Meme as Building Block for Evolutionary Optimization of Problem Instances

Liang Feng, Yew-Soon Ong, Ah-Hwee Tan and Ivor Wai-Hung Tsang*

Abstract

A significantly under-explored area of evolutionary optimization in the literature is the study of optimization methodologies that can evolve along with the problems solved. Particularly, present evolutionary optimization approaches generally start their search from scratch or the ground-zero state of knowledge, independent of how similar the given new problem of interest is to those optimized previously. There has thus been the apparent lack of automated knowledge transfers and reuse across problems. Taking the cue, this paper introduces a novel *Memetic Computational Paradigm for search*, one that models after how human solves problems, and embarks on a study towards intelligent evolutionary optimization of problems through the transfers of structured knowledge in the form of memes learned from previous problem-solving experiences, to enhance future evolutionary searches. In particular, the proposed memetic search paradigm is composed of four culture-inspired operators, namely, *Meme Learning*, *Meme Selection*, *Meme Variation* and *Meme Imitation*. The learning operator mines for memes in the form of latent structures derived from past experiences of problem-solving. The selection operator identifies the fit memes that replicate and transmit across problems, while the variation operator introduces innovations into the memes. The imitation operator, on the other hand, defines how fit memes assimilate into the search process of newly encountered problems, thus gearing towards efficient and effective evolutionary optimization. Finally, comprehensive studies on two widely studied challenging well established NP-hard routing problem domains, particularly, the capacitated vehicle routing (CVR) and capacitated arc routing (CAR), confirm the high efficacy of the proposed memetic computational search paradigm for intelligent evolutionary optimization of problems.

1 Introduction

Like gene in genetics, a meme is synonymous to memetic as being the building block of cultural know-how that is transmissible and replicable [10]. In the last

*Liang Feng, Yew-Soon Ong, Ah-Hwee Tan and Ivor Wai-Hung Tsang are with the Center for Computational Intelligence, School of Computer Engineering, Nanyang Technological University, Singapore. E-mail: {feng0039, asysong, asahtan, ivortsang}@ntu.edu.sg

decades, meme has inspired the new science of memetics which today represents the mind-universe analog to genetics in cultural evolution, stretching across the field of biology, cognition, psychology, and sociology [39].

Looking back on the history of *meme*, the term can be traced back to Dawkins [19] in his book "The selfish Gene", where he defined it as "a unit of information residing in the brain and is the replicator in human cultural evolution". Like genes that serve as "instructions for building proteins", memes are then "instructions for carrying out behavior, stored in brains". As discussed by Blackmore in her famous book "The Meme Machine", where she reaffirmed meme as information copied from one person to another and discussed on the theory of "memetic selection" as the survival of the fittest among competitive ideas down through generations [2]. Other definitions of meme that took flights from there have since emerged to include "memory item, or portion of an organism's neurally-stored information" [34], "unit of information in a mind whose existence influences events such that more copies of itself get created in other minds" [7], and "contagious information pattern that replicates by parasitically infecting human minds" [26].

In the context of computational intelligence, particularly evolutionary computation, a meme has been perceived as a form of individual learning procedure, adaptive improvement procedure or local search operator to enhance the capability of population based search algorithm [38, 37]. From the last decades, this integration has been established as an extension of the canonical evolutionary algorithm, by the names of hybrid, adaptive hybrid or Memetic Algorithm (MA) in the literature [10], where many success stories for solving complex real world search problems ranging from continuous optimization [37], combinatorial optimization [23, 45], constrained optimization [47], multi-objective optimization [29] to optimization in dynamic environment [49], etc., have been reported. In spite of the attention and successes that memetic algorithm has enjoyed, meme does seem to play much of a complimentary role in computational intelligence, i.e., in the form of a local search or refinement operator within the evolutionary search cycle.

Falling back on the basic definition of a meme by Dawkins and Blackmore, as the fundamental building blocks of culture evolution, research on memetic computation can perhaps be more meme-centric focus by treating memes as the building blocks of a given problem domain, much like gene serving as the building blocks of life. Taking the cue, this paper embarks on a study towards a novel *Memetic Computational Paradigm for search*, one that models after how human solves problems.

Today, it is well recognized that the processes of learning and the transfer of what has been learned are central to humans in problem-solving [6]. Learning has been established to be fundamental to human in functioning and adapting to the fast evolving society. Besides learning from the successes and mistakes of the past and learning to avoid making the same mistakes again, the ability of human in selecting, generalizing and drawing upon what have been experienced and learned in one context, and extending them to new problems is deem to be most remarkable [8]. Since the inception of artificial intelligence more than five decades ago, a significant number of success stories to emulate the learning, reasoning and

problem-solving intelligence of humankind have been made. The aspiration has been to develop machines that do things like humans are capable of performing intelligibly. Within the context of computational intelligence, several core learning technologies in neural and cognitive systems, fuzzy systems, probabilistic and possibilistic reasoning have been notable for their ability in emulating some of human's cultural and generalization capabilities, with many now used to enhance our daily life. *In spite of the accomplishments made in computational intelligence, the attempts to emulate the cultural intelligence of human in search, evolutionary optimization in particular, has to date received far less attention*. The learning, generalization and evolution of useful traits across related tasks or problems and the study of optimization methodology that evolves along with the problems solved has been significantly under-explored in the context of evolutionary computation. In the present study, we aspire to fill in this gap.

In particular, we propose a memetic search paradigm that takes advantage of the possible relations among problem instances, i.e., topological properties, data distribution or otherwise, to allow the effective assessments of future unseen problem instances, without the need to perform an exhaustive search each time or start the evolutionary search from a ground-zero knowledge state. The proposed memetic search paradigm is composed of four culture-inspired operators, namely, *Meme Learning*, *Meme Selection*, *Meme Variation* and *Meme Imitation*. The *learning operator* extracts memes in the form of structured knowledge from past experiences of problem-solving. In the present context, meme manifests as the instructions to intelligently bias the search (i.e., thus narrowing down the search space) on the given problems of interest. The *selection operator*, on the other hand, selects the appropriate or fit memes that shall replicate and undergo innovations via the *variation operator*, before drawing upon them for enhancing future evolutionary optimizations. Last but not least, the *imitation operator* defines the process of memes assimilation into the search process of new problems, via the generations of positive biased solutions that would lead to more efficient and effective evolutionary searches.

In this paper, we showcase the proposed memetic evolutionary search paradigm on combinatorial optimization problems. Particularly, we consider the general programming problem where there exists a number of agents and a number of tasks. Any agent can be assigned to perform any task, incurring some cost and profit that may vary, depending on the agent-task assignment. Moreover, each agent has a budget constraint and the sum of the costs of tasks assigned to it cannot exceed the constraint imposed. The objective is then to find an assignment with maximum total profit, while not exceeding the budget imposed on each agent. In the literature, several well established combinatorial optimization problems belonging to the described general programming problem include vehicle routing problem [31], generalized assignment problem [15], arc routing problem [23], etc. To demonstrate the memetic computational search paradigm for "intelligent" evolutionary optimization, the two widely studied challenging routing problem domains, namely, capacitated vehicle routing (CVR) and capacitated arc routing (CAR) are show-

cased in the current paper. To summarize, the core contributions of the current work is multi-facets, which are outlined as follows:

1. To date, almost all search methods start the optimization process from scratch, with the assumption of zero usable information, i.e., independent of how similar the current problem instance of interest is to those encountered in the pasts. The current work presents an attempt to fill this gap by embarking a study on evolutionary optimization methodology with intellectual capabilities that can evolve along with the problems solved.

2. To our knowledge, this is the first work that focuses on cultural evolutionary optimization of problem instances. It presents a novel effort on learning of useful traits from past problem solving experiences in the form of structured knowledge or latent patterns as memes, and subsequently through the mechanisms of cultural learning, selection, variation and imitation, appropriately acquired knowledge are transferred to enhance the evolutionary search on future unseen problems.

3. Beyond the formalism of simple and adaptive hybrids as memetic algorithm, this paper introduces and showcases the novel representation of acquired knowledge from past optimization experiences in the form of memes. In contrast to the manifestation of memes as refinement procedures in hybrids, here memes manifest as natural building blocks of meaningful information, and in the present context, serving as the instructions for generating solutions that would lead towards optimized solutions both effectively and efficiently.

4. This work proposes a storage of building blocks or memes to problems that share some common characteristics, such as instances from related problem domains, and supports the reuse of the memes across problems. The capacity to imitate and draw on memes from past instances of problem-solving sessions thus allows future evolutionary search to be more intelligent, leading to solutions that can be attained more efficiently on unseen problems of increasing complexity or dynamic in nature. These atomized units of memes then form societies of the mind for more effective problem solving.

5. In the experimental study, the proposed memetic evolutionary search paradigm is shown to be more efficient and effective for problem solving than conventional evolutionary optimization. Specifically, we show that the memetic paradigm enhances the search performances of existing state-of-the-art evolutionary optimization methodologies in the domains of capacitated vehicle routing and capacitated arc routing, which contain NP-hard combinatorial optimization problems of diverse properties.

The rest of this paper is organized as follows: a brief discussion on traditional evolutionary optimization and the related works is given in Section 2. Section 3 introduces the proposed memetic computational search paradigm for “intelligent”

evolutionary optimization of problems, via the transfer of structured knowledge in the form of memes from past problem-solving experiences, to enhance future evolutionary searches. Section 4 presents the brief mathematical formulations of the capacitated vehicle routing problem (CVRP) and capacitated arc routing problem (CARP), and discusses the existing state-of-the-art optimization methodologies for solving CVRP and CARP. Detailed designs of the *meme learning*, *meme selection*, *meme variation* and *meme imitation* operators for routing problems are then described in Section 5. Particularly, the culture-inspired operators are realized based on the *Hilbert-Schmidt Independence* (HSIC) [27], *Maximum Mean Discrepancy* criteria [4] and *K-means Clustering*. Last but not least, section 6 presents and analyzes the experimental results on the well established CVRP and CARP benchmarks, which serve as the representatives routing problems considered in the current work. Lastly, the conclusive remarks of this paper are drawn in Section 7.

2 Related Works

Evolutionary algorithms (EAs) are approaches that take their inspirations from the principles of natural selection and survival of the fittest in the biological kingdom. In the last decades, it is becoming well established that many effective optimizations using EA have been achieved by the use of inductive biases that fit to the structure of the problem of interest well [5]. In spite of the great deal of attention that EA has received, where many specialized EAs have been developed with the manual incorporations of human expert domain knowledge (and hence inductive biases) into the search scheme, it is worth noting here that, existing EA approaches have yet to exploit the useful traits that may exist in similar tasks or problems. In particular, the study of optimization methodology that evolves along with the problems solved has been significantly under-explored in the context of evolutionary computation. Instead, the common practice of evolutionary computation is to start the search on a given new problem of interest from ground zero state, independent of how similar the new problem instance of interest is to those encountered in the past. Thus a major impediment of existing evolutionary search methodology in the literature is the *apparent lack of automated useful knowledge transfers and reuse from past problem solving experiences as appropriate biases that can enhance evolutionary search of new problems*.

In practice, problems seldom exist in isolation, and previous related problems encountered often yield useful information that when properly harnessed, can lead to more effective future evolutionary search. To date, few works have attempted to reuse useful traits from across problems. Louis *et al.* [33], for instance, presented a study to acquire problem specific knowledge and subsequently using them to aid in the genetic algorithm (GA) search via case-based reasoning. Rather than starting anew on each problem, appropriate intermediate solutions drawn from similar problems that have been previously solved are periodically injected into the GA population. In a separate study, Cunningham and Smyth [17] also explored the

reuse of established high quality schedules from past problems to bias the search on new traveling salesman problems (TSPs). Despite these early attempts, the idea of reusing useful traits from across problems in evolutionary search did not garner significant attention. We believe this is a result of the weak generality trait of the approaches introduced in these initial studies. Particularly, both [33] and [17] considered the exact storage of past solutions or partial-solutions from previous problems solved, and subsequently regurgitate them directly into the solution population of a new evolutionary search. It is worth noting that such a simple scheme of the memory at work would proof to be futile when the nature or characteristics of the new optimization problem of interest differs, such as the problem size or dimensionality, structures, representations, etc. Thus what have been previously memorized from past problems solved cannot be directly injected into new searches for successful reuse.

In contrast to existing memory based approaches, the proposed memetic computational search paradigm addresses the non-trivial task of learning the generic building blocks or memes of useful traits from past problems solving experiences and subsequently drawing upon them through the cultural evolutionary mechanisms of meme learning, selection, variation and imitation (as opposed to a simple direct copying of past solutions in previous works) to enhance the search on new problems. In such a manner, the transfer of memes as generic building blocks of useful knowledge, can lead to enhanced search on problems of differing dimensions, topological structures, and representations, etc.

