Memetic search with interdomain learning : a realization between CVRP and CARP

Feng, Liang; Ong, Yew-Soon; Lim, Meng-Hiot; Tsang, Ivor W.

2014

Feng, L., Ong, Y., Lim, M. & Tsang, I. W. (2014). Memetic search with interdomain learning : a realization between CVRP and CARP. IEEE Transactions On Evolutionary Computation, 19(5), 644-658. https://dx.doi.org/10.1109/TEVC.2014.2362558

https://hdl.handle.net/10356/148169

https://doi.org/10.1109/TEVC.2014.2362558

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/TEVC.2014.2362558, IEEE Transactions on Evolutionary Computation

Memetic Search with Inter-Domain Learning: A Realization between CVRP and CARP

Liang Feng, Yew-Soon Ong, Meng-Hiot Lim, and Ivor W. Tsang

***Abstract*—In recent decades, a plethora of dedicated evolutionary algorithms (EAs) have been crafted to solve domain specific complex problems more efficiently. Many advanced EAs have relied on the incorporation of domain specific knowledge as inductive biases that is deemed to fit the problem of interest well. As such, the embedment of domain knowledge about the underlying problem within the search algorithms is becoming an established mode of enhancing evolutionary search performance. In this paper, we present a study on evolutionary memetic computing paradigm that is capable of learning and evolving knowledge meme that traverses different but related problem domains, for greater search efficiency. Focusing on combinatorial optimization as the area of study, a realization of the proposed approach is investigated on two NP-hard problem domains (i.e., capacitated vehicle routing problem (CVRP) and capacitated arc routing problem (CARP)). Empirical studies on well established routing problems and their respective state-of-the-art optimization solvers are presented to study the potential benefits of leveraging knowledge memes that are learned from different but related problem domains on future evolutionary search.**

***Index Terms*—Evolutionary Optimization, Memetic Computing, Cross Domain Memes, Knowledge Memes, Learning.**

I. Introduction

EVOLUTIONARY algorithms (EAs) are adaptive search approaches that take inspirations from the principles of natural selection and genetics. They have been shown to be suitable for solving nonlinear, multi-modal, and discrete NP-hard problems effectively. Due to their flexibility and ease of use, EA has been known as a universal problem solver that enjoyed significant successes in obtaining optimal or near-optimal solutions on a plethora of complex real-world optimization problems [1], [2], [3], [4], [5], [6], [7]. However, EA which involves the iterative process of reproduction, is deemed to be slow and sometimes falls short in meeting with today's competitive need for high-quality solutions promptly.

In the recent decade, it is observed that many efficient optimizations using modern advanced EAs have been achieved via the incorporation of domain specific knowledge as inductive biases that fit to the problem of interest well [8]. These dedicated EAs have been specially crafted with the embedment of human-expert domain specific knowledge about the underlying problem so as to speed up search convergence. In the recent special issues [9], [10] and journals [11] dedicated to EA research, several successes of evolutionary and memetic algorithms [12], [13], [14] that incorporate human expert knowledge have been reported on a plethora of complex applications, including quadratic assignment problem [15], feature selection [16], permutation flow shop scheduling [17], and VLSI floorplanning [18], etc.

To reduce the high reliance on humans' effort in designing advanced evolutionary algorithms, some researchers have considered a direct incorporation of solutions archived from previous searches as an alternative[1]. Cunningham and Smyth [19] explored a direct reuse of established high quality schedules from past traveling salesman problems (TSP) to bias the search on new TSPs. Louis *et al.* [20] proposed a case-injected genetic algorithm that periodically injects high-quality solutions obtained from previous searches on problem instances of the same domain to speed up future search. More recently, structured knowledge learned from archived optimized solutions has also been used to generate high quality solutions for evolutionary search on unseen problem instances of the same domain [21].

From a survey of the literature, it is worth noting that in spite of the efforts to automate the incorporation of domain knowledge into future evolutionary search, the success has been limited by several key factors. In particular, the earlier works in [19]-[21] make a strong assumption on the type of problems solved. [19] and [20] require the newly encountered problem instances to share common tasks with previous solved instances. [21] although does not require the tasks to be common among problem instances, they however are restricted to the representation used, which impedes the seamless reuse of domain knowledge across problems. To summarize, the greatest barrier to further progress can thus be attributed to the unique representations and characteristics of different problem domains. Hence, it is often the case that the information captured from a problem domain cannot be directly used in another. To date, little or no investigation has been conducted to automate the learning and evolution of knowledge from differing problem domains in the context of evolutionary optimization.

Given the restricted theoretical knowledge available in this area and the limited progress made, there is thus an appeal for evolutionary search paradigms that can draw upon useful

Liang Feng and Yew-Soon Ong are with the Center for Computational Intelligence, School of Computer Engineering, Nanyang Technological University, Singapore. E-mail: {feng0039, asysong}@ntu.edu.sg

Meng-Hiot Lim is with School of Electrical and Electronic Engineering, Nanyang Technological University, Singapore. Email: emhlim@ntu.edu.sg

Ivor W. Tsang is with Centre for Quantum Computation & Intelligent Systems, University of Technology, Sydney, Australia. Email: Ivor.Tsang@uts.edu.au

[1]Note that this is in contrast to domain-specific human crafted EAs in which domain knowledge is only captured and incorporated once as part of the algorithm design and development process.

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/TEVC.2014.2362558, IEEE Transactions on Evolutionary Computation

knowledge learned from different problems previously solved. Hence, the current work serves as a feasibility study on evolutionary paradigm that learns and evolves knowledge nuggets in the form of memes that traverse related problem domains. This is in the spirit of intelligent evolutionary optimization and enhanced search, where a meme is defined here as the basic unit of cultural transmission [22]. Through the present study, we hope to provide the following insights to the incorporation of knowledge across problem domains.

- What is the representation of a knowledge meme?
- How to learn and mine knowledge memes from the evolutionary search?
- How to evolve knowledge memes in the evolutionary search across related problem domains?
- Can evolutionary optimization benefit from the knowledge memes of related problem domains?
- How do the knowledge memes of different but related problem domains influence the evolutionary search?
- What forms of knowledge meme can lead to enhanced evolutionary search?

The current study thus present an attempt to emulate the learning and evolution of knowledge memes attained from past evolutionary optimization experiences that traverse problems of different but related domains. The aspiration has been to develop intelligent evolutionary search that functions and adapts effectively across contexts, like the way humans are capable of evolving its ability to handle related problems intelligibly and competently.

The rest of this paper is organized as follows. An overview of memetic computation and two different but related problem domains, namely the capacitated vehicle routing problem (CVRP) and the capacitated arc routing problem (CARP) are presented in Section II. These two problems serve as the case study for this paper. Section III introduces the concept of relatedness between CVRP and CARP. A meme representation common to the two independent problem domains is then proposed. Section IV showcases a realization of knowledge meme learning and evolution from CVRP to CARP and vice versa (i.e., CARP to CVRP). Subsequently, Section V presents an experimental study using well-established capacitated vehicle routing and capacitated arc routing problems to investigate the potential benefits of automating the learning and evolution of knowledge memes across the evolutionary optimization of related problem domains. A detailed analysis and discussion of the empirical results obtained is also provided in the section. Last but not least, the brief concluding remarks of this paper and our future works are presented in Section VI.

II. Preliminary

In this section, we first give a brief introduction of memetic computation. Subsequently, the mathematical formulations of the two combinatorial NP-hard optimization problems, namely the capacitated vehicle routing problem (CVRP) and the capacitated arc routing problem (CARP), are presented.

A. Memetic Computation

As with genes in genetics, a meme is synonymous to memetics as being the building block of cultural know-how that is transmissible and replicable. The term meme which can be traced back to Dawkins in his book "The Selfish Gene" [22], has inspired the new science of memetics that today represents the mind universe analog to genetics in cultural evolution, stretching across the fields of anthropology, biology, cognition, psychology, sociology and socio-biology [23], [24], [25].

In computer science and engineering, the meme-inspired computing methodology or more concisely memetic computation has become an increasing focus of research in recent years. Memetic computation has been defined as a paradigm that uses the notion of meme(s) as units of information encoded in computational representations for the purpose of problem solving [24]. In the last decades, a meme has been typically perceived as a form of individual learning procedure or local search operator to enhance the capability of population based search algorithm. This integration has been established as an extension of the canonical evolutionary algorithm, by the names of hybrid, adaptive hybrid or Memetic Algorithm. Since the establishments of memetic computation research, extensive studies on different realization of memetic computation have been considered. For instance, Nguyen *et al.* proposed a theoretic probabilistic memetic framework (PrMF) that unifies the local search frequency, intensity and selection of solutions undergoing local search under a single theme [26]. Feng *et al.* proposed a memetic multi-agent system (MeM) towards human-like social agents with memetic automaton [27]. Further, G. Iacca *et al.* [28] proposed a three stage optimal memetic exploration algorithm based on the philosophical concept of Ockham's Razor, while F. Caraffini *et al.* [29] investigated and proposed a simple but efficient parallel memetic structure, which consists of one global search operator and two local search operators with different search directions. More recently, to automate the design of memetic algorithms, F. Caraffini *et al.* [30] analyzed the separability of continuous optimization problems.

In this paper, we contribute to memetic computation by embarking a study on the feasibility of evolutionary searches that make use of useful knowledge in the form of memes learned from different but related problem domains previously solved. In contrast to memetic algorithm and existing memetic computation approaches, here meme is treated and defined as the useful traits of a problem domain of interests, and can be evolved across problem domains for enhanced evolutionary search on new encountered problems. Further, it is also worth noting here that, the current study is totally different from hyper-heuristics in the literature, which encompass a set of approaches with the goal of automating the design and turning of heuristic methods to solve computational search problems [31], [32]. In the present study, our core motivation is on evolutionary searches that can make use of knowledge learned from a different but related problem domain for enhance performance. There is no tuning and coordination of the heuristic methods.

B. Capacitated Vehicle Routing Problem

The capacitated vehicle routing problem (CVRP) was introduced by Dantzig and Ramser in [33]. It is defined on

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/TEVC.2014.2362558, IEEE Transactions on Evolutionary Computation

a connected undirected graph $G = (V, E)$, where vertex set $V = \{\mathbf{v}_i\}, i = 1 \ldots n$, n is the number of vertices, and edge set $E = \{e_{ij}\}, i, j = 1 \ldots n$ denoting the arc between vertices $\mathbf{v}_i$ and $\mathbf{v}_j$. Vertices $\mathbf{v}_d$ corresponds to the depot at which k homogeneous vehicles are based, and the remaining vertices denote the customers. Each edge e_{ij} is associated with a non-negative weight c_{ij} , which represents the travel distance from $\mathbf{v}_i$ to $\mathbf{v}_j$. Consider a demand set $D = \{d(\mathbf{v}_i)|\mathbf{v}_i \in V\}$, where $d(\mathbf{v}_i) > 0$ implies customer $\mathbf{v}_i$ requires service(i.e., known as task), the purpose of the CVRP is to design a set of least cost vehicle routes $\mathcal{R} = \{\mathcal{C}_i\},\ i = 1 \ldots k$ such that

- Each route $\mathcal{C}_i,\ i \in [1, k]$ must start and end at the depot node $\mathbf{v}_d \in V$.
- The total load of each route must be no more than the capacity W of each vehicle, $\sum_{\forall \mathbf{v}_i \in \mathcal{C}} d(\mathbf{v}_i) \leq W$.
- $\forall \mathbf{v}_i \in V$ *and* $d(\mathbf{v}_i) > 0$, there exists one and only one route $\mathcal{C}_i \in \mathcal{R}$ such that $\mathbf{v}_i \in \mathcal{C}_i$.

The objective of the CVRP is thus to minimize the overall distance $cost(R)$ traveled by all k vehicles and is defined as:

$$cost(R) = \sum_{i=1}^{k} c(\mathcal{C}_i) \qquad (1)$$

where $c(\mathcal{C}_i)$ is the summation of the distance traveled e_{ij} contained in route $\mathcal{C}_i$. An example of the CVRP and associated optimized route is given in Fig. 1(a), where the vertices represent the customers to service and dashed lines denotes shortest distance between customers.

