A Conceptual Modeling of Meme Complexes in Stochastic Search

Xianshun Chen and Yew Soon Ong

***Abstract*—In science, gene provides the instruction for making proteins, while meme is the sociocultural equivalent of a gene containing instructions for carrying out behavior. Taking inspiration from nature, we model the memeplex in search as instructions that specify the coadapted meme complexes of individuals in their lifetime. In particular, this paper presents a study on the conceptual modeling of meme complexes or memeplexes for more effective problem solving in the context of modern stochastic optimization. The memeplex representation, credit assignment criteria for meme coadaptation, and the role of emergent memeplexes in the lifetime learning process of a memetic algorithm in search are presented. A coadapted memetic algorithm that takes the proposed conceptual modeling of memeplexes into actions to solve capacitated vehicle routing problems (CVRPs) of diverse characteristics is then designed. Results showed that adaptive memeplexes provide a means of creating highly robust, self-configuring, and scalable algorithms, thus generating improved or competitive results when benchmarking against several existing adaptive or human-designed state-of-the-art memetic algorithms and metaheuristics, on a plethora of CVRP sets considered.**

***Index Terms*—Adaptive memetic algorithms, meme cooperation, memeplex, memetic algorithm (MA), search neighborhood structure, vehicle routing problems (VRPs).**

I. Introduction

In the recent decade, the underlying potential of memes with its causal effects on learning and its ability to propagate and evolve has sparked inspiration among researchers in the field of computational intelligence [1]. In memetic computation (MC), multifacet forms of a meme have been investigated and modeled in different lights [2]. Meme, for instance, has been perceived as "portion of an organism's neurally stored information" [3] that manifests into a distinct memorable unit in the mental state of an agent [4]. In addition, memes are also modeled as "constellation of activated neuronal synapses" [5] or "hierarchically arranged components of semantic memory" [6] imprinted in the brain of an autonomous agent. Inspired by the inherent nature of memes in wanting to propagate and evolve, Brodie [7], on the other hand, has perceived memes as "contagious information pattern that replicates by parasitically infecting human minds," while Durham [8] portrayed a broader notion of memes as "any kind, amount, and configuration of information in culture that shows both variation and coherent transmission." In the context of computational intelligence, MC, thus, defines *a paradigm that uses the notion of meme(s) as units of information, i.e., the building blocks, encoded in computational representations for the purpose of enhanced problem solving* [1], [2].

Memetic Algorithm (MA) represents a notable interest of *MC* that is widely established as a methodology that addresses the synergy of population-based approaches with separate lifetime learning process [9]–[11], where the latter is commonly referred to as a *meme* [12]–[15]. In particular, MAs are inspired by Neo-Darwinian principles of natural evolution and Dawkins' notion of a meme defined as a unit of cultural evolution that is capable of refinement. Recent studies on MAs have shown that they can converge to high-quality solutions more efficiently than their conventional counterparts on a wide range of domains covering problems in discrete [16]–[18], continuous [19]–[21], dynamic [22]–[24], and multiobjective optimizations [25]–[27], including embodied intelligence and evolution [28]–[30], etc.

In recent years, a significant number of dedicated MAs have been specially crafted to solve domain-specific problems effectively. While it is now well established that the incorporation or embedding of knowledge about the underlying problem within the search algorithms is beneficial for improving search performance [31]–[33], the current human-intensive process may not be the most productive practice. As opposed to the designing of a dedicated MA whenever a new problem arises, researchers are now moving toward robust self-configuring search methodology that is capable of learning and adapting to the given problem of interest while the search progresses online [1]. In essence, adaptation on the parameters and operators represents one of the key issues of the present day evolutionary search in computational intelligence. In evolutionary optimization research, for instance, some of the core issues of MA design considered by researchers involved adapting 1) the degree of lifetime learning versus population-based search [14], 2) subset of the population to undergo lifetime learning [34], 3) the amount of computational or time budget allocated for lifetime learning [14], and last but not least 4) the type of meme(s) considered in lifetime learning [13].

In MA search, a meme has generally manifested in the form of a local search solver to enhance the search intensity of population-based search algorithms. Meme adaptation in MA including *meta-Lamarckian learning* [12], *simple inheritance mechanism* [35], *cost-benefit-based adaptation scheme* [36], *self-generating MAs* [37], etc., represents one of the core design issues of MA that has garnered increasing attention in the

Manuscript received November 13, 2010; revised November 4, 2011; accepted January 27, 2012. Date of publication April 5, 2012; date of current version August 15, 2012. This work was supported in part by the Media Development Authority of Singapore, Singapore-MIT GAMBIT Game Lab and the Center for Computational Intelligence (C2I) at Nanyang Technological University. This paper was recommended by Associate Editor M.-H. Lim.

The authors are with the School of Computer Engineering, Nanyang Technological University, Singapore 639798 (e-mail: chen0469@ntu.edu.sg; asysong@ntu.edu.sg).

Digital Object Identifier 10.1109/TSMCC.2012.2188832

recent years, since the influence of meme was shown extensively to have a major impact on search performance [13]. Particularly, studies have demonstrated that the search performance obtained by MAs may not be better than that obtained by the traditional EA alone when prior knowledge on suitable problem-specific meme is not well exploited. The adaptation can be achieved based on feedback information obtained along the search such as population diversity [10], [38], or other search performance metrics [39], [40]. In this paper, our interest is on adapting the type of meme(s) to employ for lifetime learning. It is worth noting that every lifetime learning scheme or heuristic, except for uniform random search, introduces some kind of bias in the search. In discrete combinatorial search, different lifetime learning schemes have different biases and induce a unique topological move or neighborhood structure in the search space. It is these biases that make a lifetime learning scheme effective for discovery of the global optimum. However, while inducing bias in the search does generate more effective lifetime learning on certain problems, it can also lead to deteriorating search performance on problems where the incorporated bias does not fit in well, thus leading to the loss in the generality of the algorithm across diverse problems. Meme adaptation as discussed can, thus, serve to alleviate this challenge to a certain extent by adaptively selecting different lifetime learning schemes along the search.

Moving back to basics, it is worth noting that in science, gene provides the instruction for making proteins, while meme is the sociocultural equivalent of a gene containing instructions for carrying out behavior. To induce more flexible lifetime learning of neighborhoods, our interest in this paper is to raise aspirations and introduce a structural form of memetic existence, i.e., coadapted meme complexes or memeplexes, in the lifetime learning phase of MA, where few studies or none has considered to date. A memeplex is essentially a coadapted stable set of mutually assisting memes [41] that work together to achieve more than what each meme could accomplish alone [42], [43]. In this paper, we focus on the concept of *memeplex* as a self-consistent paradigm for problem solving whereby memes engage in collective learning and actions that eventually bring higher efficacy to stochastic search. Represented in the neural wiring structure or so-called memetic network, a memeplex takes the form of a distributed and dynamically evolving network with memes complementing one another on the search. Each meme, thus, assumes a functional status in the network spatial conformations that induces a unique learning bias and search neighborhood structure.

The rest of this paper is organized as follows: Section II begins with the introduction of memeplexes, and the proposed neural wiring structure is described in Sections II and III. The design of suitable credit assignment criteria to facilitate the emergence of highly cooperative memes in the memeplex for lifetime learning is discussed in Section III-B. The generation of memeplexes within modern stochastic search is then detailed in Section III-C. Section IV then presents a brief review of the capacitated vehicle routing problems (CVRPs), which is the example problem domain showcased in this paper. Subsequently, a coadapted memetic algorithm (CAMA) that takes the proposed conceptual modeling of memeplexes into action to solve CVRPs is then described. In Section V, the empirical assessment of CAMA against several existing state-of-the-art adaptive or human-designed MAs and metaheuristics, on several CVRP benchmark sets, is considered. Empirical results obtained showed that coadapted meme complexes provide a means of creating highly robust and scalable algorithms by generating improved or at least competitive search performances over existing state-of-the-art approaches. Finally, Section VI ends with a brief conclusion of the present research.

II. Memeplex in Memetic Computation

Scientists in the field of memetics have argued that memes do not work alone, but bundle themselves in so-called meme complexes or memeplexes, so as to work more effectively [41]–[43]. In science, a memeplex represents an ideology or belief system, e.g., religion, that exists in the mental state of an individual during its lifetime. Structurally, a memeplex is a coadapted stable set of mutually assisting memes [41] and can arise in the meme pool of seemingly unconnected memes [43]. Within a memeplex, memes interact to reinforce each other, forming a structure of memetic stability, with each meme being supported and promoted by the other memes in the memeplex.

In stochastic search, the concept of a memeplex creates a self-consistent paradigm whereby memes self-organize and engage in collective learning and action that will eventually give higher efficacy to lifetime learning process. In a search algorithm, memeplexes are essentially sets of instructions or memes that carry out the search behaviors or moves in the lifetime learning. Having several memes that coadapt with each other, the memeplex forges a diverse and integrated memetic ecosystem because each meme can take on a specific role such as inducing a unique learning bias and search neighborhood structure, with memes complementing one another on the search. This section presents our representation of memeplexes in search, the credit assignment criteria for coadapting memes, and the role of coadapted meme complexes in the lifetime learning process of an MA.

A. Memeplex Representation

In our study, a memeplex is perceived in the form of a neural wiring structure housed within the agent's brain that controls the agent's behavior in the lifetime, as illustrated in Fig. 1. Symbolically, a memeplex $\mathcal{M}_i$ associated with solution s_i can be built from the meme terminal set. The representation of a memeplex $\mathcal{M}$ is defined as a group of memes, i.e., $\{m_k\}^{L_{\mathcal{M}}}$.

The memes that form a memeplex $\mathcal{M}$ are taken from the meme terminal set. Each meme m_k encodes the unit of instruction to perform local refinement during the lifetime learning process. In general, a meme m_k is defined by the meme tuple $m_k = (< T_i >, <\text{Acceptance\_Strategy}>, <\text{Optional\_Depth}>)$. T_i denotes the move operator of the search neighborhood, $<$ Acceptance_Strategy $>$ denotes the acceptance strategy of the local refinement, while $<$ Optional_Depth $>$ denotes the search intensity of m_k during local refinement.

During the lifetime learning process, a memeplex $\mathcal{M}$ improves a solution s by iteratively mobilizing the memes it

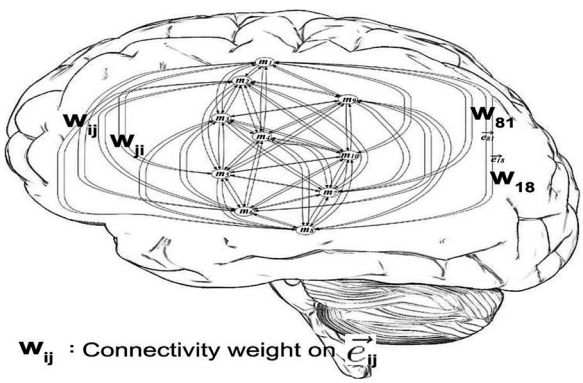

Fig. 1. Memeplex in its neural wiring structure.

encodes to conduct local refinement on s. The activation of meme $m_j \in \mathcal{M}$ at any given time instance is, thus, tightly coupled with the preceding mobilized meme $m_i \in \mathcal{M}$, which has a connectivity weight denoted by w_{ij}. In $\mathcal{M}$, each two memes $m_i, m_j \in \mathcal{M}$ are wired to each other with some connectivity weights w_{ij} and w_{ji}. The neural wiring structure of $\mathcal{M}$, which is also known as, memetic network of $\mathcal{M}$ is illustrated in Fig. 1.

