A Multi-Facet Survey on Memetic Computation

Xianshun Chen, Yew-Soon Ong, Meng-Hiot Lim, and Kay Chen Tan

***Abstract*—Memetic computation is a paradigm that uses the notion of meme(s) as units of information encoded in computational representations for the purpose of problem-solving. It covers a plethora of potentially rich meme-inspired computing methodologies, frameworks and operational algorithms including simple hybrids, adaptive hybrids and memetic automaton. In this paper, a comprehensive multi-facet survey of recent research in memetic computation is presented.**

***Index Terms*—Adaptive memetic algorithms, evolution and learning, hybridization, memes imitation, memetic algorithm design issues, memetic algorithms in uncertain environments, memetic automaton, memetic computation, multiagent system, multiobjective memetic algorithms, surrogate-assisted memetic algorithms.**

I. Introduction

FOR THE PAST three decades, many population based search techniques have surfaced to become a mainstay of optimization [1]–[4]. These techniques rely on search ideologies that work on the basis of manipulating samples representative of the search sub-regions within the solution landscape. Such inherently parallel or multi-track search algorithms derive its strength from the simultaneous explorations of multiple search regions and exploitative interactions or coupling between the different search tracks.

Typically, each search track manifests itself as an individual within a population which represents a potential solution to the problem that the algorithm is intended to solve [2], [5]–[7]. The transitional states of each individual are governed by nature-inspired processes such as evolutionary and swarm intelligence including but not limited to genetic algorithm [8], evolution strategy [9], evolutionary programming [10], genetic programming [11], differential evolution [12], estimation of distribution algorithm [13], ant colony optimization [14], particle swarm optimization [15], and artificial immune system [16], which are generally classified under two main factors; modes of replacement for each individual and the guiding principles dictating the rule of behavior or interactions of the population as a whole [17]–[19].

Individuals within a population go through processes that operate on them. These processes are usually effected through operations that manipulate the solution coding in the form of strings (binary bit string, integer permutation, etc.), trees or graphs; most notably with selection (such as ranking, roulette wheel, stochastic sampling, tournament, etc.) and reproduction (crossover, mutation, elitism, etc.). Among the three main operations, mutation serves to introduce random variations into the population, essentially a mechanism to circumvent population stagnation. What is evident is that such processes introduce well-defined procedural manipulations of the solution.

These days the evolutionary processes are typically inspired metaphorically by processes observed in nature, more specifically in population genetics. One should take care to note that such processes usually do not instill any higher level domain or problem-specific information beyond that required to evaluate the fitness of the individuals in the population. Recent trends in research on optimization have biased toward algorithms that incorporate higher level of evolutionary and adaptive behavior [17]–[21]. As with genetics, the science of memetics [22] has served as a motivational pillar and inspiration toward the possibility of *meme* developing into a proper hypothesis of the human mind. In the context of computing [23], memes denote the recurring real-world patterns or domain-specific knowledge encoded in computational representations for the purpose of effective problem-solving. In the last decade, there are signs of increasing effort in research on evolutionary methodologies inspired by population memetics that further enhances the capacity of algorithms in establishing a good balance of explorative and exploitative traits.

As with genes in genetics, a meme is synonymous to memetics as being a building block of cultural know-how that is transmissible and replicable. While genes form the "instructions for building proteins," memes are "instructions for carrying out behavior, stored in brains (or other objects)" and passed on by means such as imitation [22]. The term *meme* can be traced back to Dawkins [24] in his book *The Selfish Gene*. The term has inspired the new science of memetics which today represents the mind-universe analog to genetics in cultural evolution, stretching across the fields of anthropology, biology, cognition, psychology, sociology and socio-biology [22]. In computer science and engineering, the meme-inspired computing methodology or more concisely memetic computation has become an increasing focus of research in recent years [23]. One of the most direct applications of memetic computation in problem-solving, in recent years, has been

Manuscript received May 18, 2010; revised September 29, 2010 and January 3, 2011; accepted January 31, 2011. Date of current version September 30, 2011. This work was supported in part by the Media Development Authority of Singapore, the Singapore–MIT GAMBIT Game Laboratory, and the Center for Computational Intelligence (C2I), Nanyang Technological University.

X. Chen and Y.-S. Ong are with the School of Computer Engineering, Nanyang Technological University, 639798, Singapore (e-mail: chen0469@ntu.edu.sg; asysong@ntu.edu.sg).

M.-H. Lim is with the School of Electrical and Electronic Engineering, Nanyang Technological University, 639798, Singapore (e-mail: emhlim@ntu.edu.sg).

K. C. Tan is with the Department of Electrical and Computer Engineering, National University of Singapore, 117576, Singapore (e-mail: eletankc@nus.edu.sg).

Digital Object Identifier 10.1109/TEVC.2011.2132725

memetic algorithms [25]. It is notably one of the simplest forms of memetics inspired techniques, establishing itself as one of the key methodologies in complex optimization and operational research tackling large-scale, real-world search problems.

The term memetic algorithms (MAs) originated from Moscato [26] as being the algorithmic pairing of a population-based search method with one or more refinement methods. MAs have been successfully applied to a wide range of domains that cover problems in combinatorial optimization [27]–[29], continuous optimization [30], [31], dynamic optimization [3], [32], and multiobjective optimization [1], [33], [34]. In particular, remarkable success on significant instantiations of MAs across a wide range of real-world applications have been reported, ranging from the field of bioinformatics [35], [36], permutation flow shop scheduling [27], scheduling and routing [28], [29], to nonlinear programming problems including aerodynamic design [30], atomic and molecular structure problems [37], optimal control systems with machine learning [38]–[43], and computationally expensive and uncertain environments [44], [45].

It is clear that MA has managed to stir up a wide interest in the field of evolutionary optimization algorithms [46] and at times, it is an easy source of confusion whereby any meme-inspired paradigms are conveniently taken to be synonymous to MA [23]. There are potentially many other rich meme-inspired designs, operational models, and algorithm frameworks which form the cornerstones of memetic computation as tools for problem-solving [3], [18], [23], [47]–[51]. To capture the essence of existing and potential research in this field, it is necessary to clarify memetic computation as being 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 problem-solving. From the algorithmic point of view, memes are units of domain-specific information useful for problem-solving. For instance, in hybrid algorithms, memes are taken as instructions, rules, strategies, *a priori* knowledge, etc., for search, and in more general problem-solving context, memes are encoded in various computational representations of the problem solvers which acquire increasing level of problem-solving capability [52].

To date, work on memetic computation can be divided into several unique categories for the sake of brevity. In the present survey, we classify them from the perspective of simple hybrids, adaptive hybrids, and subsequently memetic computation that culminates into framework that integrates memes into units of information for problem-solving, henceforth memetic automaton. Simple hybrids like MAs [26] represent a form of synergistic combination between population-based and local refinement heuristics. The rationale is that, often the performance of population-based search such as evolutionary algorithms can be enhanced if problem-specific refinement techniques are incorporated. These simple hybrids incorporate domain-specific information to augment the population-based search with refinement components. The term memetic algorithm has stemmed from the fact that the local refinement procedure is akin to a meme representing some form of domain-specific *a priori* knowledge of the human expert on how the solution can be better refined [26]. Since the individual in the population is perceived as undergoing a form of continuous refinement, the process is often referred to as local search, lifetime learning or individual learning.

Although achieving remarkable success on significant instantiations of specialized MAs across a wide range of application domains, researchers have also actively ventured into the emerging field of adaptive memetic algorithms. Adaptive hybrids [17]–[21], [53] are a class of memetic computation methodologies with great capacity of acclimatizing to suit a given problem in hand, by methodically utilizing acquired information about the matching of problems to procedures, and reconfiguring itself to adapt to the domain or instances of problems being solved. The potential algorithmic improvements have been attempted by adapting several key design issues including frequency of refinement, selection of the individual subset for refinement, intensity of refinement, choice of memes to undergo refinement as well as the choice of population-based search and fitness functions [54].

To address the ever increasing complexity and dynamic nature of problem-solving, memetic automaton is conceptualized in memetic computation. A memetic automaton is a software entity that autonomously acquires increasing level of capability and intelligence through embedded memes learnt independently or via interactions [52], [55]. Memes compete and cooperate in an evolutionary process, undergoing memetic transmission, selection, replication and variation. From a system level perspective, this serves as a rudimentary illustration whereby Universal Darwinism [24], [56] is put into practice. It liberates the conceptualization of memes to be building blocks of information (knowledge, belief, emotion, etc.) encoded in computational representations suitable for problem-solving. Although not explicitly referred to as memes, artificial neural networks [57], inductive logic programming procedures [58], graphs [59], etc. may be viewed as forms of memetic representations. Unsupervised, supervised, and reinforcement learning are potential tools that can facilitate learning pertaining to memes.

The purpose of the present survey is to present a comprehensive multi-facet exposition of recent research in memetic computation. For the sake of completeness, the survey begins in Sections II and III with a brief review on the early manifestations of memetic computation within the context of evolutionary computation, often construed as memetic algorithms, hybrids and adaptive hybrids. To give a multi-facet survey of memetic computation, a review on the more recent memetic algorithms, hybrids and adaptive hybrids is provided. In particular, the current review takes focus on the recent developments of hybrids and adaptive hybrids, during which there is a clear exponential growth of research publications in the area. Subsequently, Section IV highlights "Memetic Automaton," a natural progression toward establishing "meme" as the focal point of memetic computation and pinpoints some noteworthy emerging research trends in the field that have remained under-explored. Specifically, Section IV takes a meme-centric review of ongoing research in computational intelligence that focus on reusability, transferability, pattern generalization, as exhibited by various meme-based and agent-based cultural systems, etc. Lastly, we conclude this paper with some remarks on memetic computation. It is hoped that the classifications presented in this survey will assist in identifying fresh and exciting research frontiers of memetic computation.

II. Simple Hybrids

Many of the early works in memetic computation have materialized in the form of hybridization between the population-based search and refinement procedures [60]. They are often known as genetic local search, canonical memetic algorithm or simply as hybrids. From the algorithmic perspective, two or more distinct methods when combined together in a synergistic manner with the incorporation of domain knowledge [61] can greatly enhance the problem-solving capability of the derived hybrid. Furthermore, hybrids capitalize on the complementary advantage of population-based search (more explorative) and refinement (more exploitative) in that the former provides a reliable estimate of the global optimum while the latter concentrates the search effort around the best solutions found so far by searching the neighborhoods to produce better solutions more efficiently [17]. As such, hybridization is one important feature evident in the field of memetic computation. In this section, we discuss the main issues in simple hybrids, including the types of population-based search methods and refinement procedures, the levels of hybridization, modes of inheritance as well as the types of domain-specific information incorporated into simple hybrids.

A. Types of Search Methods

Here we present some of the well-established population-based search approaches typically used in the literature, and the refinement approaches used to enhance them, in the form of hybrids. What constitutes the population-based search and refinement procedure in simple hybrids is motivated by the presence of *a priori* knowledge on the problem domain.

Typical population-based search frameworks including genetic algorithm (GA) [8], evolution strategy (ES) [9], evolutionary programming [10], genetic programming [11], differential evolution (DE) [12], estimation of distribution algorithm (EDA) [13], ant colony optimization (ACO) [14], particle swarm optimization (PSO) [15], artificial immune system [16], etc. are stochastic in nature [2]. These are generic population-based search frameworks applicable to many problem domains. However, due to the ease of use, some are observed to be favored in particular domains. ACO and GA have stronger presence in the domain of order-based discrete combinatorial optimization, while DE, EDA, PSO are more often used for handling continuous variable problems.

In the domain of multiobjective optimization (MOO) [7], [33], [48], the more popular population-based methods include the non-dominated sorting genetic algorithm-II [62], evolutionary multiobjective optimization algorithms [27], Pareto archived evolutionary strategy [63], multiobjective PSO [64], etc. It is worth noting that to date many of the notably successful multiobjective evolutionary algorithms involve some forms of hybridizations specially designed for dealing with their respective challenges [27], [33], [34]. Special-purpose reproduction operators [65], specialized fitness function management schemes [66]–[68], multiobjective refinement operators [27], [69], etc. are among some of those introduced into multiobjective MA (MOMA) hybrids, as a way of improving, the rate of convergence to the Pareto optimum solutions, spread of solutions along the Pareto front, or dealing with the presence of non-linear constraints.

The refinement procedure used in hybrids, on the other hand, comes in the form of both deterministic [70] and stochastic [71]. Among the deterministic refinement approaches, branch and bound has been used for handling combinatorial problems such as longest common subsequence [72], while Hooke–Jeeves algorithm, Nelder–Mead simplex search method, Rosenbrock algorithm, etc., for handling continuous problems such as complex engineering design [73]. To cope with the scale-up in problem complexity, such as high multi-modality, researchers have ventured into stochastic refinement to enhance search diversity in the neighborhood. In combinatorial optimization domain, some of these include the tabu search for finding low auto-correlation binary sequences [74], simulated annealing to discover the optimal resources in p2p networks [75], etc. While many of these refinement methods have been designed for single-objective optimization, others specifically for multiobjective optimization have also emerged [7], [63], [76], [77]. One of the earlier work in MOMA [76] proposed a local search operator that perturbs the solutions in a randomly chosen non-dominated improving direction of the objectives, as a means of enhancing the rate of convergence to the Pareto front. Refinement procedure that directs the search toward and along the (local) Pareto set with the aim of obtaining a wider spread of solutions in the Pareto front like the Hill Climber with Sidestep [78] is among some of the others introduced recently. Other specialized refinement procedures proposed for mitigating the impact of uncertain search environments or noisy problems include the trust-region derivative-free memetic algorithms [79]–[82].