3 Memetic Evolution

In this section we present the proposed memetic computational paradigm for "intelligent" evolutionary optimization, one that is modelled after how human solves problems, capable of evolving along with the problems solved. In the proposed memetic computational paradigm, the instructions for carrying out the behavior to act on a given problem are modeled as memes stored in brains. These memes then serve as the building blocks of past problems solving experiences that may be efficiently passed on to support the search on future unseen problems, by means of *cultural evolution*. This capacity to draw on memes from previous instances of problem-solving sessions thus allows the search to be more intelligent, leading to a more effective and efficient future search.

3.1 Cultural Operators

The proposed memetic computational paradigm is composed of four culturally-inspired operators, namely *Meme Learning*, *Meme Selection*, *Meme Variation* and *Meme Imitation* as depicted in Fig. 1, whose functions are described in what follows:

- *Meme Learning* Operator: Given that $\mathbf{p}$ corresponds to a problem instance and $\mathbf{s}^*$ denotes the optimized solution of $\mathbf{p}$, as attained by an evolutionary solver (labeled here as OS). The learning operator takes the role of modelling and generalizing the mapping from $\mathbf{p}$ to $\mathbf{s}^*$, to derive the meme. The meme learning process evolves in an incremental manner, and builds up the wealth of knowledge in the form of memes, along with the number of problem instances solved. Note the contrast to a simple storage or exact memory of specific problem instance $\mathbf{p}$ with associated solution $\mathbf{s}^*$ as considered in the previous studies based on case-based reasoning [33].

- *Meme Selection* Operator: Different prior knowledge introduces unique forms of bias into the search. Hence a certain biases would make the search more efficient on some classes of problem instances but not for others. Inappropriately harnessed knowledge, on the other hand, may lead to the possible impairments of the search. The meme selection operator thus serves to identify or select the fit memes, from the society of memes, that replicate successfully.

- *Meme Variation* Operator: The meme variation forms the intrinsic innovation tendency of the cultural evolution. Without variations, maladaptive form of bias may be introduced in the evolutionary searches involving new problem instances. For instance, a piece of knowledge as represented by the memes, which has been established as constructive based on its particular demonstration of success on a given problem instance would quickly spiral out of control via replication. This will suppress the diversity and search of the evolutionary optimization across problems. Therefore, variation is clearly essential for retaining diversity in the society of memes towards efficient and effective evolutionary search.

- *Meme Imitation* Operator: From Dawkins's book entitled "The selfish Gene" [19], ideas or memes are copied from one person to another via imitation. In the present context, memes that are learned from past problem solving experiences replicates by means of imitation and used to enhance future evolutionary search on newly encountered problems.

3.2 '*Intelligent*' Evolutionary Optimization via Memetic Evolution

In the spirit of how human solves problems, the schemata representation of meme in computing as the structured knowledge or latent patterns that is encoded in the mystical mind of nature is first identified. The problem solving experiences on the encountered problems are then captured via *meme learning* and crystallized as a

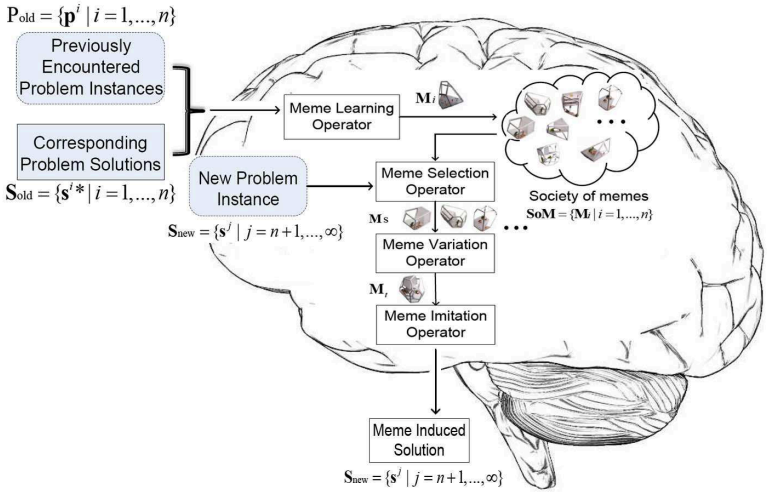


Figure 1: A depiction of how human solves problems in Memetic Computation.

part of the meme pool that form the building blocks in the society of mind [36]. In this manner, whenever a new problem comes about, the *meme selection* operator kicks in to first identify the appropriate memes from the wealth of previously accumulated memes. These memes then undergo *variations* to effect the emergence of innovative memes. Enhancements to subsequent problem-solving effectiveness and efficiency on given new problems is then achieved by means of *meme imitation*.

Referring to Fig. 1, at time step $i = 1$, the optimization solver OS is faced with the first problem instance $\mathbf{p^1}$ to search on. Since $\mathbf{p^1}$ denotes the first problem of its kind to be optimized, no memes is available for enhancing the evolutionary solver, OS, search. [1] This is equivalent to the case where a novice encounters a new problem of its kind to work on, in the absence of *a prior* knowledge that he/she can leverage upon. This condition is considered as "no relevant meme available" and the search by solver OS shall proceed normally, i.e., the *meme selection* operator remains dormant. The corresponding optimized solution attained by solver OS on $\mathbf{p^1}$ is denoted by $\mathbf{s^{1*}}$. Through the *meme learning* operator, meme $\mathbf{M_1}$ is learned and derived from the problem instance and corresponding optimized solution denoted by $(\mathbf{p^1}, \mathbf{s^{1*}})$. On subsequent unseen problem instances $j = 2, \ldots, \infty$, *meme selection* kicks in to identify the appropriate memes $\mathbf{M}$s from the society of memes denoted by $\mathbf{SoM}$. Activated memes $\mathbf{M}$s then undergo the *variation* operator to arrive at innovative meme $\mathbf{M_t}$ that serves as the instructions *imitated* to bias subsequent evolutionary optimizations by OS. In this manner, useful memes of experiences attained from previously solved problem instances are captured incrementally and archived in meme pool $\mathbf{SoM}$ to form the society of mind, which are

[1] Unless, of course, a database of memes that are learned from past problem solving experiences can be loaded and leveraged upon.

appropriately activated to enhance future search performances.

Like the memes stored in the human mind to aid in the coping of our everyday life and problem solving, memes residing in the artificial mind of the evolutionary solver play the role of positively biasing the search on newly encountered problems. In this manner, the intellectual capability of the evolutionary solver evolves along with the number of problems solved, with transferrable memes accumulating with time. When a new problem is encountered, fit memes are activated and varied to instruct and guide the intelligent solver in the search process. The mind-universe of memes thus formed the evolving domain knowledge that may be activated to solve future evolutionary search effectively and efficiently.

4 Case Studies on Routing Problems

In this section, we illustrate the proposed memetic computational paradigm for "intelligent" evolutionary optimization on the class of combinatorial optimization problem. Particularly, our interest is on the general problem scenario where there exist a number of agents and a number of tasks. Any agent can be assigned to perform any task, incurring some cost and profit that may vary depending on the agent-task assignment. Moreover, each agent has a budget and the sum of the costs of tasks assigned to it cannot exceed this budget. It is required to find an assignment in which all agents do not exceed their budget and the total profit of the assignment is maximized. It is worth highlighting the practicality of described scenarios, where a plethora of today's real world complex problems can be established to exhibit many of the properties outlined. These include the well known combinatoric problems like vehicle routing in urban waste collection or post delivery, plant location, resource scheduling, flexible manufacturing systems, generalized assignment problem [15], job scheduling problem [28], and arc routing problem [45, 23], etc. To demonstrate the memetic computational search paradigm for "intelligent" evolutionary optimization, the two widely studied challenging domains of the capacitated vehicle routing problem (CVRP) and capacitated arc routing problem (CARP) are showcased in the present section.

4.1 Capacitated Vehicle Routing Problem

The capacitated vehicle routing problem (CVRP) introduced by Dantzig and Ramser [18], is a problem to design a set of vehicle routes in which a fixed fleet of delivery vehicles of uniform capacity must service known customer demands for single commodity from a common depot at minimum cost. The CVRP can be formally defined as follows. Given a connected undirected graph $G = (V, E)$, where vertex set $V = \{v_i\}, i = 1 \dots n$, n is the number of vertices, edge set $E = \{e_{ij}\}, i, j = 1 \dots n$ denoting the arc between vertices v_i and v_j. Vertices v_d corresponds to the depot at which k homogeneous vehicles are based, and the remaining vertices denote the customers. Each arc e_{ij} is associated with a non-

negative weight c_{ij} , which represents the travel distance from v_i to v_j. Consider a demand set $D = \{d(v_i)|v_i \in V\}$, where $d(v_i) > 0$ implies customer v_i requires servicing (i.e., known as task), the CVRP consists of designing a set of least cost vehicle routes $\mathcal{R} = \{\mathcal{C}_i\},\ i = 1 \ldots k$ such that

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

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

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

where $c(\mathcal{C}_i)$ is the summation of the travel distance e_{ij} contained in route $\mathcal{C}_i$.

4.2 Capacitated Arc Routing Problem

The capacitated arc routing problem (CARP) was first proposed by Golden and Wong [24] in 1981. It can be formally stated as follows: Given a connected undirected graph $G = (V, E)$, where vertex set $V = \{v_i\}, i = 1 \ldots n$, n is the number of vertices, edge set $E = \{e_i\}, i = 1 \ldots m$ with m denoting the number of edges. Consider a demand set $D = \{d(e_i)|e_i \in E\}$, where $d(e_i) > 0$ implies edge e_i requires servicing (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 edge 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 edge 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:

1. Each travel circuit $\mathcal{C}_i,\ i \in [1, k]$ must start and end at the depot node $v_d \in V$.
2. The total load of each travel circuit must be no more than the capacity W of each vehicle, $\sum_{\forall e_i \in \mathcal{C}} d(e_i) \leq W$.
3. $\forall e_i \in E\ and\ d(e_i) > 0$, 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 edges that needed service together with the total travel cost of the remaining edges that formed the circuit:

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

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

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

4.3 Present State-of-the-art Evolutionary Optimization Methodologies

Theoretically, CVRP and CARP have been proven to be NP-hard with only explicit enumeration approaches known to solve them optimally. However, large scale problems are generally computationally intractable due to the poor scalability of most enumeration methods [14]. From a survey of the literature, meta-heuristics, heuristics and evolutionary computation have played important roles in algorithms capable of providing good solutions within tractable computational time. For CVRP, Cordeau *et al.* [16] considered a unified tabu search algorithm (UTSA) for solving VRP. Prins [41] presented an effective evolutionary algorithm with local search for the CVRP, while Reimann *et al.* [42] proposed a D-ants algorithm for CVRP which equipped ant colony algorithm with individual learning procedure. Recently, Lin *et al.* [32] takes the advantages of both simulated annealing and tabu search, and proposed a hybrid meta-heuristic algorithm for solving CVRP. Further, Chen *et al.* [11] proposed a domain-specific cooperative memetic algorithm for solving CVRP and achieved better or competitive results compared with a number of state-of-the-art memetic algorithms and meta-heuristics to date.

On the other hand, for CARP, Lacomme et al. in [30] presented the basic components that have been embedded into memetic algorithms (MAs) for solving the extended version of CARP (ECARP). Lacomme's MA (LMA) was demonstrated to outperform all known heuristics on three sets of benchmarks. Recently, Mei et al. [35] extended Lacomme's work by introducing two new local search methods, which successfully improved the solution qualities of LMA. In a separate study, a memetic algorithm with extended neighborhood search was also proposed for CARP in [45]. Further, Liang *et al.* proposed a formal probabilistic memetic algorithm for solving CARP, with new best-known solutions to date found on 9 of the benchmark problems [23].

The problem of solving CVRP and CARP can be generalized as searching for the suitable task assignments (i.e., vertices or arcs that require to be serviced) of each vehicle, and then finding the optimal service order of each vehicle for the assigned tasks. In the evolutionary search literature, the task assignment stage has been realized by means of simple task randomization [46] to more advance strategies like heuristic search [46, 11], clustering [50], etc., while the optimal service order of each vehicle is achieved via the mechanisms of evolutionary search operators. The example of an optimized CVRP or CARP solution can be illustrated in Fig. 2, where four vehicle routes, namely, $R_1 = \{0, \mathbf{v}_1, \mathbf{v}_2, \mathbf{v}_3, 0\}$, $R_2 = \{0, \mathbf{v}_6, \mathbf{v}_5, \mathbf{v}_4, 0\}$, $R_3 = \{0, \mathbf{v}_{10}, \mathbf{v}_9, \mathbf{v}_8, \mathbf{v}_7, 0\}$ and $R_4 = \{0, \mathbf{v}_{14}, \mathbf{v}_{13}, \mathbf{v}_{12}, \mathbf{v}_{11}, 0\}$,

can be observed. A '0' index value is assigned at the beginning and end of route to denote that each route starts and ends at the depot.