C. Capacitated Arc Routing Problem

The capacitated arc routing problem (CARP) was first proposed by Golden and Wong [34] in 1981. Instead of serving a set of customers (i.e., nodes, vertices) in CVRP, CARP considers a set of streets or arcs (i.e., edges). It can be formally stated as follows: Given a connected undirected graph $G = (V, E)$, where vertex set $V = \{\mathbf{v}_i\}, i = 1 \ldots n$, n is the number of vertices, arc set $E = \{e_i\}, i = 1 \ldots m$ with m denoting the number of arcs. Consider a demand set $D = \{d(e_i)|e_i \in E\}$, where $d(e_i) > 0$ implies arc e_i requires service (i.e., known as task), a travel cost vector $C_t = \{c_t(e_i)|e_i \in E\}$ with $c_t(e_i)$ representing the cost of traveling on arc e_i, a service cost vector $C_s = \{c_s(e_i)|e_i \in E\}$ with $c_s(e_i)$ representing the cost of servicing on arc e_i. A solution of CARP can be represented as a set of travel circuits $\mathcal{R} = \{\mathcal{C}_i\},\ i = 1 \ldots k$ which satisfies the following constraints:

- Each travel circuit $\mathcal{C}_i,\ i \in [1, k]$ must start and end at the depot node $\mathbf{v}_d \in V$.
- The total load of each travel circuit must be no more than the capacity W of each vehicle, $\sum_{\forall e_i \in \mathcal{C}} d(e_i) \leq W$.
- $\forall e_i \in E$ *and* $d(e_i) > 0$, there exists one and only one circuit $\mathcal{C}_i \in \mathcal{R}$ such that $e_i \in \mathcal{C}_i$.

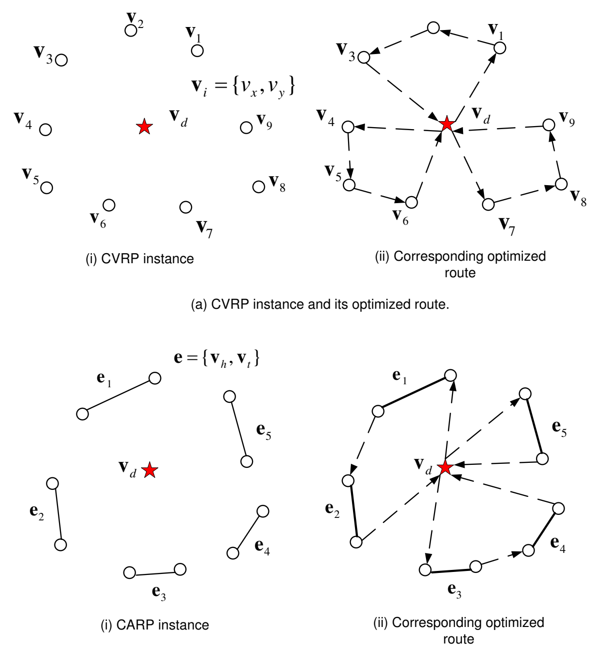
The cost of a travel circuit is then defined by the total service cost for all arcs that need service together with the total travel cost of the remaining arcs that formed the circuit:

$$cost(\mathcal{C}) = \sum_{e_i \in \mathcal{C}_s} c_s(e_i) + \sum_{e_i \in \mathcal{C}_t} c_t(e_i) \qquad (2)$$

where $\mathcal{C}_s$ and $\mathcal{C}_t$ are arc sets that need service and those that do not, respectively. The objective of CARP is then to find a valid solution $\mathcal{R}$ that minimizes the total cost:

$$C_{\mathcal{R}} = \sum_{\forall \mathcal{C}_i \in \mathcal{R}} cost(\mathcal{C}_i) \qquad (3)$$

An illustration of the CARP instance and associated optimized route is depicted in Fig. 1(b), where **full** lines denote the arcs need to be served and dashed lines gives the routes traveled by vehicles for servicing the arcs. Each arc is represented by its head and tail vertices.



(i) CVRP instance (ii) Corresponding optimized route

(a) CVRP instance and its optimized route.

(i) CARP instance (ii) Corresponding optimized route

(b) CARP instance and its optimized route.

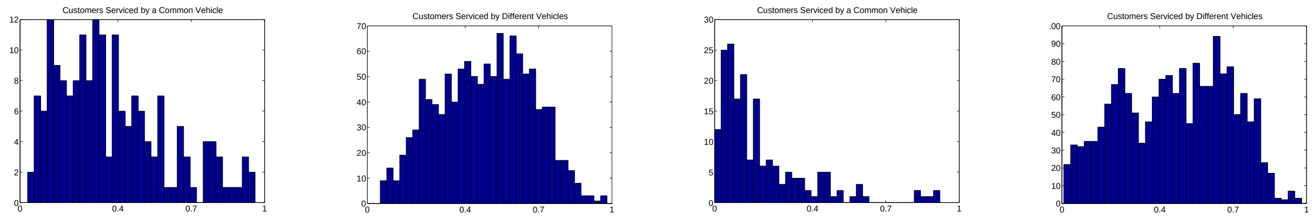
Fig. 1. An illustration of CVRP and CARP instances and their respective optimized solutions.

III. Knowledge Meme Shared by Problem Domains - CVRP & CARP

CVRP and CARP have traditionally emerged as two independent research tracks in the literature. Nevertheless, since both problem domains belong to the family of vehicle routing, it makes one wonder whether the problem-solving experiences learned on one domain could be useful to the other. Taking this cue, we begin with a study on the relatedness between problems of these two independent domains based on their optimized solution routes, since it is what both target to attain.

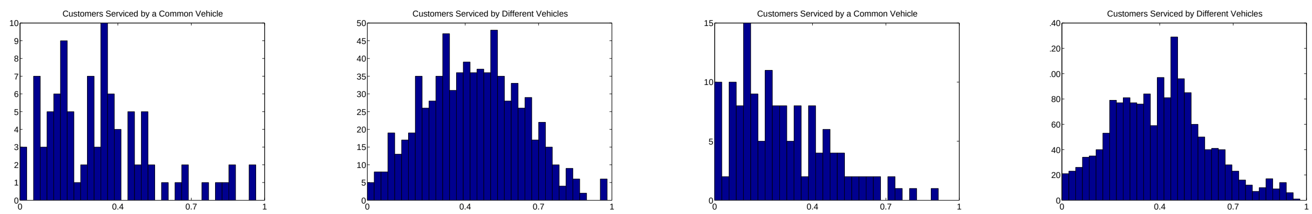
The common objective of both domains is to minimize the distances traveled by the vehicles in serving the available customers, which heavily depends on the specific assignments of customers to each vehicle. In both CVRP and CARP optimized solution routes, each vehicle can be treated as a cluster. Thus the corresponding customers assignments are actually inter-cluster and intra-cluster structure information, which are determined by the distances of the cluster members. The intra-cluster and inter-cluster distances indicate what kind of distance is it between two customers, they should be served by the same or different vehicle, respectively. Hence to study

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/TEVC.2014.2362558, IEEE Transactions on Evolutionary Computation



(a) (b) (c) (d)

Fig. 2. Pairwise distance distributions obtained from optimized "A-n54-k7" and "B-n57-k7" CVRP instances. Fig. (a) denotes the pairwise distance distribution of customers serviced by a common vehicle in "A-n54-k7"; Fig. (b) denotes the pairwise distance distribution of customers serviced by different vehicles in "A-n54-k7"; Fig. (c) presents the pairwise distance distribution of customers serviced by a common vehicle in "B-n57-k7"; Fig. (d) presents the pairwise distance distribution of customers serviced by different vehicles in "B-n57-k7".



(a) (b) (c) (d)

Fig. 3. Pairwise distance distributions obtained from optimized "E1C" and "E2C" CARP instance. Fig. (a) denotes the pairwise distance distribution of customers serviced by a common vehicle in "E1C"; Fig. (b) denotes the pairwise distance distribution of customers serviced by different vehicles in "E1C"; Fig. (c) presents the pairwise distance distribution of customers serviced by a common vehicle in "E2C"; Fig. (d) presents the pairwise distance distribution of customers serviced by different vehicles in "E2C".

the relatedness between CVRP and CARP optimized solutions, we consider the pairwise distance distributions of customers or tasks that are serviced by the same (intra) and different (inter) vehicles, which are given by the histograms of the following two distance sets:

$$D_s = \{d(t_i, t_j) | t_i, t_j \in T_s\}$$
$$D_d = \{d(t_p, t_q) | t_p, t_q \in T_d\}$$

where T_s and T_d denote the set of customers or tasks serviced by the same and different vehicles, respectively. The customer service information is extracted from the optimized solution of each problem domain independently. $d(t_i, t_j)$ gives the shortest distance between customers (i.e., vertex in CVRP and arc in CARP) t_i and t_j.

Fig 2 and Fig. 3 summarize the pairwise distributions of two CVR problem instances (e.g., labeled as "A-n54-k7" and "B-n57-k7") and two CAR problem instances (e.g., labeled as "E1C" and "E2C"), respectively. The optimized solutions are obtained using recently introduced state-of-the-art evolutionary solvers of the respective domains, i.e., [35] and [36]. In the figures, X-axis reflects the normalized range of the distance values, and Y-axis denotes the number of distances that fall within the respective range, which is given by:

$$Y_i = \sum_{j=1}^{nd} \mathbb{I}(x_j \in Bin_i) \tag{4}$$

where nd is total number of distance values, $\mathbb{I}$ denotes the indicator function. As depicted, similar trends in the pairwise distance distributions can be observed for the CVRP and CARP optimized solutions, see Fig. 2(a) versus Fig. 3(a) and Fig. 2(c) versus Fig. 3(c). These similarities imply the existence of similar structure configurations between CVRP and CARP optimized solutions. In another word, these CVRP and CARP optimized solutions bare common or similar assignment and service orders of customers, despite the differences in the problem representations.

So it is straightforward to infer that common knowledge exists in these two problem domains, and it would lead to more efficient and effective problem solving when captured from one problem domain and operate on the other. Inspired by this interesting observation, in what follows, we explore how to link these two independent routing problem domains, i.e., CVRP and CARP, and derive the shared knowledge meme in these two problem domains that can be learned and evolved for enhanced problem-solving.

A. A Common Problem Representation for CVR and CAR Problem

In this subsection, we propose to establish a common representation for CVR and CAR problem, so that the relationship between customers in these two independent domains can be conducted which makes further knowledge meme evolution across problem domain possible.

In CVRP, each customer of Fig. 1a(i) is represented as a vertex, with given cartesian coordinates ($\mathbf{v} = \{v_x, v_y\}$). On the other hand, customers or tasks in the CARP, as depicted in Fig. 1b(i), (i.e., Fig. 1), are the full line arcs ($\mathbf{e} = \{\mathbf{v}_h, \mathbf{v}_t\}$), where $\mathbf{v}_h$ and $\mathbf{v}_t$ denotes the head and tail vertex, respectively. In CARP, the distances of connected vertices are provided to describe the structure of the problem. No information on the vertex coordinates are available. As can be observed, CVRP and CARP differ in the representation of customers in a graph network. This impedes the direct incorporation of useful knowledge from one domain to the other.

In general, there are three possible means to seek a common representation involving two domains, A and B. The first is to consider the representation of A, while transforming all problem instances of domain B to A. The second is nonetheless to maintain representation of B, and transform all problem instance in A to the feature space of domain B. Lastly, all problem instances in domain A and B can be mapped to a new representation C, which is common to both. Here, we consider options 1 and 2. In particular, we derive the mapping from CARP to CVRP and use the latter as the common representation.

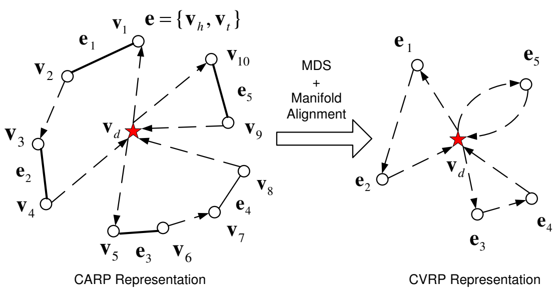


Fig. 4. Position approximation for CARP arcs via MDS and manifold alignment.