The memetic network topology of $\mathcal{M}$ can, thus, be represented as a fully connected directed graph, with the set of memes as the vertex set, and a set of directed edges $\{\vec{e}_{ij}\}$ indicating the associations or synergies between memes m_j and m_i. The memetic network can be represented using a matrix $W_{\mathcal{M}}$, which is given by

$$W_{\mathcal{M}} = \begin{pmatrix} w_{11} & w_{12} & \cdots & w_{1L} \\ w_{21} & w_{22} & \cdots & w_{2L} \\ \vdots & \vdots & \ddots & \vdots \\ w_{L1} & w_{L2} & \cdots & w_{LL} \end{pmatrix} \tag{1}$$

where w_{ij} refers to the connectivity (or synergy) from meme m_i to m_j and defines the degree to which m_i's activation favors the subsequent activation of m_j. w_{ii} denotes the individual fitness level of meme m_i. The network topology of $\mathcal{M}$ as modeled by $W_{\mathcal{M}}$ is essentially a dynamic and distributed complex network woven in memetic connection patterns. To simplify the symbol w_{ii}, we denote it as w_i in the writing. Fig. 1 illustrates the network topology that shows parts of the connectivity.

B. Meme Coadaptation

The memeplex representation allows memes within the network topology to draw their functional status from their spatial arrangement, by interacting together to achieve advance lifetime learning capabilities. The nonlinear memetic interaction within forges a neural wiring pattern that produces unique combined effects during the lifetime learning process. The interaction of one meme's activation with that of its immediate surrounding meme in the memetic network triggers a complex chain of responses leading to some higher order behavior. As a result, the emergent memeplex is unlike its component memes, insofar, as it has a distinctive effect on lifetime learning that cannot be reduced to their sum or their difference [44], [45]. The number of interactions between memes in a memetic network increases combinatorially with the number of memes, thus potentially allowing for many new and subtle types of behavior to emerge.

The arising novel and coherent structures, patterns, and properties of the emergent memeplexes must be codetermined by the problem context and the lifetime learning interactions between memeplexes [46]. Further, the emergence of memeplexes presupposes that the recurrent and persistent patterns of memetic activity can arise out of a multiplicity of relatively simple memetic interactions. As a process designed to fulfill these requirements, memetic evolution provides us with a cognitive model for the growth of powerful memeplexes. During the memetic evolution, memes are filtered out of the meme pool and get connected somehow, based on the selection pressure established in the problem context and the memeplex patterns of accumulating change observed from the causal interaction between memes in the lifetime learning. In this process, the memeplex matrix becomes the vehicle for memeplex propagations, turning the memetic network into a distributed and dynamically evolving network capable of adapting to the problem context. The emergence of memeplexes is commonly identified by the resultant memetic stability uncovered for the coadapted memes [47]. This is, in turn, determined by the fitness of the elementary memes and the synergies among them in the stabilized memetic network. In the next section, these processes and properties are discussed in greater detail.

III. Role of Memeplex in Lifetime Learning for Search

Modern stochastic searches such as MAs deploy stochastic reproduction operators (e.g., recombination operators in MAs) to generate one or more solutions, which are then subject to local refinement during lifetime learning. The role of a memeplex candidate $\mathcal{M}$ in the lifetime learning is to dynamically select and activate appropriate memes within the memetic network for the refinement of the solution s produced in the stochastic search.

The activity of a memeplex candidate $\mathcal{M}$ during a lifetime learning process, also known as $\mathcal{M}$'s life cycle, is a series of memetic activations that iterate in complex nonlinear patterns, with each interaction mobilizing a meme to refine solution s. At each iteration within $\mathcal{M}$'s life cycle, a meme m_j is activated with probability $P_{\text{activate}}(m_j)$ as given by

$$P_{\text{activate}}(m_j) = \frac{w_{ij}}{\sum_{\mu=1}^{L} w_{i\mu}} \tag{2}$$

where i is the index of the meme activated in the previous iteration, w_{ii} denotes the fitness of meme m_i, and $w_{i\mu}$ denotes the memetic synergy from m_i to m_μ. When activated, m_j induces its learning bias and search neighborhood structure, such as move operator and move acceptance criteria, to refine solution s. During the lifetime learning process, search performance of the memeplex is recorded and used to adjust each individual meme's fitness as well as the memetic synergy between memes. This processed feedback is then assimilated in credit

assignment to individual meme fitness and memetic synergy via connectivity adjustment to $W_{\mathcal{M}}$, thus allowing individual meme fitness and memetic synergy to synchronize with the changing state of the search. The detailed steps to synchronize the individual meme fitness and memetic synergy, as well as the memeplex network adaptation, with the changing state of a search, are next described in what follows.

A. Credit Assignment for Individual Memes

The fitness of individual meme m_i is represented by w_i, where w_i denotes the expected performance of m_i if it is activated. As memes get mobilized for solution improvement during the life cycle of memeplex candidate $\mathcal{M}$, the lifetime learning process monitors, interprets, and assimilates the harvested performance information from each instance of meme mobilization that led to the attained performance. Each time upon activating meme $m_i \in \mathcal{M}$, an immediate reward r_i is computed from the mobilization and the fitness of m_i is subsequently updated as in [12] with

$$w_i = \gamma w_i + r_i \tag{3}$$

$$= \gamma w_i + \frac{\Delta C_i}{\Delta T_i} \tag{4}$$

where ΔC_i and ΔT_i denote the operational gain and computational cost associated with the functional activity performed by m_i, and $0 < \gamma < 1$ is the discount factor of w_i. In (3), the discount factor γ reflects the influence of a meme's recent performance.

B. Credit Assignment for Coadapted Memes

The memetic synergy is defined between any two memes m_i and m_j, and is represented by w_{ij} and w_{ji}, where w_{ij} denotes the degree to which m_i's mobilization favors m_j's activation. The modeling of memetic synergy is crucial since emergent order cannot arise if the coexisting memes of a memeplex do not interact. Memetic synergy measures the preferences of one type of interaction over another. Through memetic synergy, memes become "acquainted" with each other and socialize in the interaction space, thus clarifying the role of and lines of communications between memes and their integration in a memetic network topology that is optimally harnessed and coordinated.

To learn the memetic synergy, a memeplex candidate $\mathcal{M}$ is decoded by retracing the memetic activation profile from $\mathcal{M}$'s network topology. Subsequently, the memetic synergy between pairs of memes (m_k, m_{k+1}) is updated [48], as given by

$$w_{k(k+1)} = w_{k(k+1)} + \frac{R}{L_{\mathcal{M}}}, \quad k = 1, \ldots, L_{\mathcal{M}} - 1 \tag{5}$$

$$= w_{k(k+1)} + \frac{\Delta C_{\mathcal{M}}}{L_{\mathcal{M}} * \Delta T_{\mathcal{M}}} \tag{6}$$

where $R = \frac{\Delta C_{\mathcal{M}}}{\Delta T_{\mathcal{M}}}$ denotes the delayed reward to the memes in $\mathcal{M}$ obtained after $\mathcal{M}$ completes its local refinement on the solution. $\Delta C_{\mathcal{M}}$ and $\Delta T_{\mathcal{M}}$ are the overall operational gain and computational cost of $\mathcal{M}$, respectively. $\Delta C_{\mathcal{M}}$ is measured by the fitness improvement before and after local refinement on the solution. $\Delta T_{\mathcal{M}}$ is measured by the number of local refinement steps made by $\mathcal{M}$ to reach the improved solution. $L_{\mathcal{M}}$ is the number of memes in $\mathcal{M}$. Fig. 2 depicts the credit assignment to the individual meme fitness and memetic synergy as $\mathcal{M}$ undergoes a series of memetic activation during its life cycle.

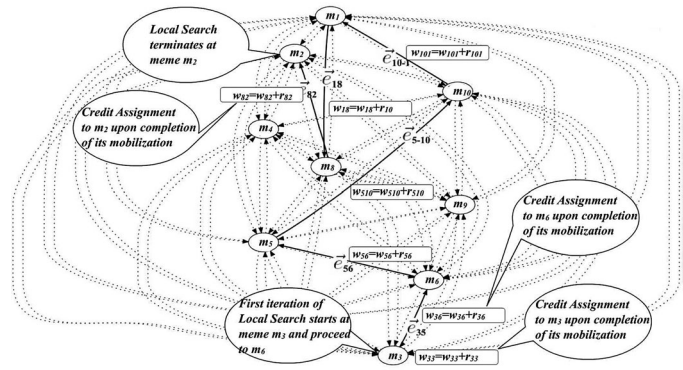

Fig. 2. Credit assignment for coadapted memes as $\mathcal{M}$ undergoes a series of memetic activation.

As illustrated in Fig. 2, the life cycle of a memeplex candidate $\mathcal{M}$ is a multiplicity of simple interactions between the connectivity adjustments for credit assignment processes described by (3) and (5), and the nonlinear interaction process of memetic activation described by (2). This memetic interaction energetics that permeate memetic network in lifetime learning for search is driven by the dynamics of the connectivity adjustments within $W_{\mathcal{M}}$ as triggered by memetic activations and propagating along the directed edges of the memetic network. The result of this process interplay is that stable patterns of memetic connections within $W_{\mathcal{M}}$ begin to emerge, allowing the gradual evolution of compatible memes that self-organize into memetic network topologies that exhibit memetic stabilities. Thus, the discovery of memetic network topologies in the form of recurrent and persistent neural wiring patterns of memeplexes gives rise to new memetic networks that are synchronized with the changing state of the search.

C. Emergence of Memeplexes

The individual meme fitness and memetic synergy denote a form of knowledge acquired during the life cycles of previous instances of memeplex candidates, which is, subsequently, employed to adaptively generate new memeplex candidate for the future state of search. When a generated memeplex candidate $\mathcal{M}$ delivers great improvement on a solution, connectivity adjustments within $W_{\mathcal{M}}$ lead to reinforcement in pattern of memetic interaction. On the contrary, when $\mathcal{M}$ does not produce desired behavior, implying that the memes that make up $\mathcal{M}$ are nonfavorable, connectivity adjustments within $W_{\mathcal{M}}$ then lead to the dissolution of $\mathcal{M}$. This process stretches across generations of memeplexes, as well as populations of stochastic solutions, and efficiently converts the revealed potential of individual memes and memetic synergies into an increasingly coherent memeplex of efficacy that is properly aligned with the structure of the solution population throughout the changing state of search.

Algorithm 1 Adaptation of Memetic Network in Memeplex Generation

```
Let A ← set of remaining unselected memes
Select seed meme m_1 ∈ A with probability P_select(m_1)
M ← Add m_1 to M
while A ≠ ∅ do
    Select meme m_k ∈ A with probability P_select(m_k)
    if accepted(m_k) then
        M ← M ⋃ m_k
        k ← k + 1
    else
        Break
    end if
end while
```

Algorithm 1 illustrates the strategy to generate a memeplex candidate $\mathcal{M}$.

