';
        if s' is better than s* then
            s* ← s';
        end if
    end if
end while
return s*;
```

to the problem of finding the optimal solution of all tasks. Since the sizes of subproblems at each layer can be kept tractable even for large-scale instances, they can be efficiently solved and then combined together to form a complete solution. Second, by virtue of the hierarchical structure, the number of tasks (virtual tasks) at each layer decreases exponentially from the bottom to the top layer of the hierarchy. Therefore, the cost of SAHiD for grouping tasks increases more slowly with the scale of CARP than the grouping costs of methods based on linear decomposition [17]-[19].

Specifically, the HD scheme starts from the bottom layer of the hierarchy, i.e., the real tasks included in a CARP instance, and employs the k-means algorithm [35] to group these real tasks. The tasks are then ordered within each group based on a best insertion heuristic (BIH) and each ordered group is treated as a virtual task at layer 2. This procedure of recursive grouping and then ordering tasks (virtual tasks) continues until only 1 virtual task remains. The general framework of HD is illustrated by Algorithm 1.

The objective of the HD scheme is to efficiently generate a good permutation of all tasks in CARP. Therefore, the Ulusoy's splitting procedure [36] is employed to split the permutation into a set of feasible routes optimally. The combination of the HD scheme and the Ulusoy's splitting procedure (HDU) is demonstrated by Algorithm 2.

The local search procedure is applied to further improve the solution obtained using HDU. First, a reverse move operator similar to the 2-opt operator is applied with the first improvement strategy. If the reverse operator fails to improve the solution, the merge-split (MS) operator [37] is employed to search the neighborhood with a larger step-size. If the MS operator finds an better solution, the reverse operator is again applied to the improved solution. Otherwise, the local search procedure terminates without changing the solution obtained by HDU. The local search procedure is depicted by Algorithm 3.

The overall framework of SAHiD is described by Algorithm 4. Note that the reconstruction phase is composed of splitting the routes and then applying HDU to obtain a new solution. The framework of SAHiD is essentially a classical iterated local search embedded with a novel HD scheme.

TABLE I. NUMBERS OF VERTICES AND VEHICLE CAPACITIES OF INSTANCES EMPLOYED FOR EXPERIMENTS

Name	|V|	Q
X-n641-k35	641	1381
X-n716-k35	716	1007
X-n801-k40	801	20
X-n895-k37	895	1816
