Multi-Factorial Evolutionary Algorithm Based on M2M Decomposition

Jiajie Mo[2], Zhun Fan[1,2], Wenji Li[2], Yi Fang[2], Yugen You[2], and Xinye Cai[3]

[1] Guangdong Provincial Key Laboratory of Digital Signal and Image Processing
Shantou, 515063, China
[2] Department of Electronic Engineering, Shantou University
Shantou, 515063, China
[3] College of Computer Science and Technology, Nanjing University of Aeronautics and Astronautics
Jiangsu, 210016, China

Abstract. This paper proposes a decomposition-based multi-objective multi-factorial evolutionary algorithm (MFEA/D-M2M). The MFEA/D-M2M adopts the M2M approach to decompose multi-objective optimization problems into multiple constrained sub-problems for enhancing the diversity of population and convergence of sub-regions. An machine learning model augmented version is also been implemented, which utilized discriminative models for pre-selecting solutions. Experimental studies on nine multi-factorial optimization (MFO) problem sets are conducted. The experimental results demonstrated that MFEA/D-M2M outperforms the vanilla MFEA on six MFO benchmark problem sets and achieved comparable results on the other three problem sets with partial intersection of global optimal.

Keywords: Multi-factorial optimization; M2M decomposition; Pre-selection

1 Introduction

With the increasing amount of incoming data streams, it is very desirable that the information systems and algorithms are capable of efficient multi-tasking[1]. Evolutionary algorithms (EAs) are population based optimization algorithms that work on Darwinian principles of natural selection or survival of the fittest[2]. The population based search enables multi-objective evolutionary algorithms (MOEAs) to achieve simultaneous convergence toward the entire Pareto Front (PF) for multi-objective optimization problems (MOPs).

Furthermore, recent researches[3–6] show that by exploiting the implicit parallelism offered by a population, a multi-factorial evolutionary algorithm (MFEA) can simultaneously and efficiently solve multiple optimization problems with one single population, where each constitutive problem contributes a unique factor influencing the evolution. Therefore, evolutionary multi-tasking[7–9] is attracting extensive attention as a new paradigm in the field of optimization and evolutionary computation.

In order to enhance the diversity of the population, M2M decomposition approach is augmented with the vanilla MFEA[8]. Furthermore, support vector machines (SVMs) are utilized as pre-selection models for finding promising solutions given only their decision variables.

The rest of this paper is organized as follows. Section 2 describes the preliminaries of vanilla MFEA. Section 3 illustrates the mechanism of combining M2M decomposition with MFEA. Section 4 describes the utilization of SVM for pre-selection. Experimental results and analysis are provided in Section 5. Section 6 concludes the paper with a brief discussion of future work.

2 Multi-Factorial Evolutionary Algorithms

Recently, MFEAs[7, 8] have been proposed to solve multiple optimization tasks with one population. In MFO, each constitutive task is considered to be contributing a unique factor influencing the evolution of the population.

Without loss of generality, consider the situation where K MOPs to be minimized simultaneously. Denoting the jth task that has M_j objevtives as T_j, its D_j dimensional search space as $\mathbf{X}_j \in \mathbb{R}^{D_j}$ and its objective function vector as $\mathbf{F}_j : \mathbf{X}_j \to \mathbb{R}^{M_j}$. The jth MOP in MFO paradigm can be defined as (1).

$$\begin{aligned} \text{minimize} \quad & \mathbf{F}_j(x) = (f_{j,1}(x), \ldots, f_{j,M_j}(x))^T \\ \text{subject to} \quad & x \in \mathbb{Y} \end{aligned} \tag{1}$$

where $\mathbb{Y}$ is the unified representation space[5] built on the search spaces $\mathbf{X}_i,\ i \in \{1, 2, \ldots, K\}$ of all constitutive MOPs.

A unified search space can be built so that a) the dimension of the unified search space $D = \max_j(D_j)$, b) the coding and decoding mapping for different tasks may be different, but all the values of a gene key are mapped into a continuous value in $[0, 1]$ and c) the coded genotypes of tasks are simply overlapped to form the chromosome. For example, the first D_j genetic key on the chromosome is the corresponding genotype of task T_j.

