Autoencoding evolutionary search with learning across heterogeneous problems

Feng, Liang; Ong, Yew-Soon; Jiang, Siwei; Gupta, Abhishek

2017

Feng, L., Ong, Y., Jiang, S. & Gupta, A. (2017). Autoencoding evolutionary search with learning across heterogeneous problems. IEEE Transactions On Evolutionary Computation, 21(5), 760-772. https://dx.doi.org/10.1109/TEVC.2017.2682274

https://hdl.handle.net/10356/147937

https://doi.org/10.1109/TEVC.2017.2682274

Autoencoding Evolutionary Search with Learning across Heterogeneous Problems

Liang Feng, Yew-Soon Ong, Siwei Jiang and Abhishek Gupta

***Abstract*—To enhance the search performance of evolutionary algorithms, reusing knowledge captured from past optimization experiences along the search process has been proposed in the literature, and demonstrated much promise. In the literature, there are generally three types of approaches for reusing knowledge from past search experiences, namely exact storage and reuse of past solutions, the reuse of model-based information, and the reuse of structured knowledge captured from past optimized solutions. In this paper, we focus on the third type of knowledge reuse for enhancing evolutionary search. In contrast to existing works, here we focus on knowledge transfer across heterogeneous continuous optimization problems with diverse properties, such as problem dimension, number of objectives, etc, that cannot be handled by existing approaches. In particular, we propose a novel autoencoding evolutionary search paradigm with learning capability across heterogeneous problems. The essential ingredient for learning structured knowledge from search experience in our proposed paradigm is a single layer denoising autoencoder, which is able to build the connections between problem domains by treating past optimized solutions as the corrupted version of the solutions for the newly encountered problem. Further, as the derived denoising autoencoder holds a closed-form solution, the corresponding reusing of knowledge from past search experiences will not bring much additional computational burden on the evolutionary search. To evaluate the proposed search paradigm, comprehensive empirical studies on the complex multi-objective optimization problems are presented, along with a real-world case study from the fibre-reinforced polymer composites manufacturing industry.**

***Index Terms*—Memetic Computation, Evolutionary Optimization, Learning, Knowledge Transfer**

I. INTRODUCTION

EVOLUTIONARY algorithms (EAs) are population-based search methods which work on Darwinian principles of natural selection or survival of the fittest [1]. Due to its strong search capability and simplicity of implementation, over the years, EA has been successfully applied to solve a variety of optimization problems in the real world, such as continuous optimization problems [2], [3], combinatorial optimization problems [4], [5], [6], [7], [8], multifactorial optimization [9], [10], etc. Despite the significant success EAs have enjoyed, it is also well known that EA involves an iterative reproduction process, which is deemed to be slow and thus restricts the practicality of EA in cases where limited computational budget is available [11], [12], [13], [14].

Liang Feng is with College of Computer Science, Chongqing University, China. E-mail: {liangf}@cqu.edu.cn

Siwei Jiang is with the Singapore Institute of Manufacturing Technology (SIMTech), Singapore. E-mail: {jiangsw}@SIMTech.a-star.edu.sg

Yew-Soon Ong and Abhishek Gupta are with the School of Computer Engineering, Nanyang Technological University (NTU), Singapore. E-mail: {asysong, abhishekg}@ntu.edu.sg

To improve the efficacy of existing EAs, reusing past search experiences from related problems has been proposed in the literature, which successfully enhanced the evolutionary search on examples such as the scheduling problem [15], traveling salesman problem [16], and arc routing problem [17]. This is because in nature, problems seldom exist in isolation, and experiences on solving previous related problems often yield useful information that when properly harnessed, can lead to more efficient problem-solving processes. Specific examples may include the exact storage and reuse of past solutions [15], [16], the reuse of model-based information [18], [19], and the reuse of structured knowledge captured from optimized solutions in past search experiences [20]. In particular, in [15], Louis *et al.* proposed to store the optimized solutions of past problems, and reuse them to aid in the genetic algorithm (GA) search via case-based reasoning. Rather than starting anew on each problem, appropriate intermediate solutions drawn from similar problems that have been previously solved are periodically injected into the GA population. In another study, Cunningham and Smyth [16] presented the direct reuse of established high quality schedules from past problems in solving traveling salesman problems. As both [15] and [16] only consider the direct insertion of past solutions, they cannot apply well on problems that bear differences in structural properties, such as problem vertex size, topological structures, representations, etc.

On the other hand, instead of reusing the exact past solutions, Martin *et al.* [18] proposed to combine pre-defined problem-specific distance metric with prior distribution mined from previous optimization experience to improve the model-directed optimization methods, e.g., estimation of distribution algorithm (EDA). Further, Roberto *et al.* [19] proposed to transfer the structural information from subproblems (previous parameter settings) to bias the construction of aggregation matrix of the EDA for solving multi-marker tagging single-nucleotide polymorphism (SNP). Next, M. Iqbal *et al.* [21] presented a study of reusing building blocks extracted from small-scale problems for more efficient problem solving on complex large-scale problems based on learning classifier system. However, since these transfer approaches are designed for model-based evolutionary optimization methods (e.g., EDA), they cannot apply with the model free evolutionary algorithms, such as genetic algorithm.

In contrast to the above mentioned two categories of approaches that reuse past search experiences for enhancing evolutionary search, the third category tries to reuse the structured common knowledge buried in the optimized solutions of past search experiences. As the structured knowledge is learned directly from the optimized solution which is independent of the solution representation, it can be reused in model-free evolutionary algorithms across problems with different size, representation, etc. In particular, in [13] and [17], a memetic computational paradigm towards intelligent evolutionary optimization that transfers the structured knowledge in the form of memes learned from previous problem-solving experiences have been presented. Taking vehicle routing and arc routing as the problem domains of study, significant improvement of evolutionary search has been observed on a variety of routing instances with different size, topologies, etc., by defining the knowledge meme as a transformation matrix captured from past optimized routing solutions. However, it is worth noting that due to the specific definition of knowledge meme, this approach can only be applied for solving combinatorial optimization problems. It will fail if continuous optimization problems are encountered, where the learning data such as problem instance, task information, etc., as required in [13], [17] is not available.

Taking this cue, our aim here is thus to embark on a study towards intelligent evolutionary search with learning capability across heterogeneous problems in continuous optimization. To the best of our knowledge, there is no or little work in the literature that considers the learning and reuse of past search experiences in model-free evolutionary search algorithms for continuous optimization problems with diverse properties, such as different problem dimensions, number of objectives, etc. In particular, in this paper, we propose an autoencoding evolutionary search paradigm which is able to derive knowledge in the form of problem solutions from past search experiences that can be injected into the current population while the search progresses. The essential ingredient of the learning component in our proposed search paradigm is a single layer denoising autoencoder derived from its conventional counterpart [22], which holds a closed-form solution that will not bring much computational burden to the evolutionary solver. Further, to evaluate the efficacy of the proposed memetic search paradigm, comprehensive empirical studies are first conducted on the complex multi-objective continuous optimization problems where prior guidance is useful for enhanced search performance, and then on a real-world case study from the fibre-reinforced polymer composites manufacturing industry.

The rest of this paper is organized as follows. A brief introduction of the deep learning autoencoder and its variant denoising autoencoder is introduced in Section II. A discussion and overview of memetic computation is also provided in the section. Further, Section III presents the proposed evolutionary search paradigm, including the theoretical derivation of the single layer denoising autoencoder and the detailed design of the search paradigm. Comprehensive empirical studies on the multi-objective benchmark problems is provided in Section IV. This is followed by a demonstration of the real-world efficacy of our proposition, with a case study on fibre-reinforced polymer composites manufacturing in Section V. Lastly, the concluding remarks of this paper and key directions for future research are discussed in Section VI.

II. Preliminary

This section begins with a brief introduction of the conventional denoising autoencoder, which serves as the basis for further derivation of the single layer denoising autoencoder in the proposed search paradigm. Subsequently, an overview of memetic computation is presented to highlight the contribution of this paper.

A. Denoising Autoencoder

Recent advances in machine learning, coupled with huge volumes of data being collected from a wide variety of clients, such as mobile devices, Internet applications, etc., have resulted in the emergence of deep learning as a method to learn a new representation and uncover the corresponding hidden structure in these ever growing data sets [23], [24], [25].

An *autoencoder* is the basic building block of deep learning networks that attempts to reproduce its input, i.e., the target output is equal to the input itself [22]. More formally, as depicted in Fig. 1, given the input vector $\mathbf{x} \in [0,1]^d$, an autoencoder maps it to a hidden representation $\mathbf{y} \in [0,1]^{d'}$ through a deterministic mapping $\mathbf{y} = s(\mathbf{W}\mathbf{x} + \mathbf{b})$, where $\mathbf{W}$ is a $d' \times d$ weight matrix, $\mathbf{b}$ is a bias vector and s is the sigmoid activation function, i.e., $s(\mathbf{x}) = \frac{1}{1+e^{(-\mathbf{x})}}$. The hidden representation $\mathbf{y}$, sometimes called the *latent representation*, is then mapped back to a reconstructed vector $\mathbf{z} \in [0,1]^d$, where $\mathbf{z} = s(\mathbf{W}'\mathbf{y} + \mathbf{b}')$, such that $\mathbf{z} \approx \mathbf{x}$. The corresponding parameters, i.e., $\mathbf{W}$, $\mathbf{W}'$, $\mathbf{b}$, $\mathbf{b}'$ are optimized to minimize the average reconstruction error as shown by:

$$\min_{\mathbf{W},\mathbf{W}',\mathbf{b},\mathbf{b}'} \frac{1}{n} L(\mathbf{x}_i, \mathbf{z}_i) \quad (1)$$

where n denotes the number of data instances, and L is a loss function, such as the square loss, Kullback-Leibler divergence, etc.

