A Cooperative Co-evolutionary Algorithm for Large-Scale Multi-Objective Optimization Problems

Minghan Li
Xidian University
Xi'an, China
lmh0921@163.com

Jingxuan Wei
Xidian University
Xi'an, China
wjx@xidian.edu.cn

ABSTRACT

A wide range of real-world problems are multi-objective optimization problems (MOPs). Multi-objective evolutionary algorithms (MOEAs) have been proposed to solve MOPs, but the search process deteriorates with the increase of MOPs' dimension of decision variables. In order to solve the problem, firstly, the decision variables are divided into different groups by adopting a fast interdependency identification algorithm; secondly, a novel cooperative co-evolutionary algorithm is used to solve MOPs. Experiment results on large-scale problems show that the proposed algorithm is effective.

CCS CONCEPTS

• **Applied computing → Multi-criterion optimization and decision-making**;

KEYWORDS

Evolutionary multi-objective optimization, Multi-objective optimization, Large-scale optimization

ACM Reference Format:
Minghan Li and Jingxuan Wei. 2018. A Cooperative Co-evolutionary Algorithm for Large-Scale Multi-Objective Optimization Problems. In *GECCO '18 Companion: Genetic and Evolutionary Computation Conference Companion, July 15–19, 2018, Kyoto, Japan.* ACM, New York, NY, USA, 6 pages. https://doi.org/10.1145/3205651.3208250

1 INTRODUCTION

Over the past years, a number of investigations has been proposed to solve MOPs. Many MOEAs have proven their effectiveness in solving MOPs, e.g., [5, 6, 27, 31, 32]. However, traditional MOEAs' search ability is severely deteriorated when the number of objectives increases, especially for many-objective optimization problems (MaOPs) which have more than three objectives. Recently, MaOPs have been widely discussed and people propose much remarkable work, e.g., [5, 8, 11–13, 15, 17, 23, 27]. In fact, empirical evidence indicates that most of the classical MOEAs significantly decrease their effectiveness when the number of decision variables of the MOP increases [7], and this kind of problems is called large-scale multi-objective optimization problems (LSMOPs). In the last several years, several algorithms are proposed to solve LSMOPs [1, 18, 29, 30], but work is still little comparing to MaOPs'.

Fortunately, in the field of single objective optimization problems (SOPs), large-scale global optimization (LSGO) is widely studied. Some remarkable work has been proposed, e.g., [2, 20, 22, 26]. The variable-grouping mechanism and the cooperate mechanism are the basic ideas of LSGO, which enlighten us to solve LSMOPs.

This paper proposes a cooperative co-evolutionary algorithm for solving LSMOPs. This algorithm firstly divides decision variables into different groups by adopting a fast interdependency identification algorithm which is proposed by [9]. Then we design a novel cooperative co-evolution algorithm to solve LSMOPs based on previous groups. Experiments are done and they prove the effectiveness of the proposed algorithm.

The reminder of this paper is organized as follows. Section 2 presents background. Section 3 describes the proposed algorithm. Section 4 presents the test suite. Finally, conclusions and experimental results are made in section 5.

2 BACKGROUND

2.1 BASIC CONCEPTS

2.1.1 Multi-objective Optimization Problems. Without loss of generality, a multi-objective optimization problem can be described as (1) [4],

$$f(x) = \begin{cases} \min f(x) = (f_1(x), f_2(x), \ldots, f_m(x)) \\ s.t.\ x \in \Omega \end{cases} \tag{1}$$

where $x = (x_1, x_2, \ldots, x_n)$ is the decision vector, Ω is the search space, $f_i(x)$ is the i-th objective function in the objective space. We call the problem many-objective optimization problem when the number of objectives is more than three.

2.1.2 Pareto dominance. A solution $x = (x_1, x_2, \ldots, x_n)$ is said to dominate (denoted by $\prec$) another solution $y = (y_1, y_2, \ldots, y_n)$ if and only if $f(x)$ is partially less than $f(y)$. That means, $\forall m \in \{1, \ldots, M\}$, we have $f_m(x) \leq f_m(y)$ and $\exists m \in \{1, \ldots, M\}$, where $f_m(x) < f_m(y)$.

2.1.3 Pareto optimal solution. A solution $x = (x_1, x_2, \ldots, x_n)$ is said to be an optimal solution if and only if there is no $y = (y_1, y_2, \ldots, y_n)$ that y dominates x with respect to solution space.

2.1.4 Pareto optimal set. For a given MOP $f(x)$, the Pareto optimal set is $P = \{x \in \Omega | \nexists y \in \Omega, f(y) \prec f(x)\}$, which is also called non-dominated solutions.