The adaptive memeplex generation process is described in what follows. First, a "seed" meme m_1 is filtered out of the meme pool, with probability $P_{\text{select}}(m_1)$ as given by

$$P_{\text{select}}(m_1) = \frac{w_1}{\sum_{\mu=1}^{|\mathcal{A}|} w_\mu} \tag{7}$$

where w_μ denotes the fitness of meme μ, and $\mathcal{A}$ denotes the set of unselected memes. From (7), it is worth noting that meme with higher fitness is given higher probability of being chosen as the seed. Once selected, the seed meme m_1 will then "germinate" and grow into a memeplex candidate $\mathcal{M}$, which harnesses and integrates a cognitive territory glued together based on individual meme fitness and memetic synergy at the current search state. This cognitive territory is defined such that any subsequent meme m_k that is to be included in $\mathcal{M}$ is chosen from $\mathcal{A}$, with probability $P_{\text{select}}(m_k)$ defined by

$$P_{\text{select}}(m_k) = \frac{[w_k]^\alpha [w_{(k-1)k}]^\beta}{\sum_{\mu\in\mathcal{A}} [w_\mu]^\alpha [w_{(k-1)\mu}]^\beta} \tag{8}$$

where w_μ denotes the fitness of meme μ, and $w_{(k-1)\mu}$ denotes the memetic connectivity from m_{k-1} to m_μ, i.e., how much mobilization of m_{k-1} favors the activation of m_μ. α and β are the parameters that control the relative weight of individual meme fitness and memetic synergy, respectively. Equation (8) indicates that a meme is selected to form the memeplex based not only on their individual performance, but also on how well they synergize with others. As a result, (8) prescribes that any subsequent candidate meme m_k for inclusion must have high fitness, as well as strong synergy with the mobilized meme m_{k-1}.

To balance between population-based search and lifetime learning in MA, a cognitive boundary $L_{\mathcal{M}}$ is enforced to limit the size of elementary memes that compose the memeplex candidate. This boundary is controlled through a simulated-annealing (SA)-like rejection mechanism. For the candidate meme m_k obtained from (8), the candidate meme will be added to $\mathcal{M}$ with probability $P_{\text{accept}}(m_k)$ defined as

$$P_{\text{accept}}(m_k) = \begin{cases} 1, & w_k > w_{k-1} \\ exp(-\frac{(w_{k-1}-w_k)}{T}), & \text{otherwise} \end{cases} \tag{9}$$

where $T = \frac{1}{k-1}$. By (9), m_k will be added to $\mathcal{M}$ if it has a higher fitness than m_{k-1}. Otherwise, a probability based on the fitness difference $(w_{k-1} - w_k)$ and T is considered for the inclusion of m_k into $\mathcal{M}$. When the size of candidate memeplex is small, T will be sufficiently large, thus allowing diverse memes, i.e., even memes with lower fitness to possess some chance of participating in the memeplex. In this case, the lower meme fitness is compensated by its strong synergy exhibited in the previous step of (8). As the set of elementary memes increases, T decreases, imposing the constraint of mainly highly qualified candidate memes to be added into the memeplex. When a selected candidate meme m_k fails to be included in $\mathcal{M}$, the scheme terminates and the current $\mathcal{M}$ is returned as the memeplex instance. Equation (9) enforces that if a meme in $\mathcal{A}$ is to be added to $\mathcal{M}$, its fitness must be justified for the additional computational cycles spent for their mobilization during the life cycle of $\mathcal{M}$.

In the next section, we present a study on the proposed concept of memeplex for effective problem solving. In particular, the role of emergent memeplexes in the lifetime learning process of an MA, which is labeled here as the CAMA, is investigated to solve the class of CVRP.

IV. Coadapted Memetic Algorithm for Solving Capacitated Vehicle Routing Problems

The vehicle routing problem (VRP) represents the cornerstone of optimization distribution networks. In VRP, a set of N customers $\{e_1, e_2, \ldots, e_l, \ldots, e_N\}$ is to be serviced by a fleet of K vehicles $\{v_1, v_2, \ldots, v_i, \ldots, v_K\}$. Each customer has demand(e_l) that must be serviced by vehicle v_i. Each vehicle v_i has a finite capacity denoted as C_v and travels a route $\tau_i = \{a_{i1}, a_{i2}, \ldots, a_{ij}, \ldots, a_{im}\}$, passing through the central depot e_0, with a_{ij} representing the jth customer visited by v_i. In addition, each vehicle may travel a maximum allowable distance of $L_{\max}$. The objective of the VRP is to find the optimal routing solution $s = \{\tau_1, \tau_2, \ldots, \tau_i, \ldots, \tau_K\}$, which minimizes the overall distance traveled by all vehicles, under the strict constraints that each customer is serviced exactly once by a single vehicle, and the total demands of all customers serviced by any vehicle v_i do not exceed vehicle capacity C_v. VRP is considered as one of the most difficult problems due to its complex combinatorial nature; it is the fusion of two NP-hard problems, namely the traveling salesman problem (TSP) and the bin packing problem. For VRP instances with few nodes, the branch and bound method is deemed to be effective and known to provide the best solutions to date [49], [50]. However, exact methods, such as branch and bound, are not viable for large-scale VRP [51]. As a result, most researchers have turned to metaheuristics to solve real-world VRPs. The drawback is that generic metaheuristics do not guarantee convergence to global optimum [52], [53]. Genetic algorithms [54]–[57], SA [58]–[60], and tabu search (TS) [58], [61] represent some of the popular metaheuristics algorithms that have been developed to handle real-life VRPs,

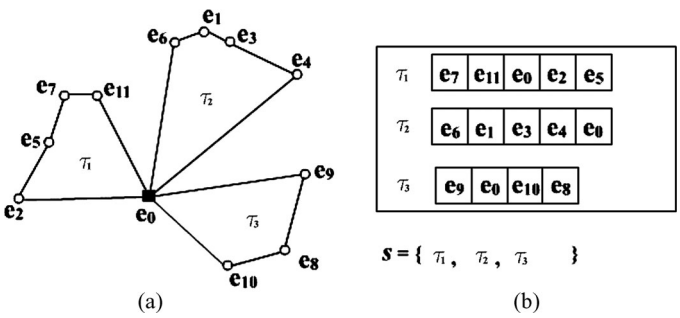

Fig. 3. Multiroute VRP solution representation.

with reasonably high degree of success. For MAs on VRP, Pereira *et al*. [62] proposed a genetic vehicle representation (GVR) with each individual solution composing of dynamic, variable-length integer vectors, where each vector defines the route of a vehicle. Prins [63] proposed an MA that uses specialized splitting procedure to transform TSP-like chromosome into respective VRP solution representation. Kubiak [64] defined some distance metrics for VRP and introduced a number of distance-preserving recombination operators based on fitness landscape analysis, while Berger and Barkaoui [65] introduced a coevolutionary MA to minimize the total traveled distance.

A. Problem Formulation

Here, we begin with a description of the multiroute VRP solution representation [62] and problem formulation. The genetic population $P(t)$ at generation t consists of a set of M candidate solutions

$$P(t) = \{s_1, s_2, \ldots, s_i, \ldots, s_M\}. \tag{10}$$

Solution $s \in P(t)$ defines a set of K routes

$$s = \{\tau_1, \tau_2, \ldots, \tau_i, \ldots, \tau_K\} \tag{11}$$

where τ_i denotes the route visited by vehicle v_i, and K is the number of vehicles in s.

Route τ_i defines an ordered set of customers

$$\tau_i = \{a_{i1}, a_{i2}, \ldots, a_{ij}, \ldots, a_{im}\} \tag{12}$$

where a_{ij} of route τ_i denotes the jth customer visited, and m is the number of customers visited by vehicle v_i.

The inclusion of depot e_0 is enforced in each route τ_i (i.e., $e_0 = a_{ij}, \exists a_{ij} \in \tau_i$). Hence, every route τ_i forms a closed TSP circuit. Fig. 3(a) illustrates an example of the VRP, and the respective multiroute solution representation is given in Fig. 3(b). Route $\tau_1 = \{e_7, e_{11}, e_0, e_2, e_5\}$ in Fig. 3(b) denotes the *closed circuit* $e_0 \rightarrow e_2 \rightarrow e_5 \rightarrow e_7 \rightarrow e_{11} \rightarrow e_0$ visitation route of vehicle v_1.

The distance traveled by vehicle v_i is denoted here as $L(\tau_i)$ and is given by

$$L(\tau_i) = \sum_{j=1}^{m-1} d(a_{ij}, a_{i(j+1)}) + d(a_{i1}, a_{im}) \tag{13}$$

where $d(a_{ij}, a_{i(j+1)})$ represents the symmetric distance between two consecutive customers a_{ij} and $a_{i(j+1)}$.

On the other hand, the total demand serviced by vehicle v_i is denoted as $D(\tau_i)$ and is given by

$$D(\tau_i) = \sum_{j=1}^{m} \text{demand}(a_{ij}) \tag{14}$$

where demand(a_{ij}) corresponds to the demand of customer a_{ij}.

The objective of the VRP is to minimize the overall distance $C(s)$ traveled by all K vehicles and is defined as

$$C(s) = \sum_{i=1}^{K} L(\tau_i) \tag{15}$$

$$= \sum_{i=1}^{K} [\sum_{j=1}^{m-1} d(a_{ij}, a_{i(j+1)}) + d(a_{i1}, a_{im})] \tag{16}$$

subjected to constraints

$$L(\tau_i) \leq L_{\max} \forall \tau_i \in s$$
$$D(\tau_i) \leq C_v \forall \tau_i \in s \tag{17}$$

where $L_{\max}$ is the maximum allowable distance traveled by each vehicle, and C_v is the capacity of each vehicle.

B. Genetic Reproduction Operators for Vehicle Routing Problem

In this section, we briefly discuss the evolutionary operators as described in [66] to solve the VRPs. In crossover, a child solution s_c is generated from partial routes of two selected parent solutions $s_i, s_j \in P(t)$. In particular, taking the depot node e_0 as the axis center, a bisecting axis ψ of random angle θ is generated to slice the VRP graph into two separate subgraphs containing nonempty customers, i.e., H_+ and H_-, and the child solution s_c is a fusion of partial routes H_+ in s_i and H_- in s_j.

The mutation operator is performed by one of the four mutation heuristics, namely, *swap*, *inversion*, *insertion*, and *displacement*, through random selection, which is purely stochastic in nature. Hence, each heuristic shares equal probability of being chosen for the mutation of s_c [66].

Since the evolutionary operators discussed thus far may lead to infeasible solutions, the *SplitRepair* procedure as introduced in [66] is used to revive any infeasible solution generated.

C. Memeplex Representation for Vehicle Routing Problem

In the problems of vehicle routing, we define the representation of a memeplex $\mathcal{M}$ as a group $\{m_k\}^{L_{\mathcal{M}}}$ of memes. These memes that form $\mathcal{M}$ are taken from the meme terminal, and each meme m_k is defined by the meme tuple $m_k = (< T_i >, <\text{Acceptance\_Strategy}>, <\text{Optional\_Depth}>)$. T_i denotes the move operator of the search neighborhood, and $<\text{Acceptance\_Strategy}>$ denotes the acceptance strategy of the local refinement. The $<\text{Optional\_Depth}>$ denotes the search intensity of m_k during local refinement. Each two memes $m_i, m_j \in \mathcal{M}$ are wired to each other via memetic connectivities w_{ij}, w_{ji}.