B. Level of Hybridization

In this section, we present the different refinement levels of hybridization used in MA. Researchers have incorporated refinement procedures at different stages of the population-based search. Most notably, it is possible to classify refinements as being incorporated before, after, or interleaved as depicted in Fig. 1.

Refinement incorporated before the population-based search in the form of initialization has been shown to enhance the efficiency of simple hybrids. Reference [83] proposed the opposition-based learning to initialize the foremost population, and showed that in contrast to random initialization, their proposed population initialization method helped to accelerate the search convergence rate. In the same spirit, [84] demonstrated search improvements with a sequential transformation method for population initialization in the context of traveling salesman problem, while [85] with a scheme based on population dispersion to effect the initialization. On the other hand, for procedures that are interleaved within the population-based search operators, refinement is conducted only after one

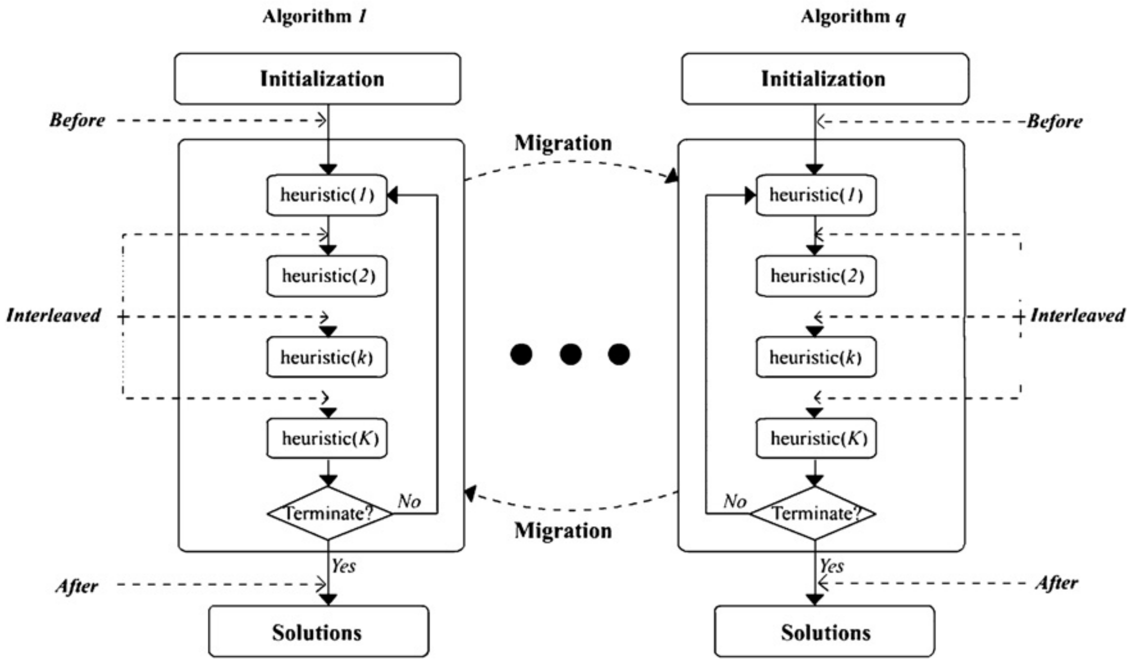

Fig. 1. Levels of hybridization.

or more reproduction operators has completed [69]. Note that the refinement process can be used more than once within the population-based search [86]. For other hybrids that incorporate the refinement process as a form of post-processing, [87] used inductive logic programming for improving the solutions in capacitated location routing problem and Steiner tree problem. The use of population-based search to coordinate interleaving refinement procedures [18] or fine-tuning the configurations of refinement procedures [88] to suit an optimization problem in hand have also been considered. These include the hyperheuristics [18], meta-Lamarckian MA [31], Indirect EAs [88], where the latter manages a collection of low-level heuristics as refinement procedures.

As illustrated in Fig. 1, the hybrids can also be deployed in the form of multi-populations or multi-islands or ensembles [89]–[93]. These independent hybrid models cooperate through the exchange of genetic or memetic information or other means. The improved robustness, diversity, and efficiency of multi-population hybrids have been reported by many [91], [92].

C. Modes of Inheritance

In this section, the modes of inheritance used in hybrids are reviewed. Two modes of inheritance have been studied closely in hybrids, namely Lamarckian and Baldwinian inheritances. Lamarckian inheritance [31], [50] characterizes that an individual passes on the learned traits acquired in its lifetime to its offspring directly. On the other hand in Baldwinian inheritance [94], [95], the learnt traits as acquired are not passed to the offspring. Instead, learnt traits result in the increased fitness of individuals, thus giving fitter learnt individuals higher chance of survival than naive Darwinian evolution.

In the context of computational intelligence, Lamarckian learning forces the genotype to reflect the result of improvement through placing the refined and improved individual back into the population to compete for reproductive opportunities. Baldwinian inheritance, on the other hand, only reflects the refinement improvement of individuals in fitness change while leaving the genotype of the population unaffected. Let $\mathbf{x}, \mathbf{x}^{opt} \in \mathbf{X}^N$ where $\mathbf{X}$ and $\mathbf{x}^{opt}$ denote initial solution vector and improved solution vector upon undergoing refinement, $\mathbf{X}^N$ is the solution space of dimension N with fitness function $f(.)$. Algorithmically, Lamarckian and Baldwinian learning effect the following changes $Lamarckian(\mathbf{x}, f(\mathbf{x})) \rightarrow (\mathbf{x}^{opt}, f(\mathbf{x}^{opt})), \{\mathbf{x}^{opt} \in \mathbf{X}^N\}$ and $Baldwinian(\mathbf{x}, f(\mathbf{x})) \rightarrow (\mathbf{x}, f(\mathbf{x}^{opt})), \{\mathbf{x} \in \mathbf{X}^N\}$, respectively. The majority of simple hybrids in memetic computation usually employ the mode of Lamarckian inheritance [31], [50] in their design.

While Lamarckian learning is known to be more effective than Baldwinian learning under static environments [96], it has the known effect of disrupting schema processing since it changes the genetic structure of individuals, thus prone to early premature convergence [97]. Baldwinian, on the other hand, tends to induce a slower schema disruption than Lamarckian inheritance, and is known for its hindering effect that tends to block out any genetic differences [98]. In practice, both modes of learning have found their places in simple hybrids where Lamarckian inheritance exhibited excellent advantage on unimodal landscape and problems with non-changing environments, while Baldwinian inheritance is deemed as more appropriate for problems subjected to dynamic and uncertain environments [44], [99], [100].

On real-world optimization problems subjected to noise or uncertainties in the design variables and environmental conditions, or time-varying fitness functions [99], such as robust aerodynamic shape design [44] and dynamic neural network evolution [100], hybrids that employed Baldwinian inheritance have reported notable search performances. Simulation studies such as [101] have also been made by some researchers to verify the higher robustness of Baldwinian inheritance than Lamarckian inheritance under dynamic environment. Further comparison between the two modes of inheritance was also reported in [102], whereby Lamarckian and Baldwinian were served as repair schemes for computational experiments on multiobjective 0/1 knapsack problems. Results obtained indicated that Baldwinian outperformed Lamarckian repair. Mixes of Baldwinian and Lamarckian inheritances in hybrids have also been investigated. One such instance in [103] reported improved efficiency in the search involving sorting network problem, whereby solutions were inherited using both learning modes.

D. Types of Domain Knowledge Incorporated

In this section, we examine the incorporation of domain-specific knowledge in simple hybrids. It is well established that the performance of hybrids correlates positively to the incorporation of domain knowledge, an outcome that is consistent with the "no free lunch" theorem [104]. A fundamental feature of hybrids is that the algorithms are typically designed to leverage or exploit any *a priori* knowledge on the inherent structure of the real-world problem. This problem-specific knowledge has been established to significantly enhance the performance of hybrids in solving the problem that they are designed for. The types of domain-specific knowledge that have been incorporated into hybrids to enhance search are in the form of specialized solution representations [105] and preprocessing schemes [106], as well as problem-specific search operators [107], etc.

Domain-specific solution representation permits flexible yet effective specialized operators to be exploited in the search [108]. The genetic vehicle representation [105] was designed in the same spirit for solving capacitated vehicle routing problems. It is worth noting that domain representation can also help narrow the search to a substantially smaller space compared to the original, where superior solutions are more likely to exist. Other successful specialized solution representations include the parallel job representation [109], the Artificial Chromosome with probability matrix [110], etc.

Domain-specific problem representation has also been used as a form of preprocessing in hybrids. In some algorithmic designs, domain-specific knowledge incorporated as part of the preprocessing steps has been used to make the search more focused, thus enhancing the overall search convergence rate. For instance, the pre-processing approach in [111] makes use of the *a priori* knowledge on redundant edges and nodes to reduce the problem size of the original generalized traveling salesman problem. Similarly, domain knowledge incorporated into fuzzy rule generation and prescreening prior to rule selection when dealing with high-dimensional pattern classification problems has led to significant improvements in the efficiency of fuzzy rule selection using MOMA [112]. Other successful incorporations of domain knowledge for preprocessing include the block-cut tree techniques proposed in [113].

Taking advantage of the enriched representation, [114] describes a crafted crossover that exploits the specialized representation for effective exchange of partial routes between individuals, in vehicle routing problems. Specialized operators such as position based mutator [115] have been proposed for solving job-shop scheduling problems more effectively. In many occasions, domain-specific information is also incorporated within the refinement process. Some of the notable schemes in single-objective, multiobjective optimization as well as uncertain, dynamic environments, include Concorde [116] which is designed for solving traveling salesman problem using the Branch and Cut method, the specially designed refinement operators [107] for accelerating the convergence rate to Pareto front, and the random multi-start variable neighborhood local search [117].

Besides leveraging on domain knowledge in the design of specialized operators, the knowledge can be harnessed in other ways, such as within the fitness evaluation process. When handling problems with computationally expensive objective and constraints functions [45], [80]–[82], in particular, surrogate-based memetic algorithms or hybrids have been widely used [45], [81], [118]–[120] to enhance the computational tractability of the algorithm. In the refinement stage of such hybrids, approximation of the problem landscape is effected on the fly within a trust-region local refinement framework that manages the interplay between the original computationally intensive objective and constraint functions with computationally cheap surrogate models [120]–[123]. With the zero-order and first-order consistency conditions imposed at the initial guess, global convergence of the trust-region local search framework that embeds surrogate models can be guaranteed [124]. Hence, surrogate-based memetic algorithm possesses the benefit of lower susceptibility in converging to false global optimum over other surrogate based evolutionary counterparts [125], since the use of surrogates is confined within the trust-region local search. Taking the spirit of optinformatics [126], a machine learning approach [127] to making geometrical and structural predictions on the feasibility of candidate solutions was introduced to effect the decision of performing further refinements on the individuals in the context of non-linear programming with active constraints.

III. Adaptive Hybrids

In recent decades, the general practice of hybridization in MAs has evolved into the emerging field of hybridization with adaptation. Adaptation of parameters and operators represents one of the most important and promising areas of research in memetic computation [17], as practitioners often encounter problems for which they have only limited insight into the structure of the problem, making it difficult to design specific MA without adaptation. These self-configuring algorithms are capable of acclimatizing to suit a given problem in hand, by methodically utilizing acquired information about the matching of problems to procedures, and reconfiguring itself to adapt to the problem as the search progresses. In contrast to simple hybrids in which domain knowledge is only captured and incorporated once by human expert during the design of MAs, adaptive algorithms handle the population diversity of the search and the various design issues by adaptable strategies and parameters for problem-solving [17], [128]. In this section, we present the population diversity management and adaptation strategies for adaptive hybrids in memetic computation.

A. Diversity Management

An important goal of adaptation in the design of search algorithms is the preservation of diversity in the solutions being explored. In single objective optimization, the problem of premature convergence, whereby the population converges around a suboptimal solution can be particularly problematic for hybrids. Premature convergence poses as a bigger challenge to hybrids in the contexts of multiobjective optimization and problems with uncertain and dynamic environments. In the former, a poorly spread Pareto front may result, while hybrids are more prone to failure in adapting to the changing environment once it has converged to some basin of attraction.

Population diversity refers to the extent of variation in the population based on the individuals' structure or performance [129]. Various measures of diversity have been used in MA, such as the extent of variations in fitness values [130], aging of solutions as defined in [131], etc. Based on these diversity measures, various techniques have been incorporated [48], [75], [132].

