This article has been accepted for inclusion in a future issue of this journal. Content is final as presented, with the exception of pagination.

广义

Solving Generalized Vehicle Routing Problem With Occasional Drivers via Evolutionary Multitasking

Liang Feng, Lei Zhou, Abhishek Gupta, Jinghui Zhong, Zexuan Zhu, Kay-Chen Tan, and Kai Qin

***Abstract*—With the emergence of *crowdshipping* and *sharing economy*, vehicle routing problem with occasional drivers (VRPOD) has been recently proposed to involve occasional drivers with private vehicles for the delivery of goods. In this article, we present a generalized variant of VRPOD, namely, the vehicle routing problem with heterogeneous capacity, time window, and occasional driver (VRPHTO), by taking the capacity heterogeneity and time window of vehicles into consideration. Furthermore, to meet the requirement in today's cloud computing service, wherein multiple optimization tasks may need to be solved at the same time, we propose a novel evolutionary multitasking algorithm (EMA) to optimize multiple VRPHTOs simultaneously with a single population. Finally, 56 new VRPHTO instances are generated based on the existing common vehicle routing benchmarks. Comprehensive empirical studies are conducted to illustrate the benefits of the new VRPHTOs and to verify the efficacy of the proposed EMA for multitasking against a state-of-art single task evolutionary solver. The obtained results showed that the employment of occasional drivers could significantly reduce the routing cost, and the proposed EMA is not only able to solve multiple VRPHTOs simultaneously but also can achieve enhanced optimization performance via the knowledge transfer between tasks along the evolutionary search process.**

***Index Terms*—Evolutionary multitasking, occasional driver, time window, vehicle routing problem (VRP).**

Manuscript received August 19, 2019; revised November 11, 2019; accepted November 14, 2019. This work was supported in part by the National Natural Science Foundation of China under Grant 61876025, Grant 61603064, Grant 61602181, and Grant 61871272, in part by the Chongqing Application Foundation and Research in Cuttingedge Technologies under Grant cstc2017jcyjAX0319, in part by the Venture and Innovation Support Program for Chongqing Overseas Returnees under Grant cx2018044, in part by the City University of Hong Kong Research Fund under Grant 9610397, and in part by the Shenzhen Scientific Research and Development Funding Program under Grant JCYJ20180307123637294 and Grant JCYJ20170302154328155. This article was recommended by Associate Editor H. A. Abbass. *(Corresponding author: Liang Feng.)*

L. Feng and L. Zhou are with the College of Computer Science, Chongqing University, Chongqing 400044, China (e-mail: liangf@cqu.edu.cn; stone_zhou@cqu.edu.cn).

A. Gupta is with the Singapore Institute of Manufacturing Technology, Agency for Science, Technology and Research, Singapore (e-mail: abhishek_gupta@simtech.a-star.edu.sg).

J. Zhong is with the School of Computer Science and Engineering, South China University of Technology, Guangzhou 510006, China (e-mail: jinghuizhong@gmail.com).

Z. Zhu is with the College of Computer Science and Software Engineering, Shenzhen University, Shenzhen 518060, China, and also with the Shenzhen University Branch, Shenzhen Institute of Artificial Intelligence and Robotics for Society, Shenzhen 518060, China (e-mail: zhuzx@szu.edu.cn).

K.-C. Tan is with the Department of Computer Science, City University of Hong Kong, Hong Kong (e-mail: kaytan@cityu.edu.hk).

K. Qin is with Department of Computer Science and Software Engineering, Swinburne University of Technology, Hawthorn, VIC 3122, Australia (e-mail: kqin@swin.edu.au).

Color versions of one or more of the figures in this article are available online at http://ieeexplore.ieee.org.

Digital Object Identifier 10.1109/TCYB.2019.2955599

I. Introduction

THE VEHICLE routing problem (VRP) and its variants represent the cornerstone of transportation, logistics, and supply chain management [1]–[3]. In VRP, the distribution of goods lies at the heart of business activity since it is often coupled with inventory and production decisions, and the delivery cost accounts for a main portion of the total logistic costs. The objective of VRP is to design a collection of routes for multiple vehicles to serve a set of customers under certain constraints with the goal of minimizing the total cost (e.g., travel cost, service cost, etc.) involved. Over the last decades, extensive research works for solving VRP have been reported in the literature. For instance, Hirabayashi *et al.* [4] proposed an exact algorithm by using branch-and-bound for solving VRP, while Clarke and Wright [5] solved VRP with a constructive heuristic which joins two routes into one iteratively until no cost savings can be achieved. Further, Osman and Salhi [6] presented a tabu search approach by the use of a compound-moves neighborhood together with a simple tabu list mechanism. Further, Prins [7] proposed a genetic algorithm hybridized with a local search procedure to intensify the search for VRP solutions. More recently, Nalepa and Blocho [8] introduced an adaptive memetic algorithm, which can dynamically configure the algorithm parameters based on current routing information. For more reviews of algorithms for solving VRP, interested readers may refer to [9].

Today, with the emergence of "crowdshipping" and "sharing economy," vehicle routing problem with occasional drivers (VRPOD) has been recently proposed to consider occasional drivers when scheduling the delivery of goods for customers [10]–[12]. In particular, VRPOD not only has a fleet of capacitated vehicles and regular drivers available to make deliveries but it may also use the services of occasional drivers who are willing to make deliveries using their own vehicles in return for a small compensation if the delivery location is not too far from their own destination. This new VRP variant has demonstrated that the delivery cost could be significantly reduced when occasional drivers are employed [13]. However, it is worth noting that, two limitations exist in the current VRPOD setting. First, variances of vehicle capacity and cost are not considered when employing the occasional drivers, which is often unrealistic in practice since the fleets and cost of vehicles are commonly heterogeneous. Second, the traditional VRPOD is solved by a single-task optimization method, which is deem slow and ignores the similarities between tasks that could be leveraged for enhanced optimization performance when properly harnessed.

Taking this cue, in this article, we present a generalized variant of VRPOD, that is, vehicle routing problem with heterogeneous capacity, time window, and occasional driver (VRPHTO). In the proposed VRPHTO, both regular and occasional drivers are considered for the delivery service using a fleet of vehicles with heterogeneous capacity. Each customer must be visited exactly once within a given customer time window. Furthermore, instead of restricting each occasional driver to be deployed only once, a time window is associated with the occasional drivers. An occasional driver can be deployed multiple times within his/her service time window. In addition, only regular drivers have to return back to the depot once the assigned services have been completed. The objective of VRPHTO is to minimize the total cost (e.g., travel cost, service cost, etc.) without violating the predefined constraints. Finally, the compensation for occasional drivers varies with the corresponding vehicle type.

Further, rather than solving the routing problems in the traditional single task mode,[1] in this article, we conceptualize a cloud-based optimization service that is capable of catering to multiple VRPHTOs requests at the same time. Given the growing popularity of cloud computing, it is conceivable to have various logistics/transport service providers in a city to upload their routing problems on the cloud—which can potentially house a centralized planning/decision-support system [14]–[16]. Recent studies in evolutionary multitask optimization have shown that instead of tackling them independently in a sequential or parallel manner, combining multiple related tasks together in a single multitasking environment can often lead to faster discovery of consistently higher quality solutions. This is said to be achieved through the automatic exploitation of intertask similarities during multitasking—which facilitates the implicit transfer of reusable knowledge across the tasks in a unified representation space. To the best of our knowledge, there is little existing work in the literature solving routing problems in a multitask environment. We believe the great barrier to further progress can be attributed to the unique representations and characteristics of different routing problems. Hence, it is often the case that the optimization solver designed for a particular routing problem cannot optimize another at the same time. Here, inspired by the recent progress of multifactorial optimization in evolutionary computation [17], we propose an evolutionary multitasking algorithm (EMA) to optimize multiple VRPHTOs simultaneously. The backbone of the proposed EMA consists of four components, namely, permutation-based common representation, split procedure, routing information exchange, and chromosome evaluation. In contrast to existing EMAs, new designs in unified search space, decoding scheme from unified space to task-specific space, and routing information exchange operators are developed for solving VRPHTO. To summarize, the core contributions of the current work are multifacets, which are outlined as follows.

1) We present a generalized variant of VRPOD, labeled as VRPHTO, with the introduction of the time window for both customers and occasional drivers. Further, a fleet of vehicles with heterogeneous capacity is considered, which makes the problem more realistic and practical.
2) To meet the requirement of a potential cloud-based optimization solver faced with multiple distinct (but possibly similar) routing problems at the same time, a novel EMA is proposed to solve multiple VRPHTOs simultaneously.
3) New benchmarks of the proposed VRPHTO, in both single and multiple task mode, are generated based on the commonly used FSMVPRTW benchmarks proposed by Liu and Shen [18].
4) Comprehensive empirical study on the newly generated benchmarks is conducted to verify the efficacy of the proposed EMA. The experimental results show that, in contrast to the state-of-the-art evolutionary solver which tackles one routing problem in a single run, the proposed EMA is able to enhance the optimization process with fast convergence speed and superior or competitive solution quality obtained, on most VRPHTO instances.

The rest of this article is organized as follows. Section II begins with a brief introduction of the VRP variants proposed in the literature. The motivation and review on recent progress of evolutionary multitasking are also presented in this section. In Section III, we give the mathematical definition for the proposed VRP variant, namely, VRPHTO. The definition of solving VRPHTO in the multitasking scenario is also presented in this section. The proposed EMA for VRPHTO is then given in Section IV. Further, Section V discusses how to generate VRPHTO benchmarks based on FSMVPRTW instances, which is followed by the empirical studies conducted to verify the efficacy of our proposed EMA for multitasking. Finally, Section VI concludes this article with several remarks and a discussion on potential future works.

II. Preliminary

In this section, we first present a brief review of the VRP variants in the literature. Subsequently, the motivation as well as recent research progress of evolutionary multitasking are discussed.

A. *Variants of the Vehicle Routing Problem*

The vehicle routing problem (VRP) is a well-known combinatorial optimization problem, which determines the routes for a fleet of vehicles to serve a set of geographically dispersed customers with the objective of minimizing the total cost involved [19], [20]. It generalizes the traveling salesman problem (TSP) and has been proven to be NP-hard. In the literature, VRP has attracted extensive attention from both

[1]Single task mode means that only a single problem is solved by the optimization algorithm in one single run. In this article, a task refers to one VRPHTO instance, which requires to optimize the routes of multiple vehicles for serving customers.

academy and industry, and many variants of VRP by considering different constraints for vehicle routing have been proposed, such as VRP with stochastic demand, in which the exact demands of customers are unknown in advance [21]; multidepot VRP, which considers multiple depots for hosting the vehicles [22]; and dynamic VRP, in which the customers need to be served are not fixed, and varying overtime [23].

In particular, one of the most popular VRP variants is the capacitated vehicle routing problem (CVRP), which was proposed by Dantzig and Ramser in 1959 [1]. In CVRP, it is assumed that there is a homogeneous fleet of vehicles with restricted capacity, based at a central depot, which is used to satisfy customers' demands by visiting them only once. Based on CVRP, another well-known VRP variant is the vehicle routing problem with time window (VRPTW), in which the service of each customer has to start within a specific time interval. In the literature, based on the requirements that exist in real-world VRP applications, there are generally two VRPTW versions, that is, VRP with hard time window, in which customers' time windows cannot be violated [24], [25], and VRP with soft time window where a time window can be violated by paying a penalty [26]. Further, by considering more requirements and constraints that exist in real-world routing scenarios, many extensions of VRPTW have been introduced in the literature. For instance, fleet size and mix VRPTW (FSMVRPTW) was proposed to consider the vehicles with heterogeneous capacities and fixed costs in VRPTW [18]. Multidepot VRPTW (MDVRPTW) was introduced for the cases where rather than a single depot for scheduling the vehicles, multiple depots with different locations are available [27]. Further, multitrip VRPTW (MTVRPTW) was presented to allow customers to be visited more than once [28], while multiobjective VRPTW (MOVRPTW) was proposed to take multiple objectives, such as routing cost, number of vehicles, customer satisfaction, environmental impact, etc., into considerations while scheduling the routes of vehicles for customers [29].