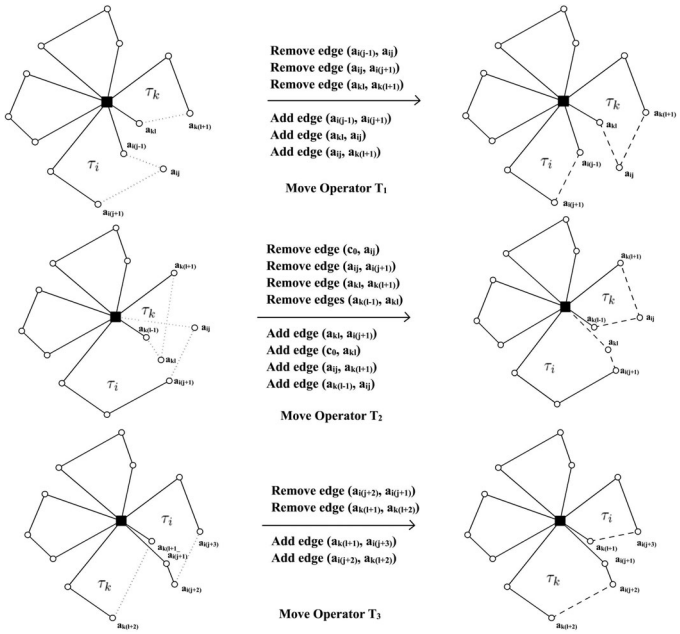

Fig. 4. Move Operators: T_1, T_2, and T_3.

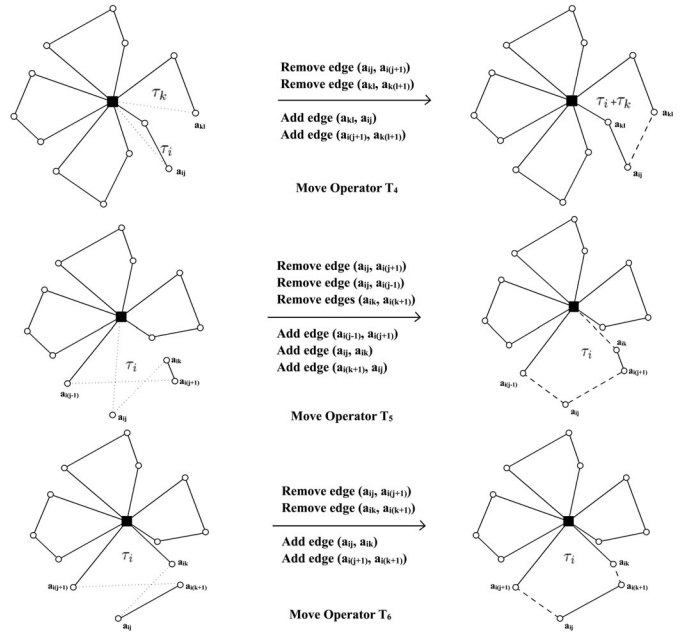

Fig. 5. Move Operators: T_4, T_5, and T_6.

For the VRPs, we define six move operators $\{T_1, T_2, T_3, T_4, T_5, \text{and}\, T_6\}$, respectively, as depicted in Figs. 4 and 5. Move operators T_1, T_2, and T_3 are clustering procedures to reassign customers between routes so as to reduce the overall traveled distance. T_4 denotes a move operator that reduces the total number of routes in the current solution by route merging. Move operators T_5 and T_6, on the other hand, serve to reduce the length of a route in the current solution by reassigning the visiting order of customers in the route. In what follows, each of the move operators is briefly detailed.

T_1: Customer τ_{kj} of route $\tau_k \in s$ is removed and reinserted into a separate route $\tau_{k'} \in s, \tau_{k'} \neq \tau_k$, if fitness improvement $\Delta C(s) > 0$.

T_2: Customer τ_{kj} of route $\tau_k \in s$ is swapped with customer $\tau_{k'l}$ of route $\tau_{k'} \in s, \tau_{k'} \neq \tau_k$, if fitness improvement $\Delta C(s) > 0$.

T_3: A subroute $\hat{\tau_k} = \{\tau_{k(j+1)}, \tau_{k(j+2)}, \ldots, \tau_{k(j+\mu)}\}$ of route $\tau_k \in s$ is swapped with subroute $\hat{\tau_{k'}} = \{\tau_{k'(l+1)}, \tau_{k'(l+2)}, \ldots, \tau_{k'(l+\nu)}\}$ of route $\tau_{k'} \in s, \hat{\tau_k} \neq \hat{\tau_{k'}}$, if fitness improvement $\Delta C(s) > 0$.

T_4: Two shorter routes $\tau_k, \tau_{k'}$ in the current solution are merged to reduce the number of routes and, thus, the overall distance traveled, if fitness improvement $\Delta C(s) > 0$.

T_5: A customer $\tau_{kj} \in \tau_k$ is reinserted after a customer $\tau_{kl} \in \tau_k, \tau_{kj} \neq \tau_{kl}$, if fitness improvement $\Delta C(s) > 0$.

T_6: A customer $\tau_{kj} \in \tau_k$ is joined with a customer $\tau_{kl} \in \tau_k, \tau_{kl} \neq \tau_{kj}$ by reversing the visiting order of customers between τ_{kj} and τ_{kl}, if fitness improvement $\Delta C(s) > 0$.

A total of 24 different memes can be assembled to form the meme terminal set from the combinatorial choices made among the six move operators, two acceptance strategies (*first improvement* or *best improvement*), and another two meme search intensity. In the next section, we present a CAMA that takes the conceptual modeling of memeplex proposed in Section III to solve CVRPs.

Algorithm 2 Outline of CAMA Algorithm

BEGIN

Create an initial population of solutions $P(t=0) = \{s_1, s_2, \ldots, s_i, \ldots, s_M\}$

Create an initial population of memeplexes $\{\mathcal{M}_1, \mathcal{M}_2, \ldots, \mathcal{M}_i, \ldots, \mathcal{M}_M\}$

While(stopping conditions are not satisfied)

For each individual **Do**

Perform genetic crossover and mutation to generate offspring solution s_c

$\mathcal{M} \leftarrow$ **Adaptive_Memeplex_Generation**

$s_c \leftarrow$ **LifetimeLearning** $(s_c, \mathcal{M})$

Credit_Assignment_for_Co-adapted_Memes

Add s_c to offspring population $P'(t)$

End For

$P(t+1) \leftarrow$ Merge$(P(t), P'(t))$

$t \leftarrow t+1$

End While

END

D. Coadapted Memetic Algorithm

The basic steps of the CAMA are outlined in Algorithm 2. The algorithm starts with the initialization of a population $P(t=0)$ of candidate solutions. At each generation, a population of

offspring is generated using evolutionary search (i.e., crossover or mutation) and, subsequently, undergoes lifetime learning defined by memeplex candidate $\mathcal{M}$ generated according to the *Adaptive_Memeplex_Generation* scheme. The lifetime learning process coupled with the *Credit_Assignment_for_Coadapted_Memes* scheme, which performs the delayed rewarding as specified in (5), synchronizes the evolution of memeplexes according to the state of search. This workflow of the memetic search proceeds until some stopping conditions are reached, for example, the predefined maximum number of fitness function evaluation has elapsed.

V. Empirical Study

In this section, an empirical study of CAMA to solve the CVRP is presented. Here, a diverse set of CVRP benchmark classes that differ in size and other characteristics [67], including customer distribution, symmetric structure, route-length restriction, and vehicle capacity, is considered. The inclusion of a multitude of VRP benchmarks with wide-ranging characteristics makes it possible to analyze the proposed CAMA. Here, we briefly describe the benchmark sets employed in our empirical study.

1) Augerat: This set of benchmarks was proposed by Augerat *et al*. [68]. We considered eight well-known Augerat benchmarks including five random instances from Augerat set A with node size ranging from 32 to 80 and three clustered instances from Augerat set B with node size ranging from 57 to 78.
2) CE: This set of benchmarks was proposed by Christofides and Eilon [69]. The four benchmark instances are modified versions of a 76-customer instance that differs in vehicle capacity.
3) Christofides: This set of benchmarks was proposed by Christofides *et al*. [58]. The node size of the problem instances ranges from 50 to 199 customers. In addition, seven instances enforce maximum route-length constraint.
4) Golden: This set of benchmarks was proposed by Golden *et al*. [70]. The set contains 20 large-scale VRP instances with customer size ranging from 200 to 483. The large instances are designed to contain highly symmetrical structures. In instances 1–8, maximum route-length constraint is applied, and the nodes are located in concentric circles around the depot.

All the results presented in this section are summaries of 30 independent runs for each VRP instance under a consistent experimental setup. Each run continues until a maximum of 300 000 fitness evaluations is reached or the solution converged to the best known result in the literature. In each run, the evolutionary search configurations of the MAs used to solve the benchmark problems are defined as follows: population size of 30, crossover probability of $p_c = 0.5$, mutation rate of $p_m = 0.5$, roulette-wheel selection scheme, and an elitism size of 2. Student's t-test with 5% p-threshold significance level was conducted to assess the statistical significance of the converged solution quality difference between two algorithms. Search results are then reported with respect to the following metrics:

- C denotes the best fitness among the converged solutions on a VRP instance;
- $\overline{C}$ denotes the average fitness of the converged solutions on a VRP instance;
- C_b denotes the best known result of a VRP instance reported in the literature;
- gap denotes the minimal percentage fitness deviation from C_b as given by $\frac{C-C_b}{C_b} \times 100\%$;
- $\overline{gap}$ denotes the average percentage fitness deviation from C_b as given by $\frac{\overline{C}-C_b}{C_b} \times 100\%$;
- p_t p-value denotes the *two-tailed* t-test difference between CAMA and the algorithm (5% p-threshold);
- p_r p-value denotes the *right-tailed* t-test difference between CAMA and the algorithm (5% p-threshold).

A. Search Quality and Effectiveness of Coadapted Memetic Algorithm

To highlight the search efficiency and quality of CAMA, the results obtained are first pitted against two state-of-the-art adaptive MAs, namely, the self-generating memetic algorithm [71] (*SGMA*) and multimeme memetic algorithm [35] (*MultiMeme*) introduced in the literature, which are re-implemented here in this study for possible assessment and comparison study.

The relative performances of the algorithms are depicted in Figs. 6–8, with the y-axis of subfigures (a) and (b) denoting gap and $\overline{gap}$, respectively, and a common x-axis denoting the VRP instances of interest (detailed tabulated values of performance metrics can be found online [72]). It can be observed that on most problems, both MultiMeme and SGMA exhibits much larger gaps compared with CAMA, i.e., both the average and minimum gaps, indicating that CAMA is superior to MultiMeme MA and SGMA in terms of convergence to solutions that has fitness qualities that are nearer to both the best known fitness and average (expected) fitness of each respective VRP instances. To further validate the observed outcome, both the two-tailed and one-tailed t-tests are conducted and the detailed statistical results are presented in Table I. The overall t-test results obtained confirmed that CAMA produces higher quality solutions than MultiMeme (exhibiting superiority on $2/3$ of the problem instances considered) and SGMA (superiority on half of the problem instances considered), outperforming these two algorithms *at* 95% *confidence level of statistical significance*.