To transform CARP to CVRP representation, we begin with a calculation on the shortest distances among all the arcs that need service using the Dijkstra's algorithm [37]. Subsequently, we approximate the position of each arc with the obtained shortest distance matrix of arcs by means of multidimensional scaling (MDS) [38], which is a well-established and popular coordinates approximation method from distance matrix. As a result, each arc is represented as a node with spatial features, like the customers in CVRP. Further, we perform manifold alignment [39] between CVRP customers and the MDS approximated CARP arc positions to derive the common feature space of CVRP and CARP, while matching the local geometry and preserving the neighborhood relationship within both CVRP and CARP. This is based on the idea that customers in CVRP and CARP who bear similar local geometry should be closed to each other in the new common space. In this way, a common problem representation between CVRP and CARP problem can be established (see Fig. 4). The pseudo-code of the proposed establishment of a common problem representation for CVR and CAR is summarized in Alg. 1.

B. Knowledge Meme Shared by CVRP and CARP

Like genes that serve as "instructions for building proteins", memes are then "instructions for carrying out behavior, stored in brains" [23], [40], [41], [24], [25]. In the present context, a knowledge meme in evolutionary optimization serves as an instruction to guide the search towards the near-optimal solution. In practice, most problems (including optimization problems) seldom exist in isolation [42]. For example, the experiences on riding a bicycle can help one to drive a

Algorithm 1: Pseudo code of the proposed establishment of common representation for CVRP and CARP

Begin:
for *given CVRP instance* I_v *and CARP instance* I_a **do**
  Calculate the shortest distance matrix SD among all the arcs of I_a by Dijkstra's algorithm.
  Approximate spatial features of arcs in I_a by means of MDS with SD.
  Perform manifold alignment between CVRP customers and the MDS approximated CARP arc positions to derive their common problem representation.
End

motorcycle more productively. Students are also able to apply what have been learned in school subsequently in their work life very successfully. Thus experiences or knowledge memes learned from solving a problem can be deployed to enhance the optimization on related problems.

With a common problem representation established between CVRP and CARP, any knowledge meme learned from one domain can be directly applied on the other. An illustration example is depicted in Fig. 5. As can be observed, the shared knowledge meme $\mathbf{M}$ of CVRP and CARP is a form of instruction mined from the optimized solution route in the common feature space. When this knowledge meme is further operated on unseen routing problem across domain, it is able to generate high quality solution routes immediately without any search process.

In both CVRP and CARP, the search for optimal solution involve first identifying suitable tasks (i.e., vertices or arcs required to be serviced) assignment for each vehicle, and then finding the optimal service orders of each vehicle for the assigned tasks. Since knowledge meme is extracted from the optimized solution routes, it contains the success of both tasks assignment and tasks servicing ordering information inside. In what follows, we will present a specific realization of the form of knowledge meme and how it is learned and evolved between CVR and CAR problem domains.

IV. Learning and Evolution of Knowledge Meme between CVRP and CARP

In this section, we present a realization of the knowledge meme and its learning and evolution between CVRP and CARP for enhanced evolutionary search.

For a given CVRP or CARP instance $\mathbf{p}$ and its optimal solution $\mathbf{s}^*$, we derive the knowledge meme as a transformation $\mathbf{M}$ on the problem instance $\mathbf{p}$, which makes the transformed task distribution align well with the optimal solution $\mathbf{s}^*$. In such a manner, the success of task assignment and task servicing ordering in $\mathbf{s}^*$ can be easily obtained via techniques such as clustering, pairwise distance comparison, etc., operating on the transformed tasks. As presented in Fig. 6, where Fig. 6(a) denotes the original task distribution of a given

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/TEVC.2014.2362558, IEEE Transactions on Evolutionary Computation

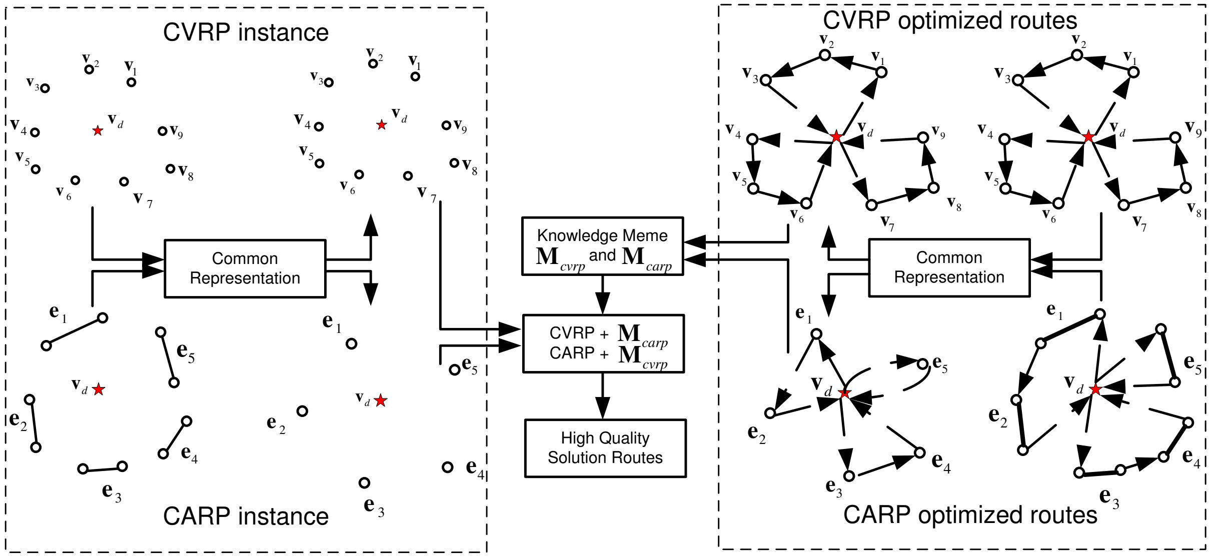


Fig. 5. An illustration example of knowledge meme shared by CVRP and CARP.

CVRP or CARP instance and Fig. 6(b) represents the obtained optimized solution. If the appropriate transformation $\mathbf{M}$ has been captured from Fig. 6(b) and deployed on Fig. 6(a), the resultant tasks distribution is depicted in Fig. 6(c). As can be observed, the transformation has re-located tasks serviced by a common vehicle to become closer to one another (as desired by the optimized solution shown in Fig. 6(b)), while tasks serviced by different vehicles to be kept further apart. Further, to match the service orders of each vehicle to that of the optimized solution, the task distribution is adapted according to the sorted pairwise distances in ascending order (e.g., the distance between v_1 and v_3 is the largest among v_1, v_2 and v_3, while the distance between v_{10} and v_9 is smaller than that of v_{10} and v_8). Thus when conducting clustering on the transformed tasks and pairwise distance comparison on the tasks assigned in each cluster, the task assignment and task service orders of Fig. 6(b) can be obtained as depicted in Fig. 6(d).

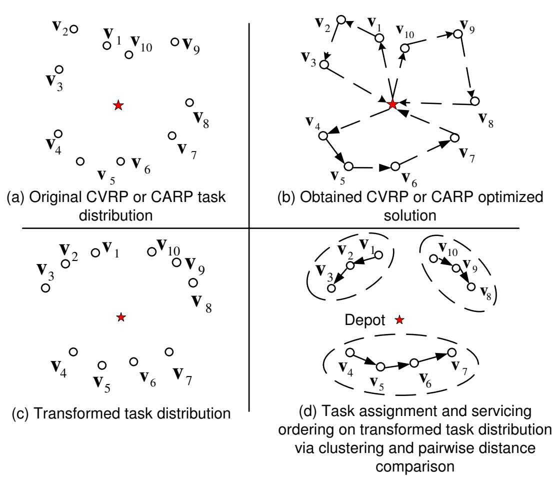


Fig. 6. An illustration example of how the success of task assignment and task servicing ordering in the optimal solution $\mathbf{s}$ are archived by a transformation.

1) Learning of Knowledge Meme from CVRP or CARP Search Experiences: In particular, for a given CVRP or CARP problem instance and its optimized solution, denoted by ($\mathbf{p}$, $\mathbf{s}^*$) in the constructed common feature space, the learning of knowledge meme $\mathbf{M}$ has been formulated as a maximization of the statistical dependency [43] between $\mathbf{p}$ and $\mathbf{s}^*$ with distance constraints as follows:

$$\begin{aligned} \max_{\mathbf{K}} \quad & tr(\mathbf{HKHY}) \\ \text{s.t.} \quad & \mathbf{K} = \mathbf{X}^T * \mathbf{M} * \mathbf{X}, \mathbf{K} \succeq 0 \\ & D_{ij} > D_{iq}, \forall (i,j,q) \in \mathcal{N} \end{aligned} \tag{5}$$

where $tr(\cdot)$ denotes the trace operation of a matrix. $\mathbf{X}$, $\mathbf{Y}$ are the matrix representations of a CARP or CVRP instance $\mathbf{p}$ and the corresponding problem solution $\mathbf{s}^*$, respectively. In $\mathbf{X}$, each column gives the location information (e.g., coordinates) of a task. Further, if task $\mathbf{v}_i$ and task $\mathbf{v}_j$ are served by the same vehicle, $\mathbf{Y}(i,j) = 1$, otherwise, $\mathbf{Y}(i,j) = -1$. Further, $\mathbf{H} = \mathbf{I} - \frac{1}{n}\mathbf{1}\mathbf{1}'$ centers the data and the labels in the feature space, $\mathbf{I}$ denotes the identity matrix, n equals to the number of tasks. $D_{ij} > D_{iq}$ is the constraint to impose that after task i, task q should be served before task j by the same vehicle. This order information can be readily obtained from optimized solution $\mathbf{s}^*$.

Let $\mathbf{T}_{ij}$ denotes a $n \times n$ matrix that takes non-zeros at $T_{ii} = T_{jj} = 1$, $T_{ij} = T_{ji} = -1$. The distance constraints $D_{ij} > D_{iq}$ in Equation 5 is then reformulated as $tr(\mathbf{KT}_{ij}) > tr(\mathbf{KT}_{iq})$. Further, slack variables ξ_{ijq} are introduced to measure the violations of distance constraints and penalize the corresponding square loss. Consequently, by substituting the constraints into Equation 5, we arrive at:

$$\begin{aligned} \min_{\mathbf{M},\xi} \quad & -tr(\mathbf{XHYHX}^T\mathbf{M}) + \frac{C}{2}\sum \xi_{ijq}^2 \\ \text{s.t.} \quad & \mathbf{M} \succeq 0 \\ & tr(\mathbf{X}^T\mathbf{MXT}_{ij}) > tr(\mathbf{X}^T\mathbf{MXT}_{iq}) - \xi_{ijq}, \\ & \forall (i,j,q) \in \mathcal{N} \end{aligned} \tag{6}$$

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/TEVC.2014.2362558, IEEE Transactions on Evolutionary Computation

where C balances between the two parts of the criterion. The first constraint enforces the learned knowledge meme denoted by matrix $\mathbf{M}$ to be positive semi-definite, while the second constraint imposes the scaled distances among the tasks to align well with the desired service orders of the optimized solution $\mathbf{s}^*$ (i.e., $\mathbf{Y}$). By configuring C based on cross-validation, Equation 6 can be solved as described in [44].

2) Evolution of Knowledge Meme Between CVRP and CARP: The evolution of knowledge meme between CVRP and CARP domains includes, the selection of knowledge meme and the assimilation of the selected knowledge meme for generating high quality routes for unseen problems.