Given the above settings, the MO-MFO paradigm is introduced for finding a set of multi-factorial optimal solutions, which is defined as (2).

$$\{x_1, x_2, \ldots, x_j, \ldots, x_K\} = \operatorname{argmin}(\mathbf{F}_1(x), \mathbf{F}_2(x), \ldots, \mathbf{F}_j(x), \ldots, \mathbf{F}_K(x)) \tag{2}$$

where x_j is a feasible solution in $\mathbf{X}_j$. The composite problem may also be referred to as a K factorial optimization problem.

In order to compare candidate solutions during the evolution of MFEAs, the following properties of a individual $p_i,\ i \in \{1, 2, \ldots, |\mathbf{P}|\}$ in the population $\mathbf{P}$, are defined:

1. Factorial Rank: The factorial rank r_j^i of p_i for task T_j is the index of p_i in the list of population members sorted by non-dominated front (NF) and crowd distance (CD) from NSGA-II[10] with respect to T_j. To be specific, p_2 is preferred over p_1 if any one of the following conditions holds: a) $NF_2 < NF_1$ or b) $NF_2 = NF_1$ and $CD_2 > CD_1$.

2. Skill Factor: The skill factor τ_i of p_i is the one task, amongst all other tasks in a K factorial environment, with which the individual is associated. If p_i is evaluated for all tasks, then $\tau_i = \mathrm{argmin}_j(r^i_j),\ j \in \{1, 2, \ldots, K\}$. Skill factor indicates which task is most preferred by p_i. The solutions in a population can be grouped into different sub-populations named task groups according to their skill factors.
3. Scalar Fitness: The scalar fitness φ_i of p_i is given by $\varphi_i = 1/r^i_{\tau_i}$. φ_i is the inverse of the best ranking index of p_i amongst all tasks, which indicates the best fitness of p_i. Performance comparisons can be performed in a simplistic manner with scalar fitness. An individual p_1 can be considered to dominate another individual p_2 in multifactorial sense simply if $\varphi_1 > \varphi_2$.

The vanilla MFEAs[7, 8] are inspired by the bio-cultural models of multifactorial inheritance[11]. Unlike the traditional MOEAs are designed to find a set of Pareto optimal solutions, MFEA are designed to find the a set of global optimal solutions of all constitutive tasks, which means that the trade off between different tasks is not a concern of MFO. Therefore, MFEA splits the population into different task groups according to skill factors of solutions. Solutions in a task group are most suited for the corresponding task. Furthermore, it is possible that the genetic material in a gene pool of a particular task group might be useful for another task. Thus, transfer of genetic materials between tasks may accelerate the overall optimization process.

In the vanilla MFEA, the implicit transfer of genetic material may occur when two parent solutions with different skill factors are selected for reproduction. Then they can have a random matting probability (rmp) to perform SBX crossover[12] and the generated offspring can randomly imitate a skill factor from either parents. These two mechanisms are named as assortative matting and selective imitation[7], respectively.

3 M2M Decomposition based MFEA

3.1 M2M Decomposition

The M2M decomposition approach is first introduced in MOEA/D-M2M[13]. This approach decomposes a MOP into multiple constrained multi-objective subproblems by dividing the objective space into multiple sub-regions with direction vectors.

To be more specific, for a MOP with M nonnegative objectives $f_1, \ldots, f_m$, K direction vectors $\lambda^1, \ldots, \lambda^K \in \mathbb{R}^M_+$ was chosen, usually uniformly distributed. Then the objective space R^M_+ can be divided into K sub-regions $\Omega^1, \ldots, \Omega^K$, where Ω^k $(k = 1, \ldots, K)$ can be defined as (3).

$$\Omega^k = \{u \in \mathbb{R}^M_+ | (a, \lambda^k) \leq (u, \lambda^j),\ \forall j = 1, \ldots, K\} \tag{3}$$

where (u, λ^j) is the acute angle between u and λ^j. In another word, $u \in \Omega^k$ if and only if λ^k has the smallest angle to u amongst all the K direction vectors.