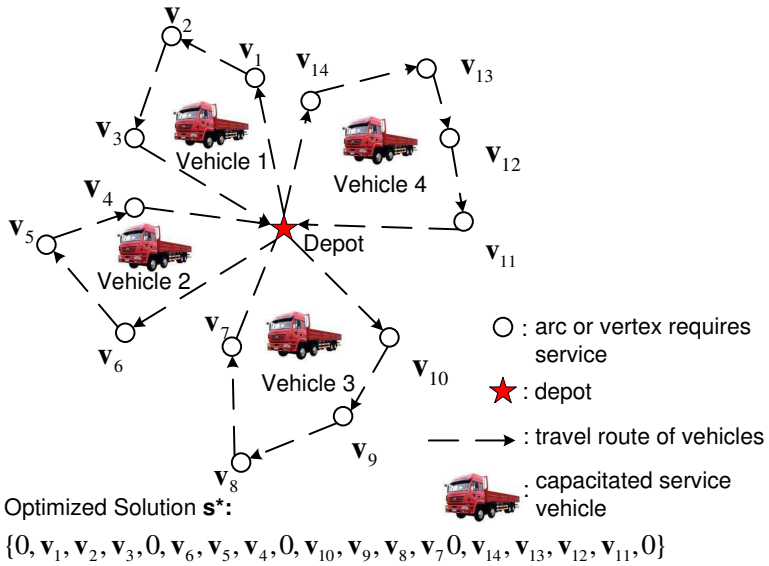


Figure 2: An example of the CARP and CVRP.

In the rest of the paper, we shall illustrate the concept of memetic evolution, which learns and extracts useful traits from past problem solving experiences as memes or structured knowledge, for "intelligent" evolutionary optimization of real world CVRPs and CARPs. In particular, we model the mapping of solved CVRP or CARP instances to their optimized solutions as memes, which serves as the instructions for carrying out the behavior that shall subsequently be used to appropriately bias the search on new unseen CVRP or CARP instances, respectively.

5 Memetic Computational Paradigm for "Intelligent" Evolutionary Optimization of Routing Problems

In this section, we present a realization of the proposed memetic computational search paradigm, particularly focusing on the cultural-inspired operators, namely, *meme learning*, *meme selection*, *meme variation* and *meme imitation*, which are designed for solving routing problems (i.e., such as CARPs and CVRPs) and other related problem domains. The "intelligent" evolutionary optimization process of routing problem instances using the mechanisms of memetic evolution is next described.

In particular, the pseudo-code and detailed workflow of the proposed memetic computational paradigm for evolutionary optimization of routing problem instances are outlined in Algo. 1 and Fig. 3, respectively. For a given new routing problem instance $\mathbf{p}_{new}^{j}$ (with data representation $\mathbf{X}_{new}^{j}$) posed to evolutionary solver OS,

the mechanisms of the *meme selection* operator kicks in to select the fit memes $\mathbf{M}$s to activate, if the $\mathbf{SoM}$ meme pool is not empty. *Meme variation*, which takes inspirations from the human's ability to generalize from past knowledge learned in previous problem solving experiences, then operates to generalize the activated $\mathbf{M}$s to arrive at $\mathbf{M_t}$. Subsequently, for given new problem instance $\mathbf{X}^j_{new}$, *meme imitation* then proceeds to positively bias the search of evolutionary optimization solver OS, with meme (i.e., generalized $\mathbf{M_t}$) induced solutions that enhance the search performances on $\mathbf{p}^j_{new}$. In our work, the meme induced solutions of tasks assignment and service orders are obtained by means of simple clustering and pairwise distance sorting (PDS), respectively, which shall be detailed later in Section 5.4. When the search on $\mathbf{p}^j_{new}$ completes, the problem instance $\mathbf{p}^j_{new}$ together with the obtained optimized solution $\mathbf{s}^{j*}_{new}$ of OS, i.e., ($\mathbf{X}^j_{new}$, $\mathbf{Y}^j_{new}$), shall then undergo the *meme learning* operation so as to update the $\mathbf{SoM}$ meme pool of the intelligent OS evolutionary solver.

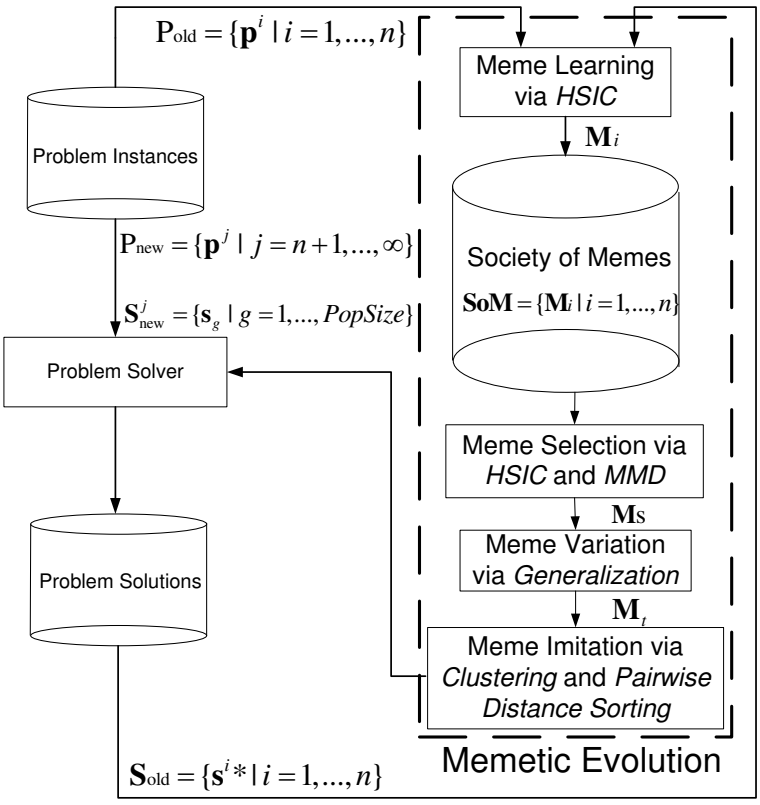


Figure 3: Memetic Computational Paradigm for "Intelligent" Evolutionary Optimization of Problems.

Algorithm 1: Pseudo code of the Memetic Computational Search Paradigm for "Intelligent" Evolutionary Optimization of Routing Problems

Begin:
for $j = 1 : \infty$ *new problem instances* $\mathbf{p}_{new}^{j}$ *or* $\mathbf{X}_{new}^{j}$ **do**
  if $\mathbf{SoM}! = \emptyset$ **then**
    /*meme pool not empty*/
    Perform *meme selection* to identify fit memes $\mathbf{M}$s $\in \mathbf{SoM}$ /*see Eqn. 11* in later Section 5.2/
    Perform *meme variation* to generalize $\mathbf{M}$s to derive $\mathbf{M_t}$.
    Empty the initial population Ω.
    for $g = 1 : Population\ Size$ **do**
      Perform *meme imitation* with $\mathbf{X}_{new}^{j}$ and $\mathbf{M_t}$ to obtain solution $\mathbf{s}_g$.
        a. $\mathbf{X}_{new}^{j'} = Transform(\mathbf{X}_{new}^{j}, \mathbf{M_t})$
        /*Fig. 6(a)→Fig. 6(b)*/
        b. $Task\ Assignment\ of\ \mathbf{s}_g = KMeans(\mathbf{X}_{new}^{j'}, Vehicle\ No., RI)$
        /Fig. 6(b)→Fig. 6(c), RI denotes random initial points*/
        c. $Service\ Order\ of\ \mathbf{s}_g = PDS(\mathbf{X}_{new}^{j'})$
        /*Fig. 6(c)→Fig. 6(d), $PDS(\cdot)$ denotes the pairwise distance sorting*/
        d. Insert $\mathbf{s}_g$ into Ω.
  else
    Proceed with the original population initialization scheme of the evolutionary solver OS.
  /*Start of Evolutionary Solver OS Search*/
  Perform reproduction and selection operations of OS with generated population $\mathbf{s}_g$s until the predefined stopping criteria are satisfied.
  /*End of Evolutionary Solver OS Search*/
  Perform *meme learning* on given $\mathbf{p}_{new}^{j}$ and corresponding optimized solution $\mathbf{s}_{new}^{j*}$ denoted by $(\mathbf{X}_{new}^{j}, \mathbf{Y}_{new}^{j})$, attained by OS evolutionary solver to derive meme $\mathbf{M}_{new}^{j}$.
  Archive the learned meme of $\mathbf{p}_{new}^{j}$ into $\mathbf{SoM}$ meme pool to update the society of meme for subsequent reuse.
End

5.1 Meme Learning Operator in Routing Problem

This subsection describes the learning and extraction of memes as building blocks of useful traits from given routing problem instances $\mathbf{p}$ and the corresponding optimized solutions $\mathbf{s}^*$ (i.e., Line 22 in Alg. 1). To begin, we refer to Fig. 2 and Fig. 4, which shall serve as the example problem instance used for our illustrations. Fig. 4(a) depicts the distribution of the tasks in the example routing problem of Fig. 2 that need to be serviced. Fig. 4(b), on the other hand, denotes the optimized routing solution of the OS evolutionary solver on problem Fig. 2 and Fig. 4(a). The dashed circles in Fig. 4(b) denote the optimized task assignments of the individual vehicles, and the arrows then indicate the optimized tasks service orders by OS.

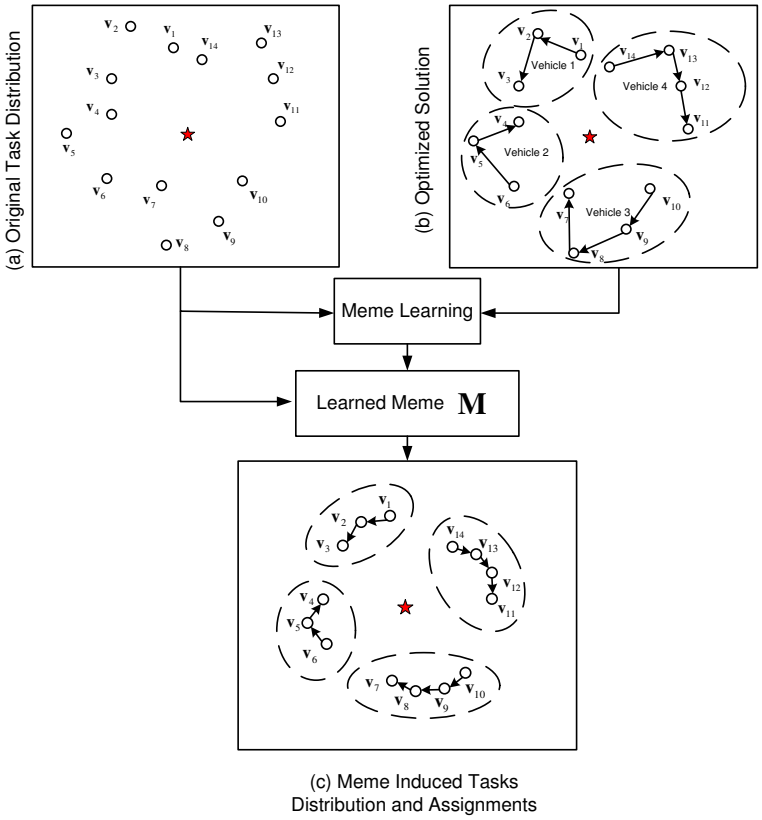


Figure 4: Meme as the instructions for "intelligent" tasks assignment and ordering of routing problems.

Here a meme $\mathbf{M}$ thus hosts the instructions for transforming the original distribution and service orders of tasks, to one that realigns well or is closer to the desired tasks distribution and tasks service orders, as defined by the optimized routing solution $\mathbf{s}^*$ attained by solver OS. Using the example routing problem instance in Fig. 2, our objective is thus to find the meme $\mathbf{M}$ that transforms the task distributions depicted in Fig. 4(a), to the desired tasks distribution of $\mathbf{s}^*$ and preserves

the corresponding tasks service orders as depicted in Fig. 4(b). In this manner, whenever a new routing problem instance is encountered, suitable learned or captured memes from previously optimized problem instances can then be deployed to realign the tasks distribution and service orders accordingly. For instance, Fig. 4(c) showcases the desirable scaled or transformed tasks distribution of Fig. 4(a) when the appropriate meme $\mathbf{M}$ is put to work. In particular, it can be observed in Fig. 4(c) that we seek for meme(s) capable of re-locating tasks serviced by a common vehicle to become closer to one another (as desired by the optimized solution $\mathbf{s}^*$ shown in Fig. 4(b)), while tasks serviced by different vehicles are kept further apart. In addition, to match the service orders of each vehicle to that of the optimized solution $\mathbf{s}^*$, the task distribution is adjusted 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).