More recently, with the development of logistics and supply chain management, transportation modes have become more flexible, which leads to new models for routing service. For instance, toward green travel, electric vehicles have been considered for vehicle routing problem [30], [31]. In contrast to traditional VRPs, the battery capacity is a new factor in defining the routing solutions [32]. For the VRP considering traffic congestion [33], traveling time between two customers is time-dependent and stochastic because of traffic congestion. Further, the soft-clustered vehicle-routing problem has been proposed to extend the classical capacitated VRP by requesting that customers are partitioned into clusters and all customers of the same cluster must be served by the same vehicle [34]. The periodic VRP has been defined as a generalization of the classical capacitated VRP in which routes are determined for a planning horizon of several days [35]. Besides only the trucks, the drones have been also considered for delivering parcels to customers, which is known as VRP with drones [36], [37]. Moreover, as aforementioned, besides regular drivers, occasional drivers have been considered in vehicle routing for today's "crowdshipping" and "sharing economy," which leads to the new VRP variant, namely, VRPOD. Dahle *et al.* [38] investigated the dynamic VRPOD, where the availability of occasional drivers is uncertain beforehand. However, as aforementioned, the present VRPOD only considers a common capacity and cost of different occasional drivers, which is unrealistic in practice. Toward more realistic VRPOD, in this article, we thus present a generalized VRPOD, which considers the time window and heterogeneous capacity of both regular and occasional drivers.

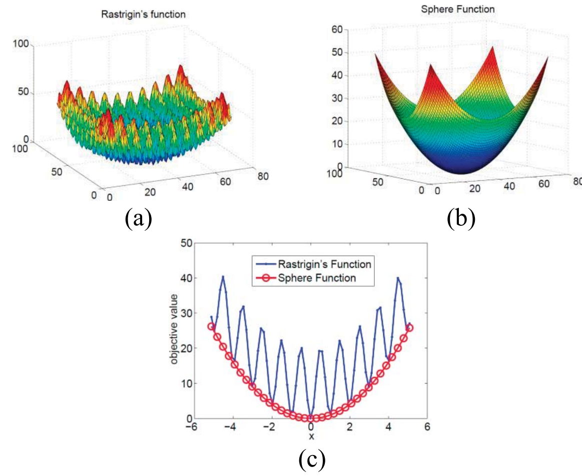


Fig. 1. Landscape of Rastrigin's function and sphere function. (a) Landscape of Rastrigin's function. (b) Landscape of Sphere function. (c) Rastrigin's and Sphere in 1 Dimension.

For more comprehensive review on the variants of the vehicle routing problem, the reader may refer to the book [39], [40], and the survey [41].

B. Evolutionary Multitasking

Today, it is well established that problems seldom exist in isolation, and the proper harness of useful traits from related problems could lead to improved optimization performance across problem domains. For example, Fig. 1 depicts the landscapes of two popular benchmark functions, that is, sphere function and rastrigin's Function [17]. As can be observed from the landscapes, the complexity of the chosen problems is not uniform. The convex sphere function is the simplest [see Fig. 1(b)], while multimodal Rastrigin's function which contains many local optima, is considerably more challenging to optimize [see Fig. 1(a)]. However, as illustrated in Fig. 1(c), by plotting the two functions in 1-D space, we note they share a common global optimum. Thus, the solutions found along the optimization process of sphere function can potentially be utilized to aid the optimization of the complex Rastrigin's functions.

Keeping the above in mind, a new search paradigm, namely, evolutionary multitasking, has been proposed by Gupta *et al.* [17], in the realm of evolutionary computation. By exploiting the latent synergies between distinct (but possibly similar) optimization problems, superior search performances of evolutionary multitasking in terms of solution quality and convergence speed have been verified on a set of continuous, discrete, and the mixtures of continuous and combinatorial tasks [17]. Due to the efficacy demonstrated by evolutionary multitasking, increasing research interests in developing EMAs for solving complex real-world problems have emerged in the literature. In particular, Yuan *et al.* [42] proposed a permutation-based multitasking algorithm to efficiently solve combinatorial problems, such as traveling salesman problem and quadratic assignment problem. Gupta *et al.* extended the work in [17], and proposed a multiobjective EMA [43]. Ding *et al.* [44] presented a generalized EMA, while Zhong *et al.* extended the multitasking in genetic programming and introduced a multifactorial genetic programming algorithm in [45]. Further, Liaw and Ting [46] extended the multitasking algorithm in [17] for solving the many-task optimization problems. Gong *et al.* [47] and Wen and Ting [48] proposed to allocate the computational resources according to the complexities of tasks for efficient evolutionary multitasking. More recently, Bali *et al.* [49] employed an online transfer parameter estimation scheme to dynamically control the extent of knowledge exchange in evolutionary multitasking.

In this article, to conceptualize an efficient cloud-based VRPHTO optimization service, we propose an EMA for addressing multiple VRPHTOs simultaneously, which will be detailed in Section IV.

III. Problem Formulation

A. Mathematical Formulation for the Proposed VRPHTO

Toward a more realistic VRP with occasional drivers, based on the existing VRPOD formulation, we further derive a formulation of VRPHTO, which takes the heterogeneous capacities and available time windows of vehicles into consideration. In particular, the proposed VRPHTO is defined on a graph $G(V, A)$, where $V = \{0, 1, \ldots, n\}$ denotes the node set, $A = \{(i, j)|i, j \in V, i \neq j\}$ represents the edge set. Each edge (i, j) is associated with a travel distance d_{ij}. In the proposed VRPHTO, we assume the velocity of all vehicles is 1. Therefore, the travel time t_{ij} of edge (i, j) is equal to d_{ij}. In addition, both d_{ij} and t_{ij} satisfy the triangle inequality.

In the graph, node 0 represents the depot, while the remaining nodes $N = \{1, \ldots, n\}$ represent the customers to be served. Each customer $i \in N$ has a positive demand q_i, a service time s_i, and a time window $[e_i, l_i]$ that gives the earliest and latest time for the vehicle to start its service. A vehicle has to wait for service if it arrives at customer i earlier than e_i. Further, a time window is also associated with the depot, that is, $[e_0, l_0] = [E, L]$, where E and L represent the earliest departure time from the depot and the latest returning time to the depot, respectively.

Further, let K be the set of available vehicles, which can be divided into T types. For the sake of simplicity, in the proposed VRPHTO, vehicles belong to the same type $t \in T$ share the same capacity Q_t, fix cost fc_t, variable cost vc_t, and vehicle time window $[ve_t, vl_t]$ which indicates the vehicle is only available during the time period from ve_t to vl_t. In other words, the vehicle cannot leave the depot before ve_t, and must complete all the assigned service tasks before vl_t. In addition, all vehicle types can be divided into two classes, that is, *TR* and *TO*, which represent the vehicles are owned by the regular drivers or occasional drivers, respectively. If a vehicle belongs to *TR*, it must go back to the depot once it completes all the services, that is, before time L.

Further, the decision variables of VRPHTO are as follows.

1) x_{ij}^k: Equals to 1 if arc (i, j) is traversed by vehicle k, or 0 otherwise.
2) a_{ik}: Arrival time of vehicle k at customer i.
3) w_{ik}: Waiting time of vehicle k at customer i.

Next, the mathematical formulation for the proposed VRPHTO is given as follows:

$$\text{Min} \sum_{k\in K,\tau(k)\in T}\sum_{j\in V} fc_{\tau(k)}x_{0j}^k + \sum_{k\in K,\tau(k)\in T}\sum_{i\in V}\sum_{j\in V} vc_{\tau(k)}d_{ij}x_{ij}^k \tag{1}$$

$$\text{Subject to} \sum_{k\in K}\sum_{j\in N} x_{ij}^k = 1 \quad \forall i \in V \tag{2}$$

$$\sum_{j\in V} x_{ij}^k = \sum_{j\in V} x_{ji}^k \quad \forall i \in V, k \in K \tag{3}$$

$$0 \leq \sum_{i\in N} x_{i0}^k \leq 1 \quad \forall k \in K, \tau(k) \in T \tag{4}$$

$$\sum_{i\in N} q_i \sum_{j\in V} x_{ij}^k \leq Q_{\tau(k)} \ \forall k \in K, \tau(k) \in T \tag{5}$$

$$x_{ij}^k\big(a_{ik} + w_{ik} + s_i + d_{ij} - a_{jk}\big) = 0 \quad \forall k \in K, (i,j) \in A \tag{6}$$

$$e_i \sum_{j\in V} x_{ij}^k \leq a_{ik} + w_{ik} \leq l_i \sum_{j\in V} x_{ij}^k \quad \forall k \in K, i \in N \tag{7}$$

$$ve_{\tau(k)} \sum_{j\in V} x_{ij}^k \leq a_{ik} + w_{ik} + s_i \leq vl_{\tau(k)} \sum_{j\in V} x_{ij}^k \tag{8}$$

$$\forall i \in N, k \in K, \tau(k) \in T$$

$$a_{ik}, w_{ik} \geq 0 \ \forall k \in K, i \in N \tag{9}$$

$$x_{ij}^k \in \{0, 1\} \ \forall k \in K, (i,j) \in A. \tag{10}$$

The objective of VRPHTO (1) is to minimize the total cost involved without violating any constraint. $\tau(k)$ is a function to get the corresponding type of vehicle k. Constraints (2) and (3) ensure that each customer can only be visited by one vehicle exactly once. Constraint (4) indicates that only regular drivers must return back to the depot when customer service is completed. Constraint (5) guarantees that the total demands of each route must not exceed the capacity of the allocated vehicle. Constraint (6) represents the relationships among the arrival time, service time, traveling time, and waiting time of vehicles.[2] Constraints (7) and (8) are the time windows for the customers and the vehicles, respectively. Constraint (9) indicates that the arrival time and waiting time are both nonnegative, and constraint (10) restrict the values of x_{ij}^k to be either 0 or 1.

[2]According to the recent study in [50] and [51], in the paper, the velocity of vehicles is assumed to be 1. However, varying velocity could be used to change the traveling time of vehicles if more traffic information is available.

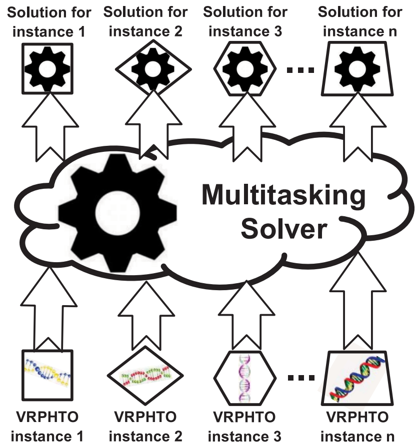


Fig. 2. Illustration of solving VRPHTO in multitasking environment.

Furthermore, as VRPOD is a variant of the capacitated VRP which has been proven to be NP-hard [13], and the consideration of heterogeneous capacity as well as time window of a vehicle will not affect the NP-hardness of the problem [52], the proposed VRPHTO is an NP-hard combinatorial optimization problem.

B. VRPHTO in Multitasking Environment

To enhance the optimization performance of the cloud-based optimization service as aforementioned, which is capable of addressing the VRPHTOs requests received from different logistics/transport service providers, in this article, instead of tackling the VRPHTOs independently in a sequential or parallel manner, we propose to combine multiple VRPHTOs together in multitasking environment and solve them via evolutionary multitasking. In particular, Fig. 2 illustrates the main concept of multitasking VRPHTO. As depicted, multiple VRPHTO instances are required to be optimized simultaneously by a single multitasking solver. Note that, multitasking has a significant distinction with multiobjective optimization (MOO), with the latter aiming to find an optimal tradeoff among several competing objectives of one single instance.

Formally, consider the scenario where n VRPHTO instances have to be solved concurrently. For the ith instance, denoted as P_i, we define the objective function as $f_i(\mathbf{x_i}) \rightarrow \mathbb{R}$, where $\mathbf{x}_i$ is a feasible solution in the search space of P_i, that is, $\mathbf{X}_i$. The objective of multitasking VRPHTO is to find $\{\mathbf{x}_1, \mathbf{x}_2, \ldots, \mathbf{x}_{n-1}, \mathbf{x}_n\} = \text{argmin}\{f_1(\mathbf{x}_1), f_2(\mathbf{x}_2), \ldots, f_{n-1}(\mathbf{x}_{n-1}), f_n(\mathbf{x}_n)\}$, where $\mathbf{x}_i$ is a feasible solution in $\mathbf{X}_i$. Note that we do not look for a tradeoff between $f_1, f_2, \ldots, f_n$, as is the case in MOO. In contrast, the aim of multitasking is to optimize each individual objective function $f_1, f_2, \ldots, f_n$, to the best possible value (within a limited amount of computation time) by exploiting the underlying relationships between them online.

IV. Proposed Method for Multitasking With VRPHTO

In this section, we first describe the workflow of the proposed EMA for solving several VRPHTO instances in a single multitasking environment. Next, the four main components in EMA, that is, permutation-based unified representation, split procedure, routing information exchange, and chromosome evaluation, are presented in detail.

A. Evolutionary Multitasking for VRPHTO

To solve VRPHTOs in multitasking mode with a single solver, according to [17], the following issues have to be considered in the algorithm design.