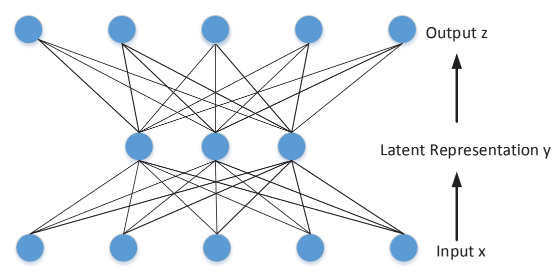


Fig. 1. Illustration of a simplest autoencoder.

Denoising autoencoder (DA) is a simple variant of the basic autoencoder described above, which corrupts the inputs before mapping them into the hidden representation. It is trained to reconstruct (or denoising) the original input $\mathbf{x}$ from its corrupted version $\tilde{\mathbf{x}}$ by minimizing Eq. 1. Usually, the hidden representation will further be used as the learned new representation for machine learning applications, such as image classification [22], speech recognition [26], etc. However, it is important to note that the hidden representation also provides a connection between the corrupted inputs $\tilde{\mathbf{x}}$ and the repaired "clean" input $\mathbf{x}$. Taking this cue, in this paper, instead of using the hidden representation as the new representation of the original data, we propose to employ it as the bridge between the corrupted input and repaired input for the purpose of reusing useful knowledge across problems towards enhanced evolutionary search on continuous optimization, which will be detailed later in Section III.

B. Memetic Computation

Today, the new science of memetics which represents the mind-universe analogue to genetics in culture evolution has stretched across the field of biology, cognition, psychology, etc., and attracted significant attention [27]. Memetic computation has been defined as a paradigm that uses the notion of meme(s) as units of information encoded in computational representations for the purpose of problem solving [28].

In the last decades, meme has been typically perceived as individual learning procedures, or local search operators that enhance the capability of population based search algorithms [29], [12], [30], [27], [31]. This integration has been established as an extension of canonical evolutionary algorithm, by the names of hybrid, adaptive hybrid or Memetic Algorithm (MA) in the literature, and used successfully for solving many real world search problems, ranging from continuous optimization [32], combinatorial optimization [6], [33], constrained optimization [34] to image processing [35], etc. However, as memes have been defined as “the basic unit of cultural transmission via imitation” in Dawkins’ book entitled “The Selfish Gene” [36], its manifestation as an individual learning procedure in MA does not embody the true nature and potential merits of memes.

To further explore the meme-centric computing paradigm for problem solving, other manifestations of meme have also emerged in the literature. For instance, Situngkir presented a structured analysis of culture by means of memetics, where a meme was regarded as the smallest unit of information [37]. Heylighen *et al.* discussed the replication, spread and reproduction operators of memes in cultural evolution [38]. Further, Feng *et al.* proposed a memetic multi-agent system (MeM) towards human-like social agents with memetic automaton [39], while Acampora *et al.* [40] introduced memetic agents as intelligent explorers to create “in time” and personalized experiences for e-Learning. More recently, memes have been modeled as a transformation matrix to be used as prior knowledge for speeding up the evolutionary search on routing problems [13].

In this paper, we contribute to memetic computation by embarking on a study towards memetic search with learning across heterogeneous problems for continuous optimization. In contrast to existing memetic computation works, in the current proposed paradigm, memes have been defined as useful knowledge that are captured and reused for enhancing the evolutionary search process across continuous optimization problems that may differ in problem dimension, number of objectives, etc, and cannot be handled by the approaches mentioned in Section I. Further, as illustrated in Fig. 2, like knowledge housed in the human mind for coping with our everyday life and problem solving, knowledge memes learned

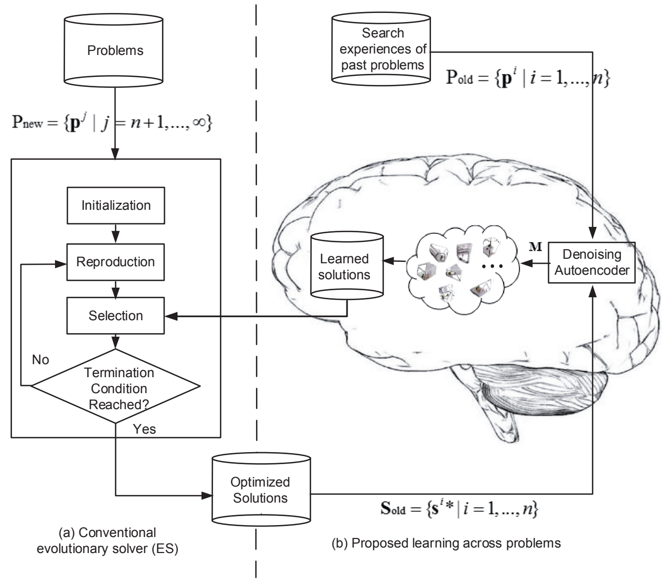


Fig. 2. Proposed memetic search with learning across heterogeneous problems for continuous optimization.

by the denoising autoencoder that residing in the artificial mind of the evolutionary solver play the role of biasing the search positively on newly encountered problems. The details of the proposed autoencoding memetic search will be presented in the next section.

III. Proposed Autoencoding Search Paradigm with Learning across Heterogeneous Problems

In many real-world domains (most commonly occurring in engineering design), it is the known semantic overlap between distinct problems that provides intuitive hints towards the viability of knowledge transfer between them. However, despite the presence of such overlap, the feature spaces of the respective problems may not be identical, which makes it difficult (if not impossible) to directly transfer solutions from one problem to the other. In this case, existing approaches such as exact storage and reuse of past solutions, would fail to conduct knowledge transfer for optimization search as the knowledge is not expressed in a sufficiently generalizable form. Keeping this in mind, here we present an autoencoding search paradigm with learning across heterogeneous problems.

In particular, in this section, we first give the theoretical derivations of the single layer denoising autoencoder which plays a key role in transferring knowledge from past search experiences in the form of problem solutions for enhancing evolutionary search. Next, the details of our proposed autoencoding search paradigm with learning capability across heterogeneous problems is presented.

A. Single Layer Denoising Autoencoder for Reusing Past Search Experiences

In continuous optimization, search experiences obtained by evolutionary algorithms usually denote the best solutions or the optimized solution sets for single objective and multi-objective

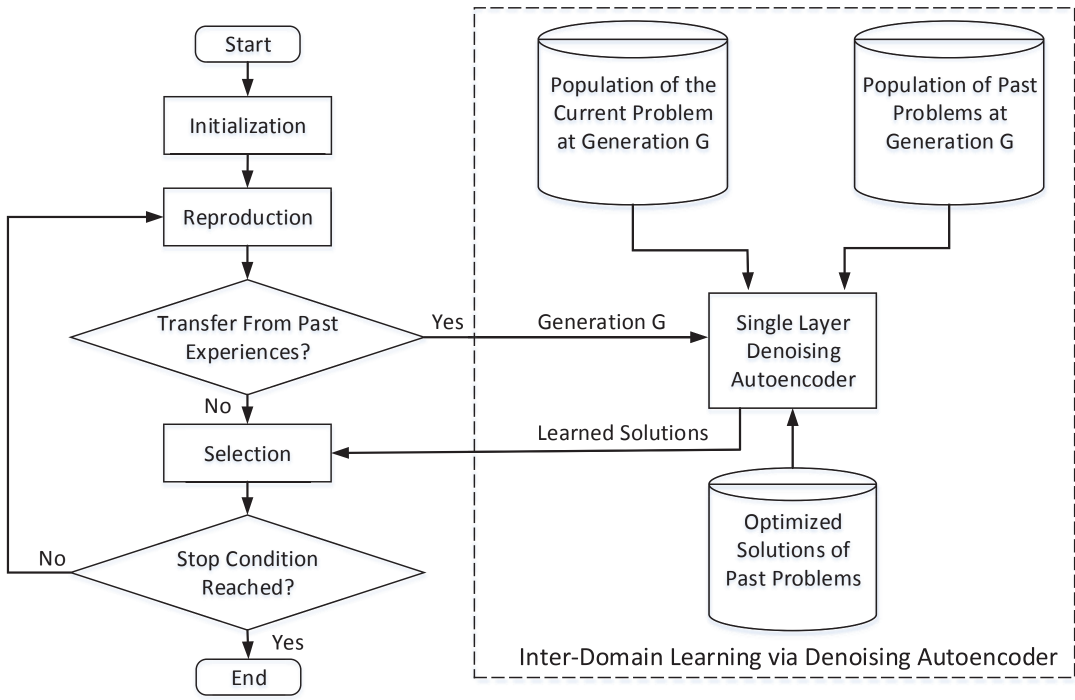


Fig. 3. Proposed autoencoding search paradigm with learning capability across heterogeneous problems.

problems, respectively. As different problems always have unique properties, such as number of objectives, dimension of variables, etc., the reusing of past search experiences via simply injecting the archived optimized solutions (e.g., [15] and [16] as discussed in Section I) is not applicable. In this section, to enable the knowledge reuse across problems with distinctive properties, we derive a single layer denoising autoencoder to build the connection between different problems, which serves as the bridge for transferring knowledge across problems in the form of solutions for the unseen problem of interest.

In particular, consider two populations of solutions for solving two different optimization problems, i.e., $\mathbf{P}_s = \{\mathbf{s}_1 \ldots, \mathbf{s}_N\}$ and $\mathbf{P}_t = \{\mathbf{t}_1 \ldots, \mathbf{t}_N\}$, where N denotes the population size. Note that $\mathbf{s}$ and $\mathbf{t}$ may have different dimensions, and we pad $\mathbf{s}$ or $\mathbf{t}$ with zeros to make both problems be of equal dimensionality. Let $\mathbf{P}_s$ and $\mathbf{P}_t$ denote the population in past search experiences and the current search population, respectively. Our key idea is to treat $\mathbf{P}_s$ as the corrupted version of $\mathbf{P}_t$, and a possible connection between these two problems will be naturally built through a denoising autoencoder.