With fitness-based diversity, adaptive hybrids attempt to maintain the population diversity by managing good fitness spread among individual solutions in the population. In the coordination of multiple refinement procedures, [40] monitored how close the average fitness is compared to the population elite. Others, on the other hand, chose to effect the search adaptation according to the sparseness of individuals [75], [132] or the super-fit individual [73] in the population. To prevent loss in diversity, additional populations using completely different fitness criteria have also been considered [91]. When dealing with problems with uncertainties, population diversity is enhanced by locating and tracking multiple moving peaks via hierarchical clustering [133] and variable relocation evolutionary process [134] which makes already converged individuals evolve more rapidly to new environmental conditions. Other fitness-based diversity schemes proposed for single and multiobjective contexts include fitness sharing [48], clustering [135], adaptive grid [62], etc.

With distance-based diversity, adaptive hybrids attempt to maintain the population diversity by managing the distance spread among individual solutions in the population while the search progresses. Some relevant works in this discipline include adaptation based on population entropy [136], diversity-adaptive parallel memetic algorithm for solving large scale combinatorial optimization problems [93], the many population management strategies proposed for handling multi-dimensional knapsack problems and weighted tardiness single-machine scheduling problems [137].

B. Design Adaptation Issues

Several core design adaptation issues of simple hybrids have been considered to date. In particular, it is now well-established [60] that potential algorithmic improvements can be achieved by adapting several key designs, including the frequency of refinement, selection of individual subset to undergo refinement, intensity of refinement, and choice of procedures to conduct refinement. In this section, we present an overview of recent works directly on these issues.

1) *Frequency of Refinement and Individual Subset Selection:* The frequency or probability of refinement defines the proportion of a population that should undergo the refinement process [138]. The aim is thus to balance the amount of computational budget allocated for population-based search versus refinement [139]. The adaptive selection of individual subset to undergo refinement helps enhance overall search productivity [60], [140], [141].

Reference [27] investigated the impact of frequency of refinement in both single and multiobjective context for permutation flowshop scheduling, where only the elite individuals of the population undergo refinements. A fitness uniform selection scheme to adapt both the frequency of refinement and individual subset selection based on population diversity was also considered for cellular hybrids [140]. In particular, the trick lies in reducing the likelihood of applying the refinement procedure on individuals falling in previously refined basins of attraction.

2) *Intensity of Refinement:* Refinement intensity of an individual defines the amount of computational budget allocated to the refinement process. Study conducted to examine the effect of this design issue had confirmed its significant impact on search performances [142].

Reference [143] proposed an adaptive refinement procedure, whereby fitness statistics obtained from the population of individuals are used to adaptively adjust the degree of refinement intensity. Reference [144] adapted by building refinement chains beginning with a fixed intensity, and subsequently using these stored chains to define the degree of refinement intensity for new individuals. It is worth noting that an extended version of the local search chain memetic algorithm for high dimensional problems [145], i.e., 1000 dimensions, was demonstrated to top the chart in the recent competition at WCCI 2010. Reference [146] also proposed a crossover-based adaptive local search that adapts the refinement intensity of the individual learning process based on some hill-climbing heuristic.

3) *Refinement Procedures or Memes:* In this subsection, we review the adaptive selection of refinement procedures or memes in memetic algorithm. The choice of memes affecting the performance of hybrids significantly was demonstrated by several researchers on a variety of problems of diverse properties [17].

Several adaptation schemes have been put forward to adapt the choice of memes in hybrids. Hyper-heuristic [18] fuses a collection of memes in the form of low-level heuristics so that the actual meme applied on each individual may differ. It is a form of a heuristic to choose memes. Adaptive choice of memes at each decision point was also proposed in multi-meme algorithms [164] and meta-Lamarckian learning [96], while the co-evolution of memes and genes in adaptive MA was considered in [19]. Reference [147] introduced a unified framework for evolutionary algorithm design that extends beyond the selection of memes to perform refinement. In particular, the probabilistic memetic framework described in [147] estimates the theoretical upper bound on computational budget for refinement as a means to control and adapt the multiple design issues of the hybrid simultaneously at runtime, from how many and what individuals that would undergo lifetime learning to which appropriate meme(s) to employ in learning. To alleviate the potentially high intensity and computational budget incurred in the refinement when solving problems with computationally expensive cost functions and uncertain environment, adaptive hill climbing [53] and trust-region based local searches [44], [45], [127] have also been introduced. Some of the refinement adaptation mechanisms proposed for handling MOO problems concentrated on the adaptive coordination of local refinement methods based on cross-dominance [20] and resource productivity criteria [21].

C. Theoretical Analysis of MA

Despite the broad research activities on the issues of MA design, there is the lack of sufficient rigorous theoretical study to date, since it is typically hard for theory to keep track with the state-of-the-art algorithms that has become increasingly elaborate and complex. Nonetheless, there are recent signs of increasing research in the design and development of formal analytical study on MA. Polynomial local search complexity theory [148], fitness landscape analysis [149], [150], frequent pattern mining [151], connectivity structure analysis [152], local optima network [153], linkage graph [154], fixed-parameter algorithmics [155] and phenotypic plasticity [156]–[158] are among some of those that have been introduced to facilitate our understandings of memetic algorithms, thus rendering the design of novel adaptive hybrid methodologies possible.

The polynomial local search (PLS) complexity theory was considered in [159] to analyze the worst case complexity of hybrids as a combination of population-based and refinement procedures, particularly in the discrete domain such as the Traveling Salesman Problem. The PLS completeness on a family of hybrids or memetic algorithms can be derived to gain insights on the expected worst case behavior of the solver on the problem of interest.

Fitness landscape analysis [149], [150], on the other hand, denotes one of the most popular tools for gaining insights into the problem fitness landscape with respect to the solver used. Local optima network [153] and connectivity structure analysis [152] represent some recent theoretical analysis tools introduced to derive knowledge or more precisely the inherent properties of the search space, thus offering insights into the problem difficulty with respect to hybridization.

In the same spirit, frequent schema mining [151], frequent subtree mining [160], and linkage graph discovery [154] were introduced for extracting latent solution patterns or recurring problem structure from online optimization data encountered along the search, thus enhancing the understanding on search dynamics and structural properties of the problem, thus improving search performances. Fixed-parameter algorithmics with kernelization techniques have also been proposed [155] for analyzing the given problem and divide the problem structure into smaller independent problem kernels that can be subsequently conquered and solved more efficiently.

On the theoretical forefront pertaining to mode of inheritance, [161] verified that the saddle-crossing ability of Baldwinian learning can help alleviate the issue of getting stuck in local optima, on simple multi-peaked and dynamic landscape. The higher short-term adaptability, phenotypic plasticity, and other benefits of the Baldwin effect have also been studied in [156]–[158].

IV. TOWARD MEMETIC AUTOMATON

To date, memes in simple and adaptive hybrids (as discussed in Sections II and III) have been established as more of a complimentary role in the "learning" phase of the evolutionary cycle. Hence, the true nature and potential merits of memes may not have been fully exploited in the context of evolutionary computation.

From the formalism of simple hybrids to adaptive hybrids, it is evident that algorithms seek to achieve greater level of adaptivity to address the ever increasing complexity and dynamic nature of problem-solving. The aim of memetic computation should therefore culminate into a meme-centric framework that seamlessly integrates memes into units of domain information useful for problem-solving.

Taking meme as focal point of interest in the context of computational intelligence, we define memetic automaton as an adaptive entity that is self-contained, uses memes as building blocks of information that facilitates problem-solving. From a software perspective, a *memetic automaton* is a software agent capable of autonomic behavior during problem-solving [162], [163]. In general, the term "agent" refers to software entity that can interact with the surroundings and adapt to a complex dynamic environment. Hence, intelligent agents are able to acquire increasing level of capability through their interactions by updating its meme pool [52]. This acquired intelligence helps the agent adapts to its changing surroundings.

Two important aspects of memes with regards to memetic automaton are representation and evolution. Memes can represent the agent's ideas and knowledge captured as memory items and abstractions [164]. The expression of such memes may be in the form of recurring real-world patterns or structures of a domain that infect other agents' perceptions, beliefs, minds, etc. Memetic evolution is also central to the behavioral aspects of memetic automaton. Dawkins in [24] coined Universal Darwinism to draw the analogy on participation of genes in genetic evolution to memes participating in a cultural evolutionary process [24]. The memetic evolutionary process is then primarily driven by imitation [22], which takes place when memes are transmitted. Individuals who face repeated tasks of having to make choices, imitate others who obtained high payoffs in the previous rounds [165]. With regards to imitation, memetic selection concerns with whom one imitates [165], while memetic transmission and variation relates to how one imitates and what is imitated or assimilated [166]. Fig. 2 gives a depiction of the imitation process among multiple agents [51], [167], [168].

In this section, we direct our attention on the memetic representation and mechanisms pertaining to memetic automaton. In particular, we give a detailed account on memetic

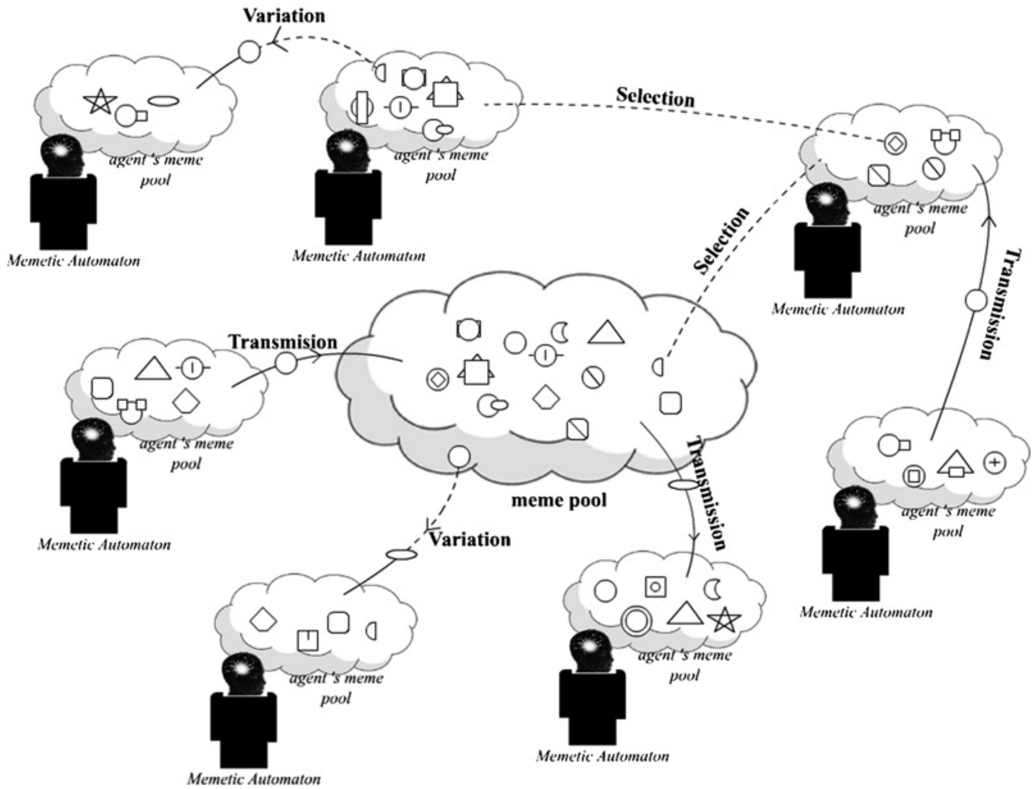

Fig. 2. Imitation process between memetic automata by means of Universal Darwinism.

representations and imitation as the principal driving force behind memetic evolution, by focusing on the various representations of memes in computational intelligence as well as all three fundamental mechanisms of Universal Darwinism, namely selection, propagation and variation.

A. Memetic Representation

An important step in memetic computation is to identify a suitable memetic representation of the memotype (i.e., the actual information content of a meme [22], [169]) for the problem being solved. To date, several recent efforts have looked into memetic representations for problem-solving, and can be delineated into the categories of symbolism, connectionism or others [23], [164].

1) *Symbolic Memetic Representation:* To date, a plethora of symbolic memes represented using schemata [49], [170], [171], tree, graph [172], logic [58], [173], etc. have been established. In schema meme, declarative knowledge has been represented in symbol structures, while procedural knowledge represented as elementary information processes called behavioral rules [174]. The agent-based memetic algorithm described in [49], [170], and [171] involves autonomous agents that perform evaluations with schema memes that reside in the agent's mind universe [171] to solve hard computational problems [174]. These formalized memes are materialized as private memory and socially-shared memory of the agents within a society [171], [175]. Other schemata-based memetic representations including the simplified definite-clause grammar in iterated learning model [167] and string-based memetic encodings in [176] and [177] for the evolution of linguistic behavior among agents and for meme-gene coevolutionary studies investigated on the NKCS fitness landscapes[1] [176], have also been established.

Memes, on the other hand, have also transpired in the form of decision rules encoded as tree or graph structures, with nodes composing of symbolic operators and operands. In [172], agents imitate the observed success of other agents by acquiring memes in the form of well-developed GP-based subtrees from other viable agents, thus enhancing their own capability in solving computationally expensive problems. In [178], tree-encoded memes representing tasks and solutions are communicated to facilitate collaborations among agents to solve the insect locomotion problem, within a distributed environment. With respect to computational agent stock markets [179], graph-encoded memes was used as the fundamental building blocks of agents in individual and social learning.