1) How to integrate multiple VRPHTOs with different properties into a unified search space?
2) How to transform a chromosome in the unified search space to be evaluated for a specific task?
3) How to determine the elitism of chromosomes whose performance varies among different VRPHTO tasks?

Keeping the above in mind, here we present the details of the proposed EMA for multitasking with VRPHTOs. First of all, based on [17], we introduce the following definitions for evolutionary multitasking.

Definition 1 (Factorial Cost): The factorial cost f_p of a chromosome p denotes its fitness or objective value on a particular task T_i. For K tasks, there will be a vector with length K, in which each dimension gives the fitness of p on the corresponding task.

Definition 2 (Factorial Rank): The factorial rank r_p simply denotes the index of a chromosome p in the list of population members sorted in ascending order with respect to their factorial costs on one specific task.

Definition 3 (Scalar Fitness): The scalar fitness φ_p of a chromosome p is defined based on its best rank over all tasks, which is given by $\varphi_p = [1/(\min_{j\in\{1,\ldots,K\}} r_p^j)]$.

Definition 4 (Skill Factor): The skill factor τ_p of a chromosome p denotes the task, amongst all other tasks, on which p is most effective, that is, $\tau_p = \text{argmin}\{r_p^j\}$, where $j\in\{1, \ldots, K\}$.

With the definitions above, the workflow of the proposed EMA for multitasking with VRPHTOs is summarized in Algorithm 1. As outlined in Algorithm 1, first of all, EMA initializes a population of chromosomes with size N_p using a permutation-based common representation and evaluates each chromosome on all the tasks to obtain the corresponding factorial cost, factorial rank, scalar fitness, and skill factor (lines 1 and 2). Second, in the while loop (lines 4-29), chromosomes p_a and p_b are selected randomly as parents to generate offspring. The skill factors of the parents and a randomly generated number are used to decide whether to perform crossover or mutation, where the routing information exchange across tasks may occur.

Further, as it is extremely computationally expensive to evaluate each chromosome on all tasks along the search, in EMA, a compromise is made to evaluate a chromosome on only a selected task where it is most likely to perform well. Specifically, if an offspring is generated by crossover, it may either be evaluated for task τ_a (i.e., skill factor of parent p_a) or task τ_b (i.e., skill factor of parent p_b) with equal probability. Otherwise, it will be evaluated on the task with respect to its parent's skill factor. Further, the offspring may undergo a local

Algorithm 1: Pseudo Code of the Proposed *EMA*.

Initialize a population P of size N_p using ***permutation-based unified representation***;
Evaluate each chromosome i on all the tasks, and obtain its *factorial cost* f_i, *factorial rank* r_i, *scalar fitness* φ_i and *skill factor* τ_i accordingly;
for *Restart:= 1 to* N_{re} **do**
  while *(Number of task evaluations < TE)* **do**
    Select parents p_a, p_b in P via binary tournament;
    Randomly generate $rand \in \{0, 1\}$;
    /* ***Routing information exchange****/
    if $\tau_a = \tau_b$ *or* $rand \leq rmp$ **then**
      Crossover(p_a, p_b) $\rightarrow$ Offspring c_a and c_b;
    else
      Mutate(p_a) $\rightarrow$ Offspring c_a;
      Mutate(p_b) $\rightarrow$ Offspring c_b;
    /* ***Split procedure*** *is performed on c before evaluating c on a particular task* */
    foreach $c \in \{c_a, c_b\}$ **do**
      if *c has 2 parents* **then**
        Randomly generate $rand \in \{0, 1\}$;
        if $rand < 0.5$ **then**
          c is evaluated only for task τ_a;
        else
          c is evaluated only for task τ_b;
      else
        c is evaluated only on the task according to its parent's *skill factor*;
      Conduct local search on c under P_{ls};
      Set the *factorial costs* of c with respect to all the unevaluated tasks as ∞ (a big value);
    Add c_a and c_b to the offspring population C;
    if n_c *(number of solutions in C) is equal to* N_p **then**
      Concatenate P and C to form population I;
      Update the *scalar fitness* φ_i and *skill factor* τ_i of every chromosome in I;
      Preserve the fittest N_p solutions in I by ***chromosome evaluation***, for survival in P;
  if *Restart* $\leq N_{re}$ **then**
    Restart by regenerating all the chromosomes in P, while preserving the N_{be} best ones.

search procedure on the evaluated task with a given probability, and its factorial costs on all the unevaluated tasks are set to be a very large number. Note that, before evaluating the chromosome on a particular task, the split procedure is performed to translate the chromosome from the unified search space into a feasible routing solution.

Finally, in each generation, N_p number of offspring will be generated and concatenated with the current population to form an intermediate population. The chromosome evaluation then kicks in to select the fittest N_p chromosomes to survive in the next generation. In addition, the chromosomes except the best N_{be} individuals, in the population will be initialized once *TE* number of evaluations have been reached. The whole algorithm terminates after N_{re} reinitialization.

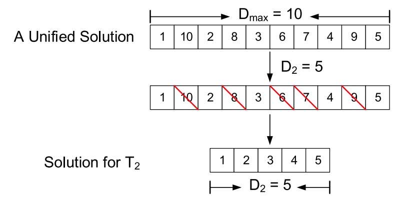


Fig. 3. Example of the reduction from $D_{\max}$ to D_i.

B. Permutation-Based Unified Representation Scheme and Decoding Exemplar

As different VRPHTOs contain unique customer set to be served, in traditional single task evolutionary search, specific chromosome encoding is required when a particular VRPHTO is encountered. In contrast, in our proposed EMA for VRPHTO, multiple VRPHTOs are solved by a single evolutionary solver. Thus, a unified representation is necessary to encode all the VRPHTOs of interests.

To encode multiple VRPHTOs in EMA, we adopt the *permutation-based representation* which is commonly used for VRPs [53], [54]. With this representation, a chromosome is encoded as a giant tour represented by a sequence in which each dimension is a customer id. Particularly, assume that K VRPHTOs are to be solved simultaneously and task P_i has D_i customers. The dimensionality of the unified chromosome representation is set as $D_{max} = max\{D_i\}$, $i = \{1, \ldots, K\}$. Each dimension is then initialized with different values that lie in the range of $[0, D_{max}]$. For evaluation on task P_i, we simply refer to the dimensions of values from 1 to D_i in the unified chromosome representation.

For instance, as depicted in Fig. 3, the dimensionality of chromosomes in the common representation is $D_{max} = 10$. Suppose a chromosome $\{1, 10, 2, 8, 3, 4, 7, 6, 9, 5\}$ is to be evaluated on task P_i with dimension $D_i = 5$. Then, the solution for P_i can be obtained by deleting the elements with the values greater than 5, which is $\{1, 2, 3, 4, 5\}$.

C. Extended Split Procedure

In the literature, the *split* approach serves as the procedure to translate a permutation-based chromosome into a feasible routing solution. For vehicle routing, the *split* operator has been widely used to partition the chromosome by inserting trip delimiters and returns the best partition as the output routing solution [53], [54]. As the proposed VRPHTO is extended from the VRPOD with additional constraints representing the heterogeneous capacities and time windows of vehicles, the *split* thus can also be extended for VRPHTO. In particular, in the present context, to guarantee the feasibility of the VRPHTO solutions, we extend the *split* by

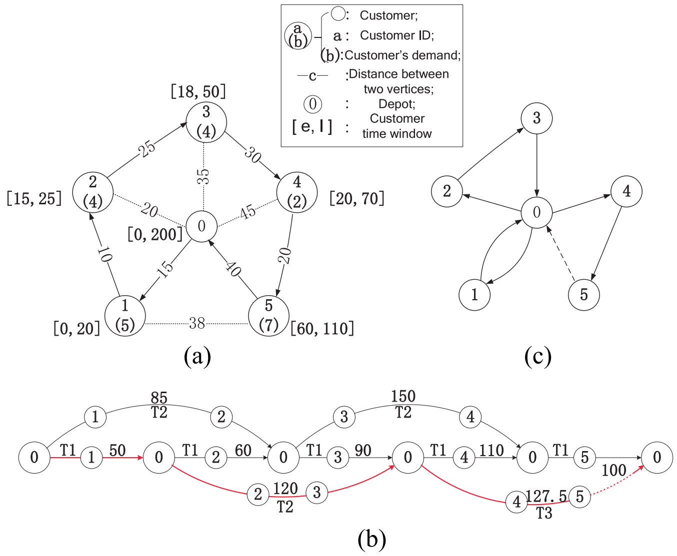


Fig. 4. Examples of the extended *split* procedure. (a) A giant tour $S = 1, 2, 3, 4, 5$. (b) All the feasible routes evaluated by *Split*. (c) The optimal solution for S.

TABLE I
PROPERTIES OF A VRPHTO EXAMPLE WITH THREE VEHICLE TYPES ("F_c," "V_c," AND "VTW" REPRESENT THE FIXED COST, THE VARIABLE COST, AND THE VEHICLE TIME WINDOW OF THE VEHICLES, RESPECTIVELY)

Vehicle Type	*Capacity*	F_c	V_c	*VTW*
$T1$	8	20	1	$[0, \infty]$
$T2$	10	40	1	$[0, \infty]$
$T3$	10	30	1.5	[10, 80]

taking the VRPHTO constraints and features into consideration. According to [54], the complexity of the *extended split* is $O(k * n^2)$, where k denotes the number of vehicle types, n gives the number of customers.

In particular, given a permutation-based chromosome S, the *extended split* first checks all the possible insertions of trip delimiters, and preserves the feasible insertions (vehicle routes) which satisfy the time window constraints of customers. Further, the vehicle type with the least cost is then assigned to the obtained routes by considering both the demands of customers and the time window of the vehicle. Finally, the feasible vehicle routes with minimal costs then form the output routing solution.

Further, Fig. 4 presents an illustrative example of the *extended split* procedure. Particularly, Fig. 4(a) gives one chromosome $S = \{1, 2, 3, 4, 5\}$, and the corresponding routing problem with detailed customer information, such as customer demand, customer time window, travel cost, etc. Table I summarizes the vehicle types considered in this example. $T1$ and $T2$ denote vehicles with regular drivers, while $T3$ represents the vehicle with occasional drivers. Further, Fig. 4(b) provides all the feasible routes obtained by *split*. The number and label "T*" associated with each route denote the corresponding cost and assigned vehicle type, respectively. Next, since the route $\{0, 4, 5, 0\}$ is served by an occasional driver of type $T3$, which does not need to go back to the depot when the service at customer 5 has been completed, the last segment $\{5, 0\}$ is denoted as a dashed line. Finally, by evaluating the cost of all the routes, the output of the *split* procedure is the minimum partition $R = \{0, 1, 0, 2, 3, 0, 4, 5, 0\}$ with a total cost $c(R) = 297.5$, which is depicted in Fig. 4(c).

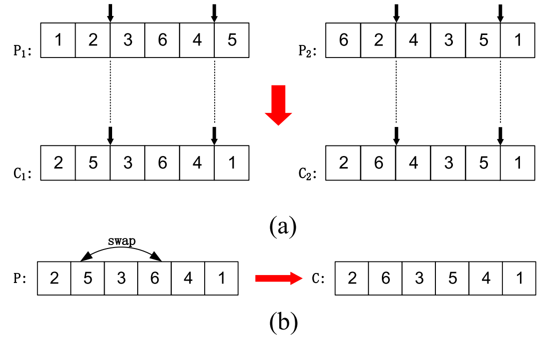


Fig. 5. Examples of (a) OX and (b) SW.

D. Routing Information Exchange Across Instances in Evolutionary Multitasking

Routing information exchange process is contained in the solution reproduction procedure in the unified solution space. In EMA, the skill factor τ_p is associated with chromosome p to indicate the VRPHTO task on which p has the best performance among multiple VRPHTOs. As different chromosomes may have various skill factors, each chromosome contains valuable routing information for a particular task. According to [17], the information exchanging across tasks happens when the genetic crossover of solutions possessing different skill factors is performed. Therefore, the design of crossover and mutation in the unified solution space for routing information exchange across VRPHTOs is vital for facilitating automatic knowledge transfer across tasks, which could enhance the problem-solving process on each task.

In particular, the popular order crossover (OX) [55] and swap mutation (SW) [56] for vehicle routing are considered here for chromosome reproduction in EMA.[3] Given two parents p_1 and p_2, OX first randomly chooses two cut-points from p_1 (or p_2). The segment between this two points is then directly copied into the child c_1 (or c_2). Next, it deletes the elements of the segment in p_2 (or p_1), and inserts the rest elements of p_2 (or p_1) into c_1 (or c_2) without changing the element orders.