Here we reconstruct the corrupted inputs with a single level mapping $\mathbf{M}$: $\mathcal{R}^d \rightarrow \mathcal{R}^d$ (d is the space dimension), that minimizes the squared reconstruction loss, which is given by:

$$\mathcal{L}_{sq}(\mathbf{M}) = \frac{1}{2N}\sum_{i=1}^{N} ||\mathbf{t}_i - \mathbf{M}\mathbf{s}_i||^2 \tag{2}$$

To simplify the notation, we assume that a constant feature is added to the input, i.e., $\mathbf{s}_i = [\mathbf{s}_i; 1]$ and $\mathbf{t}_i = [\mathbf{t}_i; 1]$, and an appropriate bias is incorporated within the mapping $\mathbf{M} = [\mathbf{M}, \mathbf{b}]$. Further, the loss in Eq. 2 can be reduced to the matrix form:

$$\mathcal{L}_{sq}(\mathbf{M}) = \frac{1}{2N}\text{tr}[(\mathbf{P}_t - \mathbf{M}\mathbf{P}_s)^{\mathrm{T}}(\mathbf{P}_t - \mathbf{M}\mathbf{P}_s)] \tag{3}$$

where $\text{tr}(\cdot)$ denote the trace operation of a matrix. The solution of Eq. 3 can be expressed as the well-known closed-form solution for ordinary least squares [41], which is given by:

$$\mathbf{M} = (\mathbf{P}_t\mathbf{P}_s^{\mathrm{T}})(\mathbf{P}_s\mathbf{P}_s^{\mathrm{T}})^{-1} \tag{4}$$

As $\mathbf{M}$ is the mapping between $\mathbf{P}_t$ and $\mathbf{P}_s$, it provides a connection for these two problems. The archived optimized solutions for problem $\mathbf{P}_s$ can be directly injected into the population of the search for solving problem $\mathbf{P}_t$ by multiplying $\mathbf{M}$. It is worth noting that, as the mapping $\mathbf{M}$ holds a closed-form solution, the transfer of knowledge from past search experiences via $\mathbf{M}$ does not incur significant additional computational cost into the evolutionary search process[1].

In what follows, the detailed designs of our proposed search paradigm with the derived single layer denoising autoencoder for learning across heterogeneous continuous optimization problems will be presented.

B. Autoencoding Search with Learning across Heterogeneous Problems

The workflow of the proposed autoencoding search with learning capability across heterogeneous problems is outlined in Fig.3. As depicted, for a given new problem instances $\mathbf{p}_{new}$, the evolutionary search process will first proceed with initialization, reproduction (i.e., crossover and mutation) process.

[1]As additional solutions are transferred across problems, extra number of fitness evaluation may be required in the target problem domain (these have nevertheless been accounted for in the various plots of convergence trends).

Subsequently, the transfer of past search experiences take place when the user defined condition is satisfied. In this paper, for simplicity, the transfer of past search experiences occurs with a fixed generation interval, i.e., g_{it}, which is user defined and has been set as 10 throughout this study.

Further, as outlined in Alg. 1, to transfer knowledge from past search experiences, with the current generation index G, the current population $\mathbf{PC}_G$ and the population archived from a past optimized problem at generation G, i.e, $\mathbf{PP}_G$ will be used as the input $\mathbf{P}_t$ (i.e., $\mathbf{P}_t = \mathbf{PC}_G$) and $\mathbf{P}_s$ (i.e., $\mathbf{P}_s = \mathbf{PP}_G$), respectively, for the denoising autoencoder. d_{pc} and d_{pp} denote the dimension of solutions in $\mathbf{PC}_G$ and $\mathbf{PP}_G$, respectively. As discussed in section III-A, solutions in $\mathbf{PC}_G$ or $\mathbf{PP}_G$ will be padded with 0 to make both problems share a common dimensionality. Subsequently, the connection between the current problem and the considered past optimized problem can then be built by $\mathbf{M}$ obtained via Eq. 4. Next, multiplying the optimized solution or solution set (i.e., line 4 and line 10 in Alg. 1) obtained from the experienced past problem with the connectivity matrix $\mathbf{M}$, leads to knowledge-induced solutions that are directly injected into the current population (with proper handling of dimensions), to bias the search process accordingly. A specific example of transferring knowledge from past search experiences via the proposed denoising autoencoder is given in Fig. 4.

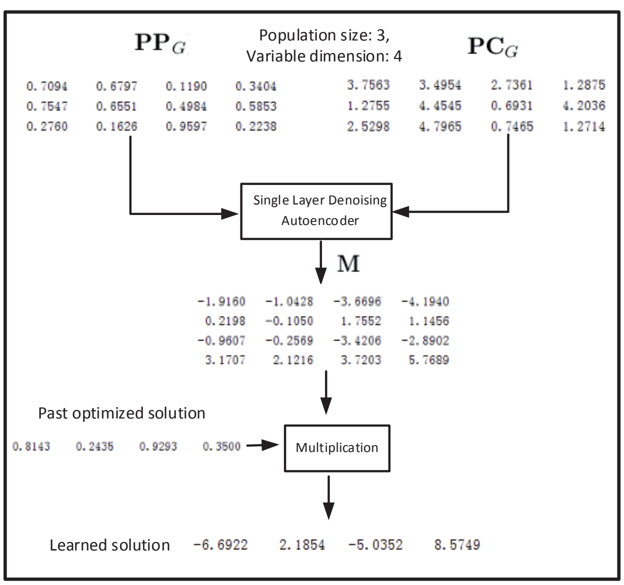


Fig. 4. An example of transferring knowledge from past search experiences via the proposed denoising autoencoder.

Last but not the least, for a positive transfer of past search experiences, the knowledge induced solutions from across problems will be merged together with the current search population which will then undergo the process of natural selection so as to form the new generation of solutions. The inherent adaptiveness encourages us to allow the principles of evolution to take over when little prior knowledge is available about the relationship between problems. If the transfer is indeed beneficial, then survival of the fittest dictates that the positively transferred genetic material survives. Else, the same principle kicks in to automatically eliminate the negatively transferred knowledge [9], [15], [17]. The whole procedure described above will repeat until the stopping criterion is satisfied.

Input : $\mathbf{PC}_G$: matrix of the current population at generation G (each row of the matrix denotes one solution, and solution dimension is d_{pc}); $\mathbf{PP}_G$: matrix of population for a past search problem at generation (each row of the matrix denotes one solution, and solution dimension is d_{pp}); $\mathbf{BS}$: optimized solution (a vector for single optimization problems) or solution set (a matrix for multi-objective problems, each row denotes one solution) of past optimized problems.

Output: $\mathbf{LS}$: Learned solutions across problems for injection in the current population;

```
1  if d_pc ≤ d_pp then
2      Pad each solution in PC_G with 0 to form PC'_G which has d'_pc = d_pp
3      Calculate M with Eq. 4 by setting P_t = PC'_G and P_s = PP_G
4      Obtain LS' by multiplying the BS with M
5      Achieve LS by referring to the first d_pc dimensions of the solutions contained in LS'
6  else
7      Pad each solution in PP_G with 0 to form PP'_G which has d'_pp = d_pc
8      Calculate M with Eq. 4 by setting P_t = PC_G and P_s = PP'_G
9      Pad each solution in BS with 0 to form BS' which has d'_pp = d_pc
10     Achieve LS by multiplying the BS' with M
```

Algorithm 1: Pseudo code of the proposed transfer of past search experiences via denoising autoencoder.

IV. Empirical Study

To verify the efficacy of the proposed autoencoding evolutionary search paradigm with learning across heterogenous problems, empirical study has been conducted in this section. In particular, we evaluate the proposed search paradigm on the complex multi-objective optimization problems.

A. Experimental Configuration

12 commonly used multi-objective problem (MOP) benchmarks, including 5 MOPs in ZDT family problems with two objectives [42] and 7 MOPs in DTLZ family problems with three objectives [43], are considered here. Further, two popular multi-objective evolutionary algorithms (MOEAs), namely *NSGAII* [44] and *SPEA2* [45], are employed as our baseline MOP solvers. Denoting A as the MO solver, three versions of A are compared in this study. One is the original MO solver A, which involves no transfer or no injection of new solutions during the search process. The second is the

TABLE I

Hypervolume median and IQR values obtained by *NSGAII*, *NSGAII+R*, *NSGAII+M1*, and *NSGAII+M2* on ZDT/DTLZ MOPs across 20 independent runs with 75000 function evaluations. ("≈", "+" and "-" denote the corresponding algorithm is statistically significant similar, better and worse than *NSGAII*, respectively.)

MOPs	*NSGAII*	*NSGAII+R*	*NSGAII+M1*	*NSGAII+M2*
ZDT1	6.323E-01±6.20E-03	5.376E-01±4.12E-02 –	6.501E-01±6.42E-03 +	6.495E-01±3.94E-03 +
ZDT2	2.864E-01±4.54E-02	1.402E-01±9.11E-02 –	3.200E-01±4.92E-03 +	3.177E-01±3.50E-03 +
ZDT3	4.925E-01±7.44E-03	4.205E-01±3.57E-02 –	5.122E-01±2.72E-03 +	5.113E-01±1.59E-03 +
ZDT4	1.817E-01±3.10E-01	0.000E-00±0.00E-00 –	6.519E-01±3.48E-03 +	6.457E-01±6.72E-03 +
ZDT6	2.215E-01±4.89E-02	9.008E-06±3.93E-03 –	3.912E-01±3.22E-03 +	3.920E-01±9.16E-04 +
DTLZ1	0.000E-00±0.00E-00	0.000E-00±0.00E-00 ≈	0.000E-00±0.00E-00 ≈	0.000E-00±0.00E-00 ≈
DTLZ2	3.411E-01±9.78E-03	3.276E-01±1.17E-02 –	3.246E-01±1.05E-02 –	3.137E-01±1.64E-02 –
DTLZ3	0.000E-00±0.00E-00	0.000E-00±0.00E-00 ≈	0.000E-00±0.00E-00 ≈	0.000E-00±0.00E-00 ≈
DTLZ4	3.432E-01±1.01E-02	3.283E-01±1.49E-02 –	3.232E-01±1.84E-02 –	3.227E-01±2.52E-02 –
DTLZ5	8.924E-02±7.75E-04	8.815E-02±6.24E-04 –	8.699E-02±1.79E-03 –	8.520E-02±2.00E-03 –
DTLZ6	0.000E-00±0.00E-00	0.000E-00±0.00E-00 ≈	9.059E-02±6.80E-03 +	9.061E-02±1.45E-02 +
DTLZ7	2.099E-01±1.77E-02	8.030E-02±3.11E-02 –	2.491E-01±1.16E-02 +	2.568E-01±9.60E-03 +
Mean	2.332E-01±3.80E-02	1.602E-01±1.92E-02	3.000E-01±5.83E-03	2.988E-01±7.04E-03

TABLE II

Hypervolume median and IQR values obtained by *SPEA2*, *SPEA2+R*, *SPEA2+M1*, and *SPEA2+M2* on ZDT/DTLZ MOPs across 20 independent runs with 75000 function evaluations. ("≈", "+" and "-" denote the corresponding algorithm is statistically significant similar, better and worse than *SPEA2*, respectively.)