Next, we proceed to analyze the success rate of CAMA, MultiMeme, and SGMA on each of the problem instance tabulated in Table II. Here, success rate is measured as the percentage of the 30 independent runs that an algorithm converged to solution(s) at a precision level of 5% fitness gap of the best known result. Note that a ✓ symbol in Table II denotes a 100% success rate on a particular problem instance. It is observed that on the Augerat, CE, and Christofides benchmarks, all three algorithms were able to achieve 100% success rate. On the Golden benchmark, which composes of large-scale VRP instances, however, the success rate is noted to drop significantly for MultiMeme and SGMA, while CAMA is able to maintain a 100% success

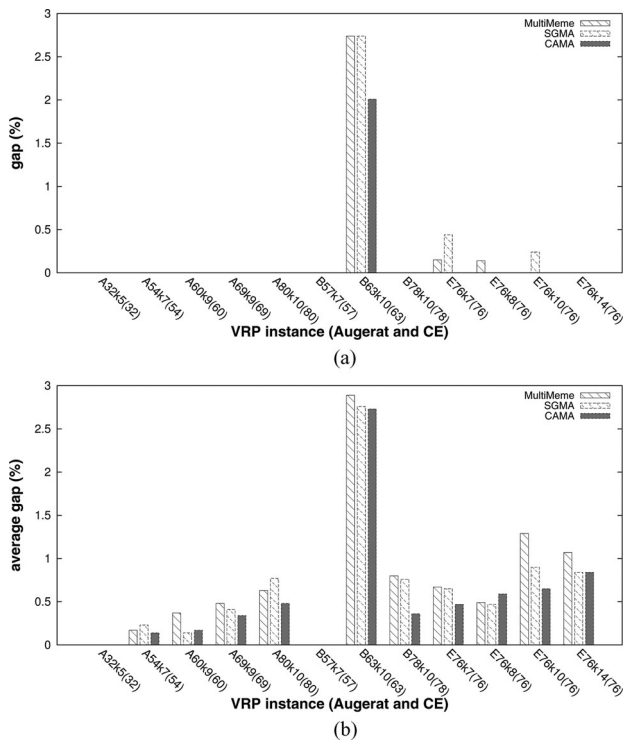

Fig. 6. Minimum and average percentage fitness gaps, $gap, \overline{gap}$, across Augerat & CE benchmarks (empty bars implies zero gap readings). (a) gap (the fitness gap in percentage between best fitness C and the best known fitness C_b). (b) $\overline{gap}$ (the fitness gap in percentage between average fitness $\overline{C}$ and the best known fitness C_b).

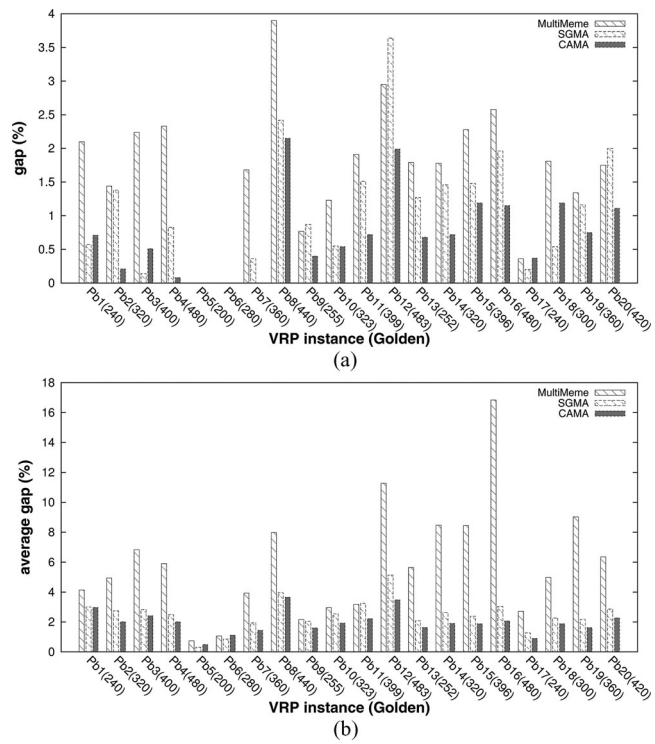

Fig. 8. Minimum and average percentage fitness gaps, $gap, \overline{gap}$, across Golden benchmarks (empty bars implies zero gap readings). (a) gap (the fitness gap in percentage between best fitness C and the best known fitness C_b). (b) $\overline{gap}$ (the fitness gap in percentage between average fitness $\overline{C}$ and the best known fitness C_b).

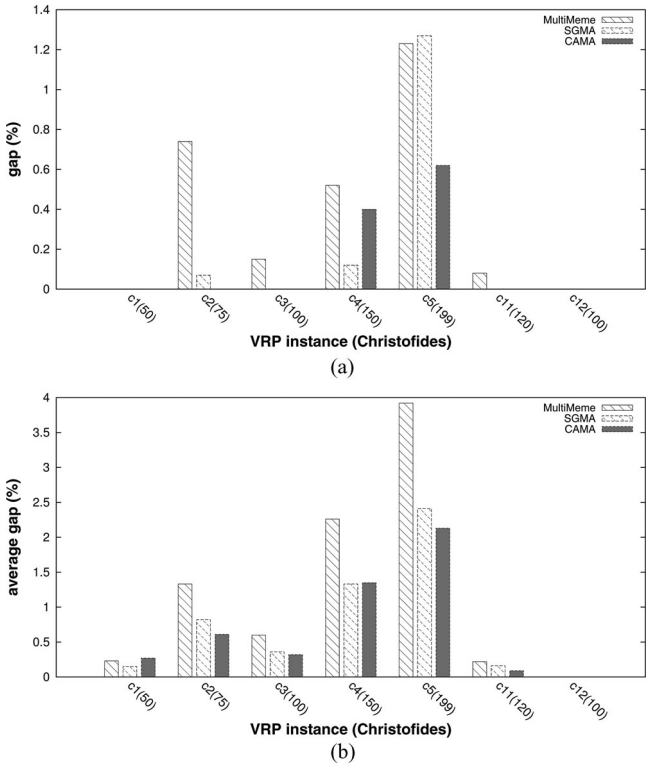

Fig. 7. Minimum and average percentage fitness gaps, $gap, \overline{gap}$, across Christofides benchmarks (empty bars implies zero gap readings). (a) gap (the fitness gap in percentage between best fitness C and the best known fitness C_b). (b) $\overline{gap}$ (the fitness gap in percentage between average fitness $\overline{C}$ and the best known fitness C_b).

rate on all problems (with the exception of instance Pb18(300) only), highlighting that CAMA was able to perform much more robustly.

B. Computational Cost and Efficiency

In this section, we analyze the computational cost and efficiency of the proposed CAMA to solve CVRP with respect to adaptive MAs and canonical MA serving as baselines for comparison. The computational cost is measured by the average wall clock time required for an algorithm in converging to solution(s) at a precision level of 5% fitness gap of the best known result for an instance.

To understand the impact of the proposed memeplex adaptation on the optimization process, we analyze the wall clock time for CAMA to converge to fitness within 5% gap, in comparison with adaptive MAs including SGMA and MultiMeme. For most of the problem instances summarized in Figs. 9 and 10, CAMA takes approximately 0.1 s to converge, except on two large-scale problems (i.e., c4 and c5) where a few seconds is incurred. Further, for the sake of conciseness, the detailed statistical results are provided in [72]. The results in Figs. 9 and 10 also showed that CAMA consistently outperforms SGMA and MultiMeme in terms of the efficiency measure, indicating that SGMA and MultiMeme require more time budgets to converge to solutions of the same quality than CAMA on the representative VRP benchmark instances of diverse classes. This trend

TABLE I
COMPARISON AMONG CAMA, SGMA, AND MULTIMEME (t-TEST), $^{(+)}$ INDICATES CAMA PERFORMS BETTER STATISTICALLY, IN TERMS OF p_t AND p_r

$i(n)$	$algorithm$	t-test on C between $algorithm$ and CAMA	
		p_t	p_r
A32k5(32)	SGMA	1.000	1.000
	MultiMeme	1.000	1.000
A54k7(54)	SGMA	0.195	0.097
	MultiMeme	0.726	0.363
A60k9(60)	SGMA	0.699	0.651
	MultiMeme	$0.026^{(+)}$	$0.013^{(+)}$
A69k9(69)	SGMA	0.242	0.121
	MultiMeme	0.062	0.031
A80k10(80)	SGMA	$0.038^{(+)}$	$0.019^{(+)}$
	MultiMeme	0.472	0.236
B57k7(57)	SGMA	1.000	1.000
	MultiMeme	1.000	1.000
B63k10(63)	SGMA	0.437	0.219
	MultiMeme	0.074	0.037
B78k10(78)	SGMA	$0.009^{(+)}$	$0.004^{(+)}$
	MultiMeme	0.065	0.033
E76k7(76)	SGMA	$0.050^{(+)}$	$0.025^{(+)}$
	MultiMeme	0.108	0.054
E76k8(76)	SGMA	0.311	0.845
	MultiMeme	0.492	0.754
E76k10(76)	SGMA	$0.039^{(+)}$	$0.019^{(+)}$
	MultiMeme	$0.040^{(+)}$	$0.020^{(+)}$
E76k14(76)	SGMA	1.000	0.500
	MultiMeme	0.279	0.140
c1(50)	SGMA	0.382	0.809
	MultiMeme	0.843	0.578
c2(75)	SGMA	0.077	0.039
	MultiMeme	$0.001^{(+)}$	$0.000^{(+)}$
c3(100)	SGMA	0.683	0.342
	MultiMeme	$0.033^{(+)}$	$0.016^{(+)}$
c4(150)	SGMA	0.880	0.560
	MultiMeme	$0.008^{(+)}$	$0.004^{(+)}$
c5(199)	SGMA	0.088	0.044
	MultiMeme	$0.015^{(+)}$	$0.007^{(+)}$
c11(120)	SGMA	0.139	0.069
	MultiMeme	$0.010^{(+)}$	$0.005^{(+)}$
c12(100)	SGMA	1.000	0.500
	MultiMeme	$0.001^{(+)}$	$0.000^{(+)}$
Pb1(240)	SGMA	0.870	0.435
	MultiMeme	$0.018^{(+)}$	$0.009^{(+)}$
Pb2(320)	SGMA	$0.003^{(+)}$	$0.001^{(+)}$
	MultiMeme	$0.003^{(+)}$	$0.001^{(+)}$
Pb3(400)	SGMA	0.110	0.055
	MultiMeme	$0.004^{(+)}$	$0.002^{(+)}$
Pb4(480)	SGMA	$0.027^{(+)}$	$0.014^{(+)}$
	MultiMeme	$0.001^{(+)}$	$0.000^{(+)}$
Pb5(200)	SGMA	0.499	0.751
	MultiMeme	0.566	0.283
Pb6(280)	SGMA	0.188	0.906
	MultiMeme	0.819	0.590
Pb7(360)	SGMA	$0.044^{(+)}$	$0.022^{(+)}$
	MultiMeme	$0.001^{(+)}$	$0.000^{(+)}$
Pb8(440)	SGMA	0.188	0.094
	MultiMeme	$0.006^{(+)}$	$0.003^{(+)}$
Pb9(255)	SGMA	$0.020^{(+)}$	$0.010^{(+)}$
	MultiMeme	0.072	0.036
Pb10(323)	SGMA	$0.008^{(+)}$	$0.004^{(+)}$
	MultiMeme	0.080	0.040
Pb11(399)	SGMA	$0.000^{(+)}$	$0.000^{(+)}$
	MultiMeme	$0.006^{(+)}$	$0.003^{(+)}$
Pb12(483)	SGMA	$0.000^{(+)}$	$0.000^{(+)}$
	MultiMeme	$0.047^{(+)}$	$0.023^{(+)}$
Pb13(252)	SGMA	$0.001^{(+)}$	$0.001^{(+)}$
	MultiMeme	$0.001^{(+)}$	$0.001^{(+)}$
Pb14(320)	SGMA	$0.000^{(+)}$	$0.000^{(+)}$
	MultiMeme	$0.000^{(+)}$	$0.000^{(+)}$
Pb15(396)	SGMA	$0.001^{(+)}$	$0.000^{(+)}$
	MultiMeme	$0.004^{(+)}$	$0.002^{(+)}$
Pb16(480)	SGMA	$0.000^{(+)}$	$0.000^{(+)}$
	MultiMeme	$0.000^{(+)}$	$0.000^{(+)}$
Pb17(240)	SGMA	$0.010^{(+)}$	$0.005^{(+)}$
	MultiMeme	$0.007^{(+)}$	$0.003^{(+)}$
Pb18(300)	SGMA	$0.004^{(+)}$	$0.002^{(+)}$
	MultiMeme	$0.009^{(+)}$	$0.004^{(+)}$
Pb19(360)	SGMA	$0.001^{(+)}$	$0.001^{(+)}$
	MultiMeme	$0.001^{(+)}$	$0.001^{(+)}$
Pb20(420)	SGMA	$0.000^{(+)}$	$0.000^{(+)}$
	MultiMeme	0.067	0.033