By applying this process on both of the parents, we can arrive at two offsprings. On the other hand, SW randomly chooses two elements from a chromosome and exchanges their values. An illustrative example of the OX and SW operator is given in Fig. 5.[4]

Moreover, to control the frequency of the information interaction, a parameter called *rmp* is defined [17]. As depicted in Algorithm 1, line 8, the *rmp* controls the frequency of two

[3]Without loss of generality, other crossover operators for VRP can also be applied.

[4]The crossover and mutation are performed in a standard manner as described in [55] and [56]. However, it is worth noting that these operators are performed in the unified solution representation based on the *skill factors* of solutions.

parents possessing different skill factors to undergo crossover. Therefore, a high value of *rmp* allows more information to be shared across two routing problem domains frequently via crossover. It is worth noting that a proper knowledge transfer across related problems is essential to enhance the corresponding problem solving process.

E. Chromosome Evaluation in Evolutionary Multitasking

The chromosome evaluation procedure determines the survival of solutions in the multitasking scenario. Usually, the elite chromosomes are preserved for the next generation to speed up the performance of EA [57], [58]. For VRPHTO, the comparison can be done easily in the single task scenario, where chromosomes with lower routing cost are preferred. However, in the multitasking environment, chromosomes for different VRPHTOs cannot be compared directly.

Therefore, the *scalar fitness* (as formalized in Definition 3) is considered here for *chromosome evaluation*. In particular, for two chromosome p_a and p_b, we consider p_a outperforms p_b if φ_{p_a} is less than φ_{p_b}.

V. Experimental Study

In this section, empirical studies are conducted to verify the efficacy of the proposed EMA for solving VRPHTO in a multitasking mode. Specifically, we first introduce how the VRPHTO instances can be generated based on the existing commonly used HVRPTW benchmarks. Subsequently, the experimental configurations, as well as the obtained results are presented and discussed.

A. Benchmark Generation

1) Generation of VRPHTO Benchmark: The generation of the proposed VRPHTO instances is based on the widely used HVPRTW benchmarks proposed by Liu and Shen [18]. The properties of the HVRPTW benchmarks are summarized in Table II, in which "VT," "*Capacity*," and "F_c" denote the type, the capacity, and the fixed cost of vehicles employed, respectively. As given in Table II, there are totally six categories of routing instances, namely, C1A, C2A, R1A, R2A, RC1A, and R2A, in the HVRPTW benchmark set. Different types of regular vehicles are employed in these six HVRPTW categories. Further, a time window or scheduling horizon is given at the central depot for each set. R1A, C1A, and RC1A have a short time window that allows only a few customers to be served, while R2A, C2A, and RC2A hold a long time window that could include more customers to be served. The customer distributions in these six categories are also different. In particular, R1A and R2A have random customer distributions, while C1A and C2A have clustered customer distributions. The fusion of random distributed customers and clustered customers are contained in RC1A and RC2A. Finally, there are totally 56 routing instances in this HVRPTW set.

For the VRPHTO instances, the occasional drivers are derived from the regular drivers in HVRPTW by changing the fixed cost (cost for vehicle maintenance and drivers' salary) and variable cost (cost for petrol consumption and drivers' reward) of vehicles. Usually, since courier companies do not need to maintain the occasional vehicles, occasional drivers always have lower fixed cost but higher variable cost, than regular drivers. Taking this cue, for a given HVRPTW instance with n types of regular drivers (denoted as regular types), the generation of VRPHTO instance is designed as follows.

1) Denote the n regular types as $\mathrm{RT}_1, \mathrm{RT}_2, \ldots, \mathrm{RT}_n$ with the capacity arranged in an ascending order.
2) Randomly select $\lfloor (n/2) \rfloor$ regular types to generate occasional types. If an occasional type is created from RT_i, it has the same capacity with RT_i. The corresponding fixed

TABLE II
Properties of the HVRPTW Benchmark ("VT," "Capacity," "F_c," "V_c," and "VTW" Represent the Vehicle Type, the Capacity, the Fixed Cost, the Variable Cost, and the Vehicle Time Window of the Vehicles, Respectively)

R1A (12 problems)			R2A (11 problems)		
VT	Capacity	F_c	VT	Capacity	F_c
A	30	50	A	300	450
B	50	80	B	400	700
C	80	140	C	600	1200
D	120	250	D	1000	2500
E	200	500	/	/	/
Time window of the depot		[0, 230]	Time window of the depot		[0, 1000]
C1A (9 problems)			C2A (8 problems)		
VT	Capacity	F_c	VT	Capacity	F_c
A	100	300	A	400	1000
B	200	800	B	500	1400
C	300	1350	C	600	2000
/	/	/	D	700	2700
Time window of the depot		[0, 1236]	Time window of the depot		[0, 3390]
RC1A (8 problems)			RC2A (8 problems)		
VT	Capacity	F_c	VT	Capacity	F_c
A	40	60	A	100	150
B	80	150	B	200	350
C	150	300	C	300	550
D	200	450	D	400	800
/	/	/	E	500	1100
/	/	/	D	1000	2500
Time window of the depot		[0, 240]	Time window of the depot		[0, 960]

TABLE III
Properties of the VRPHTO Benchmark ("VT," "Capacity," "F_c," "V_c," and "VTW" Represent the Vehicle Type, the Capacity, the Fixed Cost, the Variable Cost, and the Vehicle Time Window of the Vehicles, Respectively)

R1A (12 problems)						R2A (11 problems)					
VT	Capacity	F_c	V_c	VTW		VT	Capacity	F_c	V_c	VTW	
				V_e	V_l					V_e	V_l
A	30	50	1.0	0	230	A	300	450	1.0	0	1000
B	50	80	1.0	0	230	B	400	700	1.0	0	1000
C	80	140	1.0	0	230	C	600	1200	1.0	0	1000
D	120	250	1.0	0	230	D	1000	2500	1.0	0	1000
E	200	500	1.0	0	230	A1	300	225	1.5	186	596
B1	50	65	1.5	80	191	B1	400	575	1.5	554	917
C1	80	110	1.5	29	113	/	/	/	/	/	/
Time window of the depot				[0, 230]		Time window of the depot				[0, 1000]	
C1A (9 problems)						C2A (8 problems)					
VT	Capacity	F_c	V_c	VTW		VT	Capacity	F_c	V_c	VTW	
				V_e	V_l					V_e	V_l
A	100	300	1.0	0	1236	A	400	1000	1.0	0	3390
B	200	800	1.0	0	1236	B	500	1400	1.5	0	3390
C	300	1350	1.0	0	1236	C	600	2000	1.0	0	3390
A1	100	150	1.5	434	1014	D	700	2700	1.0	0	3390
/	/	/	/	/	/	A1	400	500	1.5	600	1600
/	/	/	/	/	/	C1	600	1667	1.5	103	872
Time window of the depot				[0, 1236]		Time window of the depot				[0, 3390]	
RC1A (8 problems)						RC2A (8 problems)					
VT	Capacity	F_c	V_c	VTW		VT	Capacity	F_c	V_c	VTW	
				V_e	V_l					V_e	V_l
A	40	60	1.0	0	240	A	100	150	1.0	0	960
B	80	150	1.0	0	240	B	200	350	1.0	0	960
C	150	300	1.0	0	240	C	300	550	1.0	0	960
D	200	450	1.0	0	240	D	400	800	1.0	0	960
B1	80	105	1.5	42	159	E	500	1100	1.0	0	960
C1	150	220	1.5	104	202	F	1000	2500	1.0	0	960
/	/	/	/	/	/	A1	100	75	1.5	328	775
/	/	/	/	/	/	B1	200	250	1.5	44	413
/	/	/	/	/	/	C1	300	420	1.5	472	875
Time window of the depot				[0, 240]		Time window of the depot				[0, 960]	

TABLE IV
SUMMARY OF THE MULTITASKING VRPHTO PROBLEMS INVESTIGATED

Multitasking Problems	$P1$	$P2$	$P3$	$P4$	$P5$	$P6$	$P7$	$P8$	$P9$	$P10$	$P11$	$P12$	$P13$	$P14$
VRPHTO Instances	$R101A$ + $R102A$	$R103A$ + $R104A$	$R105A$ + $R106A$	$R107A$ + $R108A$	$R109A$ + $R110A$	$R111A$ + $R112A$	$C101A$ + $C102A$	$C103A$ + $C104A$	$C105A$ + $C106A$	$C107A$ + $C108A$	$RC101A$ + $RC102A$	$RC103A$ + $RC104A$	$RC105A$ + $RC106A$	$RC107A$ + $RC108A$
Multitasking Problems	$P15$	$P16$	$P17$	$P18$	$P19$	$P20$	$P21$	$P22$	$P23$	$P24$	$P25$	$P26$	$P27$	
VRPHTO Instances	$R201A$ + $R202A$	$R203A$ + $R204A$	$R205A$ + $R206A$	$R207A$ + $R208A$	$R209A$ + $R210A$	$C201A$ + $C202A$	$C203A$ + $C204A$	$C205A$ + $C206A$	$C207A$ + $C208A$	$RC201A$ + $RC202A$	$RC203A$ + $RC204A$	$RC205A$ + $RC206A$	$RC207A$ + $RC208A$	

TABLE V
SUMMARY OF THE CONFIGURATION DETAILS OF EVOLUTIONARY OPERATOR AND PARAMETER IN BOTH SEA AND EMA

Item	Configurations
Population size	$N_p = 30$
Total task evaluations	$TE_{total} = 30000$
Restart times	$N_{re} = 10$
Task evaluations per Restart	$TE = 3000$
Elite chromosomes preserved per Restart	$N_{be} = 2$ in *EMA*, $N_{be} = 1$ in *SEA*
Independent run times	$Runs = 20$
Crossover	OX (Order Crossover) [55]
Mutation	Swap mutation [56]
Local search operators	Exchange [59], Or-opt [59], 2-opt move [59]
Probability of routing information interaction	$rmp = 0.3$
Local search probability	$P_{ls} = 0.1$

cost is set to be lower than RT$_i$ but higher than RT$_{i-1}$. Further, as the variable cost of occasional drivers should be higher than regular drivers, we set it as 1.5 and 1.0 for occasional and regular driver, respectively.

3) In addition, the time window of occasional driver is then randomly generated within the length of $[(L - E/\alpha)$, $(L - E/\beta)]$ that lies in the time window of the depot, that is, $[E, L]$. For simplicity, α and β are configured as 3 and 2, respectively, in the present context. The time window for regular drives are set to be the same as the depot.

The detailed vehicle properties of the generated VRPHTO benchmark are summarized in Table III. In the table, "VT," "Capacity," "F_c," "V_c," and "*VTW*" represent the type, the capacity, the fixed cost, the variable cost, and the vehicle time window of the vehicles, respectively. Particularly, in column "VT," symbols (e.g., "*A*" and "*B*") denote the regular vehicles, while symbols with suffix "1" (e.g., "*A*1" and "*B*1") gives the occasional vehicles that are generated based on the regular vehicles labeled by the corresponding symbol. For instance, the occasional vehicle "*A*1" is generated from regular vehicle "*A*."

2) Generation of Multitasking VRPHTO Benchmark: For conducting multitasking with VRPHTO, in this article, we consider to solve two tasks in a single run. In particular, based on the newly generated VRPHTO instances, we further built 27 multitasking VRPHTO problem sets by pairing the instances in order, within the same VRPHTO category. For example, for the 12 VRPHTOs, that is, from R101A to R112A, belonging to the R1A category, the VRPHTO pairs, that is, R101A and R102A, R103A and R104A, R105A and R106A, R107A and R108A, R109A and R110A, and R111A and R112A will be six benchmarks for multitasking. As can be observed, the paired VRPHTO instances belong to the same category, they mainly differ in the time windows of customers. Other properties of the VRPHTOs are as discussed in Section V-A1 and summarized in Table III. Further, please note that, since C1A and R2A have 9 and 11 instances, respectively, the last instance in these two categories has no counterpart to be paired with, we simply discarded C109A and R211A in the multitasking VRPHTO benchmark. Thus, we have totally 27 pairs of VRPHTO instances which are summarized in Table IV.

B. Experimental Setup

To evaluate the efficacy of the proposed EMA, a state-of-the-art single task evolutionary algorithm [54] (labeled as SEA hereafter) is considered here as the baseline solver.[5] For fair comparison, both SEA and the proposed EMA are configured with the same evolutionary operators as well as parameters. The only difference between SEA and EMA is the proposed multitasking mechanism lies in the latter. We implement SEA and EMA in C++ using the computer with a 3.2 GHz Intel Corei5 processor and 8 gigabyte RAM. According to [17], detailed configuration of evolutionary operator and parameter in both SEA and EMA are given in Table V.