First-order logic memes that formalize the agent's belief or pieces of knowledge for effective problem solving have also been considered. The extended well founded semantics [58], [173] represents a key ingredient for building rational epistemic agents capable of managing and revising their goals and belief, as well as knowledge production in a dynamically configurable society of agents. Further extension to fuzzy memes that takes into account the uncertain property of human thinking have also been described in [180] to better

[1]NKCS landscapes are coupled fitness landscapes in which a move by one species deforms the fitness of other linked species coexisting in an environment with N denoting the number of genes in each species, K the number of genes that affect the fitness contribution of each gene, C the number of connections between species, and S the number of species

model and simulate the complexity involving the evolution of the mind in agents and the relationships among agents within a social network.

2) *Connectionist Memetic Representation:* In this category, meme generally takes the form of "information encoded in neural structure" or "constellation of activated neuronal synapses in memory" [181] in agent. In [57] and [168], neural memes materialize as recurring patterns captured within artificial neural networks (ANN) that define the behaviors and guiding criteria of agents, to facilitate cultural transmission of successful collective and learnt behavior that constitute toward computationally preferred task-solving approach. Within the connectionist paradigm, various types of ANN architectures and learning methodologies have been used in the representation of memes. Self-organizing map was used for the mining of internal rhythmic memes in [182], while multilayer perceptrons for classification of compositional meme patterns and promoting behavior that mimics high quality expert agent's strategy [57], [182], [183]. The Widrow–Hoff ANN [168] was also investigated to equip descendent agents with capabilities to learn from their parents in a supervised manner.

3) *Other Forms of Memetic Representations:* Alternative forms of memetic representations include the Bayesian models [59], [184], Hidden Markov models [185], state-transition system models [51], [183], etc. These memetic representations take on various computational forms, based on differing assumptions and considerations about the agent environment.

Among memetic representations that operate with incomplete or uncertain information, Bayesian models such as [59] and [184] facilitate rapid learning of human behaviors in agents through a probabilistic framework for acquiring new memes via imitation-learning with Bayesian inference. This enables agents to master challenging tasks such as autonomous navigation. Dynamic Bayesian networks such as the hidden Markov models [185] are capable of representing linguistic memes from the perception of experiences encoded in continuous-valued signals. Such networks endowed agents with the capabilities of acquiring memes for decision making from other agents. Some other memetic representations include the locally weighted regression [186] and the self-model proposed in [187] for facilitating the imitation of memes in the form of complex biped locomotion movement from human experts, thus enabling agents to diagnose and recover from unanticipated situations faster, by drawing on the developed memes previously learned by successful agents.

B. Memetic Selection

Here, we describe the meme selection mechanism. Selection for the purpose of imitation is ultimately an agent's expression of its exploitative trait, i.e., selecting a meme to imitate at each decision point t. Consider a population of agents $A(t) = \{a_1, a_2, \ldots, a_p\}$ and a meme pool $M(t) = \{m_1, m_2, \ldots, m_q\}$. The memes expression matrix of the system at a decision point t can be written as

$$E(t) = [e_{ij}]_{q\times p} \tag{1}$$

$$e_{ij} = \begin{cases} 1, & \text{if } a_j \text{ expresses } m_i \\ 0, & \text{otherwise.} \end{cases} \tag{2}$$

$M(t)$ refers to the common meme pool [51], [175] as illustrated in Fig. 2. The meme pool of an agent a_i is likewise written as $M_i(t)$ [175], [183]. The goal of selection of an agent a_i is to choose a meme $m \in M(t)$ such that agent a_i through expressing m improves its chances of success in future actions [51]. To determine which meme an agent selects at a given decision point, evaluation and behavioral rules are typically defined. In what follows, we review some of the current meme selection schemes with respect to evaluation rule and behavioral rule.

1) *Evaluation Rule:* The evaluation rule assigns a qualitative or quantitative figure of merit to the memes. Qualitative estimation of payoff for memes has been realized based on nominal or ordinal scale. With nominal scale, a label is assigned to each meme, as used in the coordination strategy of multiple memes [40], [130], [188]. In contrast, the ordinal scale ranks the memes in order of payoff [189]. In quantitative evaluation, the payoff of a meme, m_i, can be quantified based on statistical measures. Two common payoff measures used are maximum $f_{\max}(m_i)$ (e.g., [190]) and average $\overline{f}(m_i)$ (e.g., [179], [191]), defined as follows:

$$f_{\max}(m_i) = \max\left\{ f\left(a_j\right) \times e_{ij} \quad \middle| \quad \text{for } j = 1, 2, \ldots, \quad p \right\}$$

$$\overline{f}(m_i) = \frac{\sum_{j=1}^{p} f\left(a_j\right) \times e_{ij}}{\sum_{j=1}^{p} e_{ij}}$$

where $f\left(a_j\right)$ is the payoff associated with agent a_j.

The memes can be further quantified in absolute or relative scales. Instances of absolute scale can be found in [175] and [179] in which the combined behavior-evaluation cost and the mean absolute percentage error are used, respectively, for defining meme fitness. In contrast to absolute scale, relative scale compares memes based on their relative statistical performance, such as the roulette wheel meme selection operation [190] based on the relative payoff of the memes.

2) *Behavioral Rule:* The behavioral rule specifies how an agent selects a meme from $M(t)$ based on its payoff. Existing behavioral rules can be differentiated into "imitate the elite" and "proportional imitation." The "imitate the elite" rule specifies that an agent will imitate the elite meme from the meme pool, according to average or maximum payoff. In implicit imitation [192] or teacher-based supervised evolutionary learning [57], beginner agents learn by imitating the memes from expert agents of the same environment, based on the payoff criterion. On the other hand, the "proportional imitation" rule specifies the degree that one agent imitates another, is positively correlated to their difference in payoffs. In [190], an agent imitates by selecting a meme probabilistically from its neighboring agents, through a localized roullete wheel selection operation based on relative payoff. Other variants of the "proportional imitation" rule include the rule-exchange mechanism proposed in [175] whereby a number of rule-based memes with low strength values are replaced by rule-based memes with high strength values between two arbitrary agents. In relation to the aforementioned rules, behavioral noise and inertia are some concepts designed to handle the uncertainty during meme selection. A noisy behavioral rule

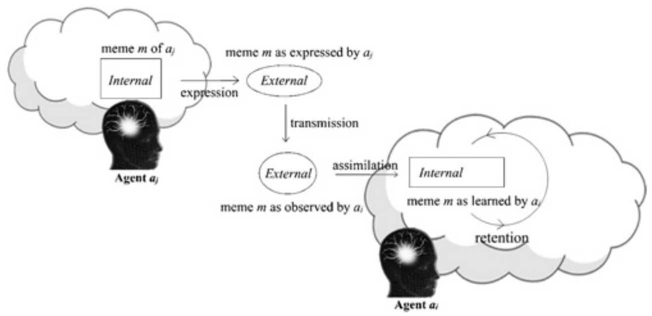

Fig. 3. Imitation via memetic transmission.

may correctly imitate the selected meme most of the times, but makes mistakes (e.g., imitate another meme instead of the selected meme) occasionally. On the other hand, for a behavioral rule with inertia, the agent is driven to imitate the selected meme although occasionally switching back to its original meme [193].

C. Memetic Propagation

In this section, we outline the imitation process with respect to memetic propagation. Fig. 3 illustrates the imitation process in which agent a_i imitates the meme expressed by agent a_j [194]. In the imitation process as illustrated, agent a_j first expresses its memes, which is observed by agent a_i in the subsequent transmission stage. Agent a_i then infers the memes from its observation based on its knowledge, belief, etc., by means of memetic assimilation. There is also a retention stage which occurs as new memes are continuously learnt and evolved by agent a_i. Memetic variation may occur during any of the four stages, which will be detailed later in Section IV-D. Reference [194] gives a detailed account on how memes are assimilated, retained, expressed, and transmitted in the imitation process from a sociological viewpoint.

1) *Meme Transmission:* A number of meme transmission mechanisms have been proposed to date. Depending on how an agent acquires a meme, the modes of transmission are classified into vertical and horizontal transmissions, as illustrated in Fig. 4.

In the vertical transmission, for instance, an agent acquires the innate behaviors from its parents, through genetic inheritance. Reference [168] showed a form of vertical transmission, whereby a descendent agent inherits the behavior and guiding criteria of its parents by assimilating their network structures. Other examples of vertical transmission in multiagent environments include [195] whereby each agent inherits the innate linguistic behaviors of their parents' peers during language evolution.

In the horizontal transmission, behaviors are transmitted via memes among peers of the same population. In the so-called agent-based memetic algorithms [49], [170] and autonomy-oriented Computing [171], [174], agents follow a horizontal transmission exchanging memes that takes the form of behavioral rules with neighboring peer agents modeled in a lattice-like cellular structure [49], [170] or a linked graph [171], during its lifetime. The rule-exchange and collective-knowledge-reuse schemes of the organizational-learning-oriented classifier

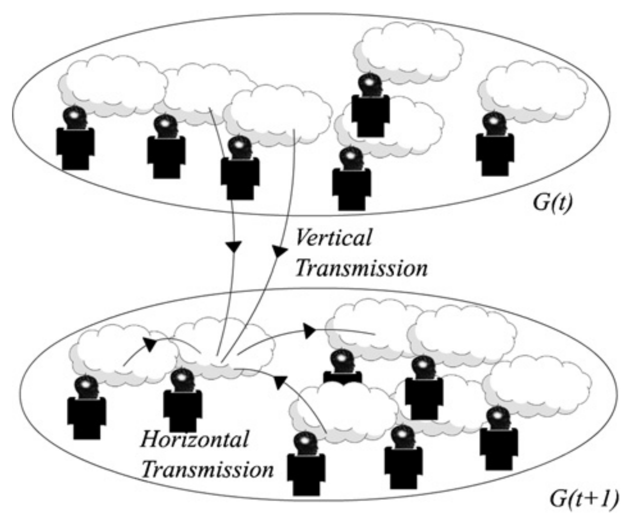

Fig. 4. Mode of transmission: vertical transmission of innate behaviors between parents in generation $G(t)$ and their child in $G(t+1)$ and horizontal transmission of memes among agents in $G(t+1)$.

system proposed in [175], and the advice exchange framework for peer agents facing problems containing similar structure in [183] represent some others in a horizontal transmission mode.

2) *Meme Expression and Assimilation:* In memetic expression and assimilation, the focus is placed on the sociotypes (which is the social expression of a meme [169], as analogous to the phenotype of a gene) instead of memotypes of the agents. The agent assimilates memes by observing the behaviors expressed by other agents. In [168], an agent assimilates the memes of other agents by inferring them from the perceived state-action-evaluation behavior expressed using the Widrow–Hoff neural network learning algorithm. Similar expression and assimilation stages exist in the iterated learning model in [167] whereby each agent indirectly acquires linguistic memes from another by learning from a set of meaning-signal pairs generated from the linguistic memes of another agent.

3) *Meme Retention:* During the process of imitation, memes are constantly injected either into the meme pool of an agent [167] or the common meme pool [51], which is usually of finite size. This results in a competition among the existing memes and the injected memes during the retention stage. At this stage, the longer a meme survives in an agent's meme pool or the common meme pool, the more it will spread through the population of agents. Various strategies in memetic computation incorporate schemes for avoiding unnecessary loss of memes of particular usefulness. In [175], memes having a higher strength value than a pre-defined threshold are not replaced when new memes are introduced into the meme pool of an agent. Other approaches include the use of automatically defined function for protecting discovered rules in tree-based memetic representation [172], and the reuse of learned policy for the automatic development of the learning structure [50].

D. Memetic Variation

Here, we discuss the issues pertaining to memetic variation taking place in the imitation and evolutionary process.

Memetic variation process refers to the self-generation and reconfiguration of memes. The process takes place during the various stages of meme propagation. In particular, variation is active during the transmission and retention stages [167], [168].

In the transmission stage, memetic variation occurs when there is no faithful copying of memes between agents [57], [168], [182] or as a result of recombination or mutation of memes [172], [175], [190]. Variation in [168] takes place due to the inherent biases that arise when an agent attempts to updates its memes by training its meme-inhabited networks based on the incomplete or imperfect training samples it observes from its parents. On the other hand, variation can be a result of memetic recombination such as in [172], where the memetic crossover is used to restructure memes that are embodied as tree-based memory. Another manifestation of memetic variations includes the memetic mutation designed for evolution of unplanned coordination among independent agents [190].

In the retention stage, variation happens when the integrity of existing memes is affected as a result of lifetime learning [51] or each agent's individual cognition toward novel or curious situations [167], [196]. In [51], each agent reconfigures a selected meme from the meme pool to meet its design goals through reinforcement learning, subsequently transferring the refined meme back to the meme pool shared with other agents. On the other hand, the invention algorithm proposed in [167] pushes agents to invent new linguistic memes for spontaneous evolution of linguistic structure. Other variation mechanisms include the intelligent adaptive curiosity mechanism proposed in [196] for propeling an agent toward innovation that maximizes its learning progress and the rule-generation mechanism proposed in [175] which creates new rule-based memes when an agent fails to find stored knowledge matching the current environmental state.

V. Discussion on Potential Future Work

Memetic algorithms, hybrids and their adaptive variants have to date enjoyed excellent success across a wide realm of real world problem domains that stretches from business, economics and finance, to design in science and engineering. In the process, many dedicated hybrids have been crafted to solve domain-specific problems better, with many more operational hybrids expected to be introduced for solving freshly identified problems in decades to come. To date, the use of meme as a form of individual learning or refinement procedure has been to enhance the search efficiency and solution precision of traditional EA. In some cases, it is worth noting the use of memetic algorithm or hybrid is inevitable.