Next, the mathematical definitions of meme $\mathbf{M}$ to achieve the transformations of tasks distribution are detailed. In particular, given $\mathbf{V} = \{\mathbf{v}_i \,|i = 1, \ldots, n\}$, n is the number of tasks, denoting the tasks of a problem instance to be assigned. The distance between any two tasks $\mathbf{v}_i = (v_{i1}, \ldots, v_{ip})^T$ and $\mathbf{v}_j = (v_{j1}, \ldots, v_{jp})^T$ in the p-dimensional space $\mathbb{R}^p$ is then given by:

$$d_M(\mathbf{v}_i, \mathbf{v}_j) = ||\mathbf{v}_i - \mathbf{v}_j||_M = \sqrt{(\mathbf{v}_i - \mathbf{v}_j)^T \mathbf{M} (\mathbf{v}_i - \mathbf{v}_j)}$$

where T denotes the transpose of a matrix or vector. Meme $\mathbf{M}$ is positive semidefinite, and can be decomposed as $\mathbf{M} = \mathbf{L}\mathbf{L}^T$ by means of singular value decomposition (SVD). By substituting this decomposition into $d_M(\mathbf{v}_i, \mathbf{v}_j)$, we arrive at:

$$d_M(\mathbf{v}_i, \mathbf{v}_j) = \sqrt{(\mathbf{L}^T\mathbf{v}_i - \mathbf{L}^T\mathbf{v}_j)^T (\mathbf{L}^T\mathbf{v}_i - \mathbf{L}^T\mathbf{v}_j)} \tag{4}$$

From equation 4, it is worth noting that the distances among the tasks are scaled by meme $\mathbf{M}$. Such a meme $\mathbf{M}$ shall then perform the realignment of tasks distribution and service orders of a given new problem instance to be closer to the optimized solution $\mathbf{s}^*$ of a related problem instance that has been previously encountered.

In what follows, the mathematical formulations for the automated learning of meme $\mathbf{M}$ are detailed. The schemata representations of a problem instance ($\mathbf{p}$), optimized solution ($\mathbf{s}^*$) and distance constraints set $\mathcal{N}$ are first defined. In particular, the data representations of the example problem instance in Fig. 2 is depicted in Fig. 5, where v_{11}, v_{12}, etc., denote the features representation of each task, and $D(\cdot)$ indicates the Euclidean distance metric. Further, if task $\mathbf{v}_i$ and task $\mathbf{v}_j$ are served by the same vehicle, $\mathbf{Y}(i, j) = 1$, otherwise, $\mathbf{Y}(i, j) = -1$.

To capture meme $\mathbf{M}$ from a given CVRP or CARP problem instance, denoted by ($\mathbf{p}$, $\mathbf{s}^*$), we formulate the learning task as a maximization of the dependency between $\mathbf{X}$ and $\mathbf{Y}$ according to the *Hilbert-Schmidt Independence Criterion* (HSIC)

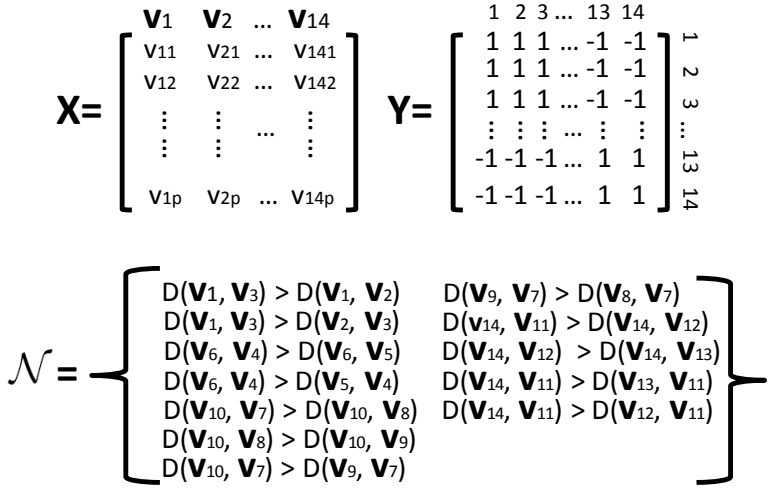


Figure 5: Data representations of a problem instance $\mathbf{p} = \mathbf{X}$, the corresponding optimized solution $\mathbf{s}^* = \mathbf{Y}$ and distance constraints set $\mathcal{N}$.

[27] with distance constraints, which is mathematically defined as:

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

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.

Further, $\mathbf{H} = \mathbf{I} - \frac{1}{n}\mathbf{I}\mathbf{I}'$ 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 upon serving task i, task q is served before task j by the same vehicle.

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 can be 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:

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

where C balances between the two parts of the criterion. The first constraint enforces the learnt 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., $\mathbf{Y}$).

To solve the learning problem in Equation 6, we first derive the minimax optimization problem by introducing dual variables α for the inequality constraints based on Lagrangian theory.

$$\begin{aligned} Lr &= tr(-\mathbf{H}\mathbf{X}^T\mathbf{M}\mathbf{X}\mathbf{H}\mathbf{Y}) + \frac{C}{2}\sum \xi_{ij}^2 \\ &\quad - \sum \alpha_{ij}(tr(\mathbf{X}^T\mathbf{M}\mathbf{X}\mathbf{T}_{ij}) - tr(\mathbf{X}^T\mathbf{M}\mathbf{X}\mathbf{T}_{iq})) \end{aligned} \tag{7}$$

Set $\frac{\partial Lr}{\partial \xi_{ij}} = 0$, we have:

$$C\sum \xi_{ijq} - \sum \alpha_{ijq} = 0 \Longrightarrow \xi_{ijq} = \frac{1}{C}\sum \alpha_{ijq} \tag{8}$$

By substituting Equation 8 into Equation 7, we reformulate the learning problem in Equation 6 as a minimax optimization problem, which is given by:

$$\begin{aligned} \max_{\alpha}\min_{\mathbf{M}} \quad & tr[(-\mathbf{X}\mathbf{H}\mathbf{Y}\mathbf{H}\mathbf{X}^T - \sum \alpha_{ijq}\mathbf{X}\mathbf{T}_{ij}\mathbf{X}^T \\ & + \sum \alpha_{ijq}\mathbf{X}\mathbf{T}_{iq}\mathbf{X}^T)\mathbf{M}] - \frac{1}{2C}\sum \alpha_{ijq}^2 \\ \text{s.t.} \quad & \mathbf{M} \succeq 0 \end{aligned} \tag{9}$$

By setting

$$\mathbf{A} = \mathbf{X}\mathbf{H}\mathbf{Y}\mathbf{H}\mathbf{X}^T + \sum \alpha_{ijq}\mathbf{X}\mathbf{T}_{ij}\mathbf{X}^T - \sum \alpha_{ijq}\mathbf{X}\mathbf{T}_{iq}\mathbf{X}^T$$

and

$$\Delta J_{ijq}^t = tr[(\mathbf{X}\mathbf{T}_{iq}\mathbf{X}^T - \mathbf{X}\mathbf{T}_{ij}\mathbf{X}^T)\mathbf{M}] - \frac{1}{C}\alpha_{ijq}$$

C is configured with a default value of 10 in the learning problem of Equation 9, and then Equation 9 can be solved as described in [51].

5.2 Meme Selection Operator in Routing Problem

Since different meme introduces unique biases into the evolutionary search, inappropriately chosen memes and hence biases can lead to negative impairments of the evolutionary search. To facilitate a positive transfer of memes [40] that would lead to enhanced evolutionary search, the *meme selection* operator (i.e., Line 5 in Alg. 1) is designed to select fit memes that shares common characteristics with the given new problem of interest to replicate.

Suppose there is a set of n unique $\mathbf{M}$s in $\mathbf{SoM}$, i.e., $\mathbf{SoM} = \{\mathbf{M}_1, \mathbf{M}_2, \ldots, \mathbf{M}_n\}$ that form the society of memes. The *meme selection* operator is here realized as to identify the weight μ_i of each meme. A fitter meme should have a higher weight and the summation of the weights of all memes equates to 1 (i.e., $\sum_{i=1}^{n} \mu_i = 1$).

In particular, the meme coefficient vector $\boldsymbol{\mu}$ is determined based on the HSIC and Maximum Mean Discrepancy criteria [4].

$$\begin{aligned} \max_{\mu} \quad & tr(\mathbf{HKHY}) + \sum_{i=1}^{m} (\mu_i)^2 Sim_i \\ \text{s.t.} \quad & \mathbf{M}_t = \sum_{i=1}^{n} \mu_i \mathbf{M}_i \\ & \mathbf{K} = \mathbf{X}^T * \mathbf{M}_t * \mathbf{X}, \mathbf{K} \succeq 0 \\ & \mu_i \geq 0 \\ & \sum_{i=1}^{n} \mu_i = 1 \end{aligned} \tag{10}$$

By substituting the constraints of Equation 10 into the objective function, we arrive at:

$$\begin{aligned} \max_{\mu} \quad & tr(\mathbf{HX}^T \mathbf{M}_t \mathbf{XHY}) + \sum_{i=1}^{m} (\mu_i)^2 Sim_i \\ \text{s.t.} \quad & \mathbf{M}_t = \sum_{i=1}^{n} \mu_i \mathbf{M}_i, \mathbf{M}_i \succeq 0 \\ & \mu_i \geq 0 \\ & \sum_{i=1}^{n} \mu_i = 1 \end{aligned} \tag{11}$$

where Sim_i is the similarity measure between two given problem instances. In the present context, $Sim_i = -(\beta * MMD_i + (1-\beta) * Dif_i)$, where MMD_i denotes the Maximum Mean Discrepancy [4], which is used to compare the distribution similarity between two given instances by measuring the distance between their corresponding means. $MMD(D_s, D_t) = ||\frac{1}{n_s}\sum_{i=1}^{s}\phi(x_i^s) - \frac{1}{n_t}\sum_{i=1}^{t}\phi(x_i^t)||$, where $\phi(\cdot)$ maps the original input to a high dimensional space. Here, for the purpose of computational efficiency, we consider a linear mapping with $\phi(\mathbf{x}) = \mathbf{x}$. Dif_i denotes the difference in vehicle capacity for two given CARP or CVRP instances. β balances between the two parts (i.e., MMD_i and Dif_i) in Sim_i. From experiences, due to the higher relevance of the task distribution features than vehicles capacity when defining the similarities between problem instances, here β is configured as 0.8. In Equation 10, the first term serves to maximize the statistical dependence between input $\mathbf{X}$ and output label $\mathbf{Y}$ for clustering [44]. The second term measures the similarity between the previous problem instances solved to the given new problem of interest.

In Equation 11, two unknown variables exist (i.e., μ and $\mathbf{Y}$). $\mathbf{Y}$ is obtained from the results of task assignment. With $\mathbf{Y}$ fixed, Equation 11 becomes a quadric programming problem of μ. To solve the optimization problem of Equation 11,

Table 1: Criteria for measuring performance.

Criterion	Definition
$Number\ of\ Fitness\ Evaluation$	Average number of fitness evaluations across all independent runs conducted
$Cpu\ Time$	Average computational cost in wall-clock time across all independent runs conducted
$Ave.Cost$	Average fitness of the solutions obtained across all independent runs conducted
$B.Cost$	Best fitness of the solutions obtained across all independent runs conducted
$Std.Dev$	Standard deviation of the solutions' fitness across all independent runs conducted
$Success\ No.$	Number of independent runs that fare above the $Ave.Cost$ value

Table 2: Properties of the "Augerat" CVRP data set.

Data Set	A-n32-k5	A-n54-k7	A-n60-k9	A-n69-k9	A-n80-k10	B-n41-k6	B-n57-k7	B-n63-k10	B-n68-k9	B-n78-k10	P-n50-k7	P-n76-k5
V	31	53	59	68	79	40	56	62	67	77	49	75
C_v	100	100	100	100	100	100	100	100	100	100	150	280
LB	784	1167	1354	1159	1763	829	1140	1496	1272	1221	554	627

we first perform clustering (e.g., K-Means) on input $\mathbf{X}$ directly to obtain the label matrix $\mathbf{Y}$. By keeping $\mathbf{Y}$ fixed, we obtained μ by maximizing Equation 11 via quadric programming solver. Next, by maintaining the chosen $\mathbf{M}$ fixed, clustering is made on the new $\mathbf{X}^{'}$ (i.e., transformed by selected $\mathbf{M}$. $\mathbf{X}' = \mathbf{L}^T\mathbf{X}$, where $\mathbf{L}$ is obtained by SVD on $\mathbf{M}$) to obtain label matrix $\mathbf{Y}$.