X-n1001-k43	1001	131

III. EXPERIMENTAL STUDIES

To analyze the effectiveness of directly employing SAHiD [16] to tackle CVRP, extensive empirical studies have been carried out. In the first group of comparisons, the performances of SAHiD on CVRPs are compared against the Clarke-Wright savings method [24] and the best known solutions. As will be shown by the results, the solution quality obtained by SAHiD is

TABLE II. COMPARED ALGORITHMS SETTINGS

Name	Problem transformation	Initial solution		Local search	Reconstruction	
		HDU	Savings	With MS	HDU	Random
The savings heuristic			√			
SAHiD	√	√		√	√	
SAHiD without MS (in the 2nd group of comparisons)	√	√			√	
SAHiD without MS (in the 3rd group of comparisons)		√			√	
Savings + SAHiD without MS			√		√	
Savings + SArandom without MS			√			√

TABLE III. PERFORMANCES OF SAHID ON CVRP UNDER DIFFERENT SETTINGS THROUGH PROBLEM TRANSFORMATION. BEST COSTS, AVERAGE COSTS, RELATIVE PERCENTAGE OF DEVIATIONS AND STANDARD DEVIATIONS ARE GIVEN. THE MINIMAL VALUES BETWEEN ALL AVERAGE COSTS AND SAVINGS RESULTS ARE MARKED WITH *

Name	SAHiD			SAHiD (without MS)			RPD (%)	Savings	BKS
	Best	Avg	Std	Best	Avg	Std			
X-n641-k35	66532	67109*	292.9	66909	67441	321.7	0.49	68232	63737
X-n716-k35	45928	46250	228.4	45908	46289	253.0	0.08	45859*	43414
X-n801-k40	76497	76855*	297.2	76592	77115	304.3	0.34	77235	73331
X-n895-k37	57589	58105*	379.2	57635	58373	451.9	0.46	59100	53946
X-n1001-k43	77569	78121	436.5	77638	78208	348.1	0.11	77735*	72402

TABLE IV. PERFORMANCES OF SAHID DIRECTLY ON CVRP UNDER DIFFERENT SETTINGS. BEST COSTS, AVERAGE COSTS AND STANDARD DEVIATIONS ARE GIVEN. THE MINIMAL VALUES BETWEEN ALL AVERAGE COSTS AND SAVINGS RESULTS ARE MARKED WITH *

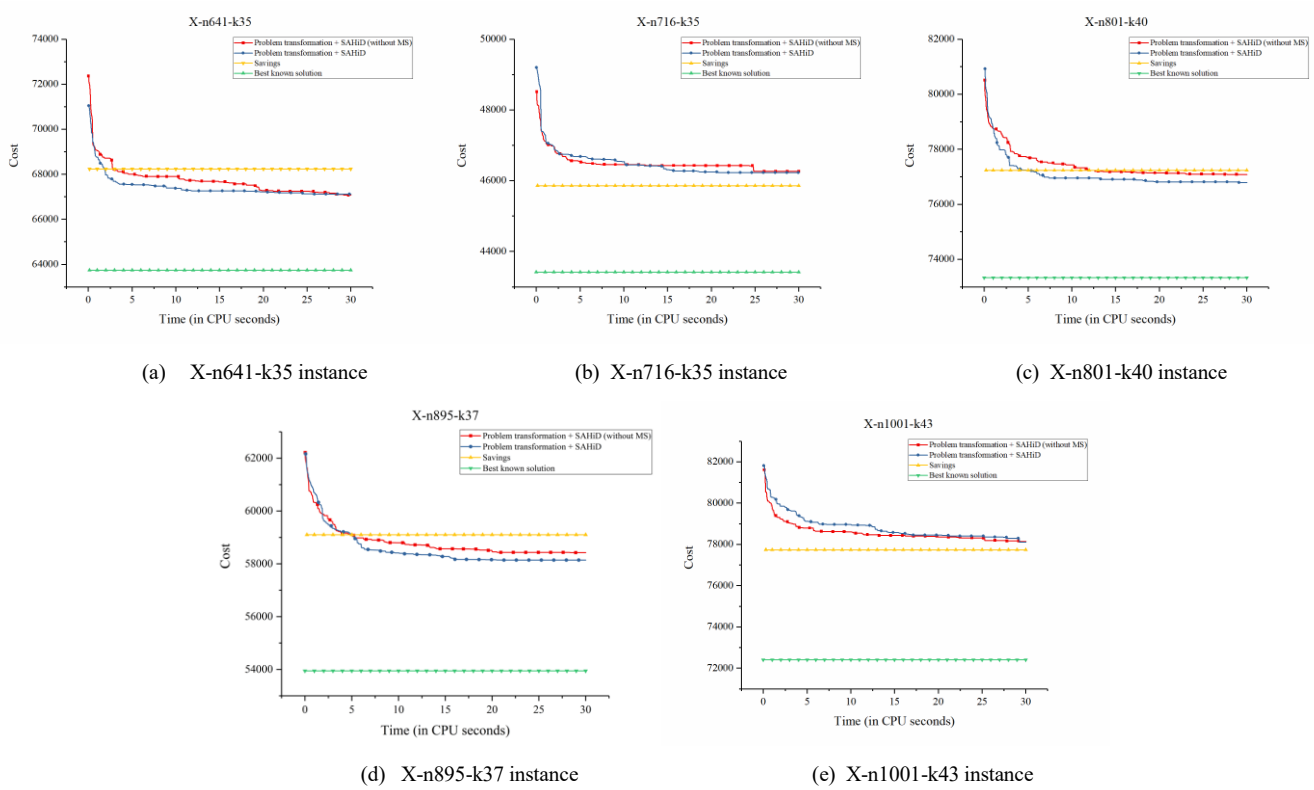
Name	SAHiD (without MS)			Savings + SAHiD (without MS)			Savings + SArandom (without MS)			Savings	BKS
	Best	Avg	Std	Best	Avg	Std	Best	Avg	Std		
X-n641-k35	67317	67746	348.1	66279	66974*	155.6	67643	67762	60.6	68232	63737
X-n716-k35	46336	46575	196.5	45534	45591*	23.6	45591	45615	11.8	45859	43414
X-n801-k40	76602	77575	477.3	76177	76315*	75.4	76682	76737	25.3	77235	73331
X-n895-k37	58444	58836	288.1	57733	57990*	128.7	58596	58630	15.2	59100	53946
X-n1001-k43	77922	78601	454.1	76843	76958*	69.8	72239	77274	14.4	77735	72402

generally undesirable. Thus, the aim of the next two groups of comparisons is to analyze the causes of the inferior performance of SAHiD on CVRP. The results obtained demonstrate that the inferior performance of SAHiD on CVRP should be attributed to the specific implementations of the merge-split (MS) operator [37] and the HD scheme in current SAHiD.