TABLE II
SUCCESS RATE OF CAMA, SGMA, AND MULTIMEME ($\checkmark \equiv 100\%$)

Benchmark	$i(n)$	Success Rate		
		CAMA	SGMA	MultiMeme
Augerat	A32k5(32)	✓	✓	✓
	A54k7(54)	✓	✓	✓
	A60k9(60)	✓	✓	✓
	A69k9(69)	✓	✓	✓
	A80k10(80)	✓	✓	✓
	B57k7(57)	✓	✓	✓
	B63k10(63)	✓	✓	✓
	B78k10(78)	✓	✓	✓
CE	E76k7(76)	✓	✓	✓
	E76k8(76)	✓	✓	✓
	E76k10(76)	✓	✓	✓
	E76k14(76)	✓	✓	✓
Christofides	c1(50)	✓	✓	✓
	c2(75)	✓	✓	✓
	c3(100)	✓	✓	✓
	c4(150)	✓	✓	✓
	c5(199)	✓	✓	66%
	c11(120)	✓	✓	✓
	c12(100)	✓	✓	✓
Golden	Pb1(240)	✓	✓	75%
	Pb2(320)	✓	✓	67%
	Pb3(400)	✓	✓	50%
	Pb4(480)	✓	✓	57%
	Pb5(200)	✓	✓	✓
	Pb6(280)	✓	✓	✓
	Pb7(360)	✓	✓	77%
	Pb8(440)	✓	97%	33%
	Pb9(255)	✓	✓	✓
	Pb10(323)	✓	✓	83%
	Pb11(399)	✓	97%	✓
	Pb12(483)	✓	77%	63%
	Pb13(252)	✓	✓	37%
	Pb14(320)	✓	✓	27%
	Pb15(396)	✓	✓	40%
	Pb16(480)	✓	✓	90%
	Pb17(240)	✓	✓	90%
	Pb18(300)	87%	53%	33%
	Pb19(360)	✓	83%	27%
	Pb20(420)	✓	✓	77%

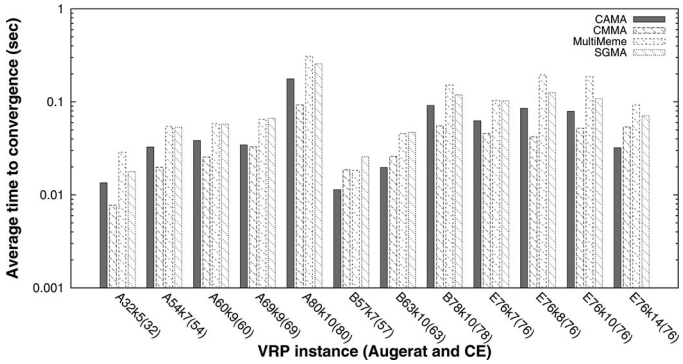

Fig. 9. Plot of average time required for convergence across Augerat and CE benchmarks.

is even more prominent on the large-scale problems as summarized in Fig. 11, where CAMA is noted to be more efficient than SGMA and MultiMeme by a factor of 2 and up to factor 10.

To gauge the computational cost imposed by the proposed memeplex adaptation in stochastic search, we assess the additional computational cost of CAMA by using the CMMA, which is a human-expert canonical MA designed to exploit the prior knowledge about the diverse memes in the lifetime learning process [66]. In Figs. 9 and 10, which summarized the results for small-to-medium-scale CVRPs, the computational cost of CAMA closely matches that of the CMMA, with only slight differences in the scale of 0.01 s. On the large-scale problems depicted in Fig. 11, the computational cost of CAMA also closely matches that of CMMA, trailing behind by 1–10 s.

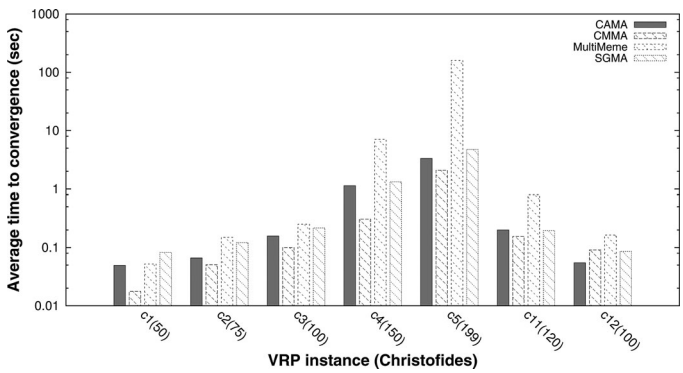

Fig. 10. Plot of average time required for convergence across Christofides benchmarks.

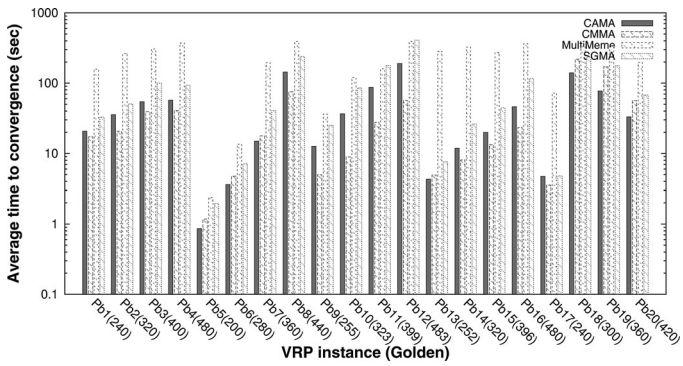

Fig. 11. Plot of average time required for convergence across Golden benchmarks.

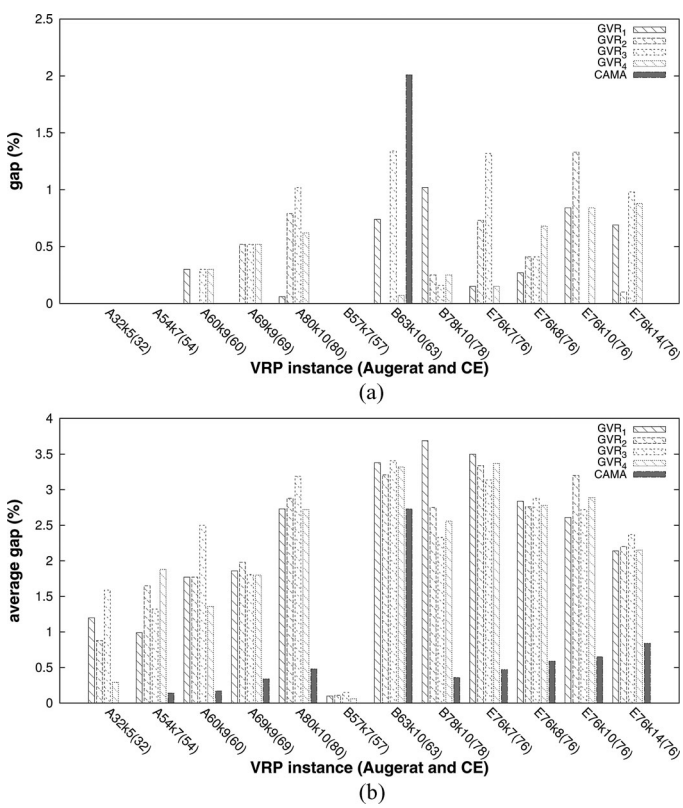

Fig. 12. Minimum and Average percentage fitness gaps, $gap, \overline{gap}$, across Augerat and CE benchmarks. (a) Minimum fitness gap. (b) Average fitness gap.

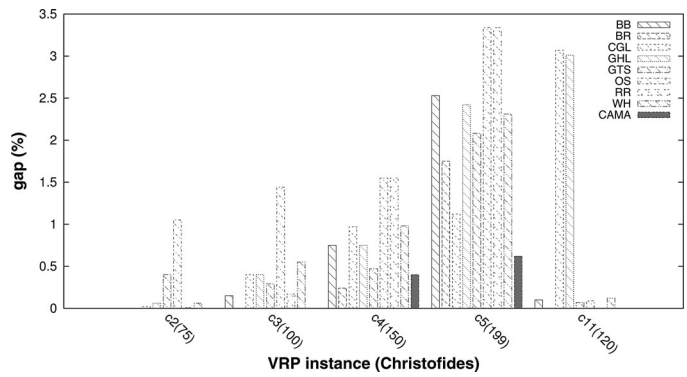

Fig. 13. Minimum percentage fitness gaps, gap, across Christofides benchmarks (c1 and c12 omitted due to their insignificant performance).

Notably, CAMA appears to take less time to converge, as observed on several large-scale problems (e.g., Pb18 with 300 customers, Pb19 with 360 customers, and Pb20 with 420 customers).

C. Coadapted Memetic Algorithm Against Other Recent State-of-the-Art Metaheuristics Algorithms in the Literature

In this section, further comparative studies of the CAMA to recent advanced metaheuristics are provided. In particular, we pit CAMA against six categories of metaheuristics search algorithms that are previously developed to solve VRPs, which range from 1) adaptive MAs, 2) VRP domain-specific MA, 3) SA, 4) TS, and 5) Genetic algorithms to 6) VRP domain-specific local search as candidate stochastic search, on their respective set of benchmarks considered. In particular, CAMA is pitted against *GVR1*, *GVR2*, *GVR3*, and *GVR4* on the Augerat and CE benchmarks, *BB*, *BR*, *CGL*, *GHL*, *GTS*, *OS*, *RR*, and *WH* on the Christofides benchmarks, and *BR*, *GTS*, *RTR*, *SEEA*, and *XK* on the Golden benchmarks. Overall, a total of 15 stochastic search algorithms are used as baselines for comparison in this study.

- *SEEA* MA proposed in [63] that uses specialized splitting procedure in place of trip delimiters.
- *BB* *Coevolved hybrid genetic algorithm* proposed in [65].
- *OS* Metastrategy SA algorithm proposed in [59].
- *GHL* *TabuRoute* proposed in [61].
- *CGL* *TS* proposed in [73].
- *GTS* *Granular TS* proposed in [74].
- *RR* *Parallel TS algorithm* using *ejection chain* proposed in [75].
- *BR* *Boneroute* MA proposed in [76].
- *XK* *Xu and Kelly TS* proposed in [77].
- GVR_x Genetic algorithm with *GVR* proposed in [62] (x=1,2,3,4).
- *RTR* *Record-to-record algorithm* proposed in [70].
- *WH* *Repeated matching heuristic* proposed in [78].