5.3 Meme Variation Operator in Routing Problem

Further, to introduce innovations into the selected fit memes during subsequent reuse, the *meme variation* operator (i.e., Line 6 in Alg. 1) then kicks in to operate on the selected memes. In the present context, we take inspirations of human's ability to generalize from past problem solving experiences. Hence *meme variation* is realized here in the form of *memes generalization*. However, it is worth noting that other alternative forms of probabilistic scheme in meme variations may also be considered since uncertainties can generate growth and variations of knowledge that we have of the world [43], hence leading to higher adaptivity capabilities for solving complex and non-trivial problems.

In particular, the generalized meme $\mathbf{M_t}$ in *meme variation* is realized as a linear combination of the selected memes:

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

5.4 Meme Imitation Operator in Routing Problem

In many routing problems, such as CVRP and CARP, the search for optimal solution is typically solved as two separate phases. The first phase involves the assignment of the tasks that require services to the appropriate vehicles. The second phase then serves to find the optimal service order of each vehicle for the assigned tasks obtained in phase 1.

In what follows, the meme induced solutions that serve as initial population to intelligently bias the OS search shall be described. For each solution $\mathbf{s}_g$, the

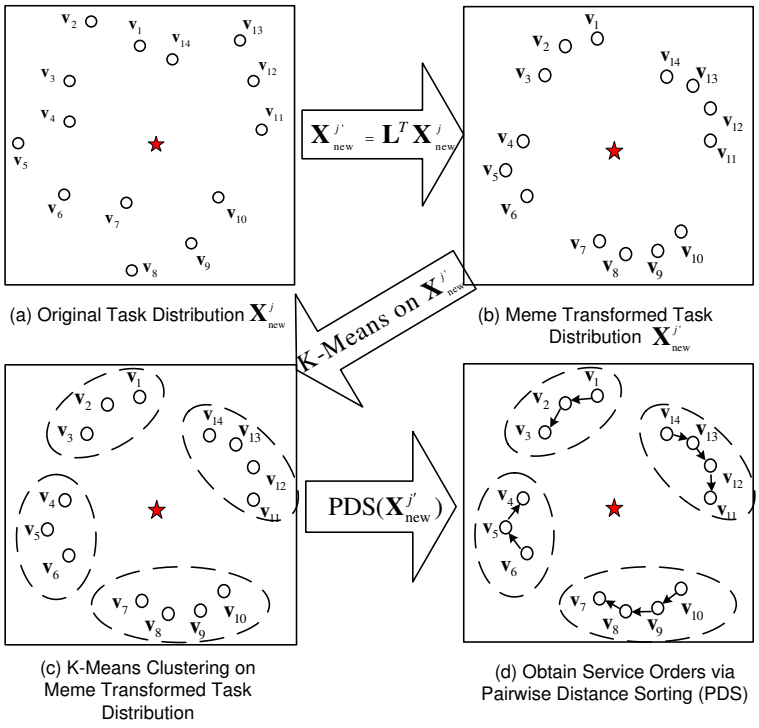


Figure 6: Illustration of meme imitation for generating intelligently biased CVRP or CARP solutions.

imitation of tasks assignments and tasks service orders using generalized meme $\mathbf{M}_t$ then lead to a transformation (i.e., Line 10 in Alg. 1) of the original tasks distribution $\mathbf{X}^{j}_{new}$ to a meme induced tasks distribution $\mathbf{X}^{j'}_{new}$, which is given by:

$$\mathbf{X}^{j'}_{new} = \mathbf{L}^T\mathbf{X}^{j}_{new} \tag{12}$$

where $\mathbf{L}$ is derived by means of singular value decomposition on $\mathbf{M}_t$. The illustrative example is depicted in Fig. 6, where Fig. 6(a) denote the original task distribution $\mathbf{X}^{j}_{new}$ and Fig. 6(b) is the resultant meme induced tasks distribution $\mathbf{X}^{j'}_{new}$ using $\mathbf{M}_t$.

In phase 1, simple clustering scheme such as K-Means clustering with random initializations is then conducted on the meme induced tasks distribution $\mathbf{X}^{j'}_{new}$ to derive the tasks assignments of the vehicles as depicted in Fig. 6(c), where the dashed circles denote the task assignments of the individual vehicles, i.e., denoting the tasks that shall be serviced by a common vehicle.

In phase 2, the service orders of each vehicle are subsequently achieved by sorting the pairwise distances among tasks in an ascending order. The two tasks with largest distance shall denote the first and last tasks to be serviced. Taking the first task as reference, the service order of the remaining tasks are defined according

Table 3: Properties of the "CE" and "Christofides" CVRP data sets.

Data Set	E-n33-k4	E-n76-k7	E-n76-k8	E-n76-k10	E-n76-k14	E-n101-k8	c50	c75	c100	c100b	c120	c150	c199
V	32	75	75	75	75	100	50	75	100	100	120	150	199
C_v	8000	220	180	140	100	200	160	140	200	200	200	200	200
LB	835	682	735	830	1021	815	524.61	835.26	826.14	819.56	1042.11	1028.42	1291.45

to the sorted orders. Taking Fig. 6(d) as example, where the arrows indicate the service orders of the tasks, the distance between $\mathbf{v}_1$ and $\mathbf{v}_3$ are largest among $\mathbf{v}_1$, $\mathbf{v}_2$ and $\mathbf{v}_3$. By assigning $\mathbf{v}_1$ as the reference task to be served, $\mathbf{v}_2$ shall be the next task to be serviced, since the distance between $\mathbf{v}_1$ and $\mathbf{v}_2$ is smaller than that of $\mathbf{v}_1$ and $\mathbf{v}_3$.

6 Experimental Study

To evaluate the efficiency and effectiveness of the proposed memetic evolutionary search paradigm, empirical studies conducted on the challenging domains of capacitated vehicle routing problems (CVRPs) and capacitated arc routing problems (CARPs) are presented in this section. These consist of problems of diverse properties in terms of vertices size, graph topologies, etc., which cannot be handled using existing case based reasoning approaches [33, 17] as previously discussed in Section 2. Two state-of-the-art evolutionary algorithms for solving CVRPs and CARPs, labeled in their respective published works as *CAMA* [9] and *ILMA* [35], are considered here as the baselines conventional evolutionary solver, of the respective domains, in the proposed memetic search paradigm. In this study, several criteria have been defined to measure the search performances, which are listed in Table 1. Among these criteria, $Number\ of\ Fitness\ Evaluation$ and $Cpu\ Time$ are used to measure the efficiency of the algorithms, while $Ave.Cost$, $B.Cost$ and $Success\ No.$ serve as the criteria for measuring the solution qualities of the algorithms.

6.1 Capacitated Vehicle Routing Problem

6.1.1 Empirical Configuration

All three commonly used CVRP benchmark data sets are investigated in the present empirical study, namely "AUGERAT" [1], "CE" [12] and "CHRISTOFIDES" [13]. The "Augerat" dataset includes 12 CVRP instances, "CE" consists of 6 CVRP instances, and "Christofides" has 7 CVRP instances. The detailed properties (e.g., number of vertices, lower bound, etc.) of the CVRP instances considered are summarized in Table 2 and Table 3, where V denotes the number of vertices that need to be serviced, C_v gives the capacity of the vehicles in each instance, and LB describes the lower bound of each problem instance. Note the diversity in the properties of the problems considered.

In CVRP, each task or vertex has a corresponding $coordinates$ and $demand$. Using the $coordinates$ of the vertex, the tasks assignment of each vehicle are

Table 4: Statistic results of *CAMA*, *CAMA-R*, and *CAMA-M* on "AUGERAT" CVRP benchmarks.

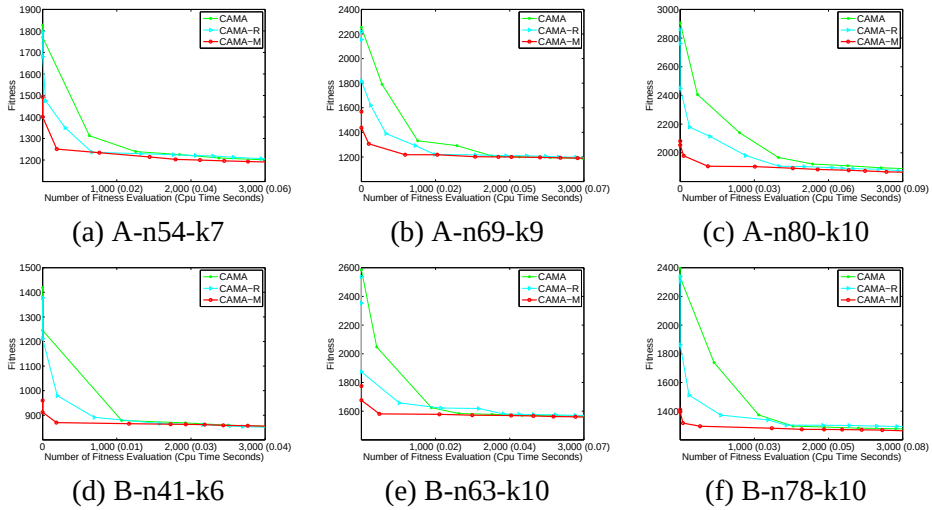
Data	$CAMA$				$CAMA-R$				$CAMA-M$ (Proposed Method)			
Set	*B.Cost*	*Ave.Cost*	*Std.Dev*	*Success No.*	*B.Cost*	*Ave.Cost*	*Std.Dev*	*Success No.*	*B.Cost*	*Ave.Cost*	*Std.Dev*	*Success No.*
1.A-n32-k5	784	748	0	30	784	784	0	30	784	784	0	30
2.A-n54-k7	1167	1169.50	3.36	19	1167	**1167**	0	30	1167	**1167**	0	**30**
3.A-n60-k9	1354	1356.73	3.59	17	1354	1355.20	1.86	21	1354	**1354.4**	1.22	**27**
4.A-n69-k9	1159	1164.17	3.07	16	1159	1162.20	2.41	24	1159	**1161.43**	2.37	**28**
5.A-n80-k10	1763	1778.73	9.30	9	1763	1777.07	7.94	10	1763	**1775.7**	8.80	**16**
6.B-n41-k6	829	**829.30**	0.47	**21**	829	829.93	0.94	10	829	829.53	0.73	17
7.B-n57-k7	1140	1140	0	30	1140	1140	0	30	1140	1140	0	30
8.B-n63-k10	1537	1537.27	1.46	29	**1496**	1528.77	15.77	**30**	**1496**	**1525.86**	17.45	**30**
9.B-n68-k9	1274	1281.47	5.56	**15**	1274	1284.80	4.51	7	**1273**	**1281.43**	5.74	**15**
10.B-n78-k10	1221	1226.07	5.48	22	1221	1226.80	6.39	19	1221	**1224.37**	3.23	**26**
11.P-n50-k7	554	556.33	2.34	14	554	554.93	1.72	23	554	**554.26**	1.01	**28**
12.P-n76-k5	627	630.70	5.34	22	627	628.87	1.61	25	627	**628.63**	1.51	**26**

Table 5: Statistic results of *CAMA*, *CAMA-R*, and *CAMA-M* on "CE" CVRP benchmarks.

Data	$CAMA$				$CAMA-R$				$CAMA-M$ (Proposed Method)			
Set	*B.Cost*	*Ave.Cost*	*Std.Dev*	*Success No.*	*B.Cost*	*Ave.Cost*	*Std.Dev*	*Success No.*	*B.Cost*	*Ave.Cost*	*Std.Dev*	*Success No.*
1.E-n33-k4	835	835	0	30	835	835	0	30	835	835	0	30
2.E-n76-k7	682	685.67	2.17	23	682	684.73	1.31	27	682	**684.66**	1.12	**28**
3.E-n76-k8	735	737.57	2.36	18	735	737.17	1.60	21	735	**737.06**	1.91	**23**
4.E-n76-k10	830	837.03	3.56	18	831	835.80	3.19	21	830	**834.73**	2.92	**27**
5.E-n76-k14	1021	**1025.67**	3.48	16	1021	1026.27	3.33	15	1021	1025.80	3.64	**17**
6.E-n101-k8	816	820.63	3.20	17	815	818.97	1.94	26	815	**818.53**	1.55	**27**

Table 6: Statistic results of *CAMA*, *CAMA-R*, and *CAMA-M* on "CHRISTOFIDES" CVRP benchmarks.