A. Benchmark Set

Since SAHiD mainly focuses on large-scale problems, it is applied to tackle large-scale CVRPs rather than smaller ones. Five large-scale CVRP instances in [14] are selected for experiments, i.e., X-n641-k35, X-n716-k35, X-n801-k40, X-n895-k37 and X-n1001-k43. Table I presents the number of vertices $|V|$ and the vehicle capacity Q of each of them.

B. Experimental Protocol



(a) X-n641-k35 instance (b) X-n716-k35 instance (c) X-n801-k40 instance

(d) X-n895-k37 instance (e) X-n1001-k43 instance

Fig. 1. Convergence curves of SAHiD and SAHiD without MS on instances X-n641-k35, X-n716-k35, X-n801-k40, X-n895-k37 and X-n1001-k43.

Three sets of empirical studies are presented. Note there are two ways available to apply SAHiD to tackle CVRP, either by transforming CVRP into CARP and applying SAHiD to the resulting CARP thereafter, or by modifying SAHiD to accommodate the problem representation of CVRP. Both approaches are considered in the following groups of comparisons. In the first and second groups, the former approach is taken based on the transformation described in [15]. Specifically, we split each customer vertex in the original CVRP instance into two vertices joined by a zero-length arc with a demand equal to the original customer demand. Thereby, the original CVRP instance is transformed into a particular CARP instance with all the required arcs having length equal to zero. The third group adapts SAHiD to CVRP before employing SAHiD to tackle CVRP.

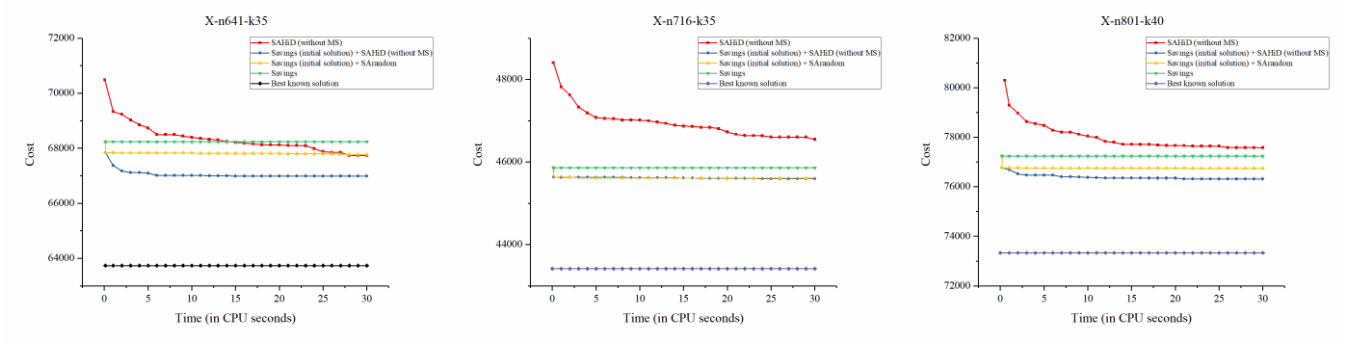
The following three groups share the same criterion for performance comparison, i.e., the solution quality within a predefined time budget. In the first group, SAHiD is compared with the best known solution and the savings method. The savings method [24] initializes back-and-forth routes $(0, i, 0)$ for $i = 1, \dots, n$ and gradually merges the two routes associated with the largest (remaining) savings. The savings are computed by $s_{ij} = c_{i0} + c_{0j} - c_{ij}$ when two routes $(0, \dots, i, 0)$ and $(0, j, \dots, 0)$ can be feasibly merged into a single route $(0, \dots, i, j, \dots 0)$, for all $i, j \geq 1$ and $i \neq j$. In the second group, SAHiD without the MS operator is compared with the original SAHiD to determine the effect of MS on CVRP. The third group is set to evaluate the performance of HDU, i.e., the core component of the current SAHiD, on CVRP. Since MS does not of comparisons, it is removed from SAHiD in the third group. The third group evaluates the performances of three algorithms on CVRP. The first algorithm is the CVRP version of SAHiD without MS. Based on the first algorithm, the second one replaces HDU with the savings method in the initialization phase. On the basis of the second algorithm, the third algorithm in this group, namely SArandom, further replaces HDU in the reconstruction phase by randomly merging virtual tasks into permutations. This random permutation scheme is set as a baseline to evaluate the improvement that can be achieved by the HD scheme in the reconstruction phase. The components of all six algorithms covered above are summarized in Table II.