Selection of Learned Knowledge Meme: Further, as more knowledge memes have been learned, the question of which knowledge meme should be selected for evolving across problem domain arises. A simple way for this selection is to choose the elite knowledge meme from the most similar problems previously solved. However, due to the enormous problem space of both domains, the sparsity of the problem instances and the differences between problems in the domains, the likelihood of a new unseen problem to bear 100% similarity to previous problem solved is very low. Thus in the current paper, we propose a generalization of multiple memes using a weighted approach of multiple memes from similar problems so as to positively bias the search towards high quality solutions robustly. Particularly, suppose there is a set of n unique $\mathbf{M}$ in the knowledge pool $\mathbf{KP}$, i.e., $\mathbf{KP} = \{\mathbf{M}_1, \mathbf{M}_2, \ldots, \mathbf{M}_m\}$. The knowledge meme selection process is formulated as to identify the weight μ_i of each knowledge meme. A fitter knowledge meme should have a higher weight and the summation of the weights of all knowledge meme equates to 1 (i.e., $\sum_{i=1}^{m} \mu_i = 1$).

In particular, the weight vector $\boldsymbol{\mu}$ is determined as:

$$\begin{aligned} \max_{\mu} \quad & tr(\mathbf{HKHY}) - \sum_{i=1}^{m} (\mu_i)^2 Dis_i \qquad (7) \\ \text{s.t.} \quad & \mathbf{M}_t = \sum_{i=1}^{m} \mu_i \mathbf{M}_i, \mu_i \geq 0, \sum_{i=1}^{m} \mu_i = 1 \\ & \mathbf{K} = \mathbf{X}^T * \mathbf{M}_t * \mathbf{X}, \mathbf{K} \succeq 0 \end{aligned}$$

where Dis_i is the discrepancy measure between two given problem instances. In the present context, $Dis_i = \beta * MMD_i + (1-\beta) * Dif_i$, where MMD_i denotes the maximum mean discrepancy [45], which is used to compare the distribution similarity between two given instances by measuring the distance between their corresponding means, that given as:

$$MMD(D_s, D_t) = ||\frac{1}{n_s}\sum_{i=1}^{s}\phi(x_i^s) - \frac{1}{n_t}\sum_{i=1}^{t}\phi(x_i^t)||$$

Dif_i denotes the difference in vehicle capacity for two given problem instances. β balances between the two parts (i.e., MMD_i and Dif_i) in Dis_i. Based on domain knowledge, the tasks distribution has a higher weightage than vehicle capacity information. This implies that $\beta > 0.5$. In this work, β is configured empirically as 0.8 to favour task distribution information over vehicle capacity information. In Equation 7, the first term serves to maximize the statistical dependence between input $\mathbf{X}$ and output label $\mathbf{Y}$ for clustering [46]. The second term measures the similarity between the previous problem instances solved and the given new problem of interest.

Since two unknown variables exist (i.e., μ and $\mathbf{Y}$) in Equation 7, it can be solved by fixing one variable alternately. When $\mathbf{Y}$ is fixed, Equation 7 becomes a quadric programming problem of μ. $\mathbf{Y}$ can be obtained by clustering (e.g., K-Means) on $\mathbf{X}$ if μ fixed. Further, as $\boldsymbol{\mu}$ obtained via solving Equation 7, the selected $\mathbf{M}_t$ is then derived as:

$$\mathbf{M}_t = \sum_{i=1}^{m} \mu_i \mathbf{M}_i, (\sum_{i=1}^{m} \mu_i = 1, \mu_i \in [0,1])$$

Assimilation of Knowledge Meme for Evolutionary Search: Subsequently, the knowledge meme $\mathbf{M}_t$ generalized from past experiences is then assimilated for enhancing evolutionary search on another problem domain via the generation of meme biased solutions. In particular, the tasks distribution of the original data $\mathbf{X}_{new}$ is first transformed or remapped to a new tasks distribution $\mathbf{X}'_{new}$ (i.e., from Fig. 6(a) to Fig. 6(c)) by:

$$\mathbf{X}'_{new} = \mathbf{L}^T \mathbf{X}_{new} \qquad (8)$$

where $\mathbf{L}$ is derived by SVD of $\mathbf{M}_t$. Further, the tasks assignment of vehicles and task service ordering of the meme biased solution are obtained by clustering on the transformed tasks and pairwise distance sorting among tasks assigned in the same cluster (Fig. 6(d)), respectively. In summary, the enhancement of evolutionary search with knowledge meme learned across problem domains is realized by injecting knowledge meme biased solutions into the population of the evolutionary search.

The overview on the workflow of knowledge meme evolution between CVRP and CARP is depicted in Fig. 7. If the solved cross domain problems $\mathbf{P}_{solved}$ are available, each problem in $\mathbf{P}_{solved}$ will undergo learning process (i.e., Equation 6) to capture the respective knowledge memes (the number of memes equals to the number of instances in $\mathbf{P}_{solved}$), which are then stored in the knowledge meme pool $\mathbf{SoM}$. For a new across domain routing problem instance $\mathbf{p}_{unseen}$ posed to the evolutionary solver, the selection and assimilation process kick in to generate knowledge meme biased routing solutions, which will be subsequently injected into the initial population of the evolutionary search to guide the search process. However, if the cross domain past solved problems are not available, the evolutionary search on the new encountered problem shall operate as routine.

V. EMPIRICAL STUDY

To investigate the feasibility of learning and evolution of knowledge memes across problem domains for enhanced evolutionary search, empirical studies conducted between the challenging NP-hard CVRP and CARP domains are presented in this section. In particular, 10 commonly used CVRP benchmark instances and 10 well-known CARP benchmark instances of diverse properties in terms of vertices size, graph topologies, etc., are considered here. The detailed properties of the CVRP and CARP instances are described in Table I

TABLE I
PROPERTIES OF THE CVRP INSTANCES.

Data	1. A-n54-k7	2. A-n69-k9	3. B-n57-k7	4. B-n78-k10	5. P-n50-k7	6. E-n76-k8	7. E-n101-k8	8. c75	9. c100b	10. c199
V	53	68	57	77	49	75	100	75	100	199
C_v	100	100	100	100	100	180	200	140	200	200
V_n	7	9	7	10	7	8	8	N/A	N/A	N/A

TABLE II
PROPERTIES OF THE CARP INSTANCES.

Data	1. E1C	2. E2C	3. E3C	4. E4B	5. E4C	6. S1C	7. S2C	8. S3C	9. S4B	10. S4C
V	77	77	77	77	77	140	140	140	140	140
C_v	160	140	135	180	130	103	120	120	160	120
V_n	10	14	17	14	19	14	27	29	27	35
E_r	51	72	87	98	98	75	147	159	190	190
E	98	98	98	98	98	190	190	190	190	190

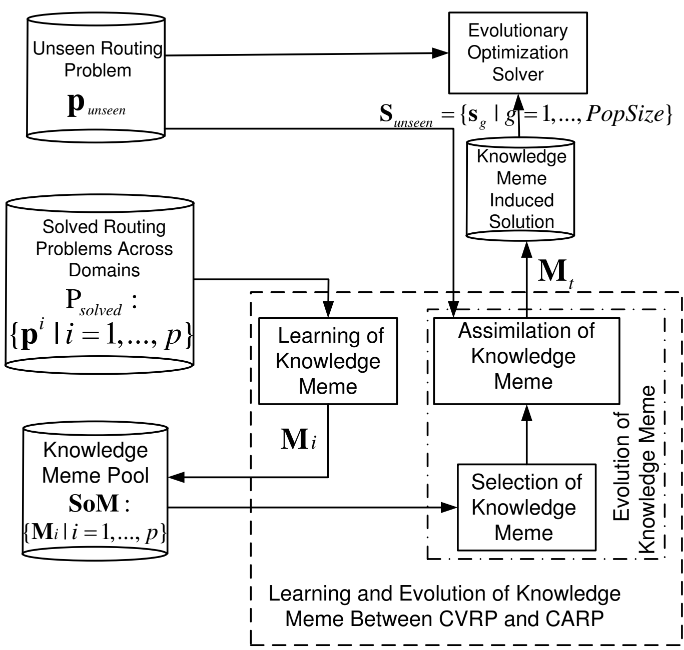


Fig. 7. An overview on the workflow of knowledge meme evolution between CVRP and CARP.

and Table II, respectively. In Table I, V denotes the number of vertices that need to be served, C_v gives the capacity of the vehicle in each problem instance, and V_n denotes the number of vehicle available[2]. In Table II, "V", "C_v", "V_n", "E_R" and "E" denote the number of vertices, capacity of vehicle, number of vehicles, number of tasks and total number of edges, of each CARP problem instance, respectively.

Further, two recent state-of-the-art evolutionary algorithms for solving CVRP and CARP, labeled in their respective published works as *CAMA* [35] and *ILMA* [36], are considered as the baseline conventional evolutionary solvers for the respective domains in the present study. In *CAMA*, the initial population is configured according to [35], as a fusion of solutions generated by *Backward Sweep* [47], *Saving* [48], *Forward Sweep* [47] and random initialization approaches, while in *ILMA*, the initial population is a fusion of chromosomes generated from *Augment_Merge* [34], *Path_Scanning* [49], *Ulusoy's Heuristic* [50] and the simple random initialization procedures. For the setup of evolutionary search with knowledge meme derived from different problem domains, the best solutions of the CARP and CVRP instances are used as the search experiences in each problem domain. *CAMA-K* and *ILMA-K* denotes the baseline solver with prior knowledge meme derived from CARP domain and CVRP domain, respectively. In particular, their initial populations are generated based on the evolved knowledge meme as discussed in section IV-2.

Last but not the least, the operator and parameter settings of *CAMA-K* and *ILMA-K* are kept the same as that of [35] and [36] for the purpose of fair comparison. In what follows, the empirical studies are presented to answer the three questions on knowledge meme transmission across problem domains for evolutionary optimization.

- Can evolutionary optimization benefit from knowledge meme across problem domains?
- How do different knowledge memes across problem domains influence the evolutionary search?
- What knowledge meme across problem domains would lead to enhanced evolutionary search?

A. Can Evolutionary Search Benefit from Different Problem Domains?

Here we assume that the optimized solutions for the CARP instances are available and use them as available problem solving experiences in the CAR problem domain to deal with unsolved CVRPs. Incidentally, when solving CARPs, the optimized solutions of the CVRPs are used as existing search experiences.

1) Solving CVRP with Knowledge Memes from CARP Domain: All results obtained on the CVRP instances by *CAMA* solver over 30 independent runs are summarized in Table III. $B.Cost$, $Ave.Cost$ and $Std.Dev$ denote the best solution with minimum cost, averaged cost of best solution obtained in each run and the standard deviation of the optimized solutions across 30 independent runs, respectively. $B.Gap$ measures the

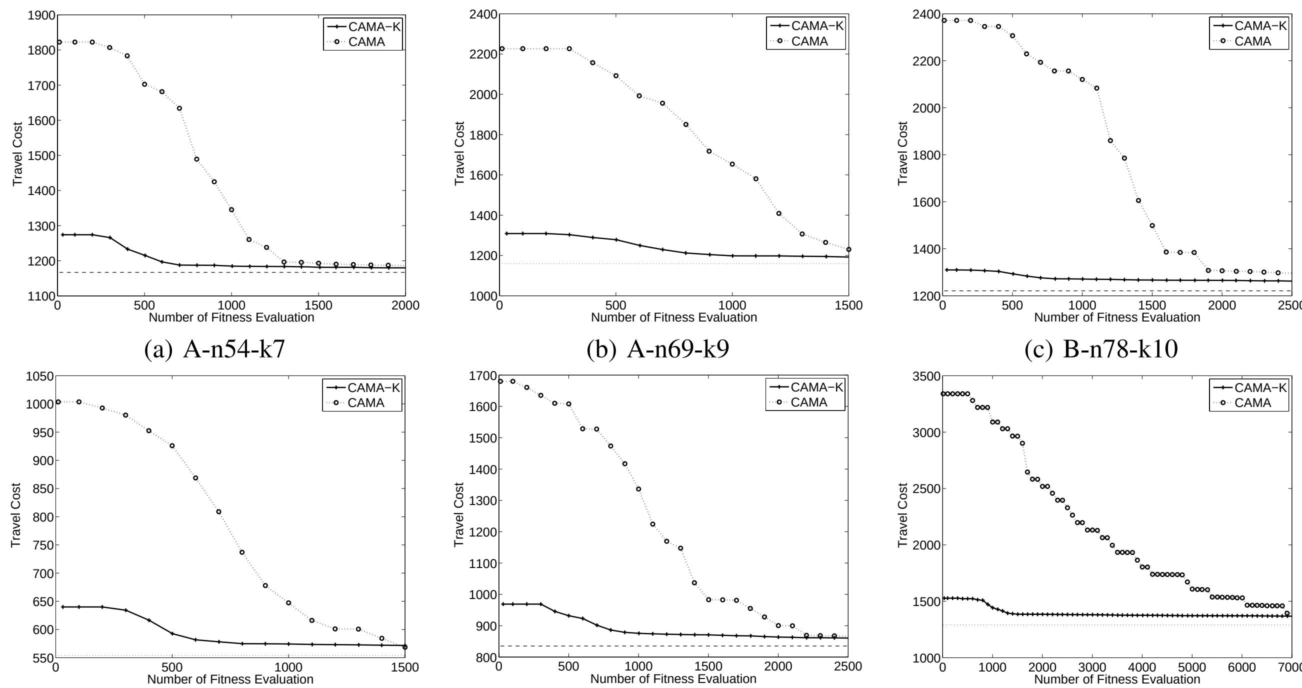
[2]The value N/A means it would be any number for the purpose of minimizing the total travel cost.