C. Results and Discussion

1) HVRPTW Versus VRPHTO: First of all, to show the benefits of "crowdshipping" and "sharing economy," we investigate the savings of travel cost obtained by the proposed VRPHTO, against the HVRPTW which has no occasional drivers for serving customers. In particular, we compare the empirical results (see Table VI) obtained by the SEA on all the 56 HVRPTW and VRPHTO instances, across 20 independent runs. In Table VI, "Vehicle Mix" denotes the set of vehicles used to serve the customers. In particular, the symbol, e.g., "*A*," "*B*," etc., gives the vehicle type, and the superscript on each symbol denotes the number of vehicles employed for service. For instance, "$A^1B^{10}C^{12}$" means there are 1, 12, and 12 vehicle of type *A*, type *B*, and type *C*, respectively, have been used to serve all the customers. Further, "TC" and "$\#_{\mathrm{OD}}$" represent the total cost of the instance and the number of occasional drivers employed, respectively. "Cost saving (CS)" gives how much percentage of cost have been saved by the employing occasional drivers, which is defined as

$$\text{Cost Saving(CS)} = \frac{\mathrm{TC_{NOD}} - \mathrm{TC_{OD}}}{\mathrm{TC_{NOD}}} \cdot 100\% \tag{11}$$

where "$\mathrm{TC_{NOD}}$" and "$\mathrm{TC_{OD}}$" represent total cost obtained by SEA on HVRPTW and VRPHTO instances, respectively.

[5]The considered single task evolutionary algorithm is one of the state-of-the-art VRP solvers, as it is able to find the best known solutions (available at http://neo.lcc.uma.es/vrp/known-best-results/) on the common VRP instances reported in [54].

TABLE VI
EMPIRICAL RESULTS OF THE SEA ON ALL THE 56 HVRPTW AND VRPHTO INSTANCES ACROSS 20 INDEPENDENT RUNS. ["VEHICLE MIX," "TC," "N_{OD}," AND "COST SAVING (CS)" REPRESENT THE COMPOSITION OF VEHICLES, THE TOTAL COST, THE NUMBER OF OCCASIONAL DRIVERS EMPLOYED AND THE PROPORTION OF COST SAVED, RESPECTIVELY]

Instances	HVRPTW		VRPHTO				
	Vehicle Mix	TC	Vehicle Mix	$\#_{OD}$	TC	Cost Saving(CS)	AvCS
$R101A$	$A^1B^{10}C^{12}$	4317.52	$B^{12}B1^2C^{10}$	2	**4293.64**	0.55%	3.78%
$R102A$	$A^1B^5C^{15}$	4173.84	$B^{10}B1^5C^8C1^1$	6	**4086.61**	2.09%	
$R103A$	B^4C^{16}	4041.70	$B^1B1^8C^{12}C1^1$	9	**3914.14**	3.16%	
$R104A$	B^4C^{16}	3952.52	$B^1B1^{11}C^9C1^2$	13	**3737.64**	5.44%	
$R105A$	B^4C^{16}	4148.82	$B^6B1^5C^{11}C1^1$	6	**4095.61**	1.28%	
$R106A$	B^4C^{16}	4053.45	$B^4B1^8C^{11}$	8	**3954.07**	2.45%	
$R107A$	$B^3C^{15}D^1$	3988.24	$B^1B1^{11}C^{11}$	11	**3839.70**	3.72%	
$R108A$	$B^3C^{15}D^1$	3931.74	$B^1B1^{13}C^8C1^2$	15	**3720.39**	5.38%	
$R109A$	B^4C^{16}	4039.40	$B^3B1^8C^{10}C1^2$	10	**3903.38**	3.37%	
$R110A$	B^1C^{18}	3973.50	$B^1B1^{13}C^8C1^2$	15	**3756.43**	5.46%	
$R111A$	B^4C^{16}	3994.84	$B1^{12}C^{10}C1^1$	13	**3824.33**	4.27%	
$R112A$	$C^{17}D^1$	3930.18	$B1^{22}C^4C1^1$	23	**3606.96**	8.22%	
$C101A$	A^{19}	7093.45	$A^{13}A1^6$	6	**6318.93**	10.92%	17.81%
$C102A$	A^{19}	7080.17	$A^{11}A1^8$	8	**5978.32**	15.56%	
$C103A$	A^{19}	7079.21	A^8A1^{11}	11	**5490.09**	22.45%	
$C104A$	A^{19}	7075.06	A^5A1^{14}	14	**5039.83**	28.77%	
$C105A$	A^{19}	7093.45	$A^{12}A1^7$	7	**6260.83**	11.74%	
$C106A$	A^{19}	7083.87	$A^{11}A1^8$	8	**6114.70**	13.68%	
$C107A$	A^{19}	7084.60	$A^{10}A1^9$	9	**5812.59**	17.95%	
$C108A$	A^{19}	7079.66	$A^{10}A1^9$	9	**5793.08**	18.17%	
$C109A$	A^{19}	7077.30	A^9A1^{10}	10	**5585.05**	21.09%	
$RC101A$	$A^3B^{13}C^4$	5173.47	$A^3B^{11}B1^8C^1$	8	**5050.14**	2.38%	3.79%
$RC102A$	$A^6B^6C^7$	5018.83	$A^3B^8B1^7C^3$	7	**4873.02**	2.91%	
$RC103A$	$A^2B^6C^8$	4850.20	$A^2B^5B1^7C^5$	7	**4725.38**	2.57%	
$RC104A$	$A^2B^4C^9$	4770.23	$A^2B^1B1^9C^6$	9	**4525.66**	5.13%	
$RC105A$	$A^6B^8C^6$	5103.69	$A^5B^8B1^6C^3$	6	**4922.05**	3.56%	
$RC106A$	$A^5B^8C^6$	4955.48	$A^2B^5B1^9C^4$	9	**4725.76**	4.64%	
$RC107A$	$A^4B^5C^8$	4825.60	$A^1B^6B1^{10}C^3$	10	**4612.61**	4.41%	
$RC108A$	$A^4B^2C^8D^1$	4724.79	$A^2B^1B1^9C^6$	9	**4500.25**	4.75%	
$R201A$	A^5	3446.78	A^4A1^1	1	3446.78	0.00%	12.23%
$R202A$	A^5	3314.93	A^4A1^1	1	**3174.99**	4.22%	
$R203A$	A^5	3141.09	A^3A1^2	2	**2861.57**	8.90%	
$R204A$	A^5	3018.14	A^1A1^4	4	**2502.35**	17.09%	
$R205A$	A^5	3218.97	A^3A1^2	2	**2915.37**	9.43%	
$R206A$	A^5	3153.52	A^2A1^3	3	**2736.42**	13.23%	
$R207A$	A^5	3079.45	A^2A1^3	3	**2596.69**	15.68%	
$R208A$	A^5	2997.24	A^1A1^4	4	**2334.88**	22.10%	
$R209A$	A^5	3125.77	A^2A1^3	3	**2808.98**	10.13%	
$R210A$	A^5	3180.09	A^2A1^3	3	**2735.90**	13.97%	
$R211A$	A^5	3020.58	A^1A1^4	4	**2422.87**	19.79%	
$C201A$	A^5	5695.02	A^4A1^1	1	**5227.58**	8.21%	8.31%
$C202A$	A^5	5685.24	A^4A1^1	1	**5212.53**	8.31%	
$C203A$	A^5	5681.55	A^4A1^1	1	**5210.45**	8.29%	
$C204A$	A^5	5677.66	A^4A1^1	1	**5204.86**	8.33%	
$C205A$	A^5	5691.36	A^4A1^1	1	**5217.25**	8.33%	
$C206A$	A^5	5689.32	A^4A1^1	1	**5215.22**	8.33%	
$C207A$	A^5	5687.35	A^4A1^1	1	**5215.01**	8.31%	
$C208A$	A^5	5686.50	A^4A1^1	1	**5209.79**	8.38%	
$RC201A$	$A^{12}B^3$	4381.05	$A^6A1^{10}B1^1$	11	**3663.73**	16.37%	22.95%
$RC202A$	$A^9B^3C^1$	4244.63	A^6A1^{12}	12	**3450.46**	18.71%	
$RC203A$	$A^5B^5C^1$	4175.68	$A^3A1^{11}B1^2$	13	**3278.12**	21.49%	
$RC204A$	$A^4B^1C^4$	4087.11	$A^1A1^{13}B1^2$	15	**3029.99**	25.86%	
$RC205A$	A^8B^5	4293.35	$A^5A1^{11}B1^1$	12	**3506.08**	18.34%	
$RC206A$	A^6B^6	4255.27	$A^2A1^{12}B1^2$	14	**3300.44**	22.44%	
$RC207A$	A^6B^6	4188.89	$A^1A1^{15}B1^1$	16	**3054.51**	27.08%	
$RC208A$	$A^4B^4C^2$	4075.04	$A1^{18}$	18	**2716.96**	33.33%	

TABLE VII
NUMERICAL RESULTS OF EMA AND SEA ("*Ave.Cost*," "*B.Cost*," "*Std.Dev*" DENOTE AVERAGED COST, THE BEST COST AND STANDARD DEVIATION ACROSS 20 INDEPENDENT RUNS, RESPECTIVELY. THE SUPERIOR PERFORMANCE IS HIGHLIGHTED IN BOLD. "≈," "+" AND "−" DENOTE EMA STATISTICALLY SIGNIFICANT SIMILAR, BETTER, AND WORSE THAN SEA, RESPECTIVELY)

Multitasking Problems	VRPHTO Instance	EMA			SEA		
		Ave.Cost	*B.Cost*	*Std.Dev*	*Ave.Cost*	*B.Cost*	*Std.Dev*
$P1$	$R101A$	**4313.16** ≈	4293.64	10.10	4316.88	4293.64	12.30
	$R102A$	**4100.83** ≈	**4081.19**	13.50	4104.31	4086.61	11.80
$P2$	$R103A$	**3932.14** ≈	**3908.81**	11.20	3932.83	3914.14	11.30
	$R104A$	**3752.29** ≈	3738.94	11.80	3752.61	**3737.64**	11.50
$P3$	$R105A$	**4112.46** ≈	4101.23	11.70	4116.09	**4095.61**	10.90
	$R106A$	3984.45 ≈	3955.73	15.70	**3983.86**	**3954.07**	13.70
$P4$	$R107A$	**3856.19** ≈	**3835.71**	12.00	3858.50	3839.70	12.40
	$R108A$	**3736.52** ≈	**3714.18**	10.90	3741.90	3720.39	12.10
$P5$	$R109A$	3917.26 ≈	**3899.64**	11.50	**3917.05**	3903.38	9.40
	$R110A$	3787.31 ≈	**3755.89**	15.30	**3783.83**	3756.43	15.60
$P6$	$R111A$	**3841.63** ≈	**3821.35**	10.50	3844.62	3824.33	14.90
	$R112A$	**3625.03** ≈	**3593.68**	14.30	3629.03	3606.96	11.70
$P7$	$C101A$	6385.94 ≈	6326.96	40.30	**6383.33**	**6318.93**	41.80
	$C102A$	6037.56 ≈	5978.32	32.10	**6036.79**	5978.32	31.90
$P8$	$C103A$	5543.79 ≈	**5488.79**	47.90	**5539.57**	5490.09	37.80
	$C104A$	**5098.19** ≈	**5034.15**	36.90	5108.92	5039.83	29.70
$P9$	$C105A$	6326.39 ≈	6311.90	13.00	**6319.78**	**6260.83**	19.20
	$C106A$	**6168.39** ≈	**6075.41**	30.30	6188.98	6114.70	36.10
$P10$	$C107A$	**5890.59** +	5812.59	51.10	5914.91	5812.59	32.30
	$C108A$	**5831.13** ≈	5793.08	44.90	5859.83	5793.08	54.70
$P11$	$RC101A$	**5067.47** ≈	**5018.65**	21.60	5072.91	5050.14	13.30
	$RC102A$	4930.70 ≈	4890.60	23.40	**4917.35**	**4873.02**	29.70
$P12$	$RC103A$	**4781.20** ≈	4742.57	25.20	4782.48	**4725.38**	18.50
	$RC104A$	**4553.24** ≈	4527.84	12.90	4560.20	**4525.66**	20.40
$P13$	$RC105A$	4967.39 ≈	4935.19	16.50	**4959.21**	**4922.05**	15.60
	$RC106A$	**4771.22** ≈	4732.71	25.90	4778.33	**4725.76**	27.40
$P14$	$RC107A$	4646.13 ≈	**4594.95**	25.20	**4644.14**	4612.61	14.60
	$RC108A$	**4525.04** ≈	**4492.32**	20.60	4529.72	4500.25	17.70
$P15$	$C201A$	5227.58 ≈	5227.58	0.00	5227.58	5227.58	0.00
	$C202A$	5212.53 ≈	5212.53	0.00	5212.53	5212.53	0.00
$P16$	$C203A$	**5210.46** ≈	5210.45	0.00	5210.47	5210.45	0.10
	$C204A$	**5204.95** ≈	5204.86	0.20	5205.28	5204.86	0.60
$P17$	$C205A$	**5217.25** ≈	5217.25	0.00	5217.26	5217.25	0.00
	$C206A$	5215.22 ≈	5215.22	0.00	5215.22	5215.22	0.00
$P18$	$C207A$	**5214.48** ≈	**5212.40**	1.00	5215.01	5215.01	0.00
	$C208A$	**5212.14** ≈	5209.79	0.80	5212.27	5209.79	0.60
$P19$	$R201A$	**3468.08** ≈	**3357.27**	45.90	3474.80	3446.78	19.70
	$R202A$	**3251.11** ≈	**3158.94**	64.80	3260.66	3174.99	59.30
$P20$	$R203A$	**2949.74** ≈	2864.29	63.10	2966.87	**2861.57**	55.50
	$R204A$	**2505.06** ≈	**2412.65**	32.80	2519.87	2502.35	7.50
$P21$	$R205A$	3060.44 ≈	2930.11	52.40	**3044.59**	**2915.37**	57.50
	$R206A$	**2846.49** ≈	**2708.62**	38.00	2850.66	2736.42	31.40
$P22$	$R207A$	2698.41 ≈	**2591.65**	73.80	**2696.86**	2596.69	70.30
	$R208A$	2413.24 −	**2331.03**	62.70	**2355.87**	2334.88	29.30
$P23$	$R209A$	**2801.59** +	**2615.27**	65.00	2833.46	2808.98	12.60
	$R210A$	2877.76 ≈	2843.57	13.50	**2872.50**	**2735.90**	45.10
$P24$	$RC201A$	**3728.71** ≈	3708.31	15.20	3730.88	**3663.73**	22.90
	$RC202A$	**3460.76** ≈	3454.03	4.70	3463.36	**3450.46**	8.00
$P25$	$RC203A$	**3307.27** ≈	**3270.96**	21.40	3311.12	3278.12	18.30
	$RC204A$	**3042.94** ≈	3030.65	10.70	3045.65	**3029.99**	15.20
$P26$	$RC205A$	**3515.90** ≈	**3470.85**	13.40	3515.99	3506.08	10.20
	$RC206A$	**3346.81** ≈	3317.16	21.50	3353.06	**3300.44**	24.40
$P27$	$RC207A$	**3067.49** ≈	**3050.27**	8.10	3069.79	3054.51	12.40
	$RC208A$	**2722.10** ≈	2716.96	2.90	2722.11	2716.96	2.20