Inspired by the division approach above, the jth constitutive MOP in MFO can be transformed into K constrained multi-objective sub-problems with K uniformly distributed direction vectors $\lambda^1, \ldots, \lambda^K$. The kth sub-problem corresponding to λ^k is defined as (4)

$$\begin{aligned} \text{minimize} \quad & \mathbf{F}_j(x) = (f_{j,1}(x), \ldots, f_{1,M_j}(x))^T \\ \text{subject to} \quad & \mathbf{F}_j(x) \in \Omega_j^k \\ & x \in \mathbb{Y} \end{aligned} \tag{4}$$

where $\mathbb{Y}$ is the aforementioned unified search space. An example of decomposing a objective space of a two objectives optimization problem is provided in Fig 1.

The pseudocode in **Algorithm 1** illustrates the allocation of solutions to sub-problem groups.

Algorithm 1: Solutions Allocation

Input: task group P, direction vectors $\lambda^1, \ldots, \lambda^K$ and size of sub-problem groups S
Output: sub-problem groups $P_1, \ldots, P_K$

```
for j ← 1 to K do
    Initialize P_j with the solutions in P whose objective vectors are
    in Ω^j.
        if (|P_j| < S) then randomly select S − |P_j| solutions from
        P and add them to P_j.
        else if (|P_j| > S) then rank the solutions in P_j using the
        non-dominated sorting method[10] and remove the S − |P_j|
        lowest ranked solutions from P_j.
        end if
end for
return P_1, ..., P_K
```

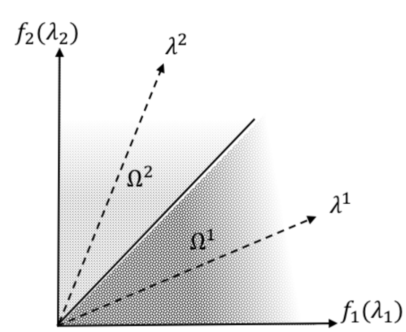

Fig. 1. Direction vectors λ^1,λ^2 divide the objective space into subregions Ω^1, Ω^2

3.2 Constructing Matting Pools

By adopting the M2M decomposition, one can simply decompose constitutive MOPs into different groups of sub-problems, and then solve sub-problems with different sub-populations by MOEAs. However, this approach is not capable of utilizing the most important feature of MFEA, namely genetic transfer, because the individuals in those separated sub-populations or sub-problem groups do not have a chance to reproduce. Prohibiting communication of genetic material between different tasks is undesirable as it constrains exploration and the power of implicit parallelism offered by the entire population[7]. Therefore, a matting pool combining mechanism is needed to ensure genetic transfer in-between different task groups.

In this section, a mechanism of randomly combing sub-problem groups from different task groups to form matting pools is illustrated. To be more specific, K randomly distributed direction vectors are adopted for dividing objective spaces of both constitutive tasks. Note that direction vectors in these two objective spaces might not be the same due to the dimension of spaces might be different. Then K matting pools are generated by randomly combining two sub-problem groups with each sub-problem group randomly picked from different tasks.

4 Pre-Selection

When function evaluation is computationally expensive, one may want to spend evaluations wisely use all evaluations on solutions that are promising, which means the solutions have a fair chance of being selected into the next generation.

Inspired by MOEA/D-SVM[14], the support vector machine is adopted as pre-selection model for MFEA/D-M2M. As illustrated in Fig 2, pre-selection is a procedure that selects unevaluated solutions given their decision variables only. Thus it is possible to select the promising ones for further function evaluation, and discard the rest.

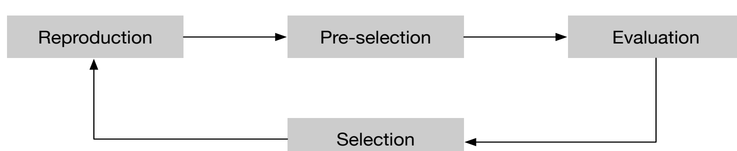

Fig. 2. Pre-selection before actual function evaluation

In order to obtain SVM models that can predict whether or not a solution is promising, the solution set containing solutions of the last generation and their offspring solutions, namely the union set of last generation, are used for training SVM.

Specifically, the decision variable vectors are regarded as feature vectors for representing solutions. and then the solutions in union population are labeled as promising if it survives the last natural selection and labeled as unpromising otherwise. To be more specific, after natural selection takes place, the solutions in current population will be labeled as promising, and solutions in the union set of last generation but not in the current population are labeled as unpromising.