Data	$CAMA$				$CAMA-R$				$CAMA-M$ (Proposed Method)			
Set	*B.Cost*	*Ave.Cost*	*Std.Dev*	*Success No.*	*B.Cost*	*Ave.Cost*	*Std.Dev*	*Success No.*	*B.Cost*	*Ave.Cost*	*Std.Dev*	*Success No.*
1.c50	524.61	525.45	2.56	27	524.61	**524.61**	0	**30**	524.61	525.73	2.90	26
2.c75	835.26	842.32	4.04	15	835.26	840.28	3.52	22	835.26	**839.53**	3.45	**26**
3.c100	826.14	829.43	2.39	11	826.14	829.73	2.08	9	826.14	**829.13**	1.94	**12**
4.c100b	819.56	819.56	0	30	819.56	819.56	0	30	819.56	819.56	0	**30**
5.c120	1042.11	1044.18	2.21	18	1042.11	1043.08	1.13	22	1042.11	**1042.83**	0.94	**27**
6.c150	1032.50	1043.27	5.67	13	1034.19	1044.73	5.87	12	**1030.67**	**1041.97**	6.27	**16**
7.c199	**1304.87**	**1321.17**	5.98	**14**	1313.46	1325.48	5.99	9	1308.92	1322.18	7.51	**14**



(a) A-n54-k7 (b) A-n69-k9 (c) A-n80-k10

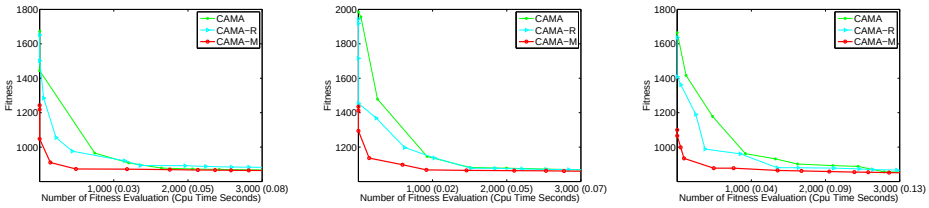
(d) B-n41-k6 (e) B-n63-k10 (f) B-n78-k10

Figure 7: Search convergence graphs of $CAMA$, $CAMA-R$, and $CAMA-M$ on representative CVRP "AUGERAT" benchmarks. Y-axis: Fitness value, X-axis: Number of Fitness Evaluation or CPU Time in Seconds.

generated using K-Means clustering. A CVRP instance is thus represented as input matrix $\mathbf{X}$, see Fig. 5, which is composed of the coordinate features for all tasks in the problem. The desired vehicle assigned for each task (i.e., task assignment) is then given by the OS optimized solution, $\mathbf{Y}$, of the respective CVRP instances.

Besides the proposed meme induced population initialization procedure, two other commonly used initialization procedures for generating the population of solution individuals in the state-of-the-art baseline *CAMA* are also investigated here to verify the efficiency and effectiveness of the proposed memetic evolutionary search paradigm. The first is the simple random approach for generating the initial population, which is labeled here as *CAMA-R*. The second is the informed heuristic population initialization procedure proposed in baseline *CAMA* [9]. In particular, the initial population is a fusion of solution generated by *Backward Sweep*, *Saving*, and *Forward Sweep* and random initialization approaches. The CAMA that employs the meme induced population initialization procedure of the proposed memetic evolution is then notated as *CAMA-M*, where the population of individuals intelligently generated based on the fit memes that have been accumulated from past CVRP solving experiences via cultural evolutionary mechanisms of the *meme learning*, *meme selection*, *meme variation* and *meme imitation*. If no memes have been accumulated so far, *CAMA-M* shall behave exactly like the baseline *CAMA*.

Last but not the least, the operator and parameter settings of *CAMA-R*, *CAMA*, *CAMA-M* are kept the same as that of [9] for the purpose of fair comparison. For *CAMA-M*, the MMD of Equation 10 is augmented with the demand of each task as one of the problem feature.



(a) E-n76-k10 (b) E-n76-k14 (c) E-n101-k8

Figure 8: Search convergence graphs of $CAMA$, $CAMA-R$, and $CAMA-M$ on representative CVRP "CE" benchmarks. Y-axis: Fitness value, X-axis: Number of Fitness Evaluation or CPU Time in Seconds.

6.1.2 Result and Discussion

All the results obtained by the respective algorithms considered, across 30 independent runs on the CVRP instances, are summarized in Table 4, Table 5 and Table 6. The $Ave.Cost$ value of *CAMA* on each CVRP instance is used as the baseline for the $Success.No$ criterion. The values in "$B.Cost$", "$Ave.Cost$" and "$Success\ No.$" denoting superior performance are then highlighted in bold font.

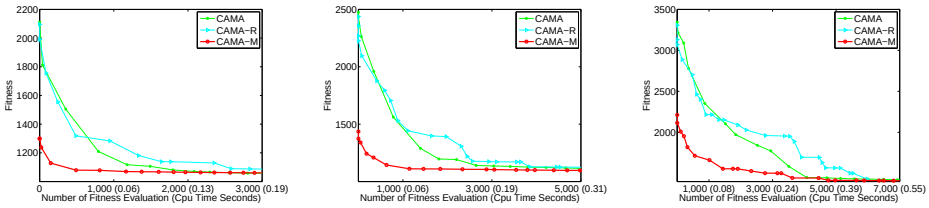
It can be observed from the results in Table 4 and Table 5 that in overall, *CAMA-R* achieved improved solution quality over *CAMA* in terms of $Ave.Cost$ and $Success\ No.$ on most of the "AUGERAT" and "CE" CVRP instances. Particularly, on instance "A-n54-k7", *CAMA-R* consistently converges to the best solution fitness across all the 30 independent runs, as denoted by "$B.Cost$". *CAMA-R* also attained an improved $B.Cost$ solution over *CAMA* on instance "B-n63-k10". However, as observed in Table 6, on the "CHRISTOFIDES" benchmarks, where the size of the problem instances scale up, relative to "AUGERAT" and "CE" (i.e., the graphs are bigger in size, with larger number of customers or vertices that require servicing), the current baseline state-of-the-art *CAMA* is noted to exhibit superior performances over *CAMA-R* in terms of $Ave.Cost$. In terms of $B.Cost$, *CAMA* is also observed to have attained superior solution quality over *CAMA-R* on instance "c199". Since the only difference between *CAMA-R* and *CAMA* lies in the heuristic bias introduced in the population initialization phase of the latter, it is possible to infer from the performances of *CAMA* and *CAMA-R* that the appropriate inductive biases made available in *CAMA* have been effective in narrowing down the search regions of the large scale problems.

Moving on next to the proposed memetic computational search paradigm, the generation of the initial population of solutions is now guided by the memes learned from past problem-solving experiences. These memes thus serve as the instructions learned from the experiences along with the CARP instances solved, which are then imitated to enhance the search on new problem instances. As discussed in Section 3, the "meme pool" of *CAMA-M* is empty when the first CVRP instance is encountered (e.g., "A-n32-k5" of "AUGERAT" benchmark problem), and thus *CAMA-M* behaves like the baseline *CAMA*. As more CVRP instances are encountered, the

meme learning, *selection*, *variation* and *imitation* mechanisms shall kick in to learn and generalize memes that would induced positive bias to the evolutionary search of new CVRP problems. It can be observed from Table 4, Table 5 and Table 6 that the *CAMA-M* converges to similar solutions attained by both *CAMA-R* and *CAMA* on the first problem instance of each CVRP benchmarks (since no memes are learned yet), but demonstrates superior performances over *CAMA-R* and *CAMA* on subsequent CVRP instances. In particular, on "AUGERAT" and "CE" benchmarks, *CAMA-M* exhibits superior performances in terms of $Ave.Cost$ and $Success\ No.$ on 13 out of total 19 CVRP instances. In addition, on "CHRISTOFIDES" benchmarks, *CAMA-M* also attained improved solution quality in terms of $Ave.Cost$ and $Success\ No.$ on 4 out of total 7 CVRP instances. Since *CAMA*, *CAMA-R* and *CAMA-M* shares a common baseline evolutionary solver, i.e., *CAMA*, and differing only in terms of the population initialization phase, the superior performance of *CAMA-M* can clearly be attributed to the effectiveness of the proposed memetic evolutionary search paradigm where meme induced solutions are intelligently generated from previous problem solving experiences for enhancing future evolutionary searches.

To assess the efficiency of the proposed memetic search paradigm, representative convergence graphs of *CAMA*, *CAMA-R* and *CAMA-M* on the CVRP benchmarks are also presented in Fig. 7, Fig. 8 and Fig. 9, respectively. Note that the Y-axis of the figures denote the actual fitness value obtained, while the X-axis gives the respective computational effort incurred in terms of both the Number of Fitness Evaluation made so far and CPU Time in seconds. As observed, *CAMA-R* is noted to converge faster than *CAMA* on most of the instances in "AUGERAT" and "CE" benchmarks (e.g., Fig. 7(a), Fig. 7(f), Fig. 8(a), Fig. 8(b), etc.). However, on the large scale "CHRISTOFIDES" benchmarks, *CAMA* poses to search more efficiently, especially from beyond 1000 number of fitness evaluations (e.g., Fig. 9(a), Fig. 9(b), Fig. 9(c), etc.). On the other hand, it can be observed that *CAMA-M* converges faster than both *CAMA-R* and *CAMA* on all the CVRP benchmarks. Particularly, on instances "B-n41-k6" (Fig. 7(d)), "B-n63-k10" (Fig. 7(e)) and "B-n78-k10" (Fig. 7(f)), etc., *CAMA-M* takes only approximately 250 number of fitness evaluations to arrive at the solution quality of *CAMA-R* and *CAMA*, which incurred more than 1500 number of fitness evaluations. On the large scale instances, such as "c120" (Fig. 9(a)), "c150" (Fig. 9(b)), etc., *CAMA-M* brings about at least 2000 number of fitness evaluations savings to arrive at the similar solution qualities of *CAMA* and *CAMA-R*.

Next, we showcase samples of the initial solutions by *CAMA*, *CAMA-R* and *CAMA-M*, as well as the converged optimized solution of *CAMA* on solving problem instance "B-n41-k6" in Fig. 10. In the present context, an initial solution of *CAMA-M*, which has been intelligently generated using the positive biases introduced by memes learned and generalized from past experiences of problem solving on instances "A-n32-k5", "A-n54-k7", "A-n60-k9" and "A-n69-k9", is illustrated. In Fig. 10, each node denotes a customer that needs to be serviced, and the nodes with the same color and shape shall be serviced by a common route or vehicle. As



(a) c120 (b) c150 (c) c199

Figure 9: Search convergence graphs of $CAMA$, $CAMA-R$, and $CAMA-M$ on representative CVRP "CHRISTOFIDES" benchmarks. Y-axis: Fitness value, X-axis: Number of Fitness Evaluation or CPU Time in Seconds.

observed, the task distributions for the initial solution of *CAMA-M* is noted to be most similar to that of the converged optimized solution of *CAMA*, as compared to that of *CAMA* and *CAMA-R*. Besides the task distributions, we also magnify portions of the figures to showcase some service orders of the initial solution in *CAMA-M*, relative to that of the converged optimized solution of baseline *CAMA*, in Fig. 10(c) and Fig. 10(d), respectively. The magnified subfigures illustrate high similarities between their respective service orders. This suggests that the service order structures of the converged optimized solution for instances "A-n32-k5", "An54-k7", "A-n60-k9" and "A-n69-k9" were successful learned and preserved by the *meme learning operator*, which are subsequently imitated as positively biased solutions (of the initial population in *CAMA-M* search) induced by the generalized meme, through the cultural evolutionary or memetic mechanisms of *meme selection*, *variation* and *imitation*.

6.2 Capacitated Arc Routing Problem

Next, we proceed to evaluate the efficiency and effectiveness of the proposed memetic evolutionary search paradigm on the Capacitated Arc Routing Problem domain.