In the last decade, the progress made in adaptive hybrids for handling multi-criteria, multiobjective, uncertain and dynamic problems, and constrained optimization tends to lag those of single-objective search. On the whole, it is nice to note that some progress to enhance our understanding on the search behavior of hybrids and the design of hybrids that come with theoretical rigors has emerged in recent years, although the effort expended remains insignificant relatively to the number of complex algorithms that have emerged. Furthermore, it is worth highlighting that to date, the manifestations of memetic computation within the community of evolutionary computation and meta-heuristics have been confined within the arena of memetic algorithms, hybrids and adaptive hybrids, i.e., a mindset that could limit the scope of what a meme-centric approach is capable of. The conceptualization of memetic automaton in memetic computation thus serves to unleash the significant number of potentially rich meme-inspired design, operational models, and algorithm frameworks that could form the cornerstones of memetic computation as tools for problem-solving.

To capture the essence of existing and potential research in this field, memetic computation is portrayed in the present survey as being a paradigm that uses the notion of meme(s) as units of information encoded in computational representations for addressing the ever increasing complexity and dynamic nature of problem-solving. In general, it is worth stressing here that a major drawback of existing optimization and computational intelligence approaches in the literature is the apparent lacking of automated knowledge transfers and reuse across problems. Particularly, most search methods tend to start 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. Like genes that serve as building blocks in genetics, memes are naturally building blocks of meaningful information, in the form of recurring real-world patterns that can be captured automatically—for instance, by means of frequent schema mining [151], frequent subtree mining [160], linkage graph discovery [154], or others. This provides a storage of building blocks or memes to common problems or sub-problems (of a complex problem that can be solved individually and independently in the spirit of divide and conquer), and supports reuse across problems. The capacity to draw on memes from past instances of problem-solving sessions thus allows the search to be more intelligent, leading to solutions that can be attained more efficiently on unseen problems of increasing complexity and dynamic in nature. These atomized units of memes, meta-memes [169], [197], or memeplex [22], [198] in computing can then be expressed in hierarchical nested relationships or conceptual entities for higher-order learning [56], thus forming societies of the mind for more effective problem solving.

In intelligent agent frameworks, moreover, memes encoded in computational representations, would form the underlying building blocks (knowledge, belief, emotion, etc.) of the mind-universe of an agent, thus competing and cooperating through an evolutionary process, undergoing memetic transmission, selection, replication, and variation. It is also worth noting that the memetic expression and assimilation stages discussed in Section IV-C have remained under-explored to date. The transmission stage in most current research has placed the focus on memotypes [51], [178], while ignoring sociotypes. In what follows, we note some similarities and differences of memetic computing to existing research pertaining to cultural algorithm [199] in the field of evolutionary computation, and transfer learning of machine learning [200].

Cultural algorithms, in particular, share commonalities to memetic computation in that both use the notion of domain-specific cultural knowledge to bias the search during problem-solving. While the former predefines an appropriate belief space representation, memetic computation on the other hand encodes high-level knowledge representation that undergoes memetic transmission, selection, replication, and/or variation. Further, memetic computing embraces sociotype transmission as a realization of meme imitation (i.e., an agent's observation of behaviors exhibited by other agents) which mirrors human behavior better and accommodates heterogeneity of agent computational representations and architectures, as opposed to the typical memotype transmission (i.e., direct copying as a form of transmission of structured knowledge among agents sharing a similar architecture).

As discussed, memes on its own is perceived as a form of structured knowledge, for example, in the form of recurrent patterns. In machine learning, transfer learning has served as a useful mechanism to promote reuse of structured knowledge or pattern that is obtained from past experiences, with the aim of improving an agent's performance on future related tasks. In the context of machine learning, in particular, transfer learning addresses the problem of insufficient training data labels, domains of disparate feature space, and different data distributions by promoting the notion of positive knowledge transfer across domains [201], etc. Thus, it is worth highlighting that transfer learning serves as a learning paradigm that could come in handy such as in the discovery, mining, and capturing of memes during the evolution. However, in contrast to the typical notion of transfer learning in machine learning and multiagent reinforcement learning, memetic computing involves the additional dimension of cultural evolution through the basic principles of memetic transmission, selection, replication, imitation, or variation, in the context of problem-solving.

VI. Concluding Remarks

Memetic computation is a broad generic framework that uses memes as units of information encoded in computational representations, and memetic algorithms as a whole is merely one aspect of the realization of memetic computation. In this survey, we outlined several aspects of memetic computation consistent with research that has been carried out in the past. We expounded on hybridization as being a foundational cornerstone for the application of memetic computation in problem-solving. This is especially relevant as the research community embarked on endeavors that resulted in many more techniques that surfaced onto the radar of optimization science in recent years. Nevertheless, debates or skepticism on the relevance or necessity of this onslaught of optimization techniques will be inevitable. As such, more productive research will be achieved by adopting a conciliatory acceptance of recent techniques that have been invented in recent years or for that matter, those that will be invented in time to come. We showed in this survey article that memes has the capacity to embrace many of the techniques known and in the context of problem-solving, memetic computation will be a useful platform to capitalize on the richness of optimization tools known to practitioners. This necessitates the availability of a flexible framework that conveniently accommodates mechanisms that engage the principle of universal Darwinism. From this perspective, multiagent computing is deemed as a framework that seamlessly espouses the notion of meme-gene coevolution. It is suggestive of the fact that an agent will serve as a powerful embodiment of mechanisms associated with meme-gene coevolution. This in our mind will be a strong supportive pillar of future memetic computation research.

References

[1] K. C. Tan, E. F. Khor, and T. H. Lee, *Multiobjective Evolutionary Algorithms and Applications (Advanced Information and Knowledge Processing)*. New York: Springer-Verlag, 2005.
[2] X. S. Yang, *Nature-Inspired Metaheuristic Algorithms*. Frome, U.K.: Luniver Press, 2008.
[3] Y. Jin and J. Branke, "Evolutionary optimization in uncertain environments: A survey," *IEEE Trans. Evol. Comput.*, vol. 9, no. 3, pp. 303–317, Jun. 2005.
[4] K. Miettinen, *Evolutionary Algorithms in Engineering and Computer Science: Recent Advances in Genetic Algorithms, Evolution Strategies, Evolutionary Programming, GE*. New York: Wiley, 1999.
[5] Y. Petalas, K. Parsopoulos, and M. Vrahatis, "Memetic particle swarm optimization," *Ann. Oper. Res.*, vol. 156, no. 1, pp. 99–127, 2007.
[6] E. Elbeltagi, T. Hegazy, and D. Grierson, "Comparison among five evolutionary-based optimization algorithms," *Adv. Eng. Inform.*, vol. 19, no. 1, pp. 43–53, 2005.
[7] Y. Jin, *Multi-Objective Machine Learning*. Berlin, Germany: Springer, 2006.
[8] M. Mitchell, *An Introduction to Genetic Algorithms (Complex Adaptive Systems)*. Cambridge, MA: MIT Press, 1998.
[9] H. G. Beyer and H. P. Schwefel, "Evolution strategies: A comprehensive introduction," *Natural Comput.*, vol. 1, no. 1, pp. 3–52, 2002.
[10] L. J. Fogel, *Intelligence Through Simulated Evolution: Forty Years of Evolutionary Programming*. New York: Wiley, 1999.
[11] R. Poli, W. Langdon, and N. McPhee. (2008). *A Field Guide to Genetic Programming* [Online]. Available: http://lulu.com
[12] K. Price, R. Storn, and J. Lampinen, *Differential Evolution: A Practical Approach to Global Optimization* (Natural Computing Series). Berlin, Germany: Springer-Verlag, 2005.
[13] P. Larranaga, *Estimation of Distribution Algorithms. A New Tool for Evolutionary Computation*. Boston, MA: Kluwer Academic, 2002, pp. 25–54.
[14] M. Dorigo and T. Stützle, *Ant Colony Optimization*. Cambridge, MA: MIT Press, 2004.
[15] J. Kennedy and R. C. Eberhart, *Swarm Intelligence*. San Francisco, CA: Morgan Kaufmann Publishers, 2001.
[16] D. Dasgupta, "Advances in artificial immune systems," *IEEE Comput. Intell. Mag.*, vol. 1, no. 4, pp. 40–49, Nov. 2006.
[17] 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. Part B: Cybern.*, vol. 36, no. 1, pp. 141–152, Feb. 2006.
[18] E. Özcan, B. Bilgin, and E. E. Korkmaz, "A comprehensive analysis of hyper-heuristics," *Intell. Data Anal.*, vol. 12, no. 1, pp. 3–23, 2008.
[19] N. Krasnogor and S. Gustafson, "A study on the use of self-generation in memetic algorithms," *Natural Comput.*, vol. 3, no. 1, pp. 53–76, 2004.
[20] A. Caponio and F. Neri, "Integrating cross-dominance adaptation in multiobjective memetic algorithms," in *Studies in Computational Intelligence*, vol. 171. 2009, pp. 325–351.
[21] P. A. N. Bosman and E. D. De Jong, "Combining gradient techniques for numerical multi-objective evolutionary optimization," in *Proc. Genet. Evol. Comput. Conf.*, vol. 1. 2006, pp. 627–634.
[22] S. Blackmore, *The Meme Machine*. New York: Oxford University Press, 1999.
[23] Y. S. Ong, M. H. Lim, and X. Chen, "Memetic computation: Past, present and future," *IEEE Comput. Intell. Mag.*, vol. 5, no. 2, pp. 24–31, May 2010.
[24] R. Dawkins, *The Selfish Gene*. Oxford, U.K.: Oxford University Press, 1976.
[25] Y. S. Ong, M. H. Lim, F. Neri, and H. Ishibuchi, "Special issue on emerging trends in soft computing: Memetic algorithm," *Soft Comput. J.*, vol. 13, nos. 8–9, pp. 1–2, Mar. 2009.

[26] P. Moscato, "On evolution, search, optimization, genetic algorithms and martial arts: Toward memetic algorithms," Tech. Rep. Caltech Concurrent Computation Program, California Instit. Technol., Pasadena, CA, Tech. Rep. 826, 1989.

[27] H. Ishibuchi, T. Yoshida, and T. Murata, "Balance between genetic search and local search in memetic algorithms for multiobjective permutation flowshop scheduling," *IEEE Trans. Evol. Comput.*, vol. 7, no. 2, pp. 204–223, Apr. 2003.

[28] T. Ke, M. Yi, and Y. Xin, "Memetic algorithm with extended neighborhood search for capacitated arc routing problems," *IEEE Trans. Evol. Comput.*, vol. 13, no. 5, pp. 1151–1166, Oct. 2009.

[29] P. P. Repoussis, C. D. Tarantilis, and G. Ioannou, "Arc-guided evolutionary algorithm for the vehicle routing problem with time windows," *IEEE Trans. Evol. Comput.*, vol. 13, no. 3, pp. 624–647, Jun. 2009.

[30] Y. S. Ong, P. B. Nair, and A. J. Keane, "Evolutionary optimization of computationally expensive problems via surrogate modeling," *Am. Instit. Aeronaut. Astronaut. J.*, vol. 41, no. 4, pp. 687–696, 2003.

[31] 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.

[32] C. K. Goh and K. C. Tan, "Evolving the tradeoffs between Pareto-optimality and robustness in multiobjective evolutionary algorithms," in *Evolutionary Computation in Dynamic and Uncertain Environments*. Berlin/Heidelberg, Germany: Springer, 2007, pp. 457–478.

[33] J. Knowles, D. Corne, and K. Deb, *Multiobjective Problem Solving From Nature: From Concepts to Applications*. Berlin, Germany: Springer, 2008.

[34] C. K. Goh, Y. S. Ong, and K. C. Tan, *Multi-Objective Memetic Algorithms*. Berlin/Heidelberg, Germany: Springer, 2009.

[35] J. M. Richer, A. Goëffon, and J. K. Hao, "A memetic algorithm for phylogenetic reconstruction with maximum parsimony," in *Proc. Eur. Conf. Evol. Comput. Mach. Learning Data Mining Bioinformatics*, 2009, pp. 164–175.

[36] C. A. Gallo, J. A. Carballido, and I. Ponzoni, "Bihea: A hybrid evolutionary approach for microarray biclustering," in *Proc. Brazilian Symp. Bioinformatics Adv. Bioinformatics Comput. Biol.*, 2009, pp. 36–47.

[37] Q. C. Nguyen, Y. S. Ong, and J. L. Kuo, "A hierarchical approach to study the thermal behavior of protonated water clusters H+(H2O)n," *J. Chem. Theory Comput.*, vol. 5, no. 10, pp. 2629–2639, 2009.

[38] H. Ishibuchi and T. Yamamoto, "Fuzzy rule selection by multiobjective genetic local search algorithms and rule evaluation measures in data mining, Fuzzy sets and systems," *Genet. Fuzzy Syst. New Develop.*, vol. 141, no. 1, pp. 59–88, 2004.