The pseudocode in **Algorithm 2** summarizes the MFEA/D-M2M-SVM as follows.

Algorithm 2: MFEA/D-M2M-SVM

Input:

MOPs, A stopping criterion,
K unit direction vectors $\lambda^1, \ldots, \lambda^K$, S the size of sub-problem group,
Genetic operators and their associated parameters.

Output: a set of multi-factorial optimal solutions.

Geneate $|P|$ individuals in $\mathbb{Y}$ to form initial population P.
for every $p_i \in P$ **do** Assign skill factor τ_i and evaluate p_i for task τ_i only.
end for
Compute scalar fitness φ_i for every p_i based on NF and CD
return $P_1, \ldots, P_K$
while(stopping condition is not satisfied) **do**
    Decompose P into subproblem groups with $\lambda^1, \ldots, \lambda^K$. $\rightarrow$ Refer **Algorithm 1**.
    for every sub-problem group of task one P_i^{T1} **do** Randomly pick a sub-problem group from task two P_j^{T2} to from a matting pool MP_i.
    for every matting pool population MP_i **do**
    while (offspring number $\leq$ matting pool size)
        Pick two parent solutions with binary tournament selection.
        Generate two offspring solutions with assortative matting.
        if (pre-selection $==$ **true**) **then**
            Use trained SVM model for predicting offsprings, SVM-predict(c).
            while (SVM-predict(c) $==$ unpromising) **do** Regenerate offspring.
            Determine skill factor τ_c and evaluate c with task τ_c only.
        end if
    end for
    Combine current population and offsprings into union.
    Perform non-dominated selection on union population.
    Select $|P|$ fittest solutions in union to form the current population.
    if (trainSVM $==$ **true**) Label solutions in last union population and train SVM model with them.
return current population P

---

5 Experimental Studies

5.1 Benchmark Problems

The Multi-Objective Multi-Factorial Optimization (MO-MFO) benchmark problem sets[1] are adopted in the experimental studies, which are the same benchmarks used in CEC2017 MFO competition. All the details of these benchmark problem sets can be found in [1, 15].

The MO-MFO benchmark contains nine different problem sets. Each problem set contains two constitutive tasks, and each task is a two- or three-objective minimization problem. For more detailed information, readers are referred to [1, 15].

The degree of intersection of the global optima: Compete Intersection (CI): the global optimal of two constitutive tasks are identical in the unified search space with respect to all variables. Partial Intersection (PI): the global optimal of the two tasks are identical in the unified search space with respect to a subset of variables only, and are different with respect to the remaining variables. No Intersection (NI): the global optimal of the two tasks are different with respect to all variables.

The similarity of the fitness landscape: High Similarity (HS): problem sets with Spearman's rank correlation coefficient[15] $R_s \geq 0.8$. Medium Similarity (MS): problem sets with $0.8 > R_s > 0.2$. Low Similarity (LS): problem sets with $R_s \leq 0.2$.

Therefore, the nine problem sets in MFO benchmark, are named by their characteristic as follows: CIHS, CIMS, CILS, PIHS, PIMS, PILS, NIHS, NIMS, NILS.

5.2 Experimental Settings

In order to demonstrate the effectiveness of MFEA/D-M2M, several experiments are conducted with comparison of the vanilla MFEA. The experimental settings are as follows:

1. Population size: 200. Number of direction vectors for each task: 10. Size of each sub-problem population: 10.
2. Random mating probability: 0.9.
3. Deferential evolution (DE) crossover probability (CR): 0.9. Deferential evolution crossover factor (F): 0.9.
4. Simulated binary crossover (SBX) probability: 0.9. Distribution index for SBX: 20.
5. Polynomial mutation probability: 1/D (D is the dimensionality of the unified representation space). Distribution index for mutation: 20.
6. The number of function evaluations at every generation: 200. Maximum function evaluation: 200,000.

The deferential evolution crossover is used for problem sets with complete or no intersection of global optimal, and the simulated binary crossover is used for problem sets with partial intersection of global optimal.