MOPs	*SPEA2*	*SPEA2+R*	*SPEA2+M1*	*SPEA2+M2*
ZDT1	6.145E-01±6.16E-03	4.550E-01±4.57E-02 –	6.559E-01±5.50E-04 +	6.532E-01±1.65E-03 +
ZDT2	1.710E-01±1.86E-01	0.000E-00±4.63E-02 –	3.233E-01±3.26E-03 +	3.163E-01±2.30E-02 +
ZDT3	4.822E-01±7.51E-03	3.839E-01±5.15E-02 –	5.135E-01±5.52E-04 +	5.118E-01±9.93E-04 +
ZDT4	7.635E-02±1.67E-01	0.000E-00±0.00E-00 –	5.421E-01±6.26E-02 +	4.688E-01±8.55E-02 +
ZDT6	1.327E-01±2.56E-02	0.000E-00±0.00E-00 –	3.965E-01±3.06E-02 +	3.955E-01±1.13E-02 +
DTLZ1	0.000E-00±0.00E-00	0.000E-00±0.00E-00 ≈	0.000E-00±0.00E-00 ≈	0.000E-00±0.00E-00 ≈
DTLZ2	3.762E-01±6.76E-03	3.660E-01±7.33E-03 –	3.492E-01±1.26E-02 –	3.366E-01±9.50E-03 –
DTLZ3	0.000E-00±0.00E-00	0.000E-00±0.00E-00 ≈	0.000E-00±0.00E-00 ≈	0.000E-00±0.00E-00 ≈
DTLZ4	2.035E-01±1.64E-01	3.528E-01±1.21E-02 ≈	3.247E-01±1.49E-01 ≈	3.227E-01±1.32E-01 ≈
DTLZ5	8.937E-02±6.16E-04	8.597E-02±1.30E-03 –	8.434E-02±2.46E-03 –	8.077E-02±4.83E-03 –
DTLZ6	0.000E-00±0.00E-00	0.000E-00±0.00E-00 ≈	6.980E-02±6.21E-02 +	3.776E-02±4.87E-02 +
DTLZ7	1.980E-01±3.14E-02	4.277E-02±2.47E-02 –	2.692E-01±1.96E-02 +	2.721E-01±1.13E-02 +
Mean	1.953E-01±4.95E-02	1.405E-01±1.58E-02	2.941E-01±2.86E-02	2.829E-01±2.74E-02

MO solver *A* equipped with the proposed search paradigm with cross problem learning capability, in which solutions learned across heterogeneous problems will be injected into the population periodically while the search progresses online (labeled as *A+M* hereafter). Next, the third is the MO solver *A* with injection of randomly generated solutions, labeled by *A+R*. The frequency and the amount of solutions for injection in *A+R* are set exactly the same as *A+M*. Please note that the configurations of evolutionary operators and parameters in *A*, *A+M* and *A+R* are kept the same, and the only difference among them is the injection of solutions during the search process.

Further, our experiments are conducted using jMetal 4.0 [46], which is a Java-based framework that is aimed at facilitating the development of metaheuristics for solving MOPs[2], and the specific experimental settings are outlined as follows.

1) Population size: Population size NP is configured as 50 in both *NSGAII* and *SPEA2* for $m = 2, 3$ objectives.
2) Maximum function evaluations: Max_FES = $75,000$
3) Independent number of runs: $runs = 20$
4) Evolutionary operators in NSGAII [44] and SPEA2 [45]:
   - SBX crossover: $p_c = 0.9$, $\eta_c = 20$
   - Polynomial mutation: $p_m = 1/n$, $\eta_m = 20$
5) Interval of injecting solution obtained by across problem learning: $G = 10$

Other parameters are kept as the default values in jMetal [46].

For the setup of the proposed search with knowledge derived across heterogeneous problems, as there are two types of MOPs, i.e., ZDT family problems with two objectives and DTLZ family problems with three objectives, we consider each type of MOP separately as the solved MOPs for providing optimized solutions as the past search experiences. In particular, we denote *A+M1* and *A+M2* as the MOP solver equipped with the proposed autoencoding search paradigm using past

[2] http://jmetal.sourceforge.net

search experiences provided by ZDT and DTLZ, respectively. Please note that, for solving a specific MOP of interest, the past search experiences for knowledge transfer will exclude this problem. In the present study, the correlations between problems are not considered, and each MOP serving as the solved problem is treated equally for knowledge transfer.

Last but not the least, all the algorithms are evaluated based on hypervolume, which is the only single set quality measure that is known to be strictly monotonic with regard to Pareto dominance [47], [48]. The reference sets in hypervolume are specified by the true Pareto fronts [46]. Further, the obtained results are compared using median values and interquartile range (IQR). In order to obtain statistically sound conclusion, the Wilcoxon rank sum test with 95% confidence level is conducted on the experimental results.

B. Results and Discussion

The hypervolume and IQR results obtained by the classical MO algorithms, i.e., *NSGAII* and *SPEA2*, and their variants which possess injection of solutions along the search that are generated randomly or via the proposed learning approach, on the ZDT/DTLZ MOPs, across 20 independent runs with 75000 function evaluations, are tabulated in Table I and II, respectively. In the tables, "$\approx$", "+" and "-" denote the corresponding algorithm is statistically significant similar, better and worse than the baseline MO algorithm, respectively.

As can be observed, in Table I and II, with the injection of randomly generated solutions, *NSGAII+R* and *SPEA2+R* consistently obtained deteriorated performance in terms of solution quality when compared to the corresponding baseline MO solvers, i.e., *NSGAII* and *SPEA2*. In particular, on totally 12 ZDT/DTLZ MOPs, *NSGAII+R* and *SPEA2+R* obtained a poorer hypervolume on 9 and 8 MOPs against *NSGAII* and *SPEA2*, respectively. The averaged hypervolume values on all the MOPs obtained by (*NSGAII*, *NSGAII+R*) and (*SPEA2*, *SPEA2+R*) are $(0.2332, 0.1602)$ and $(0.1953, 0.1405)$, respectively.

On the other hand, with regard to the proposed autoencoding memetic search paradigm, which is capable of learning from past search experiences across heterogeneous problems, superior solution quality over the baseline MO solver is observed. In particular, in Table I, *NSGAII+M1* and *NSGAII+M2* obtained higher or competitive hypervolume values on 9 out of totally 12 MOPs against *NSGAII*. Further, on MOP DTLZ6, both *NSGAII* and *NSGAII+R* obtained the median hypervolume of 0 over 20 independent runs. However, with the proposed memetic learning from past search experience across heterogeneous problems, *NSGAII+M1* and *NSGAII+M2* achieved the median hypervolume of 0.0959 and 0.0961, respectively. The averaged hypervolume value on all the MOPs obtained by *NSGAII+M1*, *NSGAII+M2* and *NSGAII* are 0.300, 0.2988, and 0.2332, respectively.

Similar results can also be observed when using *SPEA2* as the baseline MO solver. In Table II, *SPEA2+M1* and *SPEA2+M2* significantly outperformed *SPEA2* on 7 out of 12 MOPs. The averaged hypervolume values on all the MOPs of *SPEA2+M1* and *SPEA2+M2* are 0.2941 and 0.2829, respectively, while *SPEA2* only reached the hypervolume of 0.1953.

TABLE III
SUMMARIZED COMPARISON RESULTS AMONG *NSGAII*, *NSGAII+R*, *NSGAII+M1*, AND *NSGAII+M2* BASED ON **HYPERVOLUME** ON ZDT/DTLZ MOPS. (EACH TUPLE *l/t/w* DENOTES THE ALGORITHM AT THE CORRESPONDING ROW LOSES ON *l* MOPS, TIES ON *t* MOPS, AND WINS ON *w* MOPS, WHEN COMPARED TO THE ALGORITHM AT THE CORRESPONDING COLUMN, RESPECTIVELY.)

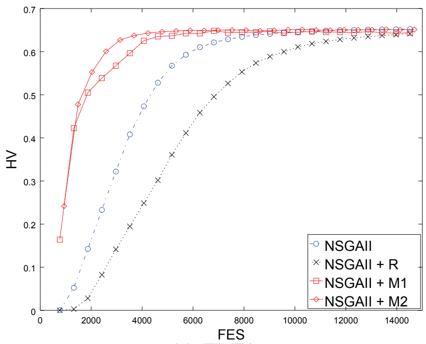
	NSGAII+M1	*NSGAII+M2*	*NSGAII*
NSGAII+R	7/4/1	7/3/2	9/3/0
NSGAII+M1		2/7/3	3/2/7
NSGAII+M2			3/2/7

TABLE IV
SUMMARIZED COMPARISON RESULTS AMONG *SPEA2*, *SPEA2+R*, *SPEA2+M1*, AND *SPEA2+M2* BASED ON **HYPERVOLUME** ON ZDT/DTLZ MOPS. (EACH TUPLE *l/t/w* DENOTES THE ALGORITHM AT THE CORRESPONDING ROW LOSES ON *l* MOPS, TIES ON *t* MOPS, AND WINS ON *w* MOPS, WHEN COMPARED TO THE ALGORITHM AT THE CORRESPONDING COLUMN, RESPECTIVELY.)