TABLE III
STATISTICAL RESULTS OF *CAMA-K* AND *CAMA* ON CVRP BENCHMARKS.("≈", "+" AND "−" DENOTE *CAMA-K* STATISTICALLY SIGNIFICANT SIMILAR, BETTER, AND WORSE THAN *CAMA*, RESPECTIVELY).

CVRP	$CAMA$					$CAMA-K$				
Instances	$B.Cost$	$Ave.Cost$	$Std.Dev$	$B.Gap$	$Ave.Gap$	$B.Cost$	$Ave.Cost$	$Std.Dev$	$B.Gap$	$Ave.Gap$
1. A-n54-k7	1167.00	1168.13	2.58	0	1.13	1167.00	**1167.00**+	0.00	0	**0**
2. A-n69-k9	1159.00	1162.87	2.81	0	3.87	1159.00	**1162.00** ≈	2.39	0	**3.00**
3. B-n57-k7	1140.00	1140.00	0.00	0	0	1140.00	1140.00 ≈	0.00	0	0
4. B-n78-k10	1221.00	1222.70	0.95	0	1.70	1221.00	**1222.60** ≈	0.44	0	**1.60**
5. P-n50-k7	554.00	556.00	2.03	0	2.00	554.00	**554.67** ≈	1.52	0	**0.67**
6. E-n76-k8	735.00	738.30	2.42	0	3.30	735.00	**736.70** ≈	1.95	0	**1.70**
7. E-n101-k8	**815.00**	819.37	3.51	**0**	4.37	817.00	**819.07** ≈	2.03	2.00	**4.07**
8. c75	835.26	840.45	2.79	0	5.19	835.26	**839.83** ≈	3.75	0	**4.57**
9. c100b	819.56	819.56	0.00	0	0	819.56	819.56 ≈	0.00	0	0
10. c199	1305.61	1318.71	7.67	14.16	27.26	**1301.00**	**1315.07**+	7.93	**9.55**	**23.62**

TABLE IV
COMPUTATIONAL COST SAVING ATTAINED BY *CAMA-K* OVER *CAMA* IN TERMS OF FITNESS EVALUATION (FE).

Data	A-n54-k7	A-n69-k9	B-n57-k7	B-n78-k10	P-n50-k7	E-n76-k8	E-n101-k8	c75	c100b	c199
FE_{saved} (%)	39.08%	50.01%	49.79%	68.07%	13.43%	46.18%	43.79%	49.96%	59.87%	76.23%



(a) A-n54-k7 (b) A-n69-k9 (c) B-n78-k10

(d) P-n50-k7 (e) c75 (f) c199

Fig. 8. Averaged search convergence graphs of *CAMA* and *CAMA-K* on representative CVRP benchmark instances. Y-axis: Travel Cost, X-axis: Number of Fitness Evaluation. (The dashed line at the bottom of each figure denotes the lower bound or best known solution of the respective benchmark reported in the literature.)

difference between the best-found value and the lower bound value of a benchmark instance, while $Ave.Gap$ gives the difference between the $Ave.Cost$ value and the lower bound value of each instance. Superior performance are highlighted in bold font. Further, in order to obtain the statistically comparison, Wilcoxon rank sum test with 95% confidence level has been conducted on the experimental results.

It can be observed from the results in Table III that overall, *CAMA-K* achieved competitive or improved solution quality over *CAMA* in terms of $Ave.Cost$ on all of the CVRP instances. In particular, with prior knowledge from the domain of CARP, *CAMA-K* obtained better $Ave.Cost$ value on 8 out of the 10 instances. Further, on instance "A-n54-k7", *CAMA-K* consistently converged to the optimal solution over all the 30 independent runs (i.e., the corresponding "$Std.Dev$" is "0".).

Subsequently, to access the efficiency of our proposed approach, the average convergence graphs on the CVRP benchmark instances are depicted in Fig. 8[3]. In the figure, the dashed line at the bottom of each figure denotes the lower bound or best known solution of the corresponding

[3]Due to page limit constraints, only representatives of each series have been shown.

TABLE V
STATISTICAL RESULTS OF *ILMA-K* AND *ILMA* ON CARP BENCHMARKS.("$\approx$", "$+$" AND "$-$" DENOTE *ILMA-K* STATISTICALLY SIGNIFICANT SIMILAR, BETTER, AND WORSE THAN *ILMA*, RESPECTIVELY).

CARP	$ILMA$					$ILMA-K$				
Instances	$B.Cost$	$Ave.Cost$	$Std.Dev$	$B.Gap$	$Ave.Gap$	$B.Cost$	$Ave.Cost$	$Std.Dev$	$B.Gap$	$Ave.Gap$
1. E1C	5595.00	5600.13	8.10	29.00	34.13	5595.00	**5598.40** $\approx$	6.56	29.00	**32.40**
2. E2C	8335.00	8356.00	37.52	92.00	113.00	8335.00	**8349.20**$+$	26.49	92.00	**106.20**
3. E3C	10292.00	10326.77	42.65	129	163.77	10292.00	**10314.50**$+$	28.86	129	**151.50**
4. E4B	8998.00	**9051.67**	57.96	114	**167.67**	8998.00	9053.17 $\approx$	49.04	114	169.17
5. E4C	**11570.00**	**11703.73**	71.50	**143**	**276.73**	11602.00	11704.93 $\approx$	81.81	175	277.93
6. S1C	8518.00	8573.33	35.04	25	80.33	8518.00	**8567.07** $\approx$	36.91	25	**74.07**
7. S2C	16504.00	16630.63	61.45	151	277.63	**16466.00**	**16608.20**$+$	67.39	**113**	**255.20**
8. S3C	**17257.00**	17391.07	75.34	**157**	291.07	17258.00	**17368.10**$+$	69.08	158	**268.10**
9. S4B	16424.00	16516.13	64.77	331	423.13	**16397.00**	**16509.10** $\approx$	63.58	**304**	416.10
10. S4C	20666.00	20809.87	72.40	291	434.87	**20624.00**	**20793.63** $\approx$	80.38	**249**	**418.63**

TABLE VI
COMPUTATIONAL COST SAVING ATTAINED BY *ILMA-K* OVER *ILMA* IN TERMS OF FITNESS EVALUATION (FE).

Data	E1C	E2C	E3C	E4B	E4C	S1C	S2C	S3C	S4B	S4C
FE_{saved} (%)	34.66%	52.72%	49.39%	−10.58%	−11.92%	25.88%	32.83%	19.85%	10.86%	7.55%

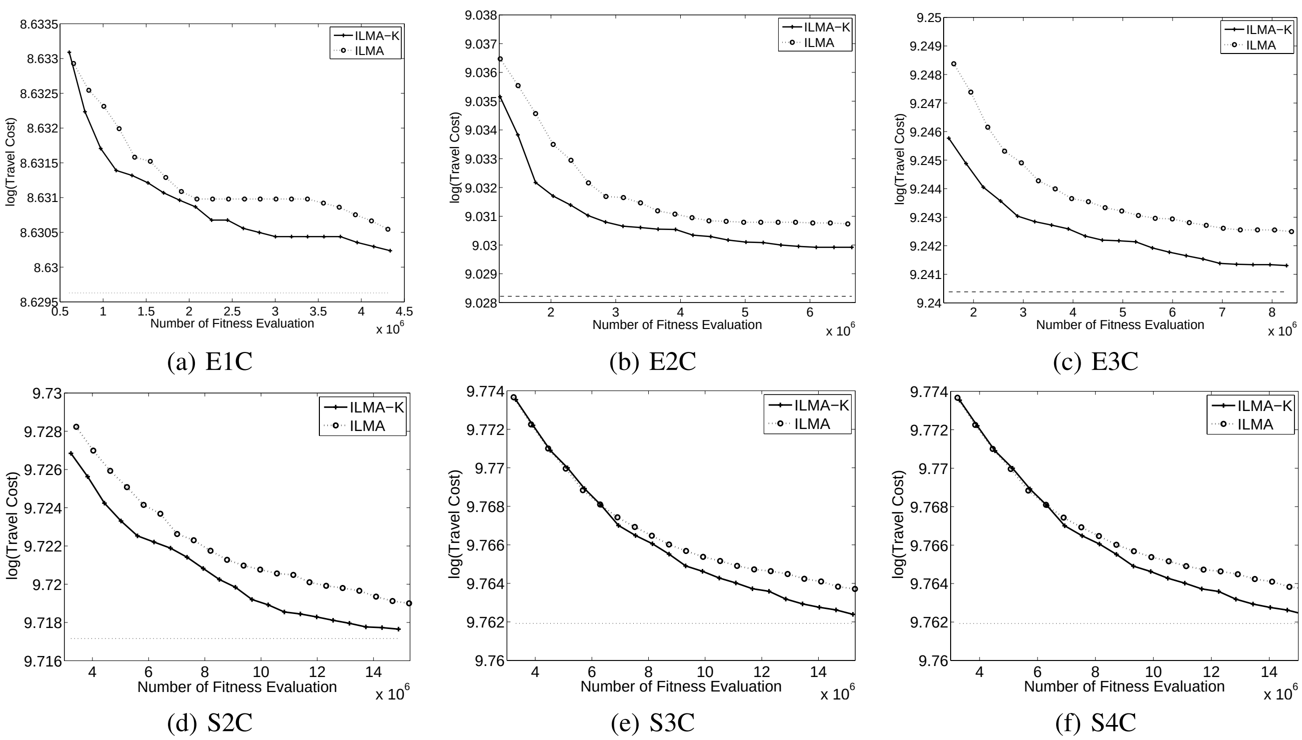


Fig. 9. Averaged search convergence graphs of *ILMA* and *ILMA-K* on representative CARP benchmark instances. Y-axis: log(Travel Cost), X-axis: Number of Fitness Evaluation. (The dashed line at the bottom of each figure denotes the lower bound or best known solution of the respective benchmark reported in the literature.)

benchmark instance reported in the literature [35]. As can be observed, on all the CVRP instances, *CAMA-K* achieved superior performance over its counterpart *CAMA*. In particular, the initial start points of *CAMA-K* on "B-n78-k10", "c75" are already very close to the respective lower bound solutions, and *CAMA-K* takes only about 1000 fitness evaluations to arrive at the solution obtained by *CAMA* over 2000 number of fitness evaluations. Further, on the larger size instance "c199", more significant number of fitness evaluations have been obtained by *CAMA-K* when compared to *CAMA*.