5.3 Performance Metric

To compare the performance of the algorithms, a popular metric - inverted generation distance (IGD)[16] is adopted. The definitions of IGD is given by (5).

$$\mathrm{IGD}(A, P^*) = \frac{1}{|P|}\sqrt{\sum_{\mathbf{x}\in P^*}(\min_{\mathbf{y}\in A} d(\mathbf{x},\mathbf{y}))^2} \tag{5}$$

where A is a set of normalized non-dominated objective vectors that are obtained for a task T_i by the algorithm, P^* is the set of uniformly distributed normalized objective vectors over the PF of T_i, and $d(\mathbf{x}, \mathbf{y})$ is the Euclidean distance between $\mathbf{x}$ and $\mathbf{y}$ in the normalized objective space.

If $|P^*|$ is large enough to represent the PF, the $\mathrm{IGD}(A, P^*)$ can measure both convergence and diversity of A to an extent.

To illustrate the convergence speed of an algorithm, the convergence curve are plotted to describe the trend of IGD values over the number of generations.

5.4 Experimental Results and Discussions

Table 1 summarized the performances of the vanilla MFEA, MFEA/D-M2M and MFEA/D-M2M-SVM on nine benchmark problem sets, in terms of average IGD values over 30 independent runs.

In Table 1, both MFEA/D-M2M and MFEA/D-M2M-SVM have achieve better performances on six problem sets, which are problem sets with complete or no intersection of global optimal. On the other problem sets with partial intersection, the proposed algorithms have achieved comparable result than the vanilla MFEA.

Table 2 shows the h-value (h) and p-value (p) from the T-test of IGD values among MFEA and MFEA/D-M2M-SVM. The significancy level of the T-test is set as 0.05.

On problem sets with complete intersection or no intersection of global optimal, such as in Fig 3 and Fig 4, MFEA/D-M2M and MFEA/D-M2M-SVM achieve better IGD convergence speed and average IGD values than the vanilla MFEA at the end of evolution. Especially in Fig 3, MFEA/D-M2M-SVM outperforms both MFEA/D-M2M and MFEA in terms of both average IGD value and convergence speed.

Table 1. Average IGD values of MFEA, MFEA/D-M2M and MFEA/D-M2M-SVM

ProblemSet-Task	MFEA	MFEA/D-M2M	MFEA/D-M2M-SVM
CIHS-T1	4.2878E-04	1.8083E-04	**1.7815E-04**
CIHS-T2	2.7627E-03	**4.9795E-04**	5.1315E-04
CIMS-T1	4.9917E-02	1.9344E-04	**1.8137E-04**
CIMS-T2	8.4888E-03	2.5233E-04	**1.9776E-04**
CILS-T1	2.6345E-04	2.5940E-04	**2.5197E-04**
CILS-T2	1.8532E-04	1.8741E-04	**1.8275E-04**
PIHS-T1	9.9586E-04	1.0479E-03	**9.8242E-04**
PIHS-T2	**3.5382E-02**	6.6400E-02	4.5432E-02
PIMS-T1	**2.9505E-03**	3.4824E-03	4.0440E-03
PIMS-T2	**9.7286E+00**	1.4166E+01	1.3686E+01
PILS-T1	**3.3238E-04**	3.6724E-04	3.5442E-04
PILS-T2	**1.0814E-02**	1.0777E-02	1.1155E-02
NIHS-T1	1.5552E+00	1.4929E+00	**1.4925E+00**
NIHS-T2	4.9591E-04	2.4923E-04	**2.4837E-04**
NIMS-T1	3.3532E-01	**1.5402E-01**	1.5517E-01
NIMS-T2	3.4444E-02	6.9613E-04	**3.0415E-04**
NILS-T1	8.3985E-04	8.9249E-04	**8.6783E-04**
NILS-T2	6.4326E-01	6.4782E-01	**6.4183E-01**

Table 2. T-test values of IGD among MFEA and MFEA/D-M2M-SVM

	h	p		h	p		h	p
CIHS-T1	1	1.1633E-13	PIHS-T1	0	7.0901E-01	NIHS-T1	1	8.3695E-16
CIHS-T2	1	9.9410E-21	PIHS-T2	1	4.8330E-04	NIHS-T2	1	9.3051E-14
CIMS-T1	1	2.9975E-04	PIMS-T1	0	7.8677E-02	NIMS-T1	1	1.5667E-03
CIMS-T2	1	2.8867E-03	PIMS-T2	1	3.5863E-05	NIMS-T2	1	9.0970E-03
CILS-T1	0	1.3100E-01	PILS-T1	0	2.8317E-01	NILS-T1	0	1.2037E-01
CILS-T2	1	1.0446E-02	PILS-T2	0	9.4582E-01	NILS-T2	1	2.0513E-21