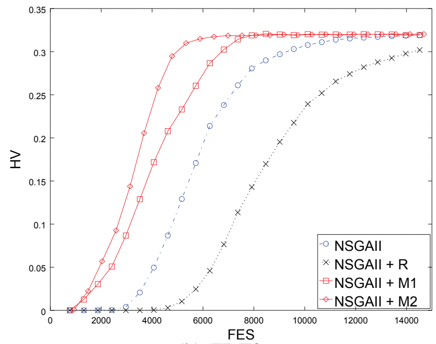
	SPEA2+M1	*SPEA2+M2*	*SPEA2*
SPEA2+R	7/2/3	7/2/3	8/4/0
SPEA2+M1		0/5/7	2/3/7
SPEA2+M2			2/3/7

Next, to provide an overview of the performance among the algorithms, Table III and IV summarized the comparison between the baseline MO solvers and their counterparts in terms of hypervolume. In the tables, each tuple *l/t/w* denotes the algorithm in the corresponding row loses on l MOPs, ties on t MOPs, and wins on w MOPs, when compared to the algorithm in the corresponding column, respectively. As can be observed, the approach of injecting randomly generated solutions along the search, performs consistently worse against the baseline MO solvers (i.e., *NSGAII* and *SPEA2*) on all the MO problems considered. This implies that a blind injection of solutions into the evolution population while the search progresses can significantly deteriorate the evolutionary search performance. In contrast, the injection of solutions generated using the knowledge learned from past search experiences on heterogeneous problems, as proposed in the autoencoding memetic search paradigm, brings about significant improvements to the search performance of both *NSGAII* and *SPEA2*.

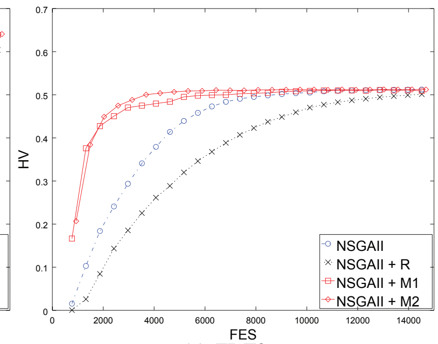
Note that the configurations of the search operators and parameters are kept the same in the MO solver with randomly solution injection (i.e., *NSGAII+R*) and the MO solver with the injection of solution learned from past search experience via the proposed learning approach (i.e., *NSGAII+M1* and *NSGAII+M2*), the superior performance obtained by the latter thus can clearly be attributed to the effectiveness of the proposed autoencoding search with knowledge transfer across heterogeneous problems. In addition, it is also worth noting that as different MO solvers have unique search schemes and biases, the consistent superior performances obtained by the injection of solutions learned from search experiences across heterogeneous problems on both *NSGAII* and *SPEA2* further confirmed the efficacy of the proposed autoencoding search paradigm for enhanced evolutionary search on continuous problems.



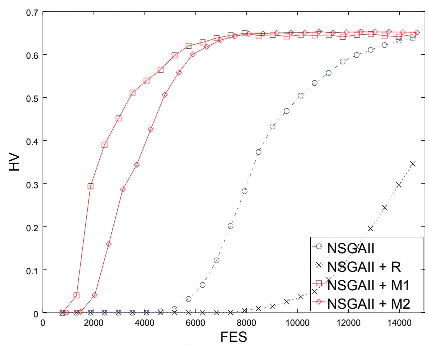
(a) ZDT1



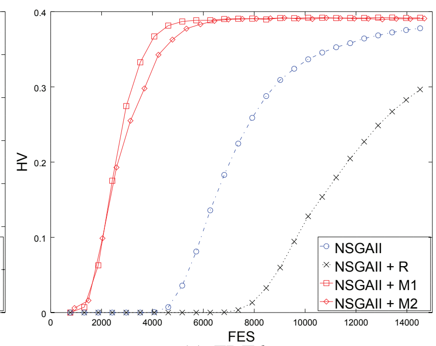
(b) ZDT2



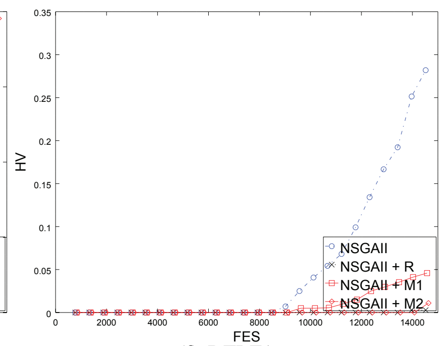
(c) ZDT3



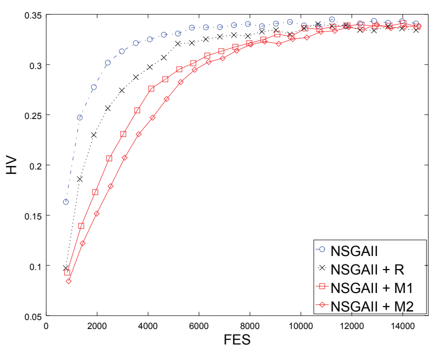
(d) ZDT4



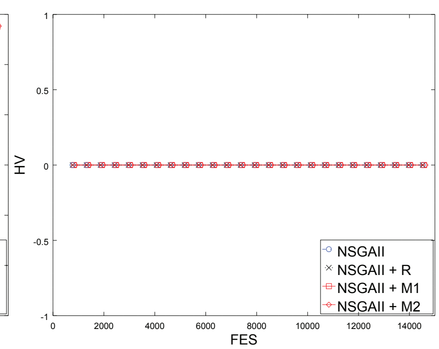
(e) ZDT6



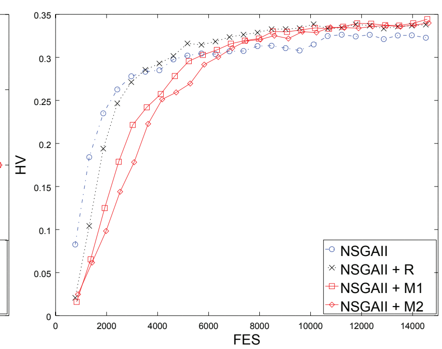
(f) DTLZ1



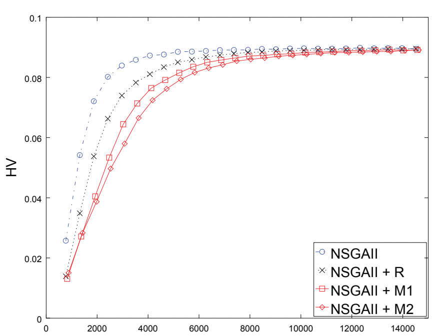
(g) DTLZ2



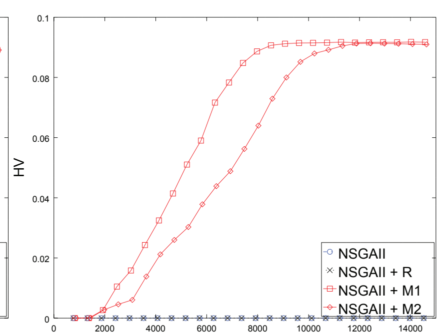
(h) DTLZ3



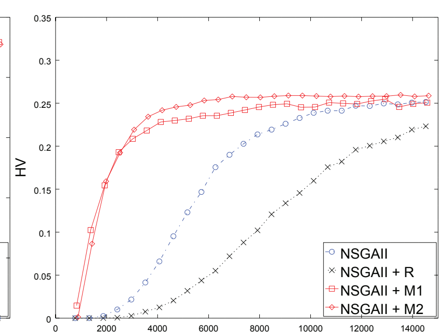
(i) DTLZ4



(j) DTLZ5



(k) DTLZ6



(l) DTLZ7

Fig. 5. The mean hypervolume (HV) values obtained by *NSGAII*, *NSGAII+R*, *NSGAII+M1*, and *NSGAII+M2* with respect to FES on representative ZDT/DTLZ MOPs across 20 independent runs.