Finally, "*AvCS*" is the averaged cost saving of all the routing instances in one category.

In general, it can be observed from Table VI that, lower cost can be obtained by employing occasional drivers with the time window in VRPHTO when compared to HVRPTW. Further, we observe that the following two factors have a great impact on the percentage of cost saving in VRPHTO, which are as follows.

1) The fixed cost of the occasional drivers.
2) The amount of the occasional drivers employed.

In VRPHTO, the cost reduction is mainly made by the saving of fixed cost when regular drivers are replaced by occasional drivers to serve customers. It can be observed from Tables III and VI that the lower fixed cost of occasional vehicle is, the more cost saving is made. For instance, as the difference between the fixed costs of the occasional and regular vehicle in R1A is less than 50, and is more than 300 in C2A, there is 8% averaged cost saving *AvCS* in C2A with only 1 occasional vehicle employed, which is much higher than the saving made in R1A that is only 3.78% with much more occasional vehicle employed.

Further, with the same fixed cost, the employment of more occasional vehicles would lead to more cost savings. This tendency can be observed clearly in Table VI on class C1A and R2A, where the value of CS increases with the number of occasional drivers. For example, there are 6 occasional vehicles employed in C101A which leads to 10.92% CS, while C104A achieved 28.77% CS with 8 more occasional vehicle employed.

2) EMA Versus SEA: To further verify the efficacy of the proposed EMA for evolutionary multitasking with VRPHTO, Table VII tabulates the results obtained by the proposed EMA and the single solver SEA on all the constructed 27 multitasking VRPHTO pairs across 20 independent runs. Note that the SEA solves the VRPHTO instances in each multitasking problem separately, while our proposed EMA, on the other hand, optimizes the two VRPHTO instances simultaneously. In the table, *B.Cost*, *Ave.Cost*, and *Std.Dev* denote the obtained best cost, averaged cost, and standard deviation, respectively. The superior performance in Table VII is highlighted in bold. Finally, in order to obtain the statistical comparison, the Wilcoxon rank-sum test with 95% confidence level has been conducted on the experimental results.

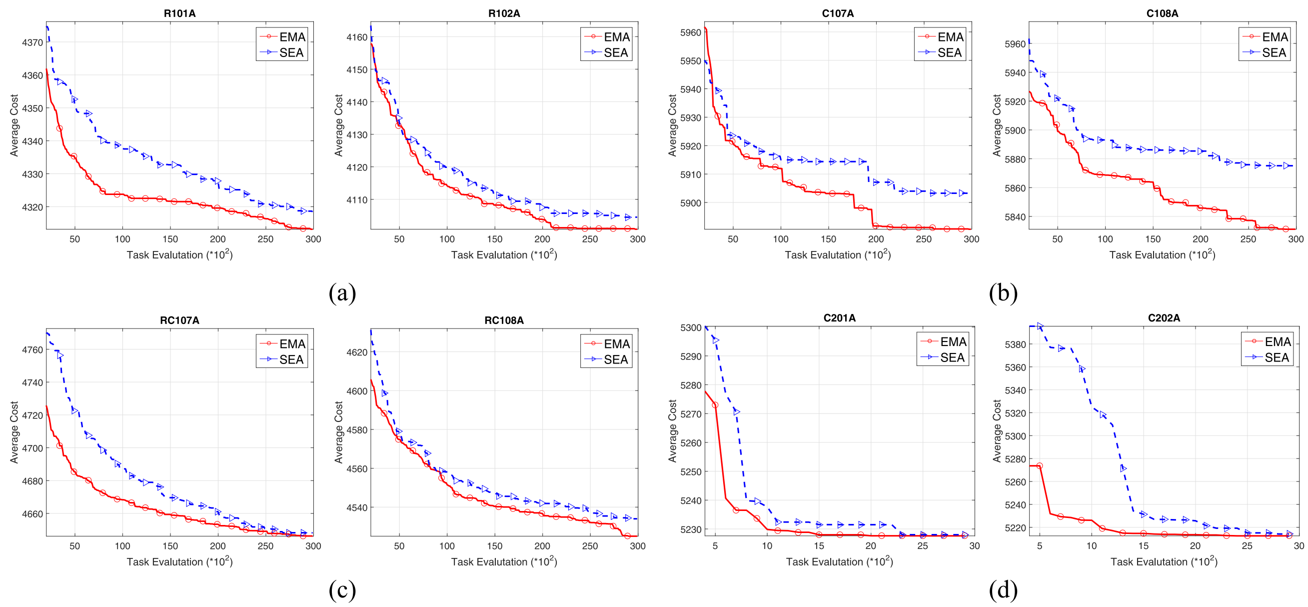


Fig. 6. Convergence traces of EMA versus the SEA on representative multitasking VRPHTOs. *Y*-axis: Averaged travel cost; *X*-axis: Task evaluation. (a) Convergence traces of *P*1. (b) Convergence traces of *P*10. (c) Convergence traces of *P*12. (d) Convergence traces of *P*15.

As can be observed in Table VII, EMA achieved superior or competitive performance against SEA on 37 out of totally 54 VRPHTO instances, in terms of *Ave.Cost*. With respect to *B.Cost*, EMA has found better best cost solutions on 25 VRPHTO instances against SEA. Further, it is noted that on class R2A, C2A, and RC2A which have long scheduling horizons, EMA obtained better or competitive average cost on 22 out of 26 instances. Particularly, when the customer distributions become more structured in routing categories like C2A and RC2A, EMA achieved superior average cost on all of the instances.

Subsequently, to access the efficiency of the proposed EMA, the representative search convergence traces of EMA and SEA on the multitasking VRPHTOs are presented in Fig. 6. The *y*-axis of the figures denotes the averaged travel cost obtained, while *x*-axis gives the respective computational effort incurred in terms of the number of task evaluations made so far. As depicted, generally, the proposed EMA converges faster than SEA on most of the VRPHTO instances. In particular, on R101A and R102A, EMA takes only about 20 000 number of task evaluations to arrives at the solutions obtained by SEA using 30 000 task evaluations. It is worth noting that on the instances of C2A, although EMA and SEA obtained the same solution quality (as presented in Table VII), EMA converges faster than SEA on all the instances, e.g., C201A and C202A in Fig. 6(d). Further, in Fig. 7, we present the best transferred solutions across tasks and the sampled existing solutions of the task in EMA, on representative VRPHTOs. As can be observed, due to the proposed multitasking, the transfer of solutions across tasks happens along the whole evolutionary search process in EMA. It is also straightforward to see that, the superior convergence speed of EMA is due to the transfer of good solutions, which possess lower objective values than the existing sampled solutions (see the solutions highlighted in the circle), across tasks.

Next, in Table VIII, we further tabulated speedup made by EMA over SEA in terms of "task evaluation" at different stages along the search on 12 representative instances. Particularly, we define the SpeedUp as

$$\text{SpeedUp} = \frac{SEA^{i}_{\text{Task Evalution}}}{EMA^{i}_{\text{Task Evalution}}} \tag{12}$$

where $A^{i}_{\text{Task Evalution}}$ denotes the number of task evaluations incurred by algorithm A to arrive at the averaged cost obtained at stage i.

In particular, here we evenly divide the evolution traces obtained by SEA and EMA and on a VRPHTO instance into eight stages according to the fitness obtained at the beginning and the end of the optimization search. Note that, a *SpeedUp* value which is larger than 1 means EMA converges faster than SEA. As shown in Table VIII, EMA achieves faster convergence at most stages of the instances. Particularly, on instances such as R112A, RC108A, C202A, etc., EMA demonstrated three times speedup over SEA, which saved three times the number of task evaluations to arrive at the same routing cost level achieved by SEA.

Finally, to provide deeper insights on the efficacy obtained by EMA, we further investigate the problem properties of the multitasking VRPHTO pairs. In particular, as customer distributions of the VRPHTO instances in the same category are identical, we focus on the customer time windows of two VRPHTOs, that is, *CTW*. In Fig. 8, we compared the customer time windows of two representative VRPHTOs pairs. The *X*-axis of the figures denotes the customer ID, while *Y*–axis gives the customer time window. As can be observed, in Fig. 8(a) (i.e., VRPHTO P10), the time windows of C107A have a great overlap with that in C108A. Further, in Fig. 8(b) (i.e., VRPHTO P12), the corresponding two VRPHTO instances (RC107A and RC108A) have relatively less overlap in customer time windows. Look back to the convergence graph

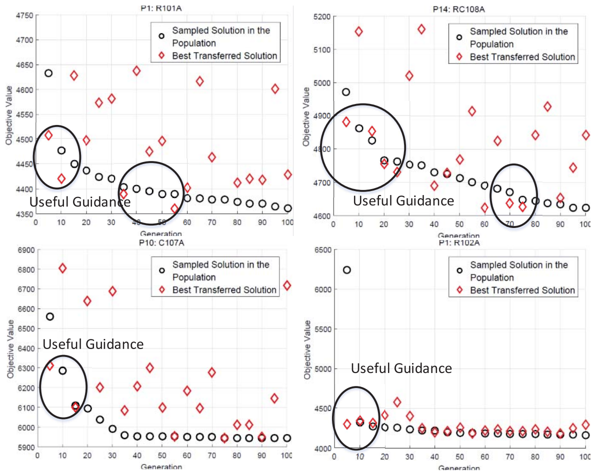


Fig. 7. Illustration of the best transferred solutions in EMA across task on representative VRPHTOs.