All algorithms included are implemented in C++ and run on the same workstation, i.e., Intel Xeon E7-4809 processor with 2.10 GHz. For all experiments presented, the results are obtained by running the algorithms 30 times independently. The common time budget is set as 30 s for each run on each instance.

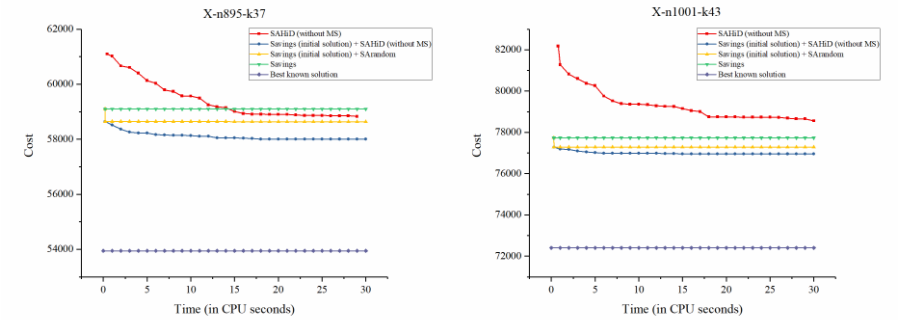
SAHiD has five user-defined parameters. To avoid misleading, each of them is named the same as [16]. Parameters α, β and p take the same values as in [16]. Parameter θ and parameter σ are modified to 101% and 2000, respectively. To keep the fairness of comparisons, the values of parameters remain the same in all algorithms presented in this paper.

C. Experimental Analysis

Fig. 1 and Fig. 2 depict the convergence curves of all six algorithms on the five test instances. Specifically, Fig. 1 compares algorithms included in the first and the second groups, and Fig. 2 compares algorithms included in the third group. Each curve corresponds to a specific run of one algorithm on one



(b) X-n641-k35 instance (b) X-n716-k35 instance (c) X-n801-k40 instance



(d) X-n895-k37 instance (e) X-n1001-k43 instance

Fig. 2. Convergence curves of SAHiD without MS, Savings + SAHiD without MS and Savings + SArandom on instances X-n641-k35, X-n716-k35, X-n801-k40, X-n895-k37 and X-n1001-k43.

instance, i.e., the run generating the median result among the 30 runs. In addition, the best-known solution of each instance obtained from [14] is also plotted. Note that the saving's method employed is the exact version without randomness, represented by a straight line on each run. Tables III and IV present the total costs of the final solutions achieved by SAHiD under different settings. The columns named "best" and "avg" provide the best and average costs for each compared algorithm among 30 runs. The column headed "std" represents the standard deviations of the costs obtained over 30 runs. The last column named "BKS" provides the cost of the best-known solution for each problem. Table III has one additional column headed "RPD", which gives the relative percentages of deviation. The value in this column is calculated by the difference between the average costs achieved by SAHiD without MS and the original SAHiD over the average cost achieved by the original SAHiD. It measures the influence of MS on SAHiD when it is applied to solve CVRP.

As illustrated in Fig. 1, all the initial costs obtained by SAHiD using HDU are larger than those obtained by the savings method. Besides, there is an obvious gap between the final costs of SAHiD and the best-known solutions. The final results of SAHiD on X-n716-k35 and X-n1001-k43 are even worse than those of the savings method. From Table III, we can see that the average costs obtained by SAHiD on X-n716-k35 and X-n1001-k43 are also higher than those of the savings method. Although SAHiD may perform better than the savings method on some instances, the advantages are quite limited. Recall that the savings method only generates the initial solution, while SAHiD is given a time budget of 30 s.