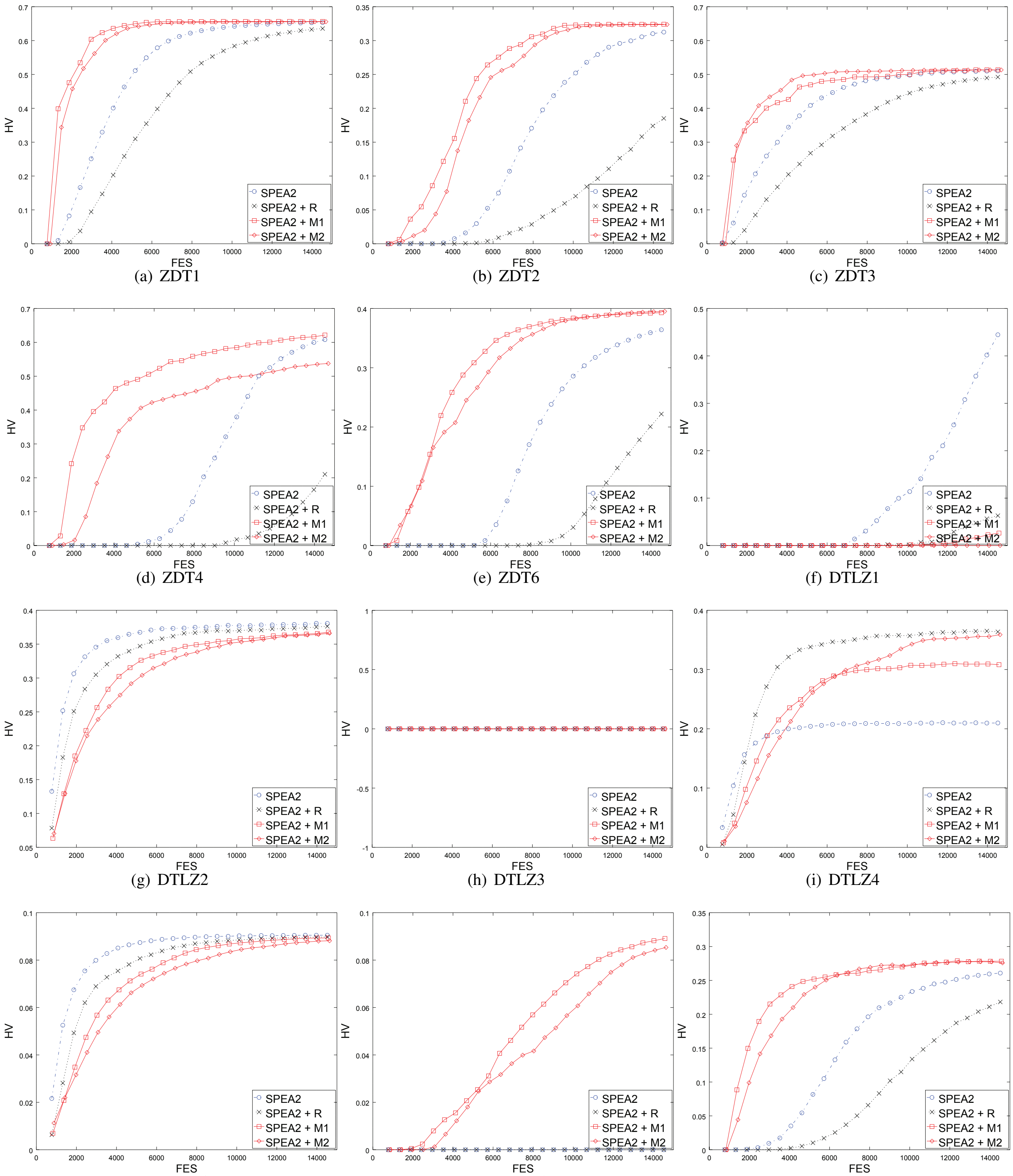
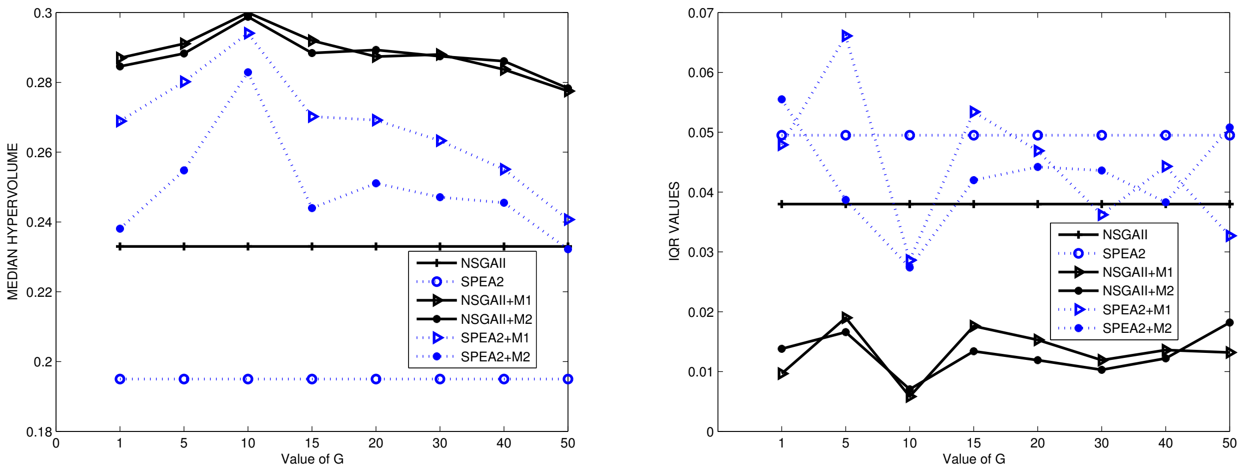


Fig. 6. The mean hypervolume (HV) values obtained by *SPEA2*, *SPEA2+R*, *SPEA2+M1*, and *SPEA2+M2* with respect to FES on representative ZDT/DTLZ MOPs across 20 independent runs.



(a) Median Hypervolume

(b) IQR Value

Fig. 7. Averaged Hypervolume median and IQR values obtained by *NSGAII*, *SPEA2*, *NSGAII+M1*, *NSGAII+M2*, *SPEA2+M1*, and *SPEA2+M2* on all the ZDT and DTLZ MOPs across 20 independent runs with 75000 function evaluations.

Further, to access the efficiency of the proposed search paradigm, we present the convergence graphs of the baseline MO solver, the MO solver with injection of randomly generated solutions, and the MO solver with injection of solutions learned from past search experiences by our proposed approach, on the representative MOPs. In particular, Fig. 5 depicts the convergence graphs of *NSGAII*, *NSGAII+R*, *NSGAII+M1*, *NSGAII+M2* and Fig. 6 gives the convergence graphs of *SPEA2*, *SPEA2+R*, *SPEA2+M1*, *SPEA2+M2*[3]. In these figures, the Y-axis denote the mean hypervolume value obtained across 20 independent runs, while the X-axis gives the respective computational effort incurred in terms of the number of fitness evaluations (FES) made so far.

In Fig. 5, it is observed that *NSGAII+R* converges slower than *NSGAII* on almost all the MOPs. In particular, on MOPs, such as ZDT2 (i.e., Fig. 5(b)) and ZDT6 (Fig. 5(e)), *NSGAII+R* incurs over $14,000$ FES to arrive at the same hypervolume obtained by *NSGAII* of around $8,000$ FES. On the other hand, using the knowledge learned from past search experiences, *NSGAII+M1* and *NSGAII+M2* converged to the Pareto front significantly faster than both *NSGAII+R* and *NSGAII* on most of the MOPs. For instance, on MOPs, such as ZDT1 (i.e., Fig. 5(a)) and ZDT3 (i.e., Fig. 5(c)), *NSGAII+M1* and *NSGAII+M2* expended only about $4,000$ FES to arrive at the hypervolume values obtained by *NSGAII* which took around $10,000$ FES. Further, on MOP DTLZ6 (i.e., Fig. 5(h)), both *NSGAII* and *NSGAII+R* obtained hypervolume approaching to 0. However, with the knowledge learned across problems, *NSGAII+M1* and *NSGAII+M2* achieved significant hypervolume improvements over the basic MO solver *NSGAII* on DTLZ6.

Similar observations have also been achieved in Fig. 6 when configuring *SPEA2* as the baseline MO solver. However, on MOP DTLZ5 and DTLZ2, we note that the proposed autoencoding search paradigm is less effective in comparison to the baseline MO solvers, i.e., *NSGAII* and *SPEA2*. This negative transfer may be due to the lack of relevant knowledge in past search experiences in relation to DTLZ5 and DTLZ2[4].

Lastly, as G defines the transfer frequency of the proposed autoencoding search paradigm, we further study how does the configuration of G affect the evolutionary search. Generally, a small value of G will greatly increase the frequency of injecting transferred solutions, while a big value of G will reduce the injection of transferred solutions significantly. Fig. 7 gives the averaged hypervolume median and IQR values obtained by *NSGAII*, *SPEA2*, *NSGAII+M1*, *NSGAII+M2*, *SPEA2+M1*, and *SPEA2+M2* on all the ZDT and DTLZ MOPs across 20 independent runs with various configuration of G. It can be observed from the figure, superior solution qualities have been obtained by the proposed autoencoding memetic search when compare to the baseline solvers, i.e., *NSGAII* and *SPEA2*, on all the configurations of G. However, while the optimal confirmation of G is in general problem dependent, fixing $G = 10$ is found to provide noteworthy results across a variety of problems encountered.

V. A Real-World Case Study in Composites Manufacturing

In this section, we consider the application of the proposed algorithm on a real-world engineering design setting from the composites manufacturing industry. In particular, we aim to carry out the computationally demanding simulation-based optimization of two distinct processes widely employed for the manufacturing of high-quality fibre-reinforced polymer (FRP) composite parts. It is noted that the *cost effective* use of composite materials is gradually becoming indispensable in automotive and aerospace industries, thereby highlighting the potential real-world implications of our proposition.

[3]Please note that, as the X-axis of these figures represents the total number of fitness evaluations incurred throughout the evolutionary search thus far, which includes the FES incurred by any of the transferred solutions have also been incorporated.

[4]Prior knowledge about the relationship between problems can be helpful for a positive transfer.

The particular manufacturing processes considered in this paper are labelled as resin transfer moulding (RTM), and injection/compression-liquid composite moulding (I/C-LCM) - both belonging to the same family of rigid-tool liquid composite moulding (LCM) methods [49].

For clarity, we shall first briefly describe the various steps involved in the RTM and I/C-LCM cycles. It will be clear through our discussion that there indeed exists the scope for knowledge transfer during the design optimization stage of the aformentioned manufacturing cycles. As a result, the manufacturer need not undertake the respective process designs from scratch (as is typically the case in traditional *tabula rasa* optimization), since the exploitation of shared knowledge can automatically lead to more effective search of the design space.

The setup of the RTM cycle consists of a rigid mould with a cavity that is shaped according to the geometry of the part to be manufactured. A preform of the dry fibrous reinforcement is first carefully placed into this cavity. Then, the mould is closed to the final (desired) thickness, after which a liquid thermoset resin is injected to fill all the remaining voids inside the mould. Furthermore, in many practical applications, heating of the mould and resin system is generally utilized as a means to control the rate of mould filling and the fluid pressure developed inside the mould during the process.