Further, to provide more intuitional insight on the resultant efficiency of search speed, we investigate how much fitness evaluation (FE) have been saved by *CAMA-K* to arrive at the converged solution obtained by its counterparts on each CVRP benchmark instance. The saving is defined as:

$$FE_{saved} = \frac{N_{cs}(A) - N_{cs}(A\text{-}K)}{N_{cs}(A)} \times 100\% \qquad (9)$$

where $N_{cs}(\cdot)$ denotes the number of fitness evaluations used by the investigating algorithm to arrive at a given solution cs. Symbol A stands for the investigating algorithm (i.e., *CAMA*

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/TEVC.2014.2362558, IEEE Transactions on Evolutionary Computation

here). If algorithm *A-K* obtained poor average convergence solution, cs is then set as the average convergence solution of *A-K*. Otherwise, cs is configured as the average convergence solution of algorithm *A*. It is worth noting that a positive FE_{saved} value means the search of *A-K* is more efficient than *A*, and on the other hand, a negative FE_{saved} value denotes *A-K*'s search is slower than its counterpart *A*. A higher FE_{saved} value denotes more fitness evaluation are saved by *A-K* to arrive at the solution quality level obtained by *A*.

The investigation results of FE_{saved} on all the CVRP benchmark instances considered are presented in Table IV. With prior knowledge meme derived from CARP domain, *CAMA-K* has brought about up to 76% saving in terms of fitness evaluations over *CAMA* (i.e., "c199").

2) Solving CARP with Knowledge Memes from CVRP Domain: On the other hand, all the results obtained on the CARP instances by *ILMA* over 30 independent runs are presented in this section. In particular, Table V gives the solution quality comparison between *ILMA-K* and *ILMA*. In order to obtain the statistically comparison, Wilcoxon rank sum test with 95% confidence level has been conducted on the experimental results. Fig. 9 presents the respective average convergence graphs of *ILMA-K* and *ILMA* to access the efficiency of the proposed approach[4]. The dashed line at the bottom of each figure denotes the lower bound or best known solution of the corresponding benchmark instance reported in the literature [36]. Further, the fitness evaluation savings obtained by *ILMA-K* over *ILMA* are summarized in Table VI.

As can be observed, with prior knowledge meme from CVRP domain, superior or competitive performance of *ILMA-K* can be observed from Table V over *ILMA*, on most of the considered CARP instances in terms of "$Ave.Cost$". Further, enhanced evolutionary search can be observed on *ILMA-K* over its counterpart *ILMA* in Fig. 9. In particular, *ILMA-K* brings about up to 52% fitness evaluation savings over *ILMA* and obtained more superior search performance on 8 out of 10 instances. Since the only difference between *ILAM* and *ILMA-K*, lies in the prior knowledge introduced in the population initialization phase of the latter, the superior performance of *ILMA-K* can clearly be attributed to the effectiveness of the knowledge meme transmission across problem domains. In summary, the achieved enhanced performance of *ILMA-K* and *CAMA-K* confirmed that evolutionary search can benefit from different but related problem domains.

B. How do the Knowledge Memes of Related Problem Domain Affect the Evolutionary Search?

To gain a better understanding of knowledge meme transmission across problem domains for enhanced evolutionary optimization, we further analyze and compare the proposed approach with knowledge meme transmission where the meme is from the most and least similar problem instance across domains. In particular, the maximum mean discrepancy in Equation 7 is used here to measure the similarity between problem instances. Let *A-BM* and *A-WM* denote the baseline evolutionary solver *A* with initial population generated by the knowledge meme derived from the most and least similar problem instance across domains, respectively. In the present context, *A* stands for *CAMA* for solving CVRP, and denotes *ILMA* for solving CARP.

The average convergence graphes on representative CVRP and CARP instances are depicted in Fig. 10 and Fig. 11, respectively. As can be observed, in Fig. 10, with knowledge meme transmitted from CARP problem domain, *CAMA-BM*, *CAMA-WM* and *CAMA-K* achieved efficient search performance than the baseline *CAMA*. Further, among *CAMA-BM*, *CAMA-WM* and *CAMA-K*, *CAMA-WM* did not perform as competitive to the other two counterparts on all the CVRP instances due to the incorporation of low similarity knowledge meme. In contrast, the proposed approach *CAMA-K* is observed to showcase superior performances, which is achieved through a generalization of the multiple highly similar memes in positively biasing the search towards high quality solutions. For CARP, in Fig. 11, negative effects of *ILMA-WM* have been observed on all the CARP instances when compared to the baseline *ILMA*. However, with knowledge meme from most similar CVRP instance, *ILMA-BM* achieved superior performance than baseline *ILMA*. Overall, with the proposed knowledge meme transmission, *ILMA-K* attained better performance than the others.

In summary, different knowledge memes will introduce unique biases into the evolutionary search, while inappropriately chosen knowledge memes can lead to negative impairments of the evolutionary search process. Further, the comparisons conducted in this section also confirmed the effectiveness of the proposed selection scheme for enhanced search.

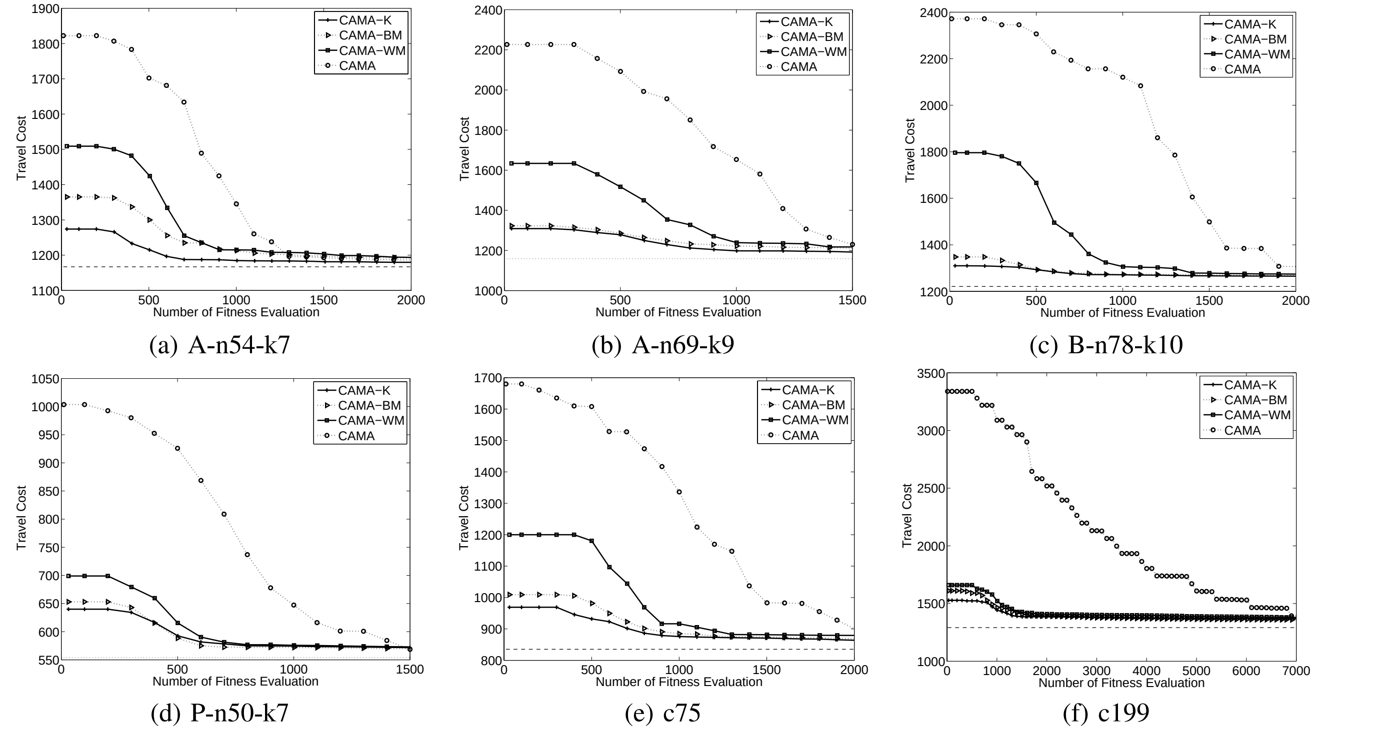
C. What forms of Knowledge Memes from Related Problem Domains Benefit Evolutionary Optimization?

In the empirical studies presented above, we observed positive as well as negative effected evolutionary search caused by knowledge meme transmission across problem domains. Here, we further study the possible reasons behind the various performances obtained, and find out what knowledge meme across problem domains would enhance the evolutionary search.

In particular, we first depict the discrepancies obtained by each CVRP instance against all CARPs and each CARP against all CVRPs obtained by Equation 7 (i.e., maximum mean discrepancy) in the established common feature space. As can be observed, in Fig. 12, from CVRP instance "1" (i.e., "A-n54-k7") to "4" (i.e., "B-n78-k10"), there is a decreasing trend on the corresponding discrepancies. Referring to the computational cost saving for *CAMA-K* over *CAMA* in Table IV, an increasing trend of reduced computational cost can be observed from "A-n54-k7" to "B-n78-k10" in general. Further, on CVRP instance "5" (i.e., "P-n50-k7"), its discrepancies go up, while the corresponding fitness evaluation (FE) saving drops to below 20%. Subsequently, when the discrepancy between CVRP instance and CARPs drops from instance 6 (i.e., "E-n76-k8") to 10 (i.e., "c199"), the respective FE saved by *CAMA-K* over *CAMA* increases again.

[4]Due to page limit constraints, only representatives of each series have been shown.

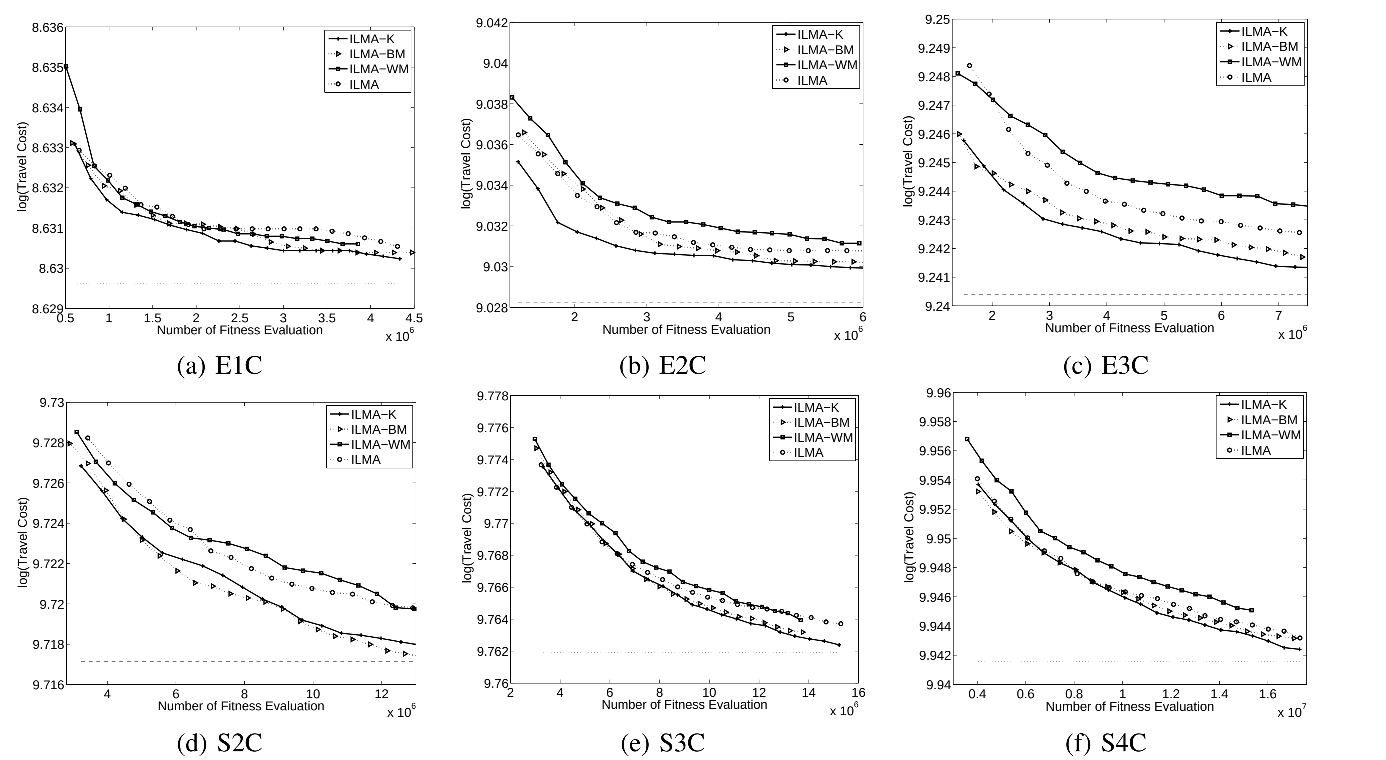
This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/TEVC.2014.2362558, IEEE Transactions on Evolutionary Computation



(a) A-n54-k7 (b) A-n69-k9 (c) B-n78-k10

(d) P-n50-k7 (e) c75 (f) c199

Fig. 10. Averaged search convergence graphs of *CAMA*, *CAMA-WM*, *CAMA-BM* and *CAMA-K* on representative CVRP benchmark instances. Y-axis: Travel Cost, X-axis: Number of Fitness Evaluation. (The dashed line at the bottom of each figure denotes the lower bound or best known solution of the respective benchmark reported in the literature.)