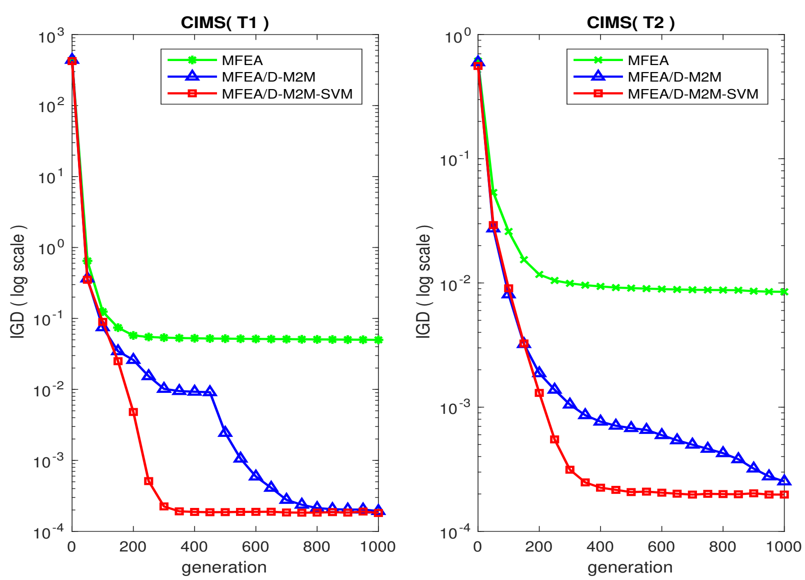

Fig. 3. Convergence curves on CIMS

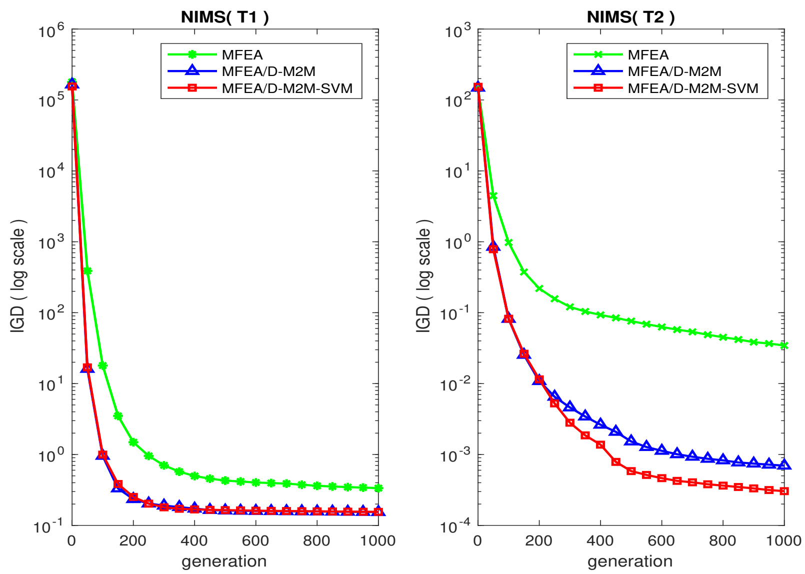

Fig. 4. Convergence curves on NIMS

6 Conclusions

This paper proposes a M2M decomposition based multi-factorial evolutionary algorithm for solving CEC2017 MFO Benchmark problems. The MFEA/D-M2M adopts the M2M approach to decompose multi-objective optimization problems into multiple constrained sub-problems to enhance the diversity of population and convergence of sub-regions. A SVM augmented version is also implemented to improve its performance. The experimental results demonstrate that the proposed algorithms have achieved better performance on both problem sets with complete intersection and no intersection of global optimal, and comparable results on the other three problem sets. The future work includes studying the decomposition approach in MFEAs and solving several real-world optimization problems to further demonstrate the effectiveness of MFEA/D-M2M.