To analyze the inferior performances of SAHiD on large-scale CVRPs within a short time period of 30 s, the following two groups of comparisons are designed.

Analysis of the MS operator:

To evaluate the influence of MS on SAHiD for CVRP, both Fig. 1 and Table III present that for CVRPs, SAHiD without MS is only slightly worse than the original SAHiD. According to Table III, RPD on all instances are below 1%, which means the contribution of MS to tackling CVRP is statistically insignificant. Moreover, the average costs of SAHiD with or without MS on X-n716-k35 and X-n1001-k43 are almost identical. Based on these results, we can conclude that the benefit of MS to CVRP is negligible.

After analyzing the strategy of the MS operator, a possible explanation of its failure on CVRPs can be provided. The MS operator searches the neighborhood with a large step-size by randomly merging routes and then applying the path-scanning heuristic [37] with 5 different rules. These rules are used to further polish the permutation obtained by the ordering technique in HDU, i.e., BIH. However, since the resulting CARP from the transformation is made up of zero-length arcs, 2 rules related to arc cost cannot be directly applied, which may largely affect the efficacy of MS in generating better permutations. Due to this difference in problem representation, MS performs much worse on CVRP than on CARP.

Analysis of HDU:

In the third group of comparisons, SAHiD is implemented with respect to the representations of CVRP such that it can be directly applied to tackle CVRPs. Due to the limited contribution of the MS operator to solving CVRP, MS is not employed in any algorithm included in this group.

According to Fig. 2 and Table IV, the SAHiD employing the savings method for initial solutions outperforms the original SAHiD which generates initial solutions based on HDU. Table IV also demonstrates that the combination of the savings method and SAHiD performs the best among SAHiDs under different settings in this group from the perspective of average costs. This demonstrates the superiority of the savings method over the current HDU on constructing initial solutions for CVRPs. Furthermore, as illustrated by Fig. 2, solutions obtained by this combination always converge to a local optimum quickly and get trapped into it. As a result, solutions after a 30 s run are only slightly better than the initial solutions generated by the savings method. This implies the ineffectiveness of the reconstruction phase, in which the HD scheme again should have played a crucial role. The results of another pair of algorithms in this group, i.e., SAHiD with the savings and SArandom with the savings, are also presented for comparisons. As it can be observed from Fig. 2, SArandom gets stuck in local optimums almost immediately after the initial solution, which means merging virtual tasks randomly almost renders the reconstruction phase useless. However, according to Fig. 2 and Table IV, replacing the HD scheme with randomly merging does not make much difference in terms of final solution quality. Moreover, the average costs of these two settings on X-n716-k35 are even almost identical. These results together strongly demonstrate the loss of efficacy of the current implementation of HDU on solving CVRPs.

After analyzing the procedure of the current HDU, we are able to give a possible explanation of its failure on CVRPs. HDU employs a greedy search heuristic named best insertion heuristic (BIH) to order the virtual tasks in a group. The idea is to successively append the nearest virtual task to the end of the current permutation, regardless of capacity constraints. When HDU is applied to CVRP, since capacity constraints are not considered, each subproblem is in essence a travelling salesman problem (TSP), and BIH can be characterized as the nearest neighbor heuristic (NN) [38] for TSP. However, the performance of NN has been shown inferior to other constructive heuristics [24], [39] on Euclidian instances, according to the results reported in [40]. Hence, HDU under current settings is not applicable to effectively solve CVRP.

IV. Conclusion

This paper presents a comparative study on directly applying the current implementation of the HD scheme (i.e., SAHiD) to generate real-time solutions to large-scale CVRPs. First, SAHiD is directly applied to tackle CVRPs based on a problem transformation. According to the experimental results, the performance of SAHiD is quite limited. Therefore, another two groups of comparisons are conducted to further analyze the underlying causes of this failure. As shown by the experimental results, the specific implementations of the current MS operator and the HD scheme jointly result in the previous failure. Specific analysis along with possible explanations are provided accordingly.

By analyzing why the current MS operator and HD scheme leads to the failure of SAHiD on CVRP, several directions of improving SAHiD on CVRP emerges naturally. First, since the MS operator is not compatible with the problem characteristics of CVRP, it can be replaced by other search operators that suit CVRP better. Second, considering that the ordering method employed by the current HD scheme (i.e., BIH) has been demonstrated to be ineffective for CVRP, replacing it with alternative constructive techniques of better efficacy (e.g., the Clarke and Wright savings method) is a promising approach to improve the fitness of SAHiD on CVRP.