(a) E1C (b) E2C (c) E3C

(d) S2C (e) S3C (f) S4C

Fig. 11. Averaged search convergence graphs of *ILMA*, *ILMA-WM*, *ILMA-BM* and *ILMA-K* on representative CARP benchmark instances. Y-axis: log(Travel Cost), X-axis: Number of Fitness Evaluation. (The dashed line at the bottom of each figure denotes the lower bound or best known solution of the respective benchmark reported in the literature.)

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/TEVC.2014.2362558, IEEE Transactions on Evolutionary Computation

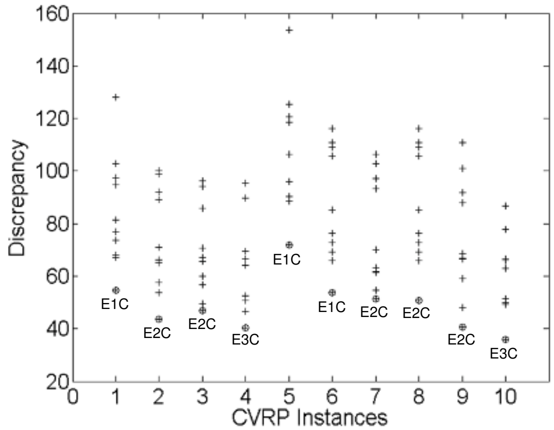


Fig. 12. Discrepancies of each CVRP instances against all CARPs. "+" denotes a discrepancy obtained between the CVRP instance and a CARP instance, and "$\oplus$" is the minimum discrepancy obtained by each CVRP instances. The name (e.g., "E1C") specified along each $\oplus$ denotes the CARP instance, on which the respective CVRP instance achieves the minimum discrepancy.

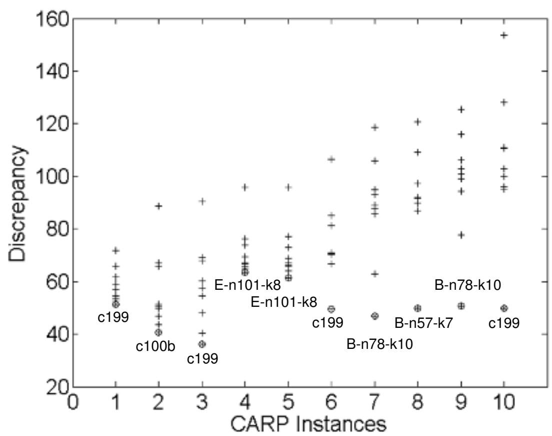


Fig. 13. Discrepancies of each CARP instances against all CVRPs. "+" denotes a discrepancy obtained between the CARP instance and a CVRP instance, and "$\oplus$" is the minimum discrepancy obtained by each CARP instances. The name (e.g., "c199") specified along each $\oplus$ denotes the CVRP instance, on which the respective CARP instance achieves the minimum discrepancy.

On the other hand, for CARP in Fig. 13, the discrepancies drop first from CARP instance "1" (i.e., "E1C") to "2" (i.e., "E2C"), and increase subsequently on the later CARP instances. Coupled with the computational cost saving for *ILMA-K* over *ILMA* in Table VI, we can see that the FE saved by *ILMA-K* goes up from 34.66% to 52.72% from "E1C" to "E2C" and drops to -11.92% at CARP instance "5" (i.e., "E4C") when the corresponding discrepancies increased. Further, the FE saving by *ILMA-K* goes up again at CARP instance "6" (i.e., "S1C"), although most of the discrepancies are increased. However, it is worth noting that from CARP instance "6" (i.e., "S1C") to "10" (i.e., "S4C"), lower minimum discrepancy can be observed compared to that of CARP instance "4" (i.e., "E4B") and "5" (i.e., "E4C"). Here, the minimum discrepancy of each CVRP instance against all the CARPs and each CARP against all the CVRPs in the established common feature space (i.e., Section III-A) is given by:

$$D_{vp} = min_{i=\{1,\ldots,t\}} MMD(vp, ap_i) \quad (10)$$
$$D_{ap} = min_{i=\{1,\ldots,t\}} MMD(ap, vp_i) \quad (11)$$

where vp and ap denote a CVRP and CARP instance, respectively. t is the number of CARP instance or CVRP instances. In the present context, t equals to 10. $MMD(\cdot)$ is the maximum mean discrepancy used in Equation 7. In Fig. 12 and Fig. 13, "$\oplus$" denotes the respective minimum discrepancy. The name specified along each $\oplus$ denotes the instance across problem domain, on which the current problem instance achieves the minimum discrepancy. As most of the discrepancies are increasing from CARP instance "6" to "10", the FE savings obtained on these instances indicate that knowledge meme from the CVRP instance with minimum discrepancy would play a dominant role via its selection weight (i.e., μ in Eq. 7) to bias the search for solving CARP.

In what follows, we further investigate the computation cost savings in terms of fitness evaluation achieved by *CAMA-K* and *ILMA-K* with respect to the minimum discrepancy obtained by each problem instance across domain. As depicted in Fig. 14 and Fig. 15, the small circles denote the particular FE saving attained by *CAMA-K* or *ILMA-K* with a minimum discrepancy obtained cross problem domain. The straight line is a linear regression of the depicted small circles. In general, an inversely-proportional relationship can be observed between FE savings and the corresponding minimum discrepancy in both CVRP and CARP problem domain.

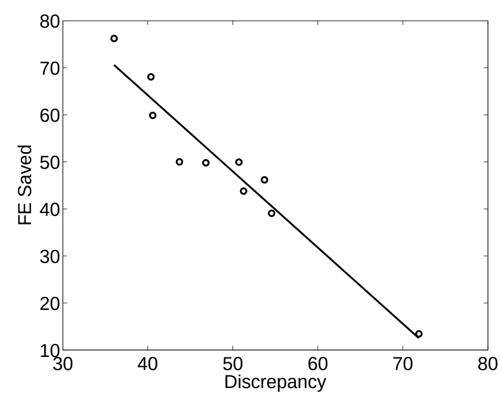


Fig. 14. FE saving attained by *CAMA-K* over *CAMA* with respect to the minimum discrepancies obtained by the CVRP instances against all CARPs.

In summary, from the observed relationship between FE savings and discrepancies between instances across problem domains, we can infer that enhanced evolutionary search would be obtained with knowledge meme transmission across domains when low discrepancy existed between the respective problem instances. If the problem instance of interest is very different from existing solved problems, the corresponding knowledge transmission would not be helpful for evolutionary search, but may even lead to negative impairments on the search process. To ensure an enhanced evolutionary search by knowledge meme transmission across problem domain, the discrepancy measure, i.e., Equation 10 and Equation 11, in the

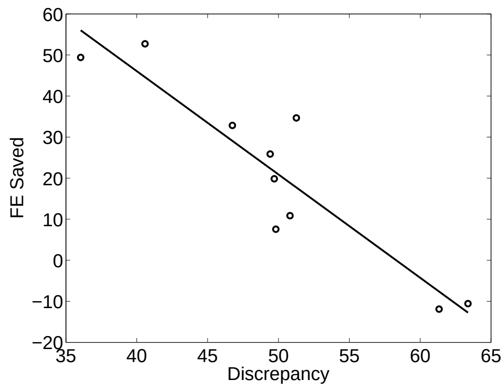


Fig. 15. FE saving attained by *ILMA-K* over *ILMA* with respect to the minimum discrepancies obtained by the CARP instances against all CVRPs.

common feature space would be used as a criterion to evaluate whether low discrepancy existed between problem domains.

VI. Conclusion

In this paper, we have presented a study on the feasibility of evolutionary paradigm capable of learning and evolving knowledge meme across different but related problem domains for the purpose of enhanced search. Based on two NP-hard combinatorial optimization problems, i.e., CVRP and CARP, we derived their common problem representation, identified their useful knowledge meme representation, and proposed how to capture and transmit the knowledge meme between CVRP and CARP evolutionary search. Empirical results show that evolutionary optimization can benefit from different but related problem domain. However, the appropriately choice of knowledge meme is crucial for enhancing the evolutionary search process. Further, by studying the performances of meme biased evolutionary search and the discrepancies between problems in different domains, we found that enhanced evolutionary optimization would be obtained from knowledge meme transmission when low discrepancy existed between the respective problems in the established common feature space.

It would be desirable that future works investigate realizations of the knowledge meme learning and evolution on more combinatorial optimization problem domains to confirm the effectiveness of the methodology and discover the relations as well as useful traits of evolutionary searches in different problem domains. Furthermore, since the discrepancy presented in the empirical study has already demonstrated a relation to positive or negative knowledge meme transmission across problem domains, greater in-depth study on the discrepancy between problems for enhanced evolutionary search would be beneficial.

Acknowledgment

This work is partially supported under the A∗STAR-TSRP funding, the Singapore Institute of Manufacturing Technology-Nanyang Technological University (SIMTech-NTU) Joint Laboratory Collaborative Research Programme on Complex Systems, the Centre for Computational Intelligence at NTU, and the Australian Research Council Future Fellowship FT130100746.