The results reported and published by the respective algorithms considered are summarized in Figs. 12–14 (detailed tabulated values of performance metrics can be found online [72]). Fig. 12 depicts the plots of gap and $\overline{gap}$, while in Figs. 13 and 14, the minimum fitness gap is reported.

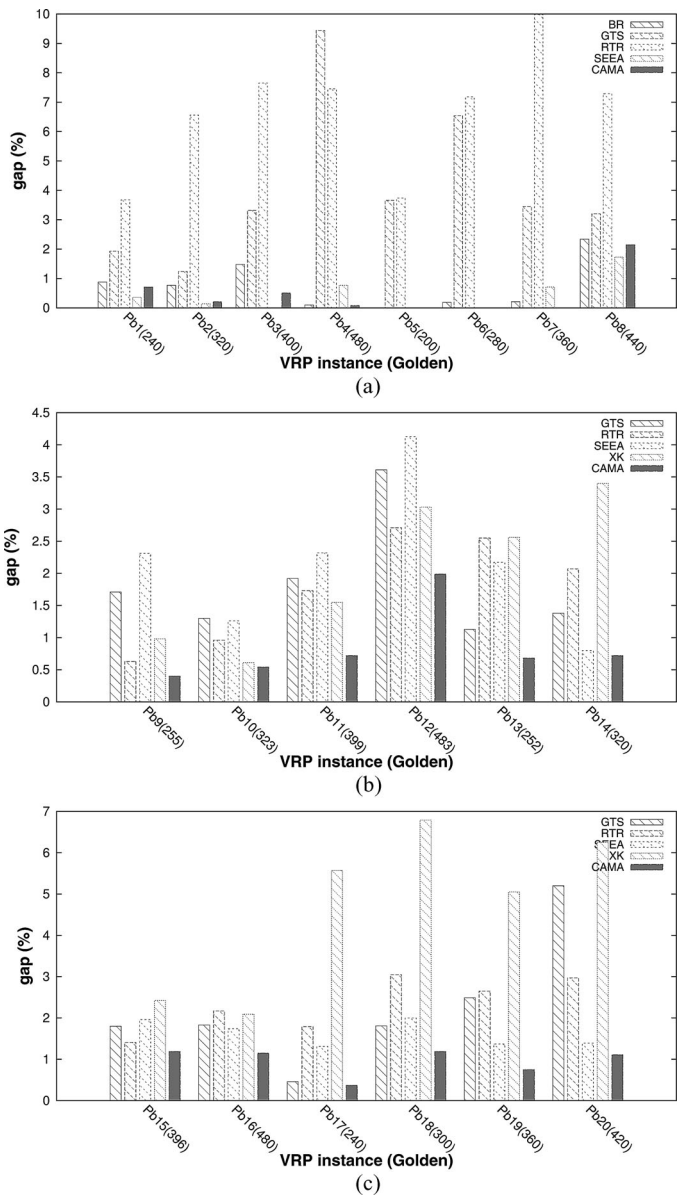

Fig. 14. Minimum percentage fitness gaps, gap, across Golden benchmarks. (a) Problem instances 1–8. (b) Problem instances 9–14. (c) Problem Instances 15–20.

It is observed that CAMA achieved performances that are superior to most of the existing stochastic search algorithms considered, in terms of both the best and average results, indicating that CAMA is able to produce both high quality and robust performance. It is notable that the reason behind the differing VRP benchmarks and performance metrics used when comparing CAMA with the respective advanced metaheuristics considered is a result of the different problem set(s) considered by the respective published works. Furthermore, for the sake of fair comparison, we used the same performance metrics reported by earlier works when assessing the CAMA against the respective advanced metaheuristics considered. It is worth highlighting here that from the best of our knowledge, none of the advanced metaheuristics has taken up the challenge of attempting to solve the plethora of diverse VRP benchmarks that we have considered in this study.

VI. Conclusion

In this paper, a conceptual modeling of meme complexes in search as a set of coadapted memes has been introduced. In particular, the proposed memeplex representation, credit assignment criteria for meme coadaptation, and the role of emergent memeplexes in the lifetime learning process of an MA have been described in great detail. Subsequently, the CAMA as a manifestation of the proposed conceptual modeling of memeplexes to solve CVRPs of diverse characteristics is then proposed and studied empirically on four sets of VRP benchmarks with diverse characteristics. Results obtained on the VRP benchmarks demonstrated that CAMA not only produced higher quality solutions compared with MultiMeme on $2/3$ of the problem instances, but also superior to SGMA on half of the problem instances considered at 95% confidence level of statistical significance. In terms of scalability, on the Golden benchmark that contains many large-scale VRP instances, CAMA also displayed a near 100% success rate, in contrast with MultiMeme and SGMA where significant degradation in the success rate is observed. Last but not least, in terms of wall clock time for solution convergence, CAMA is noted to be more efficient than SGMA and MultiMeme and closely matches the performances of an MA that was designed using prior knowledge derived from human experts on specialized problems. In summary, the conceptual modeling of meme complexes in search, through the CAMA to solve on a plethora of VRPs, is shown to be capable of attaining high quality and efficient performance more robustly when assessed against several recent adaptive MAs and human-designed state-of-the-art advanced metaheuristics.

References

[1] Y. S. Ong, M. H. Lim, and X. Chen, "Memetic computation—Past, present and future," *Comput. Intell. Mag.*, vol. 5, no. 2, pp. 24–36, 2010.
[2] X. Chen, S. Ong, M. H Lim, and K. C. Tan, "A multi-facet survey on memetic computation," *IEEE Trans. Evol. Comput.*, vol. 15, no. 5, pp. 591–607, Oct. 2011.
[3] A. Lynch, "Thought contagion as abstract evolution," *J. Ideas*, vol. 2, no. 1, pp. 3–10, 1991.
[4] D. C. Dennett, *Consciousness explained*. Boston, MA: Little, Brown, 1991.
[5] J. Delius, "Of mind memes and brain bugs, a natural history of culture," in *The Nature of Culture*. Bochum, Germany: Bochum Publications, 1989, pp. 26–79.
[6] E. O. Wilson, *Consilience : The Unity of Knowledge*. Boston, MA: Little, Brown, 1998.
[7] R. Brodie, *Virus of the Mind: The New Science of the Meme*. Seattle, WA: Integral Press, 1996.
[8] W. Durham, *Coevolution: Genes, Culture, and Human Diversity*. Stanford, CA: Stanford Univ. Press, 1991.
[9] P. Moscato, "On evolution, search, optimization, genetic algorithms and martial arts: Towards memetic algorithms," California Inst. Technol., Pasadena, Tech. Rep. Caltech Concurrent Computation Program, Rep. 826, 1989.
[10] J. Tang, M. H. Lim, and Y. S. Ong, "Diversity-adaptive parallel memetic algorithm for solving large scale combinatorial optimization problems," *Soft Comput.*, vol. 11, no. 9, pp. 873–888, 2007.
[11] M. H. Lim, Y. Yuan, and S. Omatu, "Extensive testing of a hybrid genetic algorithm for solving quadratic assignment problems," *Comput. Optim. Appl.*, vol. 23, no. 1, pp. 47–64, 2002.

[12] Y. S. Ong and A. J. Keane, "Meta-lamarckian learning in memetic algorithms," *IEEE Trans. Evol. Comput.*, vol. 8, no. 2, pp. 99–110, Apr. 2004.

[13] Y. S. Ong, M. H. Lim, N. Zhu, and K. W. Wong, "Classification of adaptive memetic algorithms: A comparative study," *IEEE Trans. Syst., Man, Cybern. B, Cybern.*, vol. 36, no. 1, pp. 141–152, Feb. 2006.

[14] Q. H. Nguyen, Y. S. Ong, and M. H. Lim, "A probabilistic memetic framework," *IEEE Trans. Evol. Comput.*, vol. 13, no. 3, pp. 604–623, Jun. 2009.

[15] V. Tirronen, F. Neri, T. Kärkkäinen, K. Majava, and T. Rossi, "An enhanced memetic differential evolution in filter design for defect detection in paper production," *Evol. Comput.*, vol. 16, no. 4, pp. 529–555, 2008.

[16] M. H. Lim, O. Cao, J. H. Li, and W. L. Ng, "Evolvable hardware using context switchable fuzzy inference processor," in *Comput. Digital Tech.*, 2004, vol. 151, no. 4, pp. 301–311.

[17] Z. Zhu, S. Jia, and Z. Ji, "Towards a memetic feature selection paradigm," *Comput. Intell. Mag.*, vol. 5, no. 2, pp. 41–53, 2010.

[18] P. Foldesi and J. Botzheim, "Modeling of loss aversion in solving fuzzy road transport traveling salesman problem using eugenic bacterial memetic algorithm," *Memetic Comput.*, vol. 2, no. 4, pp. 259–271, 2010.

[19] O. Kramer, "Iterated local search with powell' method: A memetic algorithm for continuous global optimization," *Memet. Comput.*, vol. 2, no. 1, pp. 69–83, 2010.

[20] W. Jakob. (2010). A general cost-benefit-based adaptation framework for multimeme algorithms. *Memet. Comput.* [Online]. *2(3)*, pp. 201–218. Available:http://dx.doi.org/10.1007/s12293-010-0040-9

[21] A. S. S. M. B. Ullah, R. Sarker, and C. Lokan, "Handling equality constraints with agent-based memetic algorithms," *Memet. Comput.*, vol. 3, no. 1, pp. 51–72, 2011.

[22] F. Neri and E. Mininno, "Memetic compact differential evolution for cartesian robot control," *Comput. Intell. Mag.*, vol. 5, no. 2, pp. 54–65, 2010.

[23] J. R. Gonzlez, A. D. Masegosa, and I. J. Garcla, "A cooperative strategy for solving dynamic optimization problems," *Memet. Comput.*, vol. 3, no. 1, pp. 3–14, 2011.

[24] P. Novoa-Hernndez, C. C. Corona, and D. A. Pelta, "Efficient multi-swarm PSO algorithms for dynamic environments," *Memet. Comput.*, vol. 3, no. 3, pp. 163–174, 2011.

[25] C. K. Goh, Y. S. Ong, and K. C. Tan, *Multi-Objective Memetic Algorithm*. (ser. Book Series on Studies in Computational Intelligence, vol. 171). New York: Springer-Verlag, 2009.

[26] J. Y. Chia, C. K. Goh, K. C. Tan, and V. A. Shim, "Memetic informed evolutionary optimization via data mining," *Memet. Comput.*, vol. 3, no. 2, pp. 73–87, 2011.

[27] L. T. Bui, J. Liu, A. Bender, M. Barlow, S. Wesolkowski, and H. A. Abbass, "Dmea: A direction-based multiobjective evolutionary algorithm," *Memet. Comput.*, vol. 3, no. 4, pp. 271–285, 2011.

[28] G. Acampora, V. Loia, and M. Gaeta, "Exploring e-learning knowledge through ontological memetic agents," *Comput. Intell. Mag.*, vol. 5, no. 2, pp. 66–77, 2010.

[29] A. F. T. Winfield and M. D. Erbas, "On embodied memetic evolution and the emergence of behavioural traditions in robots," *Memet. Comput.*, vol. 3, no. 4, pp. 261–270, 2011.

[30] F. Liang, Y. S. Ong, A. Tan, and X. Chen, "Towards human-like social multi-agents with memetic automaton," in *Proc. IEEE Cong. Evol. Comput.*, 2011, pp. 1092–1099.

[31] B. H. Gwee and M. H. Lim, "Ga with heuristic-based decoder for IC floorplanning," *Integrat., VLSI J.*, vol. 28, no. 2, pp. 157–172, 1999.