I/C-LCM shares much in common with RTM, except for the fact that it has one additional phase as part of the complete manufacturing cycle. In particular, unlike RTM, the mould is only partially closed prior to resin injection. Complete closure of the mould to the desired thickness occurs after the required volume of resin has been injected. Accordingly, the partial overlap between the mechanisms of the I/C-LCM and RTM cycles gives rise to a subspace, within the extent of the design space, wherein transferrable knowledge resides [10].

For a more detailed discussion on RTM and I/C-LCM, the reader is referred to [50]. We demonstrate the basic workflow of an LCM process through the illustration in Fig. 8. The primary aim of the LCM cycle is to complete stages (a), (b), and (c), as shown in Fig. 8, satisfactorily, while ensuring the minimization of cycle time (constituting objective 1) as well as the minimization of process cost (constituting objective 2). Note that the process cost is largely driven by the peripheral equipment that is needed to resist the (often very high) fluid pressure developed inside the moulds [50]. In other words, the problem of minimizing the process cost can be transformed to that of minimizing internal forces ($F_{int} \propto$ sum of fluid pressure + fibre compaction stress) within the mould. Often, practical limitations on the availability of peripheral equipment (such as a hydraulic press) also leads to the prescription of an upper bound on the maximum allowable internal force ($F_{capacity}$). Thus, a stringently constrained multi-objective optimization problem is formed for both RTM and I/C-LCM cycles, which can stated as follows:

$$minimize\ (Time,\ F_{int});\ s.t.\ F_{int} < F_{capacity}. \quad (5)$$

In the present study, $F_{capacity}$ is assumed to be 30 tons, thereby setting a restrictive constraint.

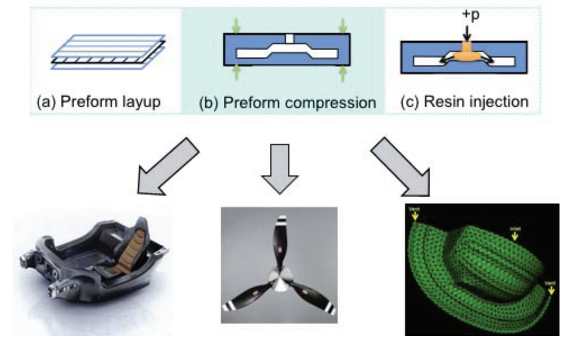


Fig. 8. Key steps of a typical LCM cycle that is widely employed for the manufacture of high-quality fibre-reinforced polymer composite parts in various applications; (a) *preform layup* involves initial preparation of the fibrous reinforcement, (b) *preform compression* involves closure of the mould walls (that are precisely machined according to the geometry of the composite part), thereby compacting the fibrous reinforcement to a high density, and (c) *resin injection* involves the introduction of a liquid polymeric resin into the mould cavity at high pressure.

During the experimental setup, it is noted that a high-fidelity simulation of an LCM cycle can be a time consuming affair, taking several minutes per evaluation for a part of complex geometry. Even for the present case study of simulating the manufacture of a FRP plate of circular planform (to be made up of glass-fibre reinforced epoxy), the runtime needed is generally several seconds per evaluation for achieving sufficiently converged results, thereby categorizing it as a computationally expensive problem from the perspective of most search-based design optimization algorithms. It is contended that the true practical value of knowledge extraction and reuse from previous problem-solving exercises, for the purpose of accelerating search on a new problem of interest, can be fully appreciated under such scenarios that are routinely encountered in engineering design settings.

For carrying out numerical simulations of the manufacture of the FRP circular plate, details of geometric parameters and material properties can be found in [10], [50]. For the sake of brevity, these details are not reproduced in this paper. However, the description of each design variable and the extent of the search space, as is required for executing the optimization algorithms, is presented in Table V.

We first optimize the I/C-LCM cycle using *NSGAII*. The embedded knowledge is then transferred across to accelerate the optimization of the RTM cycle (that is solved using algorithm *NSGAII+M* equipped with the proposed autoencoding memetic search paradigm). As has been discussed previously, since both processes belong to the same family of LCM methods, it is intuitively expected that there exists some latent form of common knowledge between them. As shown in Table V, while the RTM cycle has four design variables in all, the I/C-LCM cycle has six variables. Importantly, four of the six design variables possess *the same phenotypic meaning* as in the case of RTM. Thus, the shared knowledged is likely to be contained in these four overlapping variables.

The averaged convergence trends of the hypervolume indicator, obtained from the *NSGAII* and *NSGAII+M* algorithms (over 10 independent runs each), are presented in Fig. 9.

TABLE V
DESCRIPTION OF DESIGN VARIABLES FOR THE RTM AND I/C-LCM COMPOSITE PARTS MANUFACTURING CYCLES

Variable name	Cycle	Lower bound	Upper bound
Initial mould closure speed	RTM + I/C-LCM	1 mm/min	10 mm/min
Injection pressure	RTM + I/C-LCM	1 MPa	50 MPa
Mould temperature	RTM + I/C-LCM	293 K	373 K
Resin temperature	RTM + I/C-LCM	293 K	373 K
Injection height	I/C-LCM	0.8 cm	1 cm
Final mould closure speed	I/C-LCM	1 mm/min	10 mm/min

Note that while calculating the hypervolume, the reference set representing approximations of the ideal and nadir points are set as (40 sec, 13.5 tons) and (150 sec, 33 tons), respectively. As is clearly revealed in Fig. 9, the knowledge transfer enabled by the autoencoding memetic search paradigm provides a strong impetus to the search process, speeding up the discovery of high-quality solutions by a substantial amount. In fact, the *NSGAII+M* algorithm is found to receive a significant boost during the initial stages of evolution itself, enabling it to quickly achieve higher values of the hypervolume indicator while consuming considerably lower number of function evaluations. To demonstrate, it takes *NSGAII+M* 87.57 evaluations on average to attain a hypervolume of 0.65. On other hand, *NSGAII* alone takes an average of 224.64 evaluations to reach the same hypervolume. It is not hard to imagine that savings of approximately 137 evaluations (to reach the same level of performance) can play a vital role toward cutting down of design time, especially when faced with exorbitantly expensive computational simulations (as is commonly the case in practical engineering design environments).

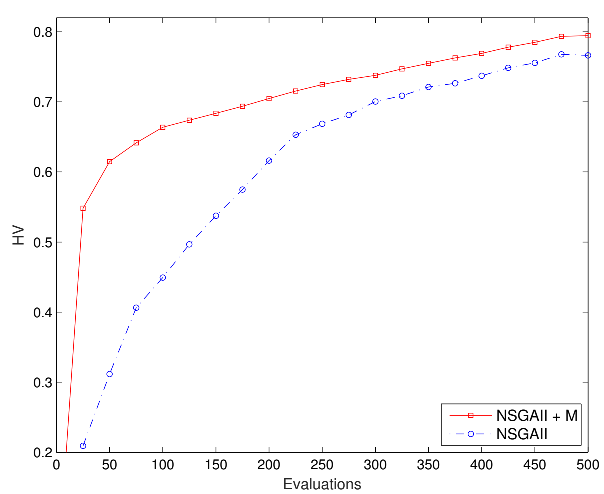


Fig. 9. Averaged convergence trends of the hypervolume idicator achieved for the RTM cycle using *NSGAII* and *NSGAII+M*; the latter being equipped with the autoencoding memetic search paradigm.

VI. CONCLUSION

In this paper, to reuse knowledge from past search experiences for enhancing evolutionary search, we have proposed an autoencoding evolutionary search paradigm with learning capability across heterogeneous problems for continuous optimization problems with diverse properties, such as different problem dimension, number of objectives, etc. In particular, to enable learning across heterogeneous problems, we have first derived a single layer denoising autoencoder which holds a close-form solution to build the connection between problems. Next, we have presented the detailed designs on the reuse of knowledge captured from past search experiences via the denoising autoencoder, in the form of problem solutions, along the evolutionary search process. The survival of the transferred solutions is then governed by the natural selection pressure of evolution. To evaluate the proposed search paradigm, we have conducted comprehensive empirical studies on the complex multi-objectives benchmarks and a real-world application from the fibre-reinforced polymer composites manufacturing industry. The obtained results confirmed the efficacy of the proposed approach for enhancing the evolutionary search when compared to the original evolutionary solver as well as the evolutionary solver with injection of randomly generated solutions while search progresses.

In the future, our works are two-fold: first of all, we would like to further study the proposed autoencoding search paradigm with more complex real-world optimization problems to explore the possible improvements on the proposed method for solving real-world applications. Secondly, we would also like to study the correlations among optimization problems, which could provide deeper insights on reusing knowledge across heterogeneous problems towards enhanced evolutionary optimization processes.

ACKNOWLEDGMENT

This work is partially supported under the National Natural Science Foundation of China (Grant No. 61603064) and the Data Science and Artificial Intelligence Center (DSAIR) at the Nanyang Technological University.