References

[1] F. Neri, G. L. Cascella, N. Salvatore, G. Acciani, and D. A. Gassi, "A hierarchical evolutionary-deterministic algorithm in topological optimization of electrical grounding grids," *WSEAS Transactions on Systems*, no. 12, pp. 2338–2345, 2005.
[2] Y. P. Chen and Y. Y. Lin, "Controlling the movement of crowds in computer graphics by using the mechanism of particle swarm optimization," *Applied Soft Computing*, vol. 9, no. 3, pp. 1170–1176, 2009.
[3] K. Tang, Y. Mei, and X. Yao, "Memetic algorithm with extended neighborhood search for capacitated arc routing problems," *IEEE Transactions on Evolutionary Computation*, vol. 13, no. 5, pp. 1151–1166, 2009.
[4] K. Tang, F. Peng, G. L. Chen, and X. Yao, "Population-based algorithm portfolios with automated constituent algorithms selection," *Information Sciences*, vol. 279, no. 0, pp. 94 – 104, 2014.
[5] Y. Mei, K. Tang, and X. Yao, "Decomposition-based memetic algorithm for multi-objective capacitated arc routing problems," *IEEE Transactions on Evolutionary Computation*, vol. 15, no. 2, pp. 151–165, 2011.
[6] ——, "A global repair operator for capacitated arc routing problem," *IEEE Transactions on Systems, Man, and Cybernetics: Part B*, vol. 39, no. 3, pp. 723–734, 2009.
[7] ——, "A memetic algorithm for periodic capacitated arc routing problem," *IEEE Transactions on Systems, Man, and Cybernetics: Part B*, vol. 41, no. 6, pp. 1654–1667, 2011.
[8] Y. C. Jin, *Knowledge Incorporation in Evolutionary Computation*, ser. Studies in Fuzziness and Soft Computing. Springer, 2010.
[9] Y. S. Ong, N. Krasnogor, and H. Ishibuchi, "Special issue on memetic algorithm," *IEEE Transactions on Systems, Man and Cybernetics - Part B*, vol. 37, no. 1, pp. 2–5, 2007.
[10] Y. S. Ong, M. H. Lim, F. Neri, and H. Ishibuchi, "Special issue on emerging trends in soft computing: memetic algorithms," *Soft Computing-A Fusion of Foundations, Methodologies and Applications*, vol. 13, no. 8-9, pp. 1–2, 2009.
[11] M. H. Lim, S. Gustafson, N. Krasnogor, and Y. S. Ong, "Editorial to the first issue, memetic computing," *Soft Computing-A Fusion of Foundations, Methodologies and App.*, vol. 1, no. 1, pp. 1–2, 2009.
[12] J. E. Smith, "Co-evolving memetic algorithms: A review and progress report," *IEEE Transactions on Systems, Man and Cybernetics - Part B*, vol. 37, no. 1, pp. 6–17, 2007.
[13] I. Paenke, Y. Jin, and J. Branke, "Balancing population-and individual-level adaptation in changing environments," *Adaptive Behavior*, vol. 17, no. 2, pp. 153–174, 2009.
[14] G. Gutin and D. Karapetyan, "A selection of useful theoretical tools for the design and analysis of optimization heuristics," *Memetic Computing*, vol. 1, no. 1, pp. 25–34, 2009.
[15] J. Tang, M. H. Lim, and Y. S. Ong, "Diversity-adaptive parallel memetic algorithm for solving large scale combinatorial optimization problemsg," *Soft Computing Journal*, vol. 11, no. 9, pp. 873–888, 2007.
[16] Z. Zhu, Y. S. Ong, and M. Dash, "Wrapper-filter feature selection algorithm using a memetic framework," *IEEE Transactions On Systems, Man and Cybernetics - Part B*, vol. 37, no. 1, pp. 70–76, 2007.
[17] S. M. K. Hasan, R. Sarker, D. Essam, and D. Cornforth, "Memetic algorithms for solving job-shop scheduling problems," *Memetic Computing*, vol. 1, no. 1, pp. 69 – 83, 2009.
[18] M. Tang and X. Yao, "A memetic algorithm for vlsi floorplanning," *IEEE Transactions on Systems, Man and Cybernetics - part B*, vol. 37, no. 1, pp. 62–69, 2007.
[19] P. Cunningham and B. Smyth, "Case-based reasoning in scheduling: Reusing solution components," *The International Journal of Production Research*, vol. 35, no. 4, pp. 2947–2961, 1997.
[20] S. J. Louis and J. McDonnell, "Learning with case-injected genetic algorithms," *IEEE Transactions on Evolutionary Computation*, vol. 8, no. 4, pp. 316–328, 2004.
[21] L. Feng, Y. S. Ong, I. W. Tsang, and A. H. Tan, "An evolutionary search paradigm that learns with past experiences," *IEEE World Congress on Computational Intelligence, Congress on Evolutionary Computation*, 2012.
[22] R. Dawkins, "The selfish gene," *Oxford: Oxford University Press*, 1976.
[23] F. Neri, C. Cotta, and P. Moscato, *Handbook of Memetic Algorithms*, ser. Studies in Computational Intelligence. Springer, 2011.
[24] Y. S. Ong, M. H. Lim, and X. S. Chen, "Research frontier: Memetic computation - past, present & future," *IEEE Computational Intelligence Magazine*, vol. 5, no. 2, pp. 24–36, 2010.

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/TEVC.2014.2362558, IEEE Transactions on Evolutionary Computation

[25] X. S. Chen, Y. S. Ong, M. H. Lim, and K. C. Tan, "A multi-facet survey on memetic computation," *IEEE Transactions on Evolutionary Computation, In Press*, no. 5, pp. 591–607, 2011.
[26] Q. H. Nguyen, Y. S. Ong, and M. H. Lim, "A probabilistic memetic framework," *IEEE Transactions on Evolutionary Computation*, vol. 13, no. 3, pp. 604–623, 2009.
[27] L. Feng, Y. S. Ong, A. H. Tan, and X. S. Chen, "Towards human-like social multi-agents with memetic automaton," *IEEE Congress on Evolutionary Computation*, pp. 1092–1099, 2011.
[28] G. Iacca, F. Neri, E. Mininno, Y. S. Ong, and M. H. Lim, "Ockham's razor in memetic computing: Three stage optimal memetic exploration," *Information Sciences*, vol. 188, no. 0, pp. 17 – 43, 2012.
[29] F. Caraffini, F. Neri, G. Iacca, and A. Mol, "Parallel memetic structures," *Information Sciences*, vol. 227, no. 0, pp. 60 – 82, 2013.
[30] F. Caraffini, F. Neri, and L. Picinali, "An analysis on separability for memetic computing automatic design," *Information Sciences*, vol. 265, no. 0, pp. 1 – 22, 2014.
[31] E. K. Burke, M. Gendreau, M. Hyde, G. Kendall, G. Ochoa, E. Özcan, and R. Qu, "Hyper-heuristics: A survey of the state of the art," *Journal of the Operational Research Society*, pp. 1695–1724, 2013.
[32] E. K. Burke, M. Hyde, G. Kendall, G. Ochoa, E. Özcan, and J. R. Woodward, *A Classification of Hyper-heuristic Approaches*. Kluwer, 2010, ch. Handbook of Meta-Heuristics, pp. 449–468.
[33] G. Dantzig and J. H. Ramser, "The truck dispatching problem," *Management Science*, vol. 6, pp. 80–91, 1959.
[34] B. Golden and R. Wong, "Capacitated arc routing problems," *Networks*, vol. 11, no. 3, pp. 305–315, 1981.
[35] X. Chen and Y. S. Ong, "A conceptual modeling of meme complexes in stochastic search," *IEEE Transactions on Systems, Man, and Cybernetics, Part C: Applications and Reviews*, no. 99, pp. 1–8, 2012.
[36] Y. Mei, K. Tang, and X. Yao, "Improved memetic algorithm for capacitated arc routing problem," *IEEE Congress on Evolutionary Computation*, pp. 1699–1706, 2009.
[37] E. W. Dijkstra, "A note on two problems in connection with graphs," *Numerische Mathematik*, vol. 1, pp. 269–271, 1959.
[38] I. Borg and P. J. F. Groenen, *Modern Multidimensional Scaling: Theory and Applications*. Springer, 2005.
[39] C. Wang and S. Mahadevan, "Manifold alignment without correspondence," *Proceedings of the 21st International Jont Conference on Artifical intelligence*, pp. 1273–1278, 2009.
[40] F. Neri and E. Mininno, "Memetic compact differential evolution for cartesian robot control," *IEEE Computational Intelligence Magazine*, no. 2, pp. 54–65, 2010.
[41] C. K. Ting and C. C. Liao, "A memetic algorithm for extending wireless sensor network lifetime," *Information Science*, vol. 180, no. 24, pp. 4818–4833.
[42] S. J. L. Pan and Q. Yang, "A survey on transfer learning," *IEEE Transactions on Knowledge and Data Engineering*, vol. 22, no. 10, pp. 1345–1359, 2010.
[43] A. Gretton, O. Bousquet, A. Smola, and B. Schölkopf, "Measuring statistical dependence with hilbert-schmidt norms," *Proceedings of Algorithmic Learning Theory*, pp. 63–77, 2005.
[44] J. Zhuang, I. Tsang, and S. C. H. Hoi, "A family of simple non-parametric kernel learning algorithms," *Journal of Machine Learning Research (JMLR)*, vol. 12, pp. 1313–1347, 2011.
[45] K. M. Borgwardt, A. Gretton, M. J. Rasch, H. P. Kriegel, B. Schölkopf, and A. J. Smola, "Integrating structured biological data by kernel maximum mean discrepancy," *International Conference on Intelligent Systems for Molecular Biology*, pp. 49–57, 2006.
[46] L. Song, A. Smola, A. Gretton, and K. M. Borgwardt, "A dependence maximization view of clustering," *Proceedings of the 24th International Conference on Machine learning*, pp. 815–822, 2007.
[47] B. E. Gillett and L. R. Miller, "A heuristic algorithm for the vehicle-dispatch problem," *Op. Res.*, vol. 22, no. 2, pp. 340–349, 1974.
[48] G. Clarke and J. Wright, "Scheduling of vehicles from a central depot to a number of delivery points," *Operations Research*, vol. 12, no. 4, pp. 568–581, 1964.
[49] B. L. Golden, J. S. DeArmon, and E. K. Baker, "Computational experiments with algorithms for a class of routing problems," *Computer & Operation Research*, vol. 10, no. 1, pp. 47–59, 1983.
[50] G. Ulusoy, "The fleet size and mix problem for capacitated arc routing," *European Journal of Operational Research*, vol. 22, no. 3, pp. 329–337, 1985.

Liang Feng received his B.S. degree in the School of Telecommunication and Information Engineering from Nanjing University of Posts and Telecommunications, China, in 2009. Currently, he is working towards the Ph.D degree at the School of Computer Engineering, Nanyang Technological University. His primary research interests include evolutionary computations, memetic computing, transfer learning, and data mining, etc.

Yew-Soon Ong received a PhD degree on Artificial Intelligence in complex design from the Computational Engineering and Design Center, University of Southampton, UK in 2003. His current research interest in computational intelligence spans across memetic computing, evolutionary design, machine learning and Big data. He is the founding Technical Editor-in-Chief of Memetic Computing Journal, founding Chief Editor of the Springer book series on studies in adaptation, learning, and optimization, Associate Editor of the IEEE Transactions on Evolutionary Computation, the IEEE Transactions on Neural Networks & Learning Systems, IEEE Computational Intelligence Magazine, IEEE Transactions on Cybernetics, Soft Computing, International Journal of System Sciences and others.

Meng-Hiot Lim is an Associate Professor with the School of Electrical & Electronics Engineering, Nanyang Technological University. Besides his mainstream research in memetic computation, he has strong interest in financial engineering, particularly in the applications of computational intelligence in financial problems. He was one of the founding directors of the Centre for Financial Engineering and M.Sc in Financial Engineering program at the Nanyang Business School, serving as a Deputy Director for several years. Dr. Lim is the President of the Memetic Computing Society, President of Uavionics Society and the Managing Editor-in-Chief of Memetic Computing Journal. He is also a co-Editor-in-Chief of the ALO Book Series by Springer. As a co-founder of three startups, he is also passionate about promoting a "can-do" brand of engineering education, inspiring students to take charge of putting what they learn into hands-on practice. He recently set up the GarageEEE lab, a hub for engineering creativity and innovation.

Ivor W. Tsang is an Australian Future Fellow and Associate Professor with the Centre for Quantum Computation & Intelligent Systems (QCIS), at the University of Technology, Sydney (UTS). He received his PhD degree in computer science from the Hong Kong University of Science and Technology in 2007. His research focuses on kernel methods, transfer learning, feature selection, big data analytics for data with trillions of dimensions, and their applications to computer vision and pattern recognition. He has more than 100 research papers published in refereed international journals and conference proceedings, including JMLR, T-PAMI, T-NN, ICML, NIPS, UAI, AISTATS, SIGKDD, IJCAI, AAAI, ICCV, CVPR, ECCV, ACL, etc. Dr. Tsang received his prestigious Australian Research Council Future Fellowship in 2013, and had previous been awarded the 2008 Natural Science Award (Class II) by the Ministry of Education, China, the IEEE Transactions on Neural Networks Outstanding 2004 Paper Award in 2006 and IEEE Transactions on Multimedia 2014 Best Paper Award. His research also earned him the Best Student Paper Award at CVPR'10, the Best Paper Award at ICTAI'11, and the Best Poster Honorable Mention at ACML'12. He was also conferred with the Microsoft Fellowship in 2005.