TABLE VIII
SPEEDUP BY EMA OVER SEA IN TERMS OF "TASK EVALUATIONS" AT THE DIFFERENT STAGES OF THE SEARCH ON REPRESENTATIVE VRPHTO INSTANCES

Instances	Speed up (Fitness)							
	Stage1	Stage2	Stage3	Stage4	Stage5	Stage6	Stage7	Stage8
$R111A$	1.40 (3975.50)	1.29 (3959.14)	1.13 (3942.78)	1.55 (3926.42)	1.30 (3910.06)	1.56 (3893.70)	1.60 (3877.34)	1.92 (3860.98)
$R112A$	1.64 (3772.40)	1.86 (3754.48)	2.09 (3736.56)	1.77 (3718.64)	1.72 (3700.72)	1.63 (3682.79)	1.48 (3664.87)	2.00 (3646.95)
$C107A$	1.05 (6121.72)	1.33 (6095.87)	1.17 (6070.02)	0.84 (6044.17)	0.97 (6018.31)	0.95 (5992.46)	0.89 (5966.61)	1.08 (5940.76)
$C108A$	1.56 (6144.28)	1.66 (6108.72)	1.94 (6073.16)	1.57 (6037.61)	2.13 (6002.05)	2.25 (5966.50)	2.46 (5930.94)	1.31 (5895.39)
$RC107A$	1.64 (4927.24)	1.77 (4891.85)	1.74 (4856.46)	1.54 (4821.08)	1.54 (4785.69)	2.32 (4750.30)	2.38 (4714.92)	1.80 (4679.53)
$RC108A$	1.35 (4764.11)	1.38 (4734.81)	1.52 (4705.51)	1.98 (4676.21)	2.23 (4646.91)	1.77 (4617.61)	1.20 (4588.32)	1.00 (4559.02)
$R203A$	0.79 (3197.64)	1.16 (3168.79)	1.00 (3139.95)	1.24 (3111.10)	0.75 (3082.26)	1.30 (3053.41)	0.99 (3024.56)	1.14 (2995.72)
$R204A$	2.47 (2785.61)	1.06 (2752.39)	1.39 (2719.17)	1.00 (2685.95)	1.29 (2652.74)	1.14 (2619.52)	1.18 (2586.30)	0.88 (2553.09)
$C201A$	0.78 (5440.94)	0.77 (5414.32)	0.76 (5387.71)	0.76 (5361.09)	0.88 (5334.47)	0.81 (5307.85)	1.97 (5281.24)	1.34 (5254.62)
$C202A$	1.57 (5517.12)	1.09 (5479.11)	1.00 (5441.09)	1.42 (5403.08)	3.08 (5365.06)	3.31 (5327.05)	3.91 (5289.03)	2.13 (5251.02)
$RC203A$	1.48 (3505.92)	1.20 (3481.57)	1.53 (3457.22)	1.44 (3432.87)	1.16 (3408.52)	1.98 (3384.17)	1.93 (3359.82)	2.87 (3335.47)
$RC204A$	1.64 (3197.73)	1.80 (3178.72)	0.87 (3159.71)	1.14 (3140.70)	0.97 (3121.69)	1.34 (3102.68)	1.81 (3083.67)	1.33 (3064.66)

in Fig. 6, EMA achieved superior performance on VRPHTO P10 when compared to VRPHTO P12. From this observation, we can infer the better optimization performance could be obtained by EMA if more similar problems are required to be solved simultaneously.

Since EMA and SEA share the same configuration of evolutionary operators and parameters, and the only difference is the multitasking paradigm incorporated in the former, the superior performance obtained by EMA in terms of both solution quality and search efficiency confirmed the efficacy of our proposed algorithm for solving multitasking VRPHTOs.

VI. CONCLUSION

In this article, to improve the practicability of VRPOD, we have proposed a generalized variant of VRP (i.e., VRPHTO) by considering the heterogeneity and time windows of both regular and occasional drivers for service. Further, with respect to the requirement in cloud computing, wherein multiple tasks need to be solved at the same time, we have also proposed a novel EMA to solve multiple VRPHTOs simultaneously. In particular, four important components of EMA, that is, permutation-based common representation, split procedure, routing information exchange, and chromosome evaluation have been introduced in detail. To illustrate the benefit of proposed VRPHTO and to evaluate the efficacy of the proposed EMA, we have generated 56 VRPHTO instances based on the widely used HVRPTW benchmark, which have been further used in the empirical study. Observation from the experimental results can be summarized as follows.

1) Routing cost can be reduced in almost all the instances when occasional drivers are employed. Particularly, the fixed cost and the employed number of occasional drivers have a great effect on the cost savings.
2) EMA is not only able to solve multiple VRPHTO tasks simultaneously but also can enhance the problem-solving process of tasks by knowledge transfer along the evolutionary search process. Greater enhancement could be obtained by EMA if more similar problems are required to be solved at the same time.

For future work, we plan to explore the similarity measure between problems, which can be used as the guidance for applying EMA on the cloud computing platform for solving multiple optimization tasks concurrently.

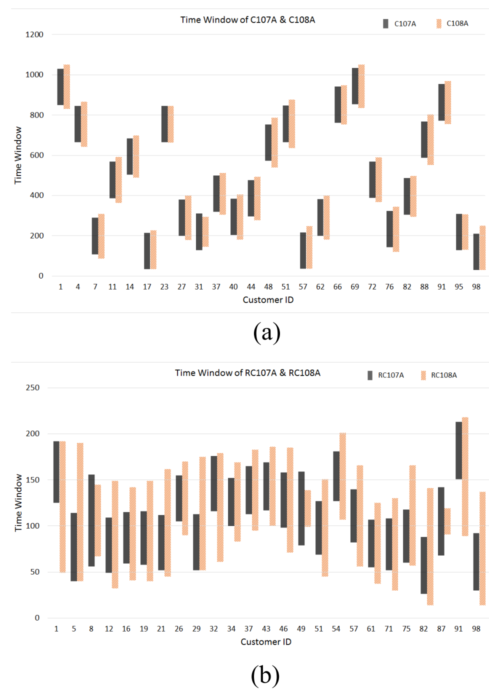


Fig. 8. Similarity Comparison of P10 and P12. y-axis: Customer time window; x-axis: Customer ID. (a) Similarity Comparison between C107A and C108A. (b) Similarity Comparison between RC107A and RC108A.

REFERENCES

[1] G. B. Dantzig and J. H. Ramser, "The truck dispatching problem," *Manag. Sci.*, vol. 6, no. 1, pp. 80–91, 1959.

[2] A. S. Azad, M. Islam, and S. Chakraborty, "A heuristic initialized stochastic memetic algorithm for MDPVRP with interdependent depot operations," *IEEE Trans. Cybern.*, vol. 47, no. 12, pp. 4302–4315, Dec. 2017.

[3] R. Shang, K. Dai, L. Jiao, and R. Stolkin, "Improved memetic algorithm based on route distance grouping for multiobjective large scale capacitated arc routing problems," *IEEE Trans. Cybern.*, vol. 46, no. 4, pp. 1000–1013, Apr. 2016.

[4] R. Hirabayashi, Y. Saruwatari, and N. Nishida, "Tour construction algorithm for the capacitated arc routing problem," *Asia–Pac. J. Oper. Res.*, vol. 9, no. 2, pp. 155–175, 1992.

[5] G. Clarke and J. W. Wright, "Scheduling of vehicles from a central depot to a number of delivery points," *Oper. Res.*, vol. 12, no. 4, pp. 568–581, 1964.

[6] I. H. Osman and S. Salhi, "Local search strategies for the vehicle fleet mix problem," in *Modern Heuristic Search Methods*. Chichester, U.K.: Wiley, 1996, pp. 131–153.

[7] C. Prins, "A simple and effective evolutionary algorithm for the vehicle routing problem," *Comput. Oper. Res.*, vol. 31, no. 12, pp. 1985–2002, 2004.

[8] J. Nalepa and M. Blocho, "Adaptive memetic algorithm for minimizing distance in the vehicle routing problem with time windows," *Soft Comput.*, vol. 20, no. 6, pp. 2309–2327, 2016.

[9] P. Toth and D. Vigo, *Vehicle Routing: Problems, Methods, and Applications*. Philadelphia, PA, USA: Soc. Ind. Appl. Math., 2014.

[10] M. Brignall, "Make a delivery and cover your petrol costs," The Guardian, London, U.K., Rep., 2011. [Online]. Available: https://www.theguardian.com/money/2011/mar/12/make-a-delivery-petrol-costs

[11] A. Barr and J. Wohl, "Exclusive: Walmart May get customers to deliver packages to online buyers," Reuters Bus., London, U.K., Rep., 2012. [Online]. Available: https://www.reuters.com/article/us-retail-walmart-delivery/exclusive-wal-mart-may-get-customers-to-deliver-packages-to-online-buyers-idUSBRE92R03820130328

[12] G. Macrina, L. D. P. Pugliese, F. Guerriero, and D. Laganà, "The vehicle routing problem with occasional drivers and time windows," in *Optimization and Decision Science: Methodologies and Applications. ODS 2017* (Springer Proceedings in Mathematics & Statistics), vol. 217, A. Sforza and C. Sterle, Eds. Cham, Switzerland: Springer, 2017.

[13] C. Archetti, M. Savelsbergh, and M. G. Speranza, "The vehicle routing problem with occasional drivers," *Eur. J. Oper. Res.*, vol. 254, no. 2, pp. 472–480, 2016.

[14] I. D. Falco, U. Scafuri, and E. Tarantino, "Mapping of time-consuming multitask applications on a cloud system by multiobjective differential evolution," *Parallel Comput.*, vol. 48, pp. 40–58, Oct. 2015.

[15] J. Ekanayake and G. Fox, "High performance parallel computing with clouds and cloud technologies," in *Cloud Computing*, D. R. Avresky, M. Diaz, A. Bode, B. Ciciani, and E. Dekel, Eds. Berlin, Germany: Springer, 2010, pp. 20–38.

[16] C. Evangelinos and C. N. Hill, "Cloud computing for parallel scientific HPC applications: Feasibility of running coupled atmosphere-ocean climate models on Amazon's EC2," *Ratio*, vol. 2, no. 2, pp. 2–34, 2008.

[17] A. Gupta, Y.-S. Ong, and L. Feng, "Multifactorial evolution: Towards evolutionary multitasking," *IEEE Trans. Evol. Comput.*, vol. 20, no. 3, pp. 343–357, Jun. 2016.

[18] F.-H. Liu and S.-Y. Shen, "The fleet size and mix vehicle routing problem with time windows," *J. Oper. Res. Soc.*, vol. 50, no. 7, pp. 721–732, 1999.

[19] K. Dorling, J. Heinrichs, G. G. Messier, and S. Magierowski, "Vehicle routing problems for drone delivery," *IEEE Trans. Syst., Man, Cybern., Syst.*, vol. 47, no. 1, pp. 70–85, Jan. 2017.

[20] J. Wang, Y. Zhou, Y. Wang, J. Zhang, C. L. P. Chen, and Z. Zheng, "Multiobjective vehicle routing problems with simultaneous delivery and pickup and time windows: Formulation, instances, and algorithms," *IEEE Trans. Cybern.*, vol. 46, no. 3, pp. 582–594, Mar. 2016.

[21] D. J. Bertsimas, "A vehicle routing problem with stochastic demand," *Oper. Res.*, vol. 40, no. 3, pp. 574–585, 1992.

[22] J. Renaud, G. Laporte, and F. F. Boctor, "A tabu search heuristic for the multi-depot vehicle routing problem," *Comput. Oper. Res.*, vol. 23, no. 3, pp. 229–235, 1996.

[23] R. W. Bent and P. V. Hentenryck, "Scenario-based planning for partially dynamic vehicle routing with stochastic customers," *Oper. Res.*, vol. 52, no. 6, pp. 977–987, 2004.

[24] M. M. Solomon, "Algorithms for the vehicle routing and scheduling problems with time window constraints," *Oper. Res.*, vol. 35, no. 2, pp. 254–265, 1987.

[25] J. Wang, T. Weng, and Q. Zhang, "A two-stage multiobjective evolutionary algorithm for multiobjective multidepot vehicle routing problem with time windows," *IEEE Trans. Cybern.*, vol. 49, no. 7, pp. 2467–2478, Jul. 2019.

[26] A. Gupta, C. K. Heng, Y. S. Ong, P. S. Tan, and A. N. Zhang, "A generic framework for multi-criteria decision support in eco-friendly urban logistics systems," *Expert Syst. Appl.*, vol. 71, pp. 288–300, Apr. 2017.

[27] Y. Kuo and C.-C. Wang, "A variable neighborhood search for the multi-depot vehicle routing problem with loading cost," *Expert Syst. Appl.*, vol. 39, no. 8, pp. 6949–6954, 2012.

[28] F. Hernandez, D. Feillet, R. Giroudeau, and O. Naud, "Branch-and-price algorithms for the solution of the multi-trip vehicle routing problem with time windows," *Eur. J. Oper. Res.*, vol. 249, no. 2, pp. 551–559, 2016.