References

[1] State Post Bureau of The People's Republic of China (2018, Jan 13). *China's Post Industry Annual Statistics of 2017 (1st ed.)* [Online]. Available: http://www.spb.gov.cn

[2] G. B. Dantzig and J. H. Ramser "The Truck Dispatching Problem", *Management Science*, vol. 6, no. 1, pp. 80–91, 1959.

[3] J. K. Lenstra and A. H. G. Rinnooy Kan, "Complexity of vehicle routing and scheduling problems," *Networks*, vol. 11, no. 2, pp. 221–227, 1981.

[4] A. Subramanian, E. Uchoa, and L. S. Ochi, "A hybrid algorithm for a class of vehicle routing problems," *Computers & Operations Research*, vol. 40, no. 10, pp. 2519–2531, Oct. 2013.

[5] T. Vidal, T. G. Crainic, M. Gendreau, and C. Prins, "A unified solution framework for multi-attribute vehicle routing problems," *Eur. J. Oper. Res.*, vol. 234, no. 3, pp. 658–673, 2014.

[6] S. Ropke and D. Pisinger, "A Unified Heuristic for a Large Class of Vehicle Routing Problems with Backhauls," *Eur. J. Oper. Res.*, vol. 171, no. 3, 2006, pp. 750–775.

[7] Y. Nagata and O. Bräysy, "Edge Assembly-Based Memetic Algorithm for the Capacitated Vehicle Routing Problem," *Networks*, vol. 54, no. 4, pp. 205–215, Dec. 2009.

[8] P. Augerat, J. M. Belenguer, E. Benavent, A. Corberan, D. Naddef, and G. Rinaldi, "Computational results with a branch and cut code for the capacitated vehicle routing problem," Research Report 949-M, Universite Joseph Fourier, France.

[9] N. Christofides and S. Eilon, "An algorithm for the vehicle dispatching problem," *Operat. Res. Q.*, vol. 20, pp. 309–318, 1969.

[10] M. L. Fisher, "Optimal solution of vehicle routing problems using Minimum K-Tree," *Oper. Res.*, vol. 42, pp. 626–642, 1994.

[11] N. Christofides, A. Mingozzi, and P. Toth, "The vehicle routing problem," *Combinatorial optimization*, vol. 11, pp. 315-338, 1979.

[12] Y. Rochat and E. D. Taillard, "Probabilistic diversification and intensification in local search for vehicle routing," *J. Heuristics*, vol. 1, no. 1, pp. 147–167, 1995.

[13] B. L. Golden, E. A. Wasil, J. P. Kelly, and I. M. Chao, "The impact of metaheuristics on solving the vehicle routing problem: Algorithms, problem sets, and computational results," *Fleet Manage. Logist.*, pp. 33–56, 1998.

[14] E. Uchoa, D. Pecin, A. Pessoa, M. Poggi, T. Vidal, A. Subramanian, "New benchmark instances for the capacitated vehicle routing problem," *Eur. J. Oper. Res.*, vol. 257, no. 3, pp. 845-858, 2017.

[15] B. L. Golden and R. T. Wong, "Capacitated arc routing problems," *Networks*, vol. 11, no. 3, pp. 305–315, 1981.

[16] K. Tang, J. Wang, X. Li, X. Yao, "A Scalable Approach to Capacitated Arc Routing Problems Based on Hierarchical Decomposition," *IEEE Trans. Cybern.*, vol. 47, no. 11, pp. 3298-3940, Nov. 2017.

[17] Y. Mei, X. Li, and X. Yao, "Decomposing large-scale capacitated arc routing problems using a random route grouping method," in *Proc. IEEE Congr. Evol. Comput. (CEC)*, Cancún, Mexico, 2013, pp. 1013–1020.

[18] Y. Mei, X. Li, and X. Yao, "Cooperative coevolution with route distance grouping for large-scale capacitated arc routing problems," *IEEE Trans. Evol. Comput.*, vol. 18, no. 3, pp. 435–449, Jun. 2014.