[39] K. Morikawa, S. Ozawa, and S. Abe, "Tuning membership functions of kernel Fuzzy classifiers by maximizing margins," *Memetic Comput. J.*, vol. 1, no. 3, pp. 221–228, 2009.

[40] 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. Part B Cybern.*, vol. 37, no. 1, pp. 28–41, Feb. 2007.

[41] K. C. Tan, E. J. Teoh, Q. Yu, and K. C. Goh, "A hybrid evolutionary algorithm for attribute selection in data mining," *Expert Syst. Appl.*, vol. 36, no. 4, pp. 8616–8630, 2009.

[42] R. Ruiz-Torrubiano and A. Suarez, "Hybrid approaches and dimensionality reduction for portfolio selection with cardinality constraints," *IEEE Comput. Intell. Mag.*, vol. 5, no. 2, pp. 92–107, May 2010.

[43] C. K. Ting, W. M. Zeng, and T. C. Lin, "Linkage discovery through data mining," *IEEE Comput. Intell. Mag.*, vol. 5, no. 1, pp. 10–13, Feb. 2010.

[44] Y. S. Ong, P. B. Nair, and K. Y. Lum, "Max-min surrogate-assisted evolutionary algorithm for robust design," *IEEE Trans. Evol. Comput.*, vol. 10, no. 4, pp. 392–404, Aug. 2006.

[45] Y. Jin, "A comprehensive survey of fitness approximation in evolutionary computation," *Soft Comput. J.*, vol. 9, no. 1, pp. 3–12, 2005.

[46] Y. S. Ong, N. Krasnogor, and H. Ishibuchi, "Special issue on memetic algorithms," *IEEE Trans. Syst. Man Cybern. Part B Cybern.*, vol. 37, no. 1, pp. 2–5, Feb. 2007.

[47] M. H. Lim, S. Gustafson, N. Krasnogor, and Y. S. Ong, "Editorial to the first issue," *Memetic Comput. J.*, vol. 1, no. 1, pp. 1–2, 2009.

[48] C. Coello Coello, G. Pulido, and E. Montes, "Current and future research trends in evolutionary multiobjective optimization," in *Information Processing with Evolutionary Algorithms* (Advanced Information and Knowledge Processing). London, U.K.: Springer, 2005, pp. 213–231.

[49] A. S. S. Ullah, R. Sarker, D. Cornforth, and C. Lokan, "An agent-based memetic algorithm (AMA) for solving constrained optimization problems," in *Proc. IEEE Congr. Evol. Comput.*, Sep. 2007, pp. 999–1006.

[50] S. Elfwing, E. Uchibe, K. Doya, and H. I. Christensen, "Evolutionary development of hierarchical learning structures," *IEEE Trans. Evol. Comput.*, vol. 11, no. 2, pp. 249–264, Apr. 2007.

[51] A. M. Farahmand, M. N. Ahmadabadi, C. Lucas, and B. N. Araabi, "Interaction of culture-based learning and cooperative co-evolution and its application to automatic behavior-based system design," *IEEE Trans. Evol. Comput.*, vol. 14, no. 1, pp. 23–57, Feb. 2010.

[52] A. H. Tan, L. Ning, and X. Dan, "Integrating temporal difference methods and self-organizing neural networks for reinforcement learning with delayed evaluative feedback," *IEEE Trans. Neural Netw.*, vol. 19, no. 2, pp. 230–244, Feb. 2008.

[53] H. Wang, D. Wang, and S. Yang, "A memetic algorithm with adaptive hill climbing strategy for dynamic optimization problems," *Soft Comput. J.*, vol. 13, nos. 8–9, pp. 763–780, 2009.

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

[55] S. Dietzold, J. Unbehauen, and S. Auer, "xOperator: An extensible semantic agent for instant messaging networks," in *The Semantic Web: Research and Applications*. Berlin/Heidelberg, Germany: Springer, 2008, pp. 787–791.

[56] R. Meuth, M. H. Lim, Y. S. Ong, and D. C. Wunsch, "A proposition on memes and meta-memes in computing for higher-order learning," *Memetic Comput. J.*, vol. 1, no. 2, pp. 85–100, 2009.

[57] G. N. Yannakakis, J. Levine, and J. Hallam, "Emerging cooperation with minimal effort: Rewarding over mimicking," *IEEE Trans. Evol. Comput.*, vol. 11, no. 3, pp. 382–396, Jun. 2007.

[58] E. Lamma, F. Riguzzi, and L. M. Pereira, "Belief revision via Lamarckian evolution," *New Generation Comput.*, vol. 21, no. 3, pp. 247–275, 2003.

[59] M. Ollis, W. H. Huang, and M. Happold, "A Bayesian approach to imitation learning for robot navigation," in *Proc. IEEE/RSJ Int. Conf. Intell. Robots Syst.*, Oct.–Nov. 2007, pp. 709–714.

[60] N. Krasnogor and J. Smith, "A tutorial for competent memetic algorithms: Model, taxonomy, and design issues," *IEEE Trans. Evol. Comput.*, vol. 9, no. 5, pp. 474–488, Oct. 2005.

[61] Y. S. Ong and A. J. Keane, "A domain knowledge based search advisor for design problem solving environments," *Eng. Applicat. Artif. Intell.*, vol. 15, no. 1, pp. 105–116, 2002.

[62] K. Deb, A. Pratap, S. Agarwal, and T. Meyarivan, "A fast and elitist multiobjective genetic algorithm: NSGA-II," *IEEE Trans. Evol. Comput.*, vol. 6, no. 2, pp. 182–197, Apr. 2002.

[63] J. D. Knowles and D. W. Corne, "M-PAES: A memetic algorithm for multiobjective optimization," in *Proc. IEEE Congr. Evol. Comput.*, vol. 1. 2000, pp. 325–332.

[64] D. Liu, K. C. Tan, C. K. Goh, and W. K. Ho, "A multiobjective memetic algorithm based on particle swarm optimization," *IEEE Trans. Syst. Man Cybern. Part B Cybern.*, vol. 37, no. 1, pp. 42–50, Feb. 2007.

[65] H. Ishibuchi, N. Tsukamoto, and Y. Nojima, "Diversity improvement by non-geometric binary crossover in evolutionary multiobjective optimization," *IEEE Trans. Evol. Comput.*, vol. 14, no. 6, pp. 985–998, Dec. 2010.

[66] Q. Zhang and H. Li, "MOEA/D: A multiobjective evolutionary algorithm based on decomposition," *IEEE Trans. Evol. Comput.*, vol. 11, no. 6, pp. 712–731, Dec. 2007.

[67] B. Soylu and M. Koksalan, "A favorable weight-based evolutionary algorithm for multiple criteria problems," *IEEE Trans. Evol. Comput.*, vol. 14, no. 2, pp. 191–205, Apr. 2010.

[68] Y. G. Woldesenbet, G. G. Yen, and B. G. Tessema, "Constraint handling in multiobjective evolutionary optimization," *IEEE Trans. Evol. Comput.*, vol. 13, no. 3, pp. 514–525, Jun. 2009.

[69] H. Ishibuchi, Y. Hitotsuyanagi, and Y. Nojima, "An empirical study on the specification of the local search application probability in multiobjective memetic algorithms," in *Proc. IEEE Congr. Evol. Comput.*, Sep. 2007, pp. 2788–2795.

[70] C. A. Floudas, *Deterministic Global Optimization: Theory, Methods and Application*. New York: Springer-Verlag, 2005.

[71] J. C. Spall, *Introduction to Stochastic Search and Optimization*. New York: Wiley, 2003.

[72] C. Blum, C. Cotta, A. J. Fernández, J. E. Gallardo, and M. Mastrolilli, "Hybridizations of metaheuristics with branch and bound derivates," in *Hybrid Metaheuristics*. Berlin, Germany: Springer, 2008, pp. 85–116.

[73] A. Caponio, F. Neri, and V. Tirronen, "Super-fit control adaptation in memetic differential evolution frameworks," *Soft Comput. J.*, vol. 13, nos. 8–9, pp. 811–831, 2009.

[74] J. E. Gallardo, C. Cotta, and A. J. Fernández, "Finding low autocorrelation binary sequences with memetic algorithms," *Appl. Soft Comput.*, vol. 9, no. 4, pp. 1252–1262, 2009.

[75] F. Neri, N. Kotilainen, and M. Vapa, "A memetic-neural approach to discover resources in P2P networks," in *Recent Advances in Evolutionary Computation for Combinatorial Optimization*, vol. 153. Berlin, Germany: Springer, 2008, pp. 113–129.

[76] H. Ishibuchi and T. Murata, "A multiobjective genetic local search algorithm and its application to flowshop scheduling," *IEEE Trans. Syst. Man Cybern. Part C Applicat. Rev.*, vol. 28, no. 3, pp. 392–403, Aug. 1998.

[77] K. C. Tan, T. H. Lee, and E. F. Khor, "Evolutionary algorithms with dynamic population size and local exploration for multiobjective optimization," *IEEE Trans. Evol. Comput.*, vol. 5, no. 6, pp. 565–588, Dec. 2001.

[78] A. Lara, G. Sanchez, C. A. C. Coello, and O. Schütze, "HCS: A new local search strategy for memetic multiobjective evolutionary algorithms," *IEEE Trans. Evol. Comput.*, vol. 14, no. 1, pp. 112–132, Feb. 2010.

[79] Y. Tenne and S. W. Armfield, "A memetic algorithm using a trust-region derivative-free optimization with quadratic modeling for optimization of expensive and noisy black-box functions," in *Studies in Computational Intelligence*, vol. 51. Berlin, Germany: Springer, 2007, pp. 389–415.

[80] Y. S. Ong, P. B. Nair, A. J. Keane, and K. W. Wong, "Surrogate-assisted evolutionary optimization frameworks for high-fidelity engineering design problems," in *Knowledge Incorporation in Evolutionary Computation* (Studies in Fuzziness and Soft Computing Series). Berlin, Germany: Springer-Verlag, 2004, pp. 307–331.

[81] S. D. Handoko, C. K. Kwoh, and Y. S. Ong, "Using classification for constrained memetic algorithm: A new paradigm," in *Proc. IEEE Int. Conf. Syst. Man Cybern.*, Oct. 2008, pp. 547–552.

[82] Q. Zhang, W. Liu, E. Tsang, and B. Virginas, "Expensive multiobjective optimization by MOEA/D with Gaussian process model," *IEEE Trans. Evol. Comput.*, vol. 14, no. 3, pp. 456–474, Jun. 2010.

[83] S. Rahnamayan, H. R. Tizhoosh, and M. M. A. Salama, "A novel population initialization method for accelerating evolutionary algorithms," *Comput. Math. Applicat.*, vol. 53, no. 10, pp. 1605–1614, 2007.

[84] R. G. Kang and C. Y. Jung, "The optimal solution of TSP using the new mixture initialization and sequential transformation method in genetic algorithm," in *Proc. PRICAI: Trends Artif. Intell.*, LNCS 4099. 2006, pp. 1181–1185.

[85] R. W. Morrison, "Dispersion-based population initialization," in *Proc. Genet. Evol. Comput. Conf.*, vol. 2723. 2003, pp. 1210–1221.

[86] A. Guillén, H. Pomares, J. González, I. Rojas, L. J. Herrera, and A. Prieto, "Parallel multiobjective memetic RBFNNs design and feature selection for function approximation problems," in *Proc. IWANN*, LNCS 4507. 2007, pp. 341–350.

[87] G. W. Klau, I. Ljubic, A. Moser, P. Mutzel, P. Neuner, U. Pferschy, G. Raidl, and R. Weiskircher, "Combining a memetic algorithm with integer programming to solve the prize-collecting Steiner tree problem," in *Proc. Genet. Evol. Comput. Conf.*, 2004, pp. 1304–1315.

[88] Q. Zhang, J. Sun, G. Xiao, and E. Tsang, "Evolutionary algorithms refining a heuristic: A hybrid method for shared-path protections in WDM networks under SRLG constraints," *IEEE Trans. Syst. Man Cybern. Part B: Cybern.*, vol. 37, no. 1, pp. 51–61, Feb. 2007.

[89] Z. Skolicki and K. D. Jong, "The influence of migration sizes and intervals on island models," in *Proc. Genet. Evol. Comput. Conf.*, 2005, pp. 1295–1302.

[90] M. Mezmaz, N. Melab, and E. G. Talbi, "Combining metaheuristics and exact methods for solving exactly multiobjective problems on the grid," *J. Math. Modeling Algorithms*, vol. 6, no. 3, pp. 393–409, 2007.

[91] T. Park and K. R. Ryu, "A dual-population genetic algorithm for adaptive diversity control," *IEEE Trans. Evol. Comput.*, vol. 14, no. 6, pp. 865–884, Dec. 2010.

[92] S. Yang and X. Yao, "Population-based incremental learning with associative memory for dynamic environments," *IEEE Trans. Evol. Comput.*, vol. 12, no. 5, pp. 542–561, Oct. 2008.

[93] J. Tang, M. H. Lim, and Y. S. Ong, "Diversity-adaptive parallel memetic algorithm for solving large scale combinatorial optimization problems," *Soft Comput. J.*, vol. 11, no. 9, pp. 873–888, 2007.

[94] J. M. Baldwin, "A new factor in evolution," in *Adaptive Individuals in Evolving Populations: Models and Algorithms*. Reading, MA: Addison-Wesley Longman Publishing, 1996, pp. 59–80.