[29] K. C. Tan, Y. H. Chew, and L. H. Lee, "A hybrid multiobjective evolutionary algorithm for solving vehicle routing problem with time windows," *Comput. Optim. Appl.*, vol. 34, no. 1, pp. 115–151, 2006.

[30] M. Schneider, A. Stenger, and D. Goeke, "The electric vehicle-routing problem with time windows and recharging stations," *Transport. Sci.*, vol. 48, no. 4, pp. 500–520, 2014.

[31] M. Keskin, G. Laporte, and B. Catay, "Electric vehicle routing problem with time-dependent waiting times at recharging stations," *Comput. Oper. Res.*, vol. 107, pp. 77–94, Jul. 2019.

[32] S. Pelletier, O. Jabali, and G. Laporte, "The electric vehicle routing problem with energy consumption uncertainty," *Transport. Res. B Methodol.*, vol. 126, pp. 225–255, Aug. 2019.

[33] G. Kim, Y. S. Ong, T. Cheong, and P. S. Tan, "Solving the dynamic vehicle routing problem under traffic congestion," *IEEE Trans. Intell. Transp. Syst.*, vol. 17, no. 8, pp. 2367–2380, Aug. 2016.

[34] T. Hintsch and S. Irnich, "Exact solution of the soft-clustered vehicle-routing problem," *Eur. J. Oper. Res.*, vol. 280, no. 1, pp. 164–178, 2020.

[35] I. Rodríguez-Martín, J.-J. Salazar-González, and H. Yaman, "The periodic vehicle routing problem with driver consistency," *Eur. J. Oper. Res.*, vol. 273, no. 2, pp. 575–584, 2019.

[36] Z. Wang and J.-B. Sheu, "Vehicle routing problem with drones," *Transport. Res. B Methodol.*, vol. 122, pp. 350–364, Apr. 2019.

[37] D. Schermer, M. Moeini, and O. Wendt, "A matheuristic for the vehicle routing problem with drones and its variants," *Transport. Res. C Emerg. Technol.*, vol. 106, pp. 166–204, Sep. 2019.

[38] L. Dahle, H. Andersson, and M. Christiansen, "The vehicle routing problem with dynamic occasional drivers," in *Proc. Int. Conf. Comput. Logist.*, 2017, pp. 49–63.

[39] P. Toth and D. Vigo, "The vehicle routing problem," in *Monographs on Discrete Mathematics and Applications*. Philadelphia, PA, USA: Soc. Ind. Appl. Math., 2002.

[40] B. L. Golden, S. Raghawan, and E. A. Wasil, *The Vehicle Routing Problem*, New York, NY, USA: Springer, 2008.

[41] K. Braekers, K. Ramaekers, and I. V. Nieuwenhuyse, "The vehicle routing problem: State of the art classification and review," *Comput. Ind. Eng.*, vol. 99, pp. 300–313, Sep. 2016.

[42] Y. Yuan, Y. S. Ong, A. Gupta, P. S. Tan, and H. Xu, "Evolutionary multitasking in permutation-based combinatorial optimization problems: Realization with TSP, QAP, LOP, and JSP," in *Proc. IEEE Region 10 Conf. (TENCON)*, 2016, pp. 3157–3164.

[43] A. Gupta, Y.-S. Ong, L. Feng, and K. C. Tan, "Multiobjective multifactorial optimization in evolutionary multitasking," *IEEE Trans. Cybern.*, vol. 47, no. 7, pp. 1652–1665, Jul. 2017.

[44] J. Ding, C. Yang, Y. Jin, and T. Chai, "Generalized multitasking for evolutionary optimization of expensive problems," *IEEE Trans. Evol. Comput.*, vol. 23, no. 1, pp. 44–58, Feb. 2019.

[45] J. Zhong, L. Feng, W. Cai, and Y.-S. Ong, "Multifactorial genetic programming for symbolic regression problems," *IEEE Trans. Syst., Man, Cybern., Syst.*, to be published.

[46] R.-T. Liaw and C.-K. Ting, "Evolutionary many-tasking based on biocoenosis through symbiosis: A framework and benchmark problems," in *Proc. IEEE Congr. Evol. Comput. (CEC)*, 2017, pp. 2266–2273.

[47] M. Gong, Z. Tang, H. Li, and J. Zhang, "Evolutionary multitasking with dynamic resource allocating strategy," *IEEE Trans. Evol. Comput.*, vol. 23, no. 5, pp. 858–869, Oct. 2019.

[48] Y.-W. Wen and C.-K. Ting, "Parting ways and reallocating resources in evolutionary multitasking," in *Proc. IEEE Congr. Evol. Comput. (CEC)*, 2017, pp. 2404–2411.

[49] K. K. Bali, Y.-S. Ong, A. Gupta, and P. S. Tan, "Multifactorial evolutionary algorithm with online transfer parameter estimation: MFEA-II," *IEEE Trans. Evol. Comput.*, to be published.

[50] S. Reil, A. Bortfeldt, and L. Mönch, "Heuristics for vehicle routing problems with backhauls, time windows, and 3D loading constraints," *Eur. J. Oper. Res.*, vol. 266, no. 3, pp. 877–894, 2018.

[51] M. Schneider, F. Schwahn, and D. Vigo, "Designing granular solution methods for routing problems with time windows," *Eur. J. Oper. Res.*, vol. 263, no. 2, pp. 493–509, 2017.

[52] C. Koç, T. Bektaş, O. Jabali, and G. Laporte, "A hybrid evolutionary algorithm for heterogeneous fleet vehicle routing problems with time windows," *Comput. Oper. Res.*, vol. 64, pp. 11–27, Dec. 2015.

[53] M. Mirabi, "A novel hybrid genetic algorithm for the multidepot periodic vehicle routing problem," *AI EDAM*, vol. 29, no. 1, pp. 45–54, 2015.

[54] C. Prins, "Two memetic algorithms for heterogeneous fleet vehicle routing problems," *Eng. Appl. Artif. Intell.*, vol. 22, no. 6, pp. 916–928, 2009.

[55] I. M. Oliver, D. J. Smith, and J. R. C. Holland, "A study of permutation crossover operators on the traveling salesman problem," in *Proc. 2nd Int. Conf. Genet. Algorithms Genet. Algorithms Appl.*, 1987, pp. 224–230.

[56] J.-Y. Potvin, C. Duhamel, and F. Guertin, "A genetic algorithm for vehicle routing with backhauling," *Appl. Intell.*, vol. 6, no. 4, pp. 345–355, 1996.

[57] G. Rudolph, "Evolutionary search under partially ordered fitness sets," in *Proc. Int. Symp. Inf. Sci. Innov. Eng. Nat. Artif. Intell. Syst.*, 2001, pp. 818–822.

[58] E. Zitzler, K. Deb, and L. Thiele, "Comparison of multiobjective evolutionary algorithms: Empirical results," *Evol. Comput.*, vol. 8, no. 2, pp. 173–195, 2000.

[59] N. Labadi, C. Prins, and M. Reghioui, "A memetic algorithm for the vehicle routing problem with time windows," *RAIRO Oper. Res.*, vol. 42, no. 3, pp. 415–431, 2008.

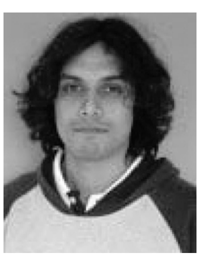
Liang Feng received the Ph.D. degree from the School of Computer Engineering, Nanyang Technological University, Singapore, in 2014.

He was a Postdoctoral Research Fellow with the Computational Intelligence Graduate Lab, Nanyang Technological University. He is currently an Assistant Professor with the College of Computer Science, Chongqing University, Chongqing, China. His research interests include computational and artificial intelligence, memetic computing, big data optimization and learning, as well as transfer learning.

Dr. Feng's research work on evolutionary multitasking won the 2019 IEEE Transactions on Evolutionary Computation Outstanding Paper Award. He is an Associate Editor of the *IEEE Computational Intelligence Magazine*, *Memetic Computing*, and *Cognitive Computation*. He is also the Founding Chair of the IEEE CIS Intelligent Systems Applications Technical Committee Task Force on "Transfer Learning & Transfer Optimization."

Lei Zhou received the B.E. degree from the School of Computer Science and Technology, Shandong University, Jinan, China, in 2014, and the Ph.D. degree from the College of Computer Science, Chongqing University, Chongqing, China, in 2019.

He is currently a Postdoctoral Research Fellow with the School of Computer Engineering, Nanyang Technological University, Singapore. His research interests include evolutionary computations, memetic computing, as well as transfer learning and optimization.



Abhishek Gupta received the Ph.D. degree in engineering science from the University of Auckland, Auckland, New Zealand, in 2014.

Over the past five years, he has been working in the area of memetic computation, with a particular focus on developing novel theories and algorithms in the topics of evolutionary transfer and multitask optimization. He is currently appointed as a Scientist with the Singapore Institute of Manufacturing Technology, Agency for Science, Technology and Research, Singapore. He also jointly serves as an Adjunct Research Scientist with the Data Science and Artificial Intelligence Research Center, School of Computer Science and Engineering, Nanyang Technological University, Singapore.

Dr. Gupta's pioneering work on evolutionary multitasking, in particular, was bestowed the 2019 IEEE Transactions on Evolutionary Computation Outstanding Paper Award by the IEEE Computational Intelligence Society (CIS). He is an Associate Editor of the IEEE Transactions on Emerging Topics in Computational Intelligence, and is also the Founding Chair of the IEEE CIS Emergent Technology Technical Committee Task Force on Multitask Learning and Multitask Optimization.

Jinghui Zhong received the Ph.D. degree from the School of Information Science and Technology, Sun Yat-sen University, Guangzhou, China, in 2012.

He is currently an Associate Professor with the School of Computer Science and Engineering, South China University of Technology, Guangzhou. From 2013 to 2016, he worked as a Postdoctoral Research Fellow with the School of Computer Engineering, Nanyang Technological University, Singapore. His research interests include evolutionary computation (e.g., genetic programming and differential evolution), machine learning, and agent-based modeling.

Zexuan Zhu received the B.S. degree in computer science and technology from Fudan University, Shanghai, China, in 2003, and the Ph.D. degree in computer engineering from Nanyang Technological University, Singapore, in 2008.

He is currently a Professor with the College of Computer Science and Software Engineering, Shenzhen University, Shenzhen, China. His research interests include computational intelligence, machine learning, and bioinformatics.

Prof. Zhu is an Associate Editor of the IEEE Transactions on Evolutionary Computation and the IEEE Transactions on Emerging Topics in Computational Intelligence, and serves as an Editorial Board Member of *Memetic Computing* and *Soft Computing*. He is also the Chair of the IEEE CIS, Emergent Technologies Task Force on Memetic Computing.

Kay-Chen Tan received the B.Eng. (First Class Hons.) degree in electronics and electrical engineering and the Ph.D. degree from the University of Glasgow, Glasgow, U.K., in 1994 and 1997, respectively.

He is a Full Professor with the Department of Computer Science, City University of Hong Kong, Hong Kong. He has published over 200 refereed articles and six books.

Prof. Tan is the Editor-in-Chief of the IEEE Transactions on Evolutionary Computation, was the Editor-in-Chief of the *IEEE Computational Intelligence Magazine* from 2010 to 2013, and currently serves as an Editorial Board Member of over ten journals. He is an Elected Member of the IEEE CIS AdCom from 2017 to 2019.

Kai Qin received the B.Eng. degree from Southeast University, Nanjing, China, in 2001, and the Ph.D. degree from Nanyang Technological University, Singapore in 2007.

From 2007 to 2012, he worked first at the University of Waterloo, Waterloo, ON, Canada, and then with INRIA, Rocquencourt, France. Since 2013, he has been a Vice-Chancellor's Research Fellow, a Lecturer and a Senior Lecturer with RMIT University, Melbourne, VIC, Australia. In 2017, he joined the Swinburne University of Technology, Melbourne, as an Associate Professor, where he is currently leading the Intelligent Data Analytics Lab and Machine Learning and Intelligent Optimization Research Group. His major research interests include evolutionary computation, machine learning, computer vision, GPU computing, services computing, and mobile computing.

Dr. Qin won the 2012 IEEE Transactions on Evolutionary Computation Outstanding Paper Award and the Overall Best Paper Award at the 18th Asia–Pacific Symposium on Intelligent and Evolutionary Systems in 2014. He is currently the Vice-Chair of the IEEE Neural Networks Technical Committee, and co-chairs the IEEE Emergent Technologies Task Forces on "Collaborative Learning and Optimization" and "Multitask Learning and Multitask Optimization."