6.2.1 Empirical Configuration

The well-established *egl* benchmark is used in the present experimental study on CARP. The data set was generated by Eglese based on data obtained from the winter gritting application in Lancashire [21, 22, 31]. This data set has been commonly used as benchmark dataset in the literature for CARP solving [23, 35]. It consists of two series of datasets (i.e., "E" and "S" series) with a total of 24 instances. In particular, CARP instances in "E" series have smaller number of vertices, task or edges than that in "S" series, thus the problem structures of "E" series are deem to be simpler than "S" series. The detailed properties of each *egl* instance are presented in Table 7 and Table 8. "$|V|$", "$|E_R|$", "E" and "LB" denote the number of vertices, number of tasks, total number of edges and lower bound, of each problem

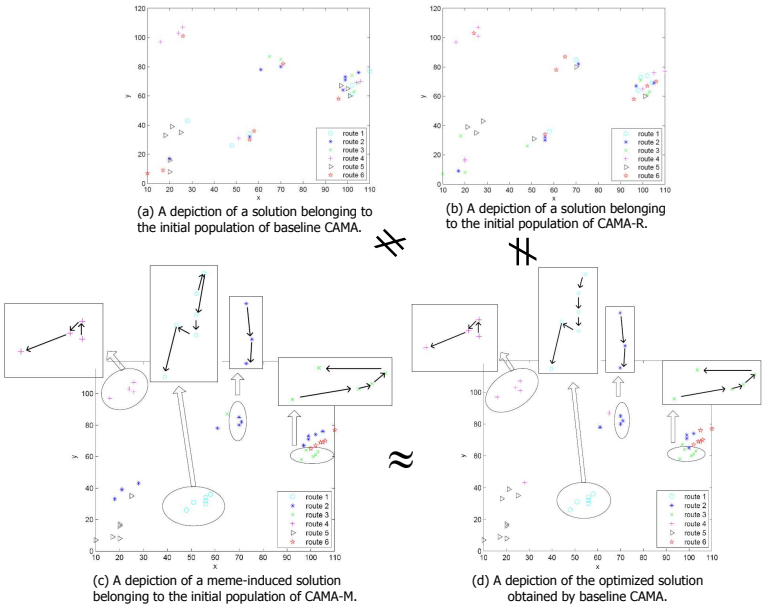


Figure 10: Illustration of the optimized solution and initial solutions of the algorithms considered on "B-n41-k6" CVRP benchmark.

Table 7: Properties of the *egl* "E" Series CARP benchmarks.

	"E" Series											
Data Set	E1A	E1B	E1C	E2A	E2B	E2C	E3A	E3B	E3C	E4A	E4B	E4C
V	77	77	77	77	77	77	77	77	77	77	77	77
E_r	51	51	51	72	72	72	87	87	87	98	98	98
E	98	98	98	98	98	98	98	98	98	98	98	98
LB	3548	4498	5566	5018	6305	8243	5898	7704	10163	6048	8884	11427

instance, respectively.

In traditional CARP, each task is represented by a corresponding *head vertex*, *tail vertex*, *travel cost* and *demand (service cost)*. The shortest distance matrix of the vertices is first derived by means of the Dijkstra's algorithm [20], i.e., using the distances available between the vertices of a CARP. The coordinate features (i.e., locations) of each task are then approximated by means of multidimensional scaling [3]. In this manner, each task is represented as a node in the form of coordinates. A CARP instance in the current setting is thus represented by input vector $\mathbf{X}$ composing of the coordinate features of all tasks in the problem. Such a representation would allow standard clustering approaches, such as the K-Means algorithm to be conducted on the CARP in task assignments, and allow the pairwise distances sorting among tasks available for preserving service orders. The label information of each task in $\mathbf{Y}$ belonging to the CARP instance, is as defined by the optimized solution of CARP.

For empirical comparisons, three solution population initialization procedures

Table 8: Properties of the *egl* "S" Series CARP benchmarks. Y-axis: Fitness value, X-axis: Number of Fitness Evaluation or CPU Time in Seconds.

	"S" Series											
Data Set	S1A	S1B	S1C	S2A	S2B	S2C	S3A	S3B	S3C	S4A	S4B	S4C
V	140	140	140	140	140	140	140	140	140	140	140	140
E_r	75	75	75	147	147	147	159	159	159	190	190	190
E	190	190	190	190	190	190	190	190	190	190	190	190
LB	5018	6384	8493	9824	12968	16353	10143	13616	17100	12143	16093	20375

Table 9: Statistic results of *ILMA*, *ILMA-R*, and *ILMA-M* on egl "E"-Series CARP benchmarks.

Data	$ILMA$				$ILMA-R$				$ILMA-M$ (Proposed Method)			
Set	*B.Cost*	*Ave.Cost*	*Std.Dev*	*Success No.*	*B.Cost*	*Ave.Cost*	*Std.Dev*	*Success No.*	*B.Cost*	*Ave.Cost*	*Std.Dev*	*Success No.*
1.E1A	3548	3548	0	30	3548	3548	0	30	3548	3548	0	30
2.E1B	4498	4517.63	12.45	9	4498	4517.80	13.19	11	4498	**4513.27**	14.93	**14**
3.E1C	5595	5599.33	7.56	22	5595	5601.73	8.84	18	5595	**5598.07**	8.58	**26**
4.E2A	5018	5018	0	30	5018	5018	0	30	5018	5018	0	30
5.E2B	6317	6341.53	20.15	14	6317	6344.03	22.38	10	6317	**6337.90**	11.90	**16**
6.E2C	8335	8359.87	36.61	21	8335	8355.07	39.26	24	8335	**8349.97**	26.16	**25**
7.E3A	5898	5921.23	30.07	20	5898	5916.93	30.50	21	5898	**5910.97**	30.57	**25**
8.E3B	7777	7794.77	23.08	22	7777	7792.17	29.95	23	**7775**	**7788.70**	15.74	**25**
9.E3C	10292	**10318.73**	40.89	**20**	10292	10327.07	33.46	15	10292	10319.16	36.15	**20**
10.E4A	6461	6471.37	15.16	**21**	**6458**	6481.77	22.77	15	6461	**6469.80**	10.27	**21**
11.E4B	8995	9060.67	45.29	10	8993	9067.93	50.54	12	**8988**	**9053.97**	41.49	**15**
12.E4C	**11555**	**11678.47**	73.57	**16**	11594	11728.30	82.39	10	11576	11697.27	76.98	14

Table 10: Statistic results of *ILMA*, *ILMA-R*, and *ILMA-M* on egl "S"-Series CARP benchmarks.

Data	$ILMA$				$ILMA-R$				$ILMA-M$ (Proposed Method)			
Set	*B.Cost*	*Ave.Cost*	*Std.Dev*	*Success No.*	*B.Cost*	*Ave.Cost*	*Std.Dev*	*Success No.*	*B.Cost*	*Ave.Cost*	*Std.Dev*	*Success No.*
1.S1A	5018	5023.93	18.14	27	5018	5025.97	26.97	27	5018	5023.67	25.39	27
2.S1B	6388	6404.07	22.96	20	6388	6403.30	20.89	20	6388	**6392.80**	14.65	**27**
3.S1C	8518	8577.63	44.18	**11**	8518	8581.67	33.98	8	8518	**8576.53**	33.12	**11**
4.S2A	9920	10037.43	61.51	14	9925	10050.30	54.24	11	**9896**	**10010.20**	67.13	**20**
5.S2B	13191	13260.03	45.37	16	13173	13257.90	48.94	15	**13147**	**13245.56**	53.02	**22**
6.S2C	16507	**16605.10**	65.26	**16**	16480	16626.43	62.90	10	**16468**	16615.40	76.79	10
7.S3A	10248	10342.77	47.56	11	10278	10369.40	52.42	13	**10239**	**10339.40**	53.29	**18**
8.S3B	13764	13912.97	79.85	12	13779	13899.70	76.96	17	**13749**	**13881.33**	85.78	**21**
9.S3C	17274	17371.10	79.12	20	17277	17402.43	74.37	12	**17261**	**17355.03**	48.23	**21**
10.S4A	12335	12498.47	67.72	15	12407	12534.47	63.23	11	**12320**	**12489.43**	83.91	**19**
11.S4B	**16378**	16542.93	89.65	17	16443	16540.43	87.52	19	16415	**16512.43**	57.54	**20**
12.S4C	20613	20794.80	77.51	14	20589	20841.13	85.53	5	**20564**	**20774.20**	86.78	**17**

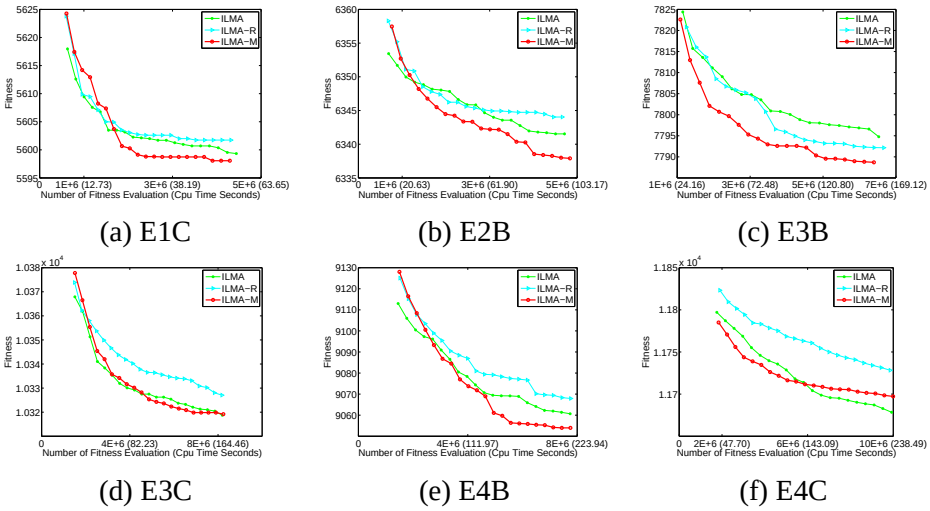


Figure 11: Search convergence graphs of $ILMA$, $ILMA-R$, and $ILMA-M$ on representative CARP "E"-Series benchmarks. Y-axis: Fitness value, X-axis: Number of Fitness Evaluation or CPU Time in Seconds.

based on *ILMA* are also considered here. The first is a simple random approach, which is labeled here as *ILMA-R*. The second is the informed heuristic based population generation procedure used in the baseline state-of-the-art *ILMA* [35] for CARP. There, the initial population is formed by a fusion of chromosomes generated from Augment_Merge [24], Path_Scanning [25], Ulusoy's Heuristic [48] and the simple random initialization procedures. The last is our proposed solution generation using past problem solving experiences with memetic evolution, which is labeled here as *ILMA-M*. In *ILMA-M*, the meme pool is empty at start, while new memes shall be accumulated with increasing CARP instances optimized. Thus, once again, the *ILMA-M* performs exactly like *ILMA* on the first problem instance encountered, since it is meant to serve as an intelligent *ILMA* whose intellectual shall increase with increasing problems solved.

To facilitate a fair comparison and verify the benefits of the learning from past experiences, the evolutionary operators of *ILMA* and its variants are configured as consistent to that reported in [35]. Further, in *ILMA-M*, the MMD of Equation 10 is augmented with the demand of each task.

6.2.2 Result and Discussion

Table 9 and Table 10 tabulate the results that measure the solution quality on the "E"-Series and "S"-Series *egl* CARP datasets as obtained by the respective algorithms, across 30 independent runs. The $Ave.Cost$ value of *ILMA* on each problem instance is used as threshold level in the $Success\ No.$ criterion. In the tables, the method with superior performance with respect to "$B.Cost$", "$Ave.Cost$" and "$Success\ No.$" are highlighted in bold font.

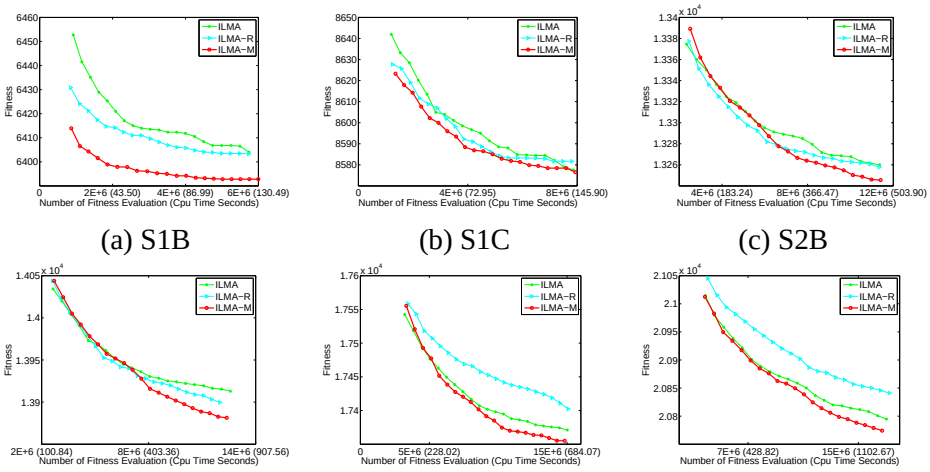
As can be observed from Table 9 and Table 10, the *ILMA-R* has attained competitive performance on solution quality over *ILMA-R*, on instances such as "E1-A", "E1-B", "E2-A", "S1-A", "S1-B" and "S2-A", etc. However, with the incorporation of heuristic information as inductive search bias in the baseline state-of-the-art *ILMA*, it is noted to attain improved performance in terms of solution quality over *ILMA-R* on the large scale instances (i.e., instances with greater service arcs, travel edges, or number of vertices, etc.), which include "E3-C", "E4-A", "E4-B", "E4-C", "S3-C", "S4-A", and "S4-C".