[95] G. Mayley, "The evolutionary cost of learning," in *Proc. 4th Int. Conf. Simul. Adaptive Behav.*, 1996, pp. 458–467.

[96] T. Sasaki and M. Tokoro, "Comparison between Lamarckian and Darwinian evolution on a model using neural networks and genetic algorithms," *Knowl. Inform. Syst.*, vol. 2, no. 2, pp. 201–222, 2000.

[97] L. D. Whitley, V. S. Gordon, and K. E. Mathias, "Lamarckian evolution, the Baldwin effect and function optimization," in *Proc. 3rd Int. Conf. PPSN*, 1994, pp. 6–15.

[98] T. El-Mihoub, A. Hopgood, L. Nolle, and A. Battersby, "Self-adaptive Baldwinian search in hybrid genetic algorithms," in *Computational Intelligence, Theory and Applications*. Berlin/Heidelberg, Germany: Springer, 2006, pp. 597–602.

[99] T. Sasaki and M. Tokoro, "Adaptation toward changing environments: Why Darwinian in nature," in *Proc. Eur. Conf. Artif. Life*, 1997, pp. 145–153.

[100] S. Bernhard and K. Martin, "A model for the dynamic interaction between evolution and learning," *Neural Process. Lett.*, vol. 10, no. 3, pp. 181–193, 1999.

[101] I. Paenke, B. Sendhoff, J. Rowe, and C. Fernando, "On the adaptive disadvantage of Lamarckianism in rapidly changing environments," in *Advances in Artificial Life*. 2007, pp. 355–364.

[102] H. Ishibuchi, S. Kaige, and K. Narukawa, "Comparison between Lamarckian and Baldwinian repair on multiobjective 0/1 knapsack problems," in *Evolutionary Multi-Criterion Optimization* (Lecture Notes in Computer Science). Berlin/Heidelberg, Germany: Springer, 2005, pp. 370–385.

[103] S. S. Choi and B. R. Moon, "A graph-based Lamarckian-Baldwinian hybrid for the sorting network problem," *IEEE Trans. Evol. Comput.*, vol. 9, no. 1, pp. 105–114, Feb. 2005.

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

[105] F. Pereira, J. Tavares, P. Machado, and E. Costa, "GVR: A new genetic representation for the vehicle routing problem," in *Proc. Irish Conf. Artif. Intell. Cognitive Sci.*, 2002, pp. 285–320.

[106] R. G. Kang and C. Y. Jung, "The improved initialization method of genetic algorithm for solving the optimization problem," in *Neural Information Processing* (Lecture Notes in Computer Science). Berlin/Heidelberg, Germany: Springer, 2006, pp. 789–796.

[107] S. F. Adra, T. J. Dodd, I. A. Griffin, and P. J. Fleming, "Convergence acceleration operator for multiobjective optimization," *IEEE Trans. Evol. Comput.*, vol. 13, no. 4, pp. 825–847, Aug. 2009.

[108] J. Schönberger, "Memetic algorithm vehicle routing," in *Operational Freight Carrier Planning Basic Concepts, Optimization Models and Advanced Memetic Algorithms*. Berlin/Heidelberg, Germany: Springer, 2005, pp. 65–76.

[109] J. Montgomery, C. Fayad, and S. Petrovic, "Solution representation for job shop scheduling problems in ant colony optimization," in *Ant Colony Optimization and Swarm Intelligence* (Lecture Notes in Computer Science). Berlin/Heidelberg, Germany: Springer, 2006, pp. 484–491.

[110] Y. W. Wang, C. H. Liu, and C. Y. Fan, "Develop a sub-population memetic algorithm for multiobjective scheduling problems," in *Proc. 2nd ICCAE*, 2010, pp. 579–583.

[111] G. Gutin and D. Karapetyan, "Generalized traveling salesman problem reduction algorithms," *Algorithmic Oper. Res.*, vol. 4, no. 2, pp. 144–154, 2008.

[112] H. Ishibuchi and T. Yamamoto, "Fuzzy rule selection by multiobjective genetic local search algorithms and rule evaluation measures in data mining," *Fuzzy Sets Syst.*, vol. 141, no. 1, pp. 59–88, 2004.

[113] S. Kersting, G. Raidl, and I. Ljubiæ, "A memetic algorithm for vertex-biconnectivity augmentation," in *Applications of Evolutionary Computing* (Lecture Notes in Computer Science). Berlin/Heidelberg, Germany: Springer, 2002, pp. 102–111.

[114] G. B. Alvarenga, G. R. Mateus, and G. D. Tomi, "Finding near optimal solutions for vehicle routing problems with time windows using hybrid genetic algorithm," in *Proc. Int. Workshop Freight Transportation*, 2003, p. 113.

[115] T. Witkowski, S. Elzway, A. Antczak, and P. Antczak, "Representation of solutions and genetic operators for flexible job shop problem," in *Advanced Intelligent Computing Theories and Applications. With Aspects of Contemporary Intelligent Computing Techniques* (Communications in Computer and Information Science). Berlin/Heidelberg, Germany: Springer, 2007, pp. 256–265.

[116] D. Applegate, R. Bixby, V. Chv´atal, and W. Cook, "Implementing the Dantzig–Fulkerson–Johnson algorithm for large traveling salesman problems," *Math. Programming Series B*, vol. 97, nos. 1–2, pp. 91–153, 2003.

[117] T. Yoshida and H. Mori, "Application of multiobjective memetic algorithm with solution diversity to probabilistic distribution network expansion planning," *IEEJ Trans. Power Energy*, vol. 129, no. 12, pp. 1495–1502, 2009.

[118] M. D. Schmidt and H. Lipson, "Coevolution of fitness predictors," *IEEE Trans. Evol. Comput.*, vol. 12, no. 6, pp. 736–749, Dec. 2008.

[119] S. H. Chen, P. C. Chang, Q. Zhang, and C. B. Wang, "A guided memetic algorithm with probabilistic models," *Int. J. Innovative Comput. Inform. Control*, vol. 5, no. 12, pp. 4753–4764, 2009.

[120] Y. Tenne and S. W. Armfield, "A versatile surrogate-assisted memetic algorithm for optimization of computationally expensive functions and its engineering applications," in *Studies in Computational Intelligence*, vol. 92. Berlin/Heidelberg, Germany: Springer, 2008, pp. 43–72.

[121] Z. Zhou, Y. S. Ong, P. B. Nair, A. J. Keane, and K. Y. Lum, "Combining global and local surrogate models to accelerate evolutionary optimization," *IEEE Trans. Syst. Man Cybern. Part C Applicat. Rev.*, vol. 37, no. 1, pp. 66–76, Jan. 2007.

[122] D. Lim, Y. Jin, Y. S. Ong, and B. Sendhoff, "Generalizing surrogate-assisted evolutionary computation," *IEEE Trans. Evol. Comput.*, vol. 14, no. 3, pp. 329–355, Jun. 2010.

[123] Y. Tenne and S. Armfield, "A framework for memetic optimization using variable global and local surrogate models," *Soft Comput. J.*, vol. 13, no. 8, pp. 781–793, 2009.

[124] N. M. Alexandrov, J. E. Dennis, R. M. Lewis, and V. Torczon, "A trust-region framework for managing the use of approximation models in optimization," *Struct. Multidisciplinary Optimization*, vol. 15, no. 1, pp. 16–23, 1998.

[125] A. Ratle, "Kriging as a surrogate fitness landscape in evolutionary optimization," *Artif. Intell. Eng. Design Anal. Manuf.*, vol. 15, no. 1, pp. 37–49, 2001.

[126] M. N. Le, Y. S. Ong, and Q. H. Nguyen, "Optinformatics for schema analysis of binary genetic algorithms," in *Proc. 10th Annu. Conf. GECCO*, 2008, pp. 1121–1122.

[127] S. D. Handoko, C. K. Kwoh, and Y. S. Ong, "Feasibility structure modeling: An effective chaperon for constrained memetic algorithms," *IEEE Trans. Evol. Comput.*, vol. 14, no. 5, pp. 740–758, Oct. 2009.

[128] S. García, J. R. Cano, and F. Herrera, "A memetic algorithm for evolutionary prototype selection: A scaling up approach," *Pattern Recognit.*, vol. 41, no. 8, pp. 2693–2709, 2008.

[129] E. Burke, S. Gustafson, G. Kendall, and N. Krasnogor, "Advanced population diversity measures in genetic programming," in *Proc. 7th PPSN*, 2002, pp. 341–350.

[130] 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.*, Sep. 2007, pp. 2374–2381.

[131] V. Cutello, N. Krasnogor, G. Nicosia, and M. Pavone, "Immune algorithm versus differential evolution: A comparative case study using high dimensional function optimization," in *Proc. Adaptive Natural Comput. Algorithms*, 2007, pp. 93–101.

[132] V. Tirronen, F. Neri, T. Karkkainen, K. Majava, and T. Rossi, "A memetic differential evolution in filter design for defect detection in paper production," in *Proc. EvoWorkshops EvoCoMnet EvoFIN EvoIASP EvoINTERACTION EvoMUSART EvoSTOC EvoTransLog: Applicat. Evol. Comput.*, 2007, pp. 320–329.

[133] S. Yang and C. Li, "A clustering particle swarm optimizer for locating and tracking multiple optima in dynamic environments," *IEEE Trans. Evol. Comput.*, vol. 14, no. 6, pp. 959–974, Dec. 2010.

[134] Y. G. Woldesenbet and G. G. Yen, "Dynamic evolutionary algorithm with variable relocation," *IEEE Trans. Evol. Comput.*, vol. 13, no. 3, pp. 500–513, Jun. 2009.

[135] O. Kramer and H. Danielsiek, "DBSCAN-based multiobjective niching to approximate equivalent Pareto-subsets," in *Proc. GECCO*, 2010, pp. 503–510.

[136] J. Tang, M. H. Lim, and Y. S. Ong, "Adaptation for parallel memetic algorithm based on population entropy," in *Proc. 8th Annu. Conf. GECCO*, 2006, pp. 575–582.

[137] K. Sorensen and M. Sevaux, "MA|PM: Memetic algorithms with population management," in *Proc. Comput. Oper. Res.*, vol. 33. 2006, pp. 1214–1225.

[138] D. Sudholt, "Local search in evolutionary algorithms: The impact of the local search frequency," in *Algorithms and Computation* (Lecture Notes in Computer Science). Berlin/Heidelberg, Germany: Springer, 2006, pp. 359–368.

[139] W. E. Hart, "Adaptive global optimization with local search," Ph.D. dissertation, Dept. Comput. Sci. Eng., Univ. California, San Diego, 1994.

[140] Q. H. Nguyen, Y. S. Ong, M. H. Lim, and N. Krasnogor, "Adaptive cellular memetic algorithms," *Evol. Comput.*, vol. 17, no. 2, pp. 231–256, 2009.

[141] Q. H. Nguyen, Y. S. Ong, M. H. Lim, and N. Krasnogor, "A study on the design issues of memetic algorithm," in *Proc. IEEE Congr. Evol. Comput.*, Sep. 2007, pp. 2390–2397.

[142] S. Droste, T. Jansen, and I. Wegener, "On the analysis of the (1+ 1) evolutionary algorithm," *Theor. Comput. Sci.*, vol. 276, nos. 1–2, pp. 51–81, 2002.

[143] D. Molina, "Adaptive local search parameters for real-coded memetic algorithm," in *Proc. IEEE Congr. Evol. Comput.*, Sep. 2005, pp. 888–895.

[144] D. Molina, M. Lozano, and F. Herrera, "Memetic algorithm with local search chaining for continuous optimization problems: A scalability test," in *Proc. 9th Int. Conf. Intell. Syst. Des. Applicat.*, 2009, pp. 830–837.

[145] D. Molina, M. Lozano, and F. Herrera, "MA-SW-chains: Memetic algorithm based on local search chains for large scale continuous global optimization," in *Proc. IEEE Congr. Evol. Comput.*, Jul. 2010, pp. 1–8.

[146] N. Noman and H. Iba, "Accelerating differential evolution using an adaptive local search," *IEEE Trans. Evol. Comput.*, vol. 12, no. 1, pp. 107–125, Feb. 2008.

[147] 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.

[148] A. Sinha, Y. P. Chen, and D. E. Goldberg, "Designing efficient genetic and evolutionary algorithm hybrids," in *Recent Advances in Memetic Algorithms* (Studies in Fuzziness and Soft Computing). Berlin/Heidelberg, Germany: Springer, 2005, pp. 259–288.

[149] P. Merz, "NK-fitness landscapes and memetic algorithms with greedy operators and k-opt local search," in *Proc. Recent Adv. Memetic Algorithms*, 2005, pp. 209–228.

[150] S. S. Choi, Y. K. Kwon, and B. R. Moon, "Properties of symmetric fitness functions," *IEEE Trans. Evol. Comput.*, vol. 11, no. 6, pp. 743–757, Dec. 2007.

[151] M. N. Le and Y. S. Ong, "A frequent pattern mining algorithm for understanding genetic algorithms," in *Proc. 4th Int. Conf. Intell. Comput. Adv. Intell. Comput. Theories Applicat. Aspects Artif. Intell.*, 2008, pp. 131–139.