REFERENCES

[1] T. Back, U. Hammel, and H. P. Schwefel, “Evolutionary computation: Comments on the history and current state,” *IEEE Transactions on Evolutionary Computation*, vol. 1, pp. 3–17, 1997.
[2] D. E. Goldberg, *Genetic Algorithms in Search, Optimization and Machine Learning*. Addison-Wesley Longman Publishing Co., Inc., 1989.
[3] R. Chelouah and P. Siarry, “A continuous genetic algorithm designed for the global optimization of multimodal functions,” *Journal of Heuristics*, vol. 6, no. 2, pp. 191–213, 2000.
[4] L. Bianchi, M. Birattari, M. Chiarandini, M. Manfrin, M. Mastrolilli, L. Paquete, O. Rossi-Doria, and T. Schiavinotto, “Metaheuristics for the vehicle routing problem with stochastic demands,” *Parallel Problem Solving from Nature PPSN VIII. Lecture Notes in Computer Science.*, vol. 3242, pp. 1 – 10, 2004.
[5] K. C. Tan, C. Y. Cheong, and C. K. Goh, “Solving multiobjective vehicle routing problem with stochastic demand via evolutionary computation,” *European Journal of Operational Research*, vol. 177, no. 2, pp. 813 – 839, 2007.
[6] K. Tang, Y. Mei, and X. Yao, “Memetic algorithm with extended neighborhood search for capacitated arc routing problems,” *IEEE Transactions on Evolutionary Computation*, vol. 13, no. 5, pp. 1151–1166, 2009.
[7] M. J. N. Su, M. J. Zhang and K. C. Tan, “Automatic design of scheduling policies for dynamic multi-objective job shop scheduling via cooperative coevolution genetic programming,” *IEEE Transactions on Evolutionary Computation*, vol. 18, no. 1, pp. 193–208, 2014.
[8] C. H. Liu and C. K. Ting, “Computational intelligence in music composition: A survey,” *IEEE Transactions on Emerging Topics in Computational Intelligence*, vol. 1, no. 1, pp. 2–15, 2017.

[9] A. Gupta, Y. S. Ong, and L. Feng, "Multifactorial evolution: Towards evolutionary multitasking," *IEEE Transactions on Evolutionary Computation*, 2015.

[10] A. Gupta, Y. S. Ong, L. Feng, and K. C. Tan, "Multi-objective multifactorial optimization in evolutionary multitasking," *IEEE Transactions on Cybernetics*, 2016.

[11] Y. C. Jin, *Knowledge Incorporation in Evolutionary Computation*, ser. Studies in Fuzziness and Soft Computing. Springer, 2010.

[12] F. Neri, C. Cotta, and P. Moscato, *Handbook of Memetic Algorithms*, ser. Studies in Computational Intelligence. Springer, 2011.

[13] L. Feng, Y. S. Ong, A. H. Tan, and I. W. Tsang, "Memes as building blocks: A case study on evolutionary optimization + transfer learning for routing problems," *Memetic Computing*, vol. 7, no. 3, pp. 159–180, 2015.

[14] K. Utkarsh, A. Trivedi, D. Srinivasan, and T. Reindl, "A consensus-based distributed computational intelligence technique for real-time optimal control in smart distribution grids," *IEEE Transactions on Emerging Topics in Computational Intelligence*, vol. 1, no. 1, pp. 51–60, 2017.

[15] S. J. Louis and J. McDonnell, "Learning with case-injected genetic algorithms," *IEEE Transactions on Evolutionary Computation*, vol. 8, no. 4, pp. 316–328, 2004.

[16] P. Cunningham and B. Smyth, "Case-based reasoning in scheduling: Reusing solution components," *The International Journal of Production Research*, vol. 35, no. 4, pp. 2947–2961, 1997.

[17] L. Feng, Y. S. Ong, M.-H. Lim, and I. W. Tsang, "Memetic search with inter-domain learning: A realization between cvrp and carp," *Transactions on Evolutionary Computation*, vol. 19, no. 5, pp. 644–658, 2015.

[18] M. Pelikan and M. W. Hauschild, "Learn from the past: Improving model-directed optimization by transfer learning based on distance-based bias," *Missouri Estimation of Distribution Algorithms Laboratory, University of Missouri in St. Louis, MO, United States, Tech. Rep*, p. 2012007, 2012.

[19] R. Santana, A. Mendiburu, and J. A. Lozano, "Structural transfer using edas: An application to multimarker tagging snp selection," *IEEE Congress on Evolutionary Computation*, pp. 1–8, 2012.

[20] L. Feng, Y. S. Ong, I. W. Tsang, and A. H. Tan, "An evolutionary search paradigm that learns with past experiences," *IEEE World Congress on Computational Intelligence, Congress on Evolutionary Computation*, 2012.

[21] M. Iqbal, W. Browne, and M. J. Zhang, "Reusing building blocks of extracted knowledge to solve complex, large-scale boolean problems," *IEEE Transactions on Evolutionary Computation*, vol. 18, pp. 465–580, 2014.

[22] P. Vincent, H. Larochelle, I. Lajoie, Y. Bengio, and P. A. Manzagol, "Stacked denoising autoencoders: Learning useful representations in a deep network with a local denoising criterion," *Journal of Machine Learning Research*, vol. 11, pp. 3371–3408, 2010.

[23] Y. Bengio, "Learning deep architectures for ai," *Found. Trends Mach. Learn.*, vol. 2, no. 1, pp. 1–127, 2009.

[24] J. Schmidhuber, "Deep learning in neural networks: An overview," *Neural Networks*, vol. 61, pp. 85–117, 2015, published online 2014; based on TR arXiv:1404.7828 [cs.NE].

[25] M. I. Jordan and T. M. Mitchell, "Machine learning: Trends, perspectives, and prospects," *Science*, vol. 349, no. 6245, pp. 255–260, 2015.

[26] X. Lu, Y. Tsao, S. Matsuda, and C. Hori, "Speech enhancement based on deep denoising autoencoder," in *INTERSPEECH*. ISCA, 2013, pp. 436–440.

[27] X. S. Chen, Y. S. Ong, M. H. Lim, and K. C. Tan, "A multi-facet survey on memetic computation," *IEEE Transactions on Evolutionary Computation, In Press*, no. 5, pp. 591–607, 2011.

[28] Y. S. Ong, M. H. Lim, and X. S. Chen, "Research frontier: Memetic computation - past, present & future," *IEEE Computational Intelligence Magazine*, vol. 5, no. 2, pp. 24–36, 2010.

[29] P. Moscato, "A gentle introduction to memetic algorithms," in *Handbook of Metaheuristics*. Kluwer Academic Publishers, 2003, pp. 105–144.

[30] C. C. F. Neri, "Memetic algorithms and memetic computing optimization: A literature review," *Swarm and Evolutionary Computation*, no. 5, pp. 1–14, 2012.

[31] J. K. Chong, "A novel multi-objective memetic algorithm based on opposition-based self-adaptive differential evolution," *Memetic Computing*, vol. 8, no. 2, pp. 147–165, 2016.

[32] Y. S. Ong and A. J. Keane, "Meta-lamarckian learning in memetic algorithm," *IEEE Transactions on Evolutionary Computation, In Press*, no. 2, pp. 99–110, 2004.

[33] Y. Feng, J. Yang, C. Wu, M. Lu, and X. Zhao, "Solving 0–1 knapsack problems by chaotic monarch butterfly optimization algorithm with gaussian mutation," *Memetic Computing*, pp. 1–16, 2016.

[34] A. B. Ullah, R. Sarker, D. Cornforth, and C. Lokan, "Ama: a new approach for solving constrained real-valued optimization problems," *Soft Computing*, vol. 13, no. 8, pp. 741–762, 2009.

[35] L. Jiao, M. Gong, S. Wang, B. Hou, Z. Zheng, and Q. Wu, "Natural and remote sensing image segmentation using memetic computing," *IEEE Computational Intelligence Magazine*, vol. 5, no. 2, pp. 78–91, 2010.

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

[37] H. Situngkir, "On selfish memes: culture as complex adaptive system," *Journal of Social Complexity*, vol. 2, no. 1, pp. 20–32, 2004.

[38] F. Heylighen and K. Chielens, "Cultural evolution and memetics," *Encyclopedia of Complexity and System Science, B. Meyers, ed.: Springer*, 2008.

[39] L. Feng, Y. S. Ong, A. H. Tan, and X. S. Chen, "Towards human-like social multi-agents with memetic automaton," in *IEEE Congress on Evolutionary Computation*, 2011, pp. 1092–1099.

[40] G. Acampora, V. Loia, and M. Gaeta, "Exploring e-learning knowledge through ontological memetic agent," *Computational Intelligence Magazine*, vol. 5, no. 2, pp. 66–77, 2010.

[41] C. Bishop, *Pattern Recognition and Machine Learning*. Springer, 2006.

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

[43] K. Deb, L. Thiele, M. Laumanns, and E. Zitzler, "Scalable test problems for evolutionary multiobjective optimization," *Evolutionary Multiobjective Optimization*, pp. 105–145, 2005.

[44] K. Deb, A. Pratap, S. Agarwal, and T. Meyarivan, "A fast and elitist multiobjective genetic algorithm: NSGA-II," *IEEE Transactions Evolutionary Computation*, vol. 6, no. 2, pp. 182–197, 2002.

[45] E. Zitzler, M. Laumanns, and L. Thiele, "SPEA2: Improving the strength Pareto evolutionary algorithm," in *Proceedings of Evolutionary Methods for Design, Optimisation and Control with Application to Industrial Problems (EUROGEN)*, 2001, pp. 95–100.

[46] J. Durillo, A. Nebro, and E. Alba, "The jmetal framework for multi-objective optimization: Design and architecture," in *Proceedings of IEEE Congress Evolutionary Computation (CEC)*, 2010, pp. 1–8.

[47] E. Zitzler and L. Thiele, "Multiobjective evolutionary algorithms: A comparative case study and the strength pareto approach," *IEEE Transactions Evolutionary Computation*, vol. 3, no. 4, pp. 257–271, 1999.

[48] J. Bader and E. Zitzler, "HypE: An algorithm for fast hypervolume-based many-objective optimization," *Evolutionary Computation*, vol. 19, no. 1, pp. 45–76, 2011.

[49] A. Gupta, "Numerical modelling and optimization of non-isothermal, rigid tool liquid composite moulding processes," Ph.D. dissertation, Department of Engineering Science, University of Auckland, 2013.

[50] A. Gupta, P. Kelly, S. Bickerton, and W. Walbran, "Simulating the effect of temperature elevation on clamping force requirements during rigid-tool liquid composite moulding processes," *Composites Part A: Applied Science and Manufacturing*, vol. 43, no. 12, pp. 2221–2229, 2012.