[32] Y. Wang, T. H. Cheng, and M. H. Lim, "A tabu search algorithm for static routing and wavelength assignment problem," *IEEE Commun. Lett.*, vol. 9, no. 9, pp. 841–843, Sep. 2005.

[33] R. Meuth, E. Saad, D. Wunsch, and J. Vian, "Memetic mission management," *Comput. Intell. Mag.*, vol. 5, no. 2, pp. 32–40, 2010.

[34] A. Caponio, G. L. Cascella, F. Neri, N. Salvatore, and M. Sumner, "A fast adaptive memetic algorithm for online and offline control design of PMSM drives," *IEEE Trans. Syst., Man, Cybern. B, Cybern.*, vol. 37, no. 1, pp. 28–41, Feb. 2007.

[35] N. Krasnogor, B. P. Blackburne, E. K. Burke, and J. D. Hirst, "Multimeme algorithms for protein structure prediction," in *Proceedings of the Parallel Problem Solving from Nature VII. Lecture Notes in Computer Science*. New York: Springer-Verlag, 2002, pp. 769–778.

[36] W. Jakob, "A cost-benefit-based adaptation scheme for multimeme algorithms," in *Proc. Parallel Process. Appl. Math.*, 2007, pp. 509–519.

[37] N. Krasnogor, "Self-generating metaheuristics in bioinformatics: The protein structure comparison case," in *Genetic Programming and Evolvable Machines*. Norwell, MA: Kluwer, 2004, pp. 181–201.

[38] F. Neri, V. Tirronen, T. Karkkainen, and T. Rossi, "Fitness diversity based adaptation in multimeme algorithms: A comparative study," in *Proc. IEEE Congr. Evol. Comput.*, 2008, pp. 2374–2381.

[39] Q. H. Nguyen, Y. S. Ong, and M. H. Lim, "Non-genetic transmission of memes by diffusion," in *Proc. Genetic Evol. Comput. Conf.*, 2008, pp. 1017–1024.

[40] W. Jakob, "Towards an adaptive multimeme algorithm for parameter optimisation suiting the engineers' needs," in *Proc. Int. Conf. Parallel Problem Solv. Nature*, 2006, pp. 132–141.

[41] R. Dawkins, *The Selfish Gene*. Oxford, U.K.: Clarendon, 1989.

[42] H. Situngkir. (2004). On selfish memes: Culture as complex adaptive system. [Online]. Available:http://cogprints.org/3471/

[43] S. J. Blackmore, *The Meme Machine*. New York: Oxford Univ. Press, 1999.

[44] G. H. Lewes, *Problems of Life and Mind,* vol. 2. London, U.K.: Kegan Paul, Trench, Turbner, 1875.

[45] D. Blitz, *Emergent Evolution: Qualitative Novelty and the levels of reality*. Norwell, MA: Kluwer, 1992.

[46] J. Goldstein, "Emergence as a construct: History and issues," *Emergence*, vol. 1, no. 1, pp. 49–72, 1999.

[47] V. Robu, H. Halpin, and H. Shepherd. (2009, Sep.). Emergence of consensus and shared vocabularies in collaborative tagging systems. *ACM Trans. Web (TWEB)* [Online]. *3(4)*, pp. 1–34. Available:http://eprints.ecs.soton.ac.uk/18192/

[48] G. Kendall and E. Soubeiga, "Choice function and random hyperheuristics," in *Proceedings of the fourth Asia-Pacific Conference on Simulated Evolution And Learning, SEAL*. New York: Springer-Verlag, 2002, pp. 667–671.

[49] T. H. Cormen, C. E. Leiserson, and R. L. Rivest, *Introduction to Algorithms*. Cambridge, MA: MIT Press, 1990.

[50] J. F. Bard and L. Huang, "A branch and cut algorithm for the VRP with satellite facilities," *IIE Trans.*, vol. 30, no. 9, pp. 821–834, Sep. 1998.

[51] P. Toth and D. Vigo, "Exact solution of the vehicle routing problem," in *Fleet Management and Logistic*. Norwell, MA: Kluwer, 1998, pp. 1–31.

[52] D. H. Wolpert and W. G. Macready, "No free lunch theorems for optimization," *IEEE Trans. Evol. Comput.*, vol. 1, no. 1, pp. 67–82, Apr. 1997.

[53] M. Gendreau, G. Laporte, and J. Y. Potvin, *Local Search in Combinatorial Optimization*. Princeton, NJ: Princeton Univ. Press, 2003.

[54] S. R. Thangiah, "Vehicle routing with time windows using genetic algorithms," Slippery Rock Univ., Slipper Rock, PA, Tech. Rep., 1993.

[55] J. Y. Potvin, D. Dube, and C. Robillard, "Hybrid approach to vehicle routing using neural networks and genetic algorithms," *Appl. Intell.*, vol. 6, no. 3, pp. 241–252, 1996.

[56] K. Zhu, "A new genetic algorithm for VRPTW," presented at the Int. Conf. Artif. Intell., Las Vegas, NV, 2000.

[57] S. J. Louis, X. Yin, and Z. Y. Yuan, "Multiple vehicle routing with time windows using genetic algorithms," in *Proc. IEEE Congr. Evol. Comput.*, 1999, pp. 1804–1808.

[58] N. Christofides, A. Mingozzi, and P. Toth, "The vehicle routing problem," in *Combinatorial Optimization*, N. Christofides, A. Mingozzi, P. Toth, and C. Sandi, Eds., Wiley, Chichester, 1979, pp. 315–338.

[59] I. H. Osman, "Metastrategy simulated annealing and tabu search algorithms for the vehicle routing problem," *Ann. Oper. Res.*, vol. 41, no. 4, pp. 421–451, Dec. 1993.

[60] A. S. Alfa, S. S. Heragu, and M. Chen, "A 3-opt based simulated annealing algorithm for vehicle routing problem," *Comput. Oper. Res.*, vol. 21, pp. 635–639, 1991.

[61] M. Gendreau, A. Hertz, and G. Laporte, "A tabu search heuristic for the vehicle routing problem," *Manage. Sci.*, vol. 40, no. 10, pp. 1276–1290, 1994.

[62] F. B. Pereira, J. Tavares, P. Machado, and E. Costa, "GVR: A new genetic representation for the vehicle routing problem," in *Irish Conference on Artificial Intelligence and Cognitive Science*. New York: Springer-Verlag, 2002, pp. 95–102.

[63] C. Prins. (2004, Oct.). A simple and effective evolutionary algorithm for the vehicle routing problem. *Comput. Operat. Res.* [Online]. *31(12)*, pp. 1985–2002. Available:http://dx.doi.org/10.1016/S0305-0548(03)00158-8

[64] M. Kubiak, "Systematic construction of recombination operators for the vehicle routing problem," *Found. Comput. Decis. Sci.*, vol. 29, pp. 205–226, 2004.

[65] J. Berger and M. Barkaoui, "A hybrid genetic algorithm for the capacitated vehicle routing problem," in *Genetic and Evolutionary Computation Conference,* vol. 2723/2003. Berlin/Heidelberg, Germany: Springer-Verlag, 2003, pp. 646–656.
[66] X. Chen, Y. S. Ong, and M. H. Lim, "Cooperating memes for vehicle routing problems," *Int. J. Innovat. Comput., Inf. Control*, vol. 7, no. 11, pp. 1–10, 2011.
[67] B. D. Diaz. (2009). "VRP benchmarks," World Wide Web electronic publication, [Online]. Available:http://neo.lcc.uma.es/radi-aeb/WebVRP/
[68] P. Augerat, J. M. Belenguer, E. Benavent, A. Corber, D. Naddef, and G. Rinaldi, "Computational results with a branch and cut code for the capacitated vehicle routing problem," Univ. Joseph Fourier, Grenoble, France, Res. Rep. 949-M, 1995.
[69] N. Christofides and S. Eilon, "An algorithm for the vehicle dispatching problem," *Operat. Res. Q.*, vol. 20, pp. 309–318, 1969.
[70] B. L. Golden, E. A. Wasil, J. P. Kelly, and I. M. Chao, "The impact of metaheuristics on solving the vehicle routing problem: Algorithms, problem sets, and computational results," *Fleet Manage. Logist.*, pp. 33–56, 1998.
[71] N. Krasnogor, "Self generating metaheuristics in bioinformatics: The proteins structure comparison case," *Genet. Programm. Evolv. Mach.*, vol. 5, no. 2, pp. 181–201, 2004.
[72] X. Chen and Y. S. Ong. (2012). A Conceptual Modeling of Meme Complexes in Stochastic Search: Detailed Statistical Results. [Online]. Available: http://www.c2i.ntu.edu.sg/Courses/papers/report/RPT-CAMA-12-11.pdf
[73] J. F. Cordeau, M. Gendreau, and G. Laporte, "A tabu search heuristic for the periodic and multi-depot vehicle routing problems," *Networks*, vol. 30, pp. 105–119, 1994.
[74] P. Toth and D. Vigo, "The granular tabu search and its application to the VRP," *INFORMS J. Comput.*, vol. 13, no. 4, pp. 333–346, 1998.
[75] C. Rego and C. Rouchairol, "A parallel tabu search algorithm using ejection chains for the vehicle routing problem," in *Metaheuristics: Theory and Applications*. Boston, MA: Kluwer, 1996, pp. 661–675.
[76] C. D. Tarantilis and C. T. Kiranoudis, "Boneroute: An adaptive memory-based method for effective fleet management," *Ann. Operat. Res.*, vol. 115, no. 1–4, pp. 227–241, 2002.
[77] J. Xu and J. Kelly, "A network flow-based tabu search for the vehicle routing problem," *Transp. Sci.*, vol. 30, pp. 379–393, 1996.
[78] P. Wark and J. Holt, "A repeated matching heuristic for the vehicle routing problem," *J. Operat. Res. Soc.*, vol. 45, pp. 1156–1167, 1994.

Xianshun Chen received the Bachelor's degree in microelectronics from the School of Electrical and Electronic Engineering, Nanyang Technological University, Singapore, where he is currently working toward the Doctorate degree in the field of competent memetic algorithms with the School of Computer Engineering.

His research interests include the design and development of competent and innovative memetic computing frameworks and various soft-computing techniques as well as their applications.

Yew Soon Ong received the B.S. and M.S. degrees in electrical and electronics engineering from Nanyang Technological University (NTU), Singapore, in 1998 and 1999, respectively, and the Ph.D. degree in artificial intelligence in complex design from the Computational Engineering and Design Center, University of Southampton, Southampton, U.K., in 2003.

He is currently an Associate Professor and the Director of the Center for Computational Intelligence, School of Computer Engineering, NTU. His current research interests include computational intelligence include memetic computing, evolutionary design, machine learning, agent-based systems, and cloud computing.

Dr. Ong is the co-founding Technical Editor-in-Chief of the *Memetic Computing Journal*, the Chief Editor of the Springer Book Series on Studies in Adaptation, Learning, and Optimization, an Associate Editor of the IEEE COMPUTATIONAL INTELLIGENCE MAGAZINE, the IEEE TRANSACTIONS ON SYSTEMS, MAN AND CYBERNETICS PART B, *Soft Computing*, *International Journal of Systems Science*, and others. He is also the Chair of the IEEE Computational Intelligence Society Emergent Technology Technical Committee and has served as a Guest Editor of several Journals such as the IEEE TRANSACTIONS ON EVOLUTIONARY COMPUTATION.