Turning then to the proposed memetic evolutionary search paradigm, it can be seen from Table 9 and Table 10 that, *ILMA-M* performed competitively to *ILMA* on instances "E1-A" and "S1-A" as expected since these are the first encountered problem instances of *ILMA-M* on the "E" and "S" *egl* CARP benchmarks, respectively, where no memes are available in the meme pool. As more problem instances are optimized, the *ILMA-M* is observed to demonstrate superior performance over *ILMA* in terms of $Ave.Cost$ on 18 out of total 24 *egl* benchmarks. In terms of $B.Cost$, *ILMA-M* achieved 2 and 8 improved solution qualities over *ILMA* and *ILMA-R* on the "E" and "S" series *egl* datasets, respectively.

Similarly, to demonstrate the efficiency of our proposed memetic evolution, the search convergence traces of *ILMA*, *ILMA-R*, and *ILMA-M* on several representative instances of "E" and "S" series *egl* benchmarks are depicted in Fig. 11 and Fig. 12, respectively. As can be observed, for *ILMA* and *ILMA-R*, on the simpler problems such as "E1-C" (Fig. 11(a)), "E2-B" (Fig. 11(b)), "S1-C" (Fig. 12(b)) and "S2-B" (Fig. 12(c)), etc., *ILMA* and *ILMA-R* demonstrated competitive convergence speed. Nevertheless, as the complexity of the problems increase, i.e., such as "E3-C" (Fig. 11(d)), "E4B" (Fig. 11(e)), "E4C" (Fig. 11(f)), "S3-C" (Fig. 12(e)) and "S4-C" (Fig. 12(f)), etc., *ILMA* consistently converges faster than *ILMA-R* with improved solution attained. Further, for our proposed memetic evolutionary search paradigm, on both the "E" and "S" series *egl* benchmarks, *ILMA-M* consistently converges more efficiently than *ILMA* and *ILMA-R* on almost all the instances presented. Overall, *ILMA-M* is noted to bring about at least 2×10^6 savings in the number of fitness function evaluations to arrive at the solutions attained by both *ILMA* and *ILMA-R* on most of the CARP instances (e.g., Fig. 11(a), Fig. 11(c), Fig. 11(e), Fig. 12(c), Fig. 12(d), etc.). For instance, it is worth noting that on instance "S1-B" (Fig. 11(a)), *ILMA-M* used up a total of 1.5×10^6 number of fitness function evaluations to converge at the solution incurred by *ILMA* and *ILMA-R*, which otherwise used up a significant large fitness evaluations of approximately 6×10^6. This equates to a total savings of 4 times by *ILMA-M* over *ILMA* and *ILMA-R*.

7 Conclusion

In this paper, we have proposed a *Memetic Computational Paradigm for search* to model how human solves problems and presented a novel study towards intelligent



(a) S1B (b) S1C (c) S2B

(d) S3B (e) S3C (f) S4C

Figure 12: Search convergence graphs of $ILMA$, $ILMA-R$, and $ILMA-M$ on representative CARP "S"-Series benchmarks. Y-axis: Fitness value, X-axis: Number of Fitness Evaluation or CPU Time in Seconds.

evolutionary optimization of problems through the transfers of structured knowledge in the form of memes learned from previous problem-solving experiences, to enhance future evolutionary searches. In particular, the four culture-inspired operators, namely, *Meme Learning*, *Meme Selection*, *Meme Variation* and *Meme Imitation* have been realized based on the HSIC, MMD criterion and K-means Clustering in the context of combinatorial routing problems. In contrast to existing works, the proposed novel memetic search paradigm enables knowledge transfer and reuse in evolutionary optimization across problems of different size, structure, or representation, etc. Last but not least, comprehensive studies on two widely studied NP-hard routing problems, namely, CVRP and CARP, confirmed the efficiency and effectiveness of the proposed paradigm for intelligent evolutionary optimization across problems.

References

[1] P. Augerat and J. M. Belenguer et. al. Computational results with a branch and cut code for the capacitated vehicle routing problem. *Research Report*, pages 949–M, 1995.

[2] S. Blackmore. The meme machine. *Oxford University Press*, 1999.

[3] I. Borg and P. J. F. Groenen. *Modern Multidimensional Scaling: Theory and Applications*. Springer, 2005.

[4] 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*, pages 49–57, 2006.

[5] P. Bosman and E. de Jong. Adaptation of a success story in gas: Estimation-of-distribution algorithms for tree-based optimization problems. In *Success in Evolutionary Computation*, volume 92 of *Studies in Computational Intelligence*, pages 3–18. Springer, 2008.

[6] J. D. Bransford, A. L. Brown, and R. R. Cocking. *How People Learn: Brain, Mind, Experience, and School*. National Academies Press, 2000.

[7] R. Brodie. Virus of the mind: The new science of the meme. *Seattle: Integral Press*, 1996.

[8] J. P. Byrnes. *Cognitive development and learning in instructional contexts*. Allyn and Bacon (Boston), 1996.

[9] 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,*, (99):1–8, 2012.

[10] 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*, (5):591–607, 2011.

[11] X. S. Chen, Y. S. Ong, M. H. Lim, and S. P. Yeo. Cooperating memes for vehicle routing problems. *International Journal of Innovative Computing*, 7(11), 2011.

[12] N. Christofides and S. Eilon. An algorithm for the vehicle dispatching problem. *Operational Research Quarterly*, 20:309 – 318, 1969.

[13] N. Christofides, A. Mingozzi, and P. Toth. The vehicle routing problem. *Combinatorial Optimization*, pages 315–338, 1979.

[14] N. Christofides, A. Mingozzi, and P. Toth. Exact algorithm for the vehicle routing problem, based on spanning tree and shortst path relaxations. *Mathematical Programming*, 20:255–282, 1981.

[15] P. C. Chu and J. E. Beasley. A genetic algorithm for the generalised assignment problem. *Computers & Operations Research*, 24(1):17–23, 1997.

[16] J. F. Cordeau, G. Laporte, and A. Mercier. A unified tabu search heuristic for vehicle routing problems with time windows. *Journal of The Operational Research Society*, 52:928–936, 2001.

[17] P. Cunningham and B. Smyth. Case-based reasoning in scheduling: Reusing solution components. *The International Journal of Production Research*, 35(4):2947–2961, 1997.

[18] G. Dantzig and J. H. Ramser. The truck dispatching problem. *Management Science*, 6:80–91, 1959.

[19] R. Dawkins. The selfish gene. *Oxford: Oxford University Press*, 1976.

[20] E. W. Dijkstra. A note on two problems in connection with graphs. *Numerische Mathematik*, 1:269–271, 1959.

[21] R. W. Eglese. Routing winter gritting vehicles. *Discrete Applied Mathematics*, 48(3):231–244, 1994.

[22] R. W. Eglese and L. Y. O. Li. A tabu search based heuristic for arc routing with a capacity constraint and time deadline. *in Metaheuristics: theory and applications, I. H. Osman and J. P. Kelly, Eds. Boston: Kluwer Academic Publishers*, pages 633–650, 1996.

[23] L. Feng, Y. S. Ong, Q. H. Nguyen, and A. H. Tan. Towards probabilistic memetic algorithm: An initial study on capacitated arc routing problem. *IEEE Congress on Evolutionary Computation 2010*, pages 18–23, 2010.

[24] B. Golden and R. Wong. Capacitated arc routing problems. *Networks*, 11(3):305–315, 1981.

[25] B. L. Golden, J. S. DeArmon, and E. K. Baker. Computational experiments with algorithms for a class of routing problems. *Computer & Operation Research*, 10(1):47–59, 1983.

[26] G. Grant. Memetic lexicon. *in Principia Cybernetica Web*, 1990.

[27] A. Gretton, O. Bousquet, A. Smola, and B. Schölkopf. Measuring statistical dependence with hilbert-schmidt norms. *Proceedings Algorithmic Learning Theory*, pages 63–77, 2005.

[28] M. T. Jensen. Generating robust and flexible job shop schedules using genetic algorithms. *IEEE Transactions on Evolutionary Computation*, 7(3):275–288, 2003.

[29] J. D. Knowles and D. W. Corne. M-paes: a memetic algorithm for multiobjective optimization. *IEEE Congress on Evolutionary Computation*, 1:325 – 332, 2000.

[30] P. Lacomme, C. Prins, and W. Ramdane-Chérif. Competitive memetic algorithms for arc routing problem. *Annals of Operational Research*, 141(1–4):159–185, 2004.

[31] L. Y. O. Li and R. W. Eglese. An interactive algorithm for vehicle routing for winter-gritting. *Journal of the Operational Research Society*, 47(2):217–228, 1996.

[32] S. W. Lin, Z. J. Lee, K. C. Ying, and C. Y. Lee. Applying hybrid meta-heuristics for capacitated vehicle routing problem. *Expert Systems with Applications*, 36(2, Part 1), 2009.

[33] S. J. Louis and J. McDonnell. Learning with case-injected genetic algorithms. *IEEE Transactions on Evolutionary Computation*, 8(4):316–328, 2004.

[34] A. Lynch. Thought contagion as abstract evolution. *Journal of Ideas*, 2:3–10, 1991.

[35] Y. Mei, K. Tang, and X. Yao. Improved memetic algorithm for capacitated arc routing problem. *IEEE Congress on Evolutionary Computation*, pages 1699–1706, 2009.

[36] M. Minsky. *The society of mind*. Simon & Schuster, Inc., 1986.

[37] Q. C. Nguyen, Y. S. Ong, and M. H. Lim. A probabilistic memetic framework. *IEEE Transactions on Evolutionary Computation*, 13(3):604–623, 2009.

[38] Y. S. Ong and A. J. Keane. Meta-lamarckian learning in memetic algorithms. *IEEE Transactions on Evolutionary Computation*, 8(2):99–110, 2004.

[39] Y. S. Ong, M. H. Lim, and X. S. Chen. Research frontier: Memetic computation - past, present & future. *IEEE Computational Intelligence Magazine*, 5(2):24–36, 2010.

[40] S. J. Pan and Q. Yang. A survey on transfer learning. *IEEE Transactions on Knowledge and Data Engineering*, 22(10):1345–1359, 2010.

[41] C. Prins. A simple and effective evolutionary algorithm for the vehicle routing problem. *Computer & Operations Research*, 31:1985–2002, 2004.

[42] M. Reimann, K. Doerner, and R. F. Hartl. D-ants: savings based ants divide and conquer the vehicle routing problem. *Computer & Operations Research*, 31:563–591, 2004.

[43] M. A. Runco and S. Pritzker. *Encyclopedia of Creativity*. Academic Press, 1999.

[44] 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*, pages 815–822, 2007.

[45] K. Tang, Y. Mei, and X. Yao. Memetic algorithm with extended neighborhood search for capacitated arc routing problems. *IEEE Transactions on Evolutionary Computation*, 13(5):1159–1166, 2009.

[46] K. Tang, Y. Mei, and X. Yao. Memetic algorithm with extended neighborhood search for capacitated arc routing problems. *IEEE Transactions on Evolutionary Computation*, 13(5):1151–1166, 2009.

[47] A. Barkat Ullah, R. Sarker, D. Cornforth, and C. Lokan. Ama: a new approach for solving constrained real-valued optimization problems. *Soft Computing*, 13(8):741–762, 2009.

[48] G. Ulusoy. The fleet size and mix problem for capacitated arc routing. *Eur. J. Oper. Res.*, 22(3):329–337, 1985.

[49] H. F. Wang, D. W. Wang, and S. X. Yang. A memetic algorithm with adaptive hill climbing strategy for dynamic optimization problems. *Soft Computing*, 13(8-9):763–780, 2009.

[50] N. Yoshiike and Y. Takefuji. Vehicle routing problem using clustering algorithm by maximum neural networks. *Intelligent Processing and Manufacturing of Materials*, 2:1109 – 1113, 1999.

[51] J. Zhuang, I. Tsang, and S. C. H. Hoi. A family of simple non-parametric kernel learning algorithms. *Journal of Machine Learning Research (JMLR)*, 12:1313–1347, 2011.