[152] M. N. Le, Y. S. Ong, Y. Jin, and B. Sendhoff, "Lamarckian memetic algorithms: Local optimum and connectivity structure analysis," *Memetic Comput. J.*, vol. 1, no. 3, pp. 175–190, 2009.

[153] S. Verel, G. Ochoa, and M. Tomassini, "Local optima networks of NK landscapes with neutrality," *IEEE Trans. Evol. Comput.*, vol. 14, no. 6, p. 1, 2010.

[154] S. S. Choi, K. Jung, and B. R. Moon, "Lower and upper bounds for linkage discovery," *IEEE Trans. Evol. Comput.*, vol. 13, no. 2, pp. 201–216, Apr. 2009.

[155] R. Niedermeier, *Invitation to Fixed Parameter Algorithms*. Oxford, U.K.: Oxford University Press, 2006.

[156] I. Paenke, J. Branke, and Y. Jin, "On the influence of phenotype plasticity on genotype diversity," in *Proc. IEEE Symp. Found. Comput. Intell.*, Apr. 2007, pp. 33–41.

[157] I. Paenke, T. J. Kawecki, and B. Sendhoff, "The influence of learning on evolution: A mathematical framework," *Artif. Life*, vol. 15, no. 2, pp. 227–245, 2009.

[158] E. Borenstein, I. Meilijson, and E. Ruppin, "The effect of phenotypic plasticity on evolution in multipeaked fitness landscapes," *J. Evol. Biol.*, vol. 19, no. 5, pp. 1555–1570, 2006.

[159] N. Krasnogor and J. Smith, "Memetic algorithms: The polynomial local search complexity theory perspective," *J. Math. Modeling Algorithms*, vol. 7, no. 1, pp. 3–24, 2008.

[160] Y. Kameya, J. Kumagai, and Y. Kurata, "Accelerating genetic programming by frequent subtree mining," in *Proc. 10th Annu. Conf. Genet. Evol. Comput.*, 2008, pp. 1203–1210.

[161] R. Mills and R. A. Watson, "On crossing fitness valleys with the Baldwin effect," in *Proc. 10th Int. Conf. Simul. Synthesis Living Syst.*, 2006, pp. 493–499.

[162] D. Vernon, G. Metta, and G. Sandini, "A survey of artificial cognitive systems: Implications for the autonomous development of mental capabilities in computational agents," *IEEE Trans. Evol. Comput.*, vol. 11, no. 2, pp. 151–180, Apr. 2007.

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

[164] A. Lynch, "Thought contagion as abstract evolution," *J. Ideas*, vol. 2, no. 1, pp. 3–10, 1991.
[165] J. Apesteguia, S. Huck, and J. Oechssler, "Imitation—theory and experimental evidence (experimental)," *J. Econ. Theory*, vol. 1, no. 136, pp. 217–235, 2003.
[166] B. D. Argall, S. Chernova, M. Veloso, and B. Browning, "A survey of robot learning from demonstration," *Robot. Autonomous Syst.*, vol. 57, no. 5, pp. 469–483, 2009.
[167] S. Kirby, "Spontaneous evolution of linguistic structure: An iterated learning model of the emergence of regularity and irregularity," *IEEE Trans. Evol. Comp.*, vol. 5, no. 2, pp. 102–110, Apr. 2001.
[168] G. Baldassarre, "Cultural evolution of 'guiding criteria' and behavior in a population of neural-network agents," *J. Memetics Evol. Models Inform. Transmission*, vol. 4, pp. 1–8, Mar. 2001.
[169] G. Grant, "Memetic lexicon," in *Principia Cybernetica Web*, 1990 [Online]. Available: http://pespmc1.vub.ac.be/MEMIN.html
[170] J. Liu, W. Zhong, and L. Jiao, "A multiagent evolutionary algorithm for constraint satisfaction problem," *IEEE Trans. Syst. Man Cybern. Part B*, vol. 36, no. 1, pp. 54–73, Feb. 2006.
[171] J. Liu, "Autonomy-oriented computing: The nature and implications of a paradigm for self-organized computing," in *Proc. 4th ICNC, 5th Int. Conf. FSKD*, 2008, pp. 18–20.
[172] B. E. Eskridge and D. F. Hougen, "Imitating success: A memetic crossover operator for genetic programming," in *Proc. IEEE CEC*, vol. 1. Jun. 2004, pp. 809–815.
[173] J. J. Alferes, P. Dell'Acqua, E. Lamma, I. J. A. Leite, L. M. Pereira, and F. Riguzzi, "A logic based approach to multiagent systems," *ALP Newsletter*, vol. 14, no. 3, 2001.
[174] X. Xie and J. Liu, "Multiagent optimization system for solving the traveling salesman problem (TSP)," *IEEE Trans. Syst. Man Cybern. Part B*, vol. 39, no. 2, pp. 489–502, Apr. 2009.
[175] K. Takadama, T. Terano, and K. Shimohara, "Nongovernance rather than governance in a multiagent economic society," *IEEE Trans. Evol. Comput.*, vol. 5, no. 5, pp. 535–545, Oct. 2001.
[176] L. Bull, "Coevolutionary species adaptation genetic algorithms: A continuing SAGA on coupled fitness landscapes," in *Proc. Adv. Artif. Life*, 2005, pp. 322–331.
[177] L. Bull, O. Holland, and S. Blackmore, "On meme-gene coevolution," *Artif. Life*, vol. 6, no. 3, pp. 227–235, 2000.
[178] E. Vallejo and F. Ramos, "Evolving insect locomotion using cooperative genetic programming," in *Proc. MICAI: Adv. Artif. Intell.*, LNCS 2000. pp. 170–181.
[179] C. H. Yeh and S. H. Chen, "Toward an integration of social learning and individual learning in agent-based computational stock markets: The approach based on population genetic programming," *J. Manage. Econ.*, pp. 363–393, 2001.
[180] S. Hassan, L. Garmendia, and J. Pavón, "Agent-based social modeling and simulation with fuzzy sets," in *Innovations in Hybrid Intelligent Systems* (Advances in Soft Computing). Berlin/Heidelberg, Germany: Springer, 2007, pp. 40–47.
[181] 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.
[182] J. Martins and E. Miranda, "A connectionist architecture for the evolution of rhythms," in *Applications of Evolutionary Computing* (Lecture Notes in Computer Science). Berlin/Heidelberg, Germany: Springer, 2006, pp. 696–706.
[183] L. Nunes and E. Oliveira, "Cooperative learning using advice exchange," in *Adaptive Agents and Multi-Agent Systems: Adaptation and Multi-Agent Learning*. Berlin/Heidelberg, Germany: Springer-Verlag, 2003, pp. 33–48.
[184] R. P. N. Rao, A. P. Shon, and A. N. Meltzoff, "A Bayesian model of imitation in infants and robots," in *Imitation and Social Learning in Robots, Humans, and Animals*. Cambridge, MA: Cambridge University Press, 2004, pp. 217–247.
[185] K. M. Squire and S. E. Levinson, "HMM-based concept learning for a mobile robot," *IEEE Trans. Evol. Comput.*, vol. 11, no. 2, pp. 199–212, Apr. 2007.
[186] J. Nakanishi, J. Morimoto, G. Endo, G. Cheng, S. Schaal, and M. Kawato, "Learning from demonstration and adaptation of biped locomotion," *Robot. Autonomous Syst.*, vol. 47, nos. 2–3, pp. 79–91, 2004.
[187] J. C. Bongard, "Accelerating self-modeling in cooperative robot teams," *IEEE Trans. Evol. Comput.*, vol. 13, no. 2, pp. 321–332, Apr. 2009.
[188] A. V. Kononova, K. J. Hughes, M. Pourkashanian, and D. B. Ingham, "Fitness diversity based adaptive memetic algorithm for solving inverse problems of chemical kinetics," in *Proc. IEEE CEC*, Sep. 2007, pp. 2366–2373.
[189] J. Cervantes and C. R. Stephens, "Limitations of existing mutation rate heuristics and how a rank GA overcomes them," *IEEE Trans. Evol. Comput.*, vol. 13, no. 2, pp. 369–397, Apr. 2009.
[190] H. Ishibuchi, R. Sakamoto, and T. Nakashima, "Evolution of unplanned coordination in a market selection game," *IEEE Trans. Evol. Comput.*, vol. 5, no. 5, pp. 524–534, Oct. 2001.
[191] S. Kwang Mong and W. Yuanshi, "Evolutionary asymmetric games for modeling systems of partially cooperative agents," *IEEE Trans. Evol. Comput.*, vol. 9, no. 6, pp. 603–614, Dec. 2005.
[192] B. Price and C. Boutilier, "Accelerating reinforcement learning through implicit imitation," *J. Artif. Intell. Res.*, vol. 19, no. 1, pp. 569–629, 2003.
[193] C. Alós-Ferrer, "Learning, bounded memory, and inertia," *Econ. Lett.*, vol. 101, no. 2, pp. 134–136, 2008.
[194] F. Heylighen and K. Chielens, "Cultural evolution and memetics," in *Encyclopedia of Complexity and System Science*. Berlin, Germany: Springer, 2009.
[195] T. Gong, "Exploring the roles of horizontal, vertical, and oblique transmissions in language evolution," *Adaptive Behav. Animals Animats Softw. Agents Robots Adaptive Syst.*, vol. 18, nos. 3–4, pp. 356–376, 2010.
[196] P. Y. Oudeyer, F. Kaplan, and V. V. Hafner, "Intrinsic motivation systems for autonomous mental development," *IEEE Trans. Evol. Comput.*, vol. 11, no. 2, pp. 265–286, Apr. 2007.
[197] L. Song, M. Lim, and Y. Ong, "Neural meta-memes framework for combinatorial optimization," in *Swarm, Evolutionary, and Memetic Computing* (Lecture Notes in Computer Science), B. Panigrahi, S. Das, P. Suganthan *et al.*, Eds. Berlin/Heidelberg, Germany: Springer, 2010, pp. 198–205.
[198] N. Krasnogor, "Self generating metaheuristics in bioinformatics: The proteins structure comparison case," *Genet. Programming Evolvable Mach.*, vol. 5, no. 2, pp. 181–201, 2004.
[199] R. G. Reynolds and B. Peng, "Cultural algorithms: Modeling of how cultures learn to solve problems," in *Proc. 16th IEEE ICTAI*, Nov. 2004, pp. 166–172.
[200] M. E. Taylor and P. Stone, "Transfer learning for reinforcement learning domains: A survey," *J. Mach. Learning Res.*, vol. 10, no. 1, pp. 1633–1685, 2009.
[201] C. W. Seah, I. W. Tsang, Y. S. Ong, and K. K. Lee, "Predictive distribution matching SVM for multi-domain learning," in *Proc. Eur. Conf. Mach. Learning Knowl. Discovery Databases: Part I*, 2010, pp. 231–247.

Xianshun Chen received the Bachelors degree in electrical and electronics engineering from Nanyang Technological University, Singapore. He is currently pursuing the Doctorate degree in the field of competent memetic algorithms from the School of Computer Engineering, Nanyang Technological University.

His current 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 2002.

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

Dr. Ong is the 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, and others. He is the Chair of the IEEE Computational Intelligence Society Emergent Technology Technical Committee and has served as a guest editor of several journals, including the IEEE TRANSACTIONS ON EVOLUTIONARY COMPUTATION.

Meng-Hiot Lim is currently an Associate Professor with the School of Electrical and Electronic Engineering, Nanyang Technological University, Singapore. He holds a concurrent appointment of Deputy Director with the highly regarded M.Sc. in Financial Engineering Program as well as the Center for Financial Engineering, a multi-disciplinary research center anchored at the Nanyang Business School, Singapore. His current research interests include memetic computation, particularly pertaining to aspects of memes-inspired problem-solving.

He is an Associate Editor of the IEEE TRANSACTIONS ON SYSTEMS, MAN AND CYBERNETICS—PART C. He is the Managing Editor-in-Chief of *Memetic Computing*, a journal by Springer, and is the Editor-in-Chief of a *Book Series on Adaptation, Learning, and Optimization*. He is the Founding President of the Memetic Computing Society.

Kay Chen Tan is currently an Associate Professor with the Department of Electrical and Computer Engineering, National University of Singapore, Singapore. He has published over 90 journal papers, over 100 papers in conference proceedings, and coauthored 5 books. He has been invited to be a keynote/invited speaker for 19 international conferences, and has served in the international program committees for over 100 conferences and has been involved in the organizing committees for over 30 international conferences.

He is currently a Distinguished Lecturer of the IEEE Computational Intelligence Society and the Editor-in-Chief of the IEEE COMPUTATIONAL INTELLIGENCE MAGAZINE. He serves as an associate editor/editorial board member of 15 international journals, such as the IEEE TRANSACTIONS ON EVOLUTIONARY COMPUTATION, IEEE TRANSACTIONS ON COMPUTATIONAL INTELLIGENCE AND AI IN GAMES, *Evolutionary Computation* (MIT Press), *European Journal of Operational Research*, *Journal of Scheduling*, and *International Journal of Systems Science*. He received the Recognition Award in 2008 from the International Network for Engineering Education and Research for his outstanding contributions to engineering education and research. He is currently a fellow of the NUS Teaching Academic.