References

1. Yuan Yuan, Yew-Soon Ong, Liang Feng, AK Qin, Abhishek Gupta, Bingshui Da, Qingfu Zhang, Kay Chen Tan, Yaochu Jin, and Hisao Ishibuchi. Evolutionary multitasking for multi-objective continuous optimization: Benchmark problems, performance metrics and baseline results. Technical report, Nanyang Technological University, 2016.
2. Thomas Back, Ulrich Hammel, and H-P Schwefel. Evolutionary computation: Comments on the history and current state. *IEEE transactions on Evolutionary Computation*, 1(1):3–17, 1997.

3. A. Gupta, Y. S. Ong, B. Da, L. Feng, and S. D. Handoko. Landscape synergy in evolutionary multitasking. In *2016 IEEE Congress on Evolutionary Computation (CEC)*, pages 3076–3083, July 2016.
4. Bingshui Da, Abhishek Gupta, Yew-Soon Ong, and Liang Feng. Evolutionary multitasking across single and multi-objective formulations for improved problem solving. In *Evolutionary Computation (CEC), 2016 IEEE Congress on*, pages 1695–1701. IEEE, 2016.
5. Abhishek Gupta, Yew Soon Ong, B Da, L Feng, and Stephanus D Handoko. Landscape synergy in evolutionary multitasking. In *Evolutionary Computation (CEC), 2016 IEEE Congress on*, pages 3076–3083. IEEE, 2016.
6. Abhishek Gupta and Yew-Soon Ong. Genetic transfer or population diversification? deciphering the secret ingredients of evolutionary multitask optimization. *IEEE Symposium Series on Computational Intelligence (SSCI)*, pages 1–7, 2016.
7. Abhishek Gupta, Yew-Soon Ong, and Liang Feng. Multifactorial evolution: toward evolutionary multitasking. *IEEE Transactions on Evolutionary Computation*, 20(3):343–357, 2016.
8. Abhishek Gupta, Yew-Soon Ong, Liang Feng, and Kay Chen Tan. Multiobjective multifactorial optimization in evolutionary multitasking. *IEEE transactions on cybernetics*, 47(7):1652–1665, 2017.
9. Lei Zhou, Liang Feng, Jinghui Zhong, Yew-Soon Ong, Zexuan Zhu, and Edwin Sha. Evolutionary multitasking in combinatorial search spaces: A case study in capacitated vehicle routing problem. In *2016 IEEE Symposium Series on Computational Intelligence (SSCI)*, pages 1–8. IEEE, 2016.
10. Kalyanmoy Deb, Amrit Pratap, Sameer Agarwal, and TAMT Meyarivan. A fast and elitist multiobjective genetic algorithm: Nsga-ii. *IEEE transactions on evolutionary computation*, 6(2):182–197, 2002.
11. Yew-Soon Ong and Abhishek Gupta. Evolutionary multitasking: a computer science view of cognitive multitasking. *Cognitive Computation*, 8(2):125–142, 2016.
12. Ram Bhushan Agrawal, K Deb, and RB Agrawal. Simulated binary crossover for continuous search space. *Complex systems*, 9(2):115–148, 1995.
13. Hai-Lin Liu, Fangqing Gu, and Qingfu Zhang. Decomposition of a multiobjective optimization problem into a number of simple multiobjective subproblems. *IEEE Transactions on Evolutionary Computation*, 18(3):450–455, 2014.
14. Xi Lin, Qingfu Zhang, and Sam Kwong. A decomposition based multiobjective evolutionary algorithm with classification. In *Evolutionary Computation (CEC), 2016 IEEE Congress on*, pages 3292–3299. IEEE, 2016.
15. Bingshui Da, Yew-Soon Ong, Liang Feng, AK Qin, Abhishek Gupta, Zexuan Zhu, Chuan-Kang Ting, Ke Tang, and Xin Yao. Evolutionary multitasking for single-objective continuous optimization: Benchmark problems, performance metric, and baseline results. *Nanyang Technological University, Singapore, Tech. Rep*, 2016.
16. David A Van Veldhuizen and Gary B Lamont. Multiobjective evolutionary algorithm research: A history and analysis. Technical report, Citeseer, 1998.