[19] Y. Mei, X. Li, and X. Yao, "Variable neighborhood decomposition for large scale capacitated arc routing problem," in *Proc. IEEE Congr. Evol. Comput. (CEC)*, Beijing, China, 2014, pp. 1313–1320.

[20] W.-L. Pearn, A. Assad, and B. L. Golden, "Transforming arc routing into node routing problems," *Comput. Oper. Res.*, vol. 14, no. 4, pp. 285–288, 1987.

[21] R. Baldacci and V. Maniezzo, "Exact methods based on node-routing formulations for undirected arc-routing problems," *Networks*, vol. 47, no. 1, pp. 52–60, 2006.

[22] H. Longo, D. A. M. Poggi, and E. Uchoa, "Solving capacitated arc routing problems using a transformation to the CVRP," *Comput. Oper. Res.*, vol. 33, no. 6, pp. 1823–1837, 2006.

[23] P. Toth and D. Vigo, "Branch-and-bound algorithms for the capacitated VRP," in *The Vehicle Routing Problem*, P. Toth and D. Vigo, Eds. Philadelphia, PA, USA: SIAM, 2001, pp. 29–51.

[24] G. Clarke and J. W. Wright, "Scheduling of vehicles from a central depot to a number of delivery points," *Oper. Res.*, vol. 12, pp. 568–581, 1964.

[25] R.H. Mole and S. R. Jameson, "A sequential route-building algorithm employing a generalised savings criterion,"*Operation Research Quarterly*, vol. 27, 1976, pp. 503-511.

[26] S. Kirkpatrik, C. Gelatt, and M. Vecchi, "Optimization by Simulated Annealing," *Science*, vol. 220, pp. 671-680, 1983.

[27] F. Glover, "Future paths for integer programming and links to artificial intelligence," *Comput. Oper. Res.*, vol. 13, pp. 533–549, May 1986.

[28] J. Baxter, "Depot location: A technique for the avoidance oflocal optima," *Eur. J. Oper. Res.*, vol. 18, pp. 208–214, 1984.

[29] N. Mladenovic and P. Hansen, "Variable Neighborhood Search", *Computers and Operations Research*, vol. 24, pp. 1097-1100, 1997.

[30] M. Reimann, K. Doerner, and R. Hartl, "D-ants: Savings based ants divide and conquer the vehicle routing problem," *Computers & Operations Research*, vol. 31, no. 4, pp. 563– 591, 2003.

[31] J. Holland, *Adaption in Natural and Artificial Systems.* Ann Arbor, MI: Univ. Michigan Press, 1975.

[32] F. Glover, "Heuristics for integer programming using surrogate constraints," *Dec. Sci.*, vol. 8, pp. 156–166, 1977.

[33] R. W. Eglese, "Routeing winter gritting vehicles," *Discr. Appl. Math.*, vol. 48, no. 3, pp. 231–244, 1994.

[34] V. Maniezzo, "Algorithms for large directed CARP instances: Urban solid waste collection operational support," Dept. Comput. Sci., Univ. Bologna, Bologna, Italy, Tech. Rep. UBLCS-2004-16, 2004.

[35] J. MacQueen, "Some methods for classification and analysis of multivariate observations," in *Proc. 5th Berkeley Symp. Math. Stat. Probab.*, Berkeley, CA, USA, 1967, pp. 281–297.

[36] G. Ulusoy, "The fleet size and mix problem for capacitated arc routing," *Eur. J. Oper. Res.*, vol. 22, no. 3, pp. 329–337, 1985.

[37] K. Tang, Y. Mei, and X. Yao, "Memetic algorithm with extended neighborhood search for capacitated arc routing problems," *IEEE Trans. Evol. Comput.*, vol. 13, no. 5, pp. 1151–1166, Oct. 2009.

[38] M. Bellmore and G. Nemhauser, "The traveling sales man problem: A survey," *Operation Research*, vol. 16, pp. 538-558, 1968.

[39] N. Christofides, "Worst-case analysis of a new heuristic for the traveling salesman problem," Tech. Rep. 388, Carnegie Mellon University, Pittsburgh, PA, Apr. 1976.

[40] D. Johnson and L. McGeoch, *The Traveling Salesman Problem: A Case Study in Local Optimization.* New York: Wiley, 1997.

[41] P. Toth and D. Vigo, editors. *The Vehicle Routing Problem.* Society for Industrial & Applied Mathematics (SIAM), 2001