GECCO '18 Companion, July 15–19, 2018, Kyoto, Japan

ACM ISBN 978-1-4503-5764-7/18/07…$15.00
https://doi.org/10.1145/3205651.3208250

2.1.5 Pareto front. Given MOP $f(x)$ and its Pareto optimal set P, the Pareto front is $PF = \{f(x), x \in P\}$.

2.1.6 Variable Interdependency Identification. Suppose $f(x)$ is a given objective. If x_i interacts with x_j for this objective, then we have a function,

$$\frac{\partial f}{\partial x} = g(x_{sub}) \tag{2}$$

where $x_j \in x_{sub}$, $x_{sub} \subseteq x$,$g(x)$ is another function. According to the Newton-Leibniz formula, we have

$$f(x)|_{x_i=b} - f(x)|_{x_i=a} = \int_a^b \frac{\partial f}{\partial x_i}\, dx \tag{3}$$

where $f(x)|_{x_i=b}$=$f(x_1, \ldots, x_{i-1}, b, \ldots, x_n)$. So we have

$$f(x)|_{x_i=b} - f(x)|_{x_i=a} = \int_a^b g(x_{sub})\, dx \tag{4}$$

As $x_j \in x_{sub}$, if we perturb x_j, then the difference value obtained from the left part of (4) would change. On the contrary, if x_i is separable, $g(x_{sub})$ is a function of x_i or a constant and the perturbation of all other variables cannot affect the left part of (4). Therefore, by simply investigating a variable's difference value with a perturbation method, the variable interdependency information of a problem can be obtained [9, 20, 22].

2.2 Related Work

2.2.1 Cooperative Coevolution. The framework of cooperative coevolution (CC) is originally introduced by Potter and De Jong [21]. Many cooperative coevolutionary numerical optimization algorithms consist of three basic ingredients [2]. The first is a decomposition method which is used to divide the N-dimensional decision vector into groups of variables $G_1 \ldots G_m$. Each such group is optimized with a separate subpopulation. The second is the cooperation step. A representative individual from each of the other subpopulations is selected. Then a population of complete N-dimensional candidate solutions is constructed by concatenating the representatives to each element of the current subpopulation. The last is the optimizer. The optimizer is applied to the population for optimizing the decision variables in the current group. By combing with different EAs, a variant of algorithms has been proposed under CC, e.g., [14, 16]. However, little work applies CC to solve MOPs.

2.2.2 Variable Grouping Strategy. As demonstrated above, CC needs a decomposition method to divide the decision vector into groups of variables. In SOPs, many algorithms have been developed. [26] proposes a new cooperative coevolution framework that is capable of optimizing large-scale nonseparable problems. A random grouping scheme and adaptive weighting are introduced in problem decomposition and coevolution. It is particularly good at dealing with nonseparable problems. However, as the algorithm is designed to solve nonseparable problems, they use a random grouping strategy and this may works not well on separable problems.

Later, detection-based static decomposition strategies are developed, e.g., [2, 20, 22]. However, both of them require heavy computational cost in the identification process, resulting in unsatisfactory performance in the optimization.

2.2.3 Related Work on LSMOPs. CCGDE3 [1] is a cooperative coevolution algorithm combined with GDE3 for solving MOPs with a large number of decision variables. The results confirm that it is effective and efficient in tackling LSMOPs. However, their scheme assigns each decision variable to its corresponding group in a random way, since they assume this will increase the chance of optimizing some interacting variables together.

In [30], variable grouping is used in the so-called Weighted Optimization Framework to tackle LSMOPs without using CC. A set of weights is applied to the groups of decision variables. The original decision variables and the weight-variables are optimized in turns. The authors report a superior performance compared to the SMPSO [19] and NSGA-II [6] algorithms. However, they use differential grouping (DG) mechanism [20] to group variables that can become computationally expensive. Besides, in their work only one of the objective functions is considered in the DG algorithm.

Ma et al. propose a decision variable analysis based MOEA, known as MOEA/DVA, for solving large-scale MOPs [18]. The decision variable analysis method based on dominance relationships is designed to divide the decision variables into three groups. Experiments show that this decision variable analysis method works efficiently on LSMOPs with two or three objectives. MOEA/DVA simply treats the variables related to both convergence and diversity as diversity related variables, as the variable analysis strategy is not able to further distinguish them. Following the basic idea of MOEA/DVA, Zhang et al. propose a decision variable clustering based evolutionary algorithm for LSMOPs [29]. Their empirical observation indicates that these decision variables still can be further divided into two groups. Based on the grouping results, the groups of variables are optimized separately. It has been shown that decision variable clustering is effective for solving LSMOPs. However, their grouping strategy have rather high computational cost.

In order to deal with the limitations of above algorithms, this paper proposes a cooperative coevolutionary algorithm for LSMOPs. We adopt a fast interdependency identification algorithm with which decision variables are fastly divided into different groups, and present a novel cooperative coevolutionary approach based on these groups.

3 PROPOSED ALGORITHM

3.1 Fast Interdependency Identification

We adopt the fast interdependency identification approach of [9], and extend it to MOPs. The basic idea is based on (4). For a simple example, let $f(x) = x_1x_2$, where $f(x)$ is an objective and it has two interacting decision variables, we have $\frac{\partial f}{\partial x_1} = x_2$, so $g(x_{sub}) = x_2$, $D_{x_1}(x) = f(x)|_{x_1=b} - f(x)|_{x_1=a} = \int_a^b x_2\, dx$. So if we perturb x_2 and get the new solution x', $D_{x_1}(x') = \int_a^b x_2{}'\, dx$ would be different from $D_{x_1}(x)$. Therefore, by investigating a variable's difference value with a perturbation method, we can identify the interdependency of decision variables.

The first stage is identification of separable and nonseparable variables. Let $D_{x_i}(x) = f(x)|_{x_i=a+\Delta} - f(x)|_{x_i=a}$. By perturbing all other variables, a new solution x' is produced. Then a new difference value $D_{x_i}(x')$ is calculated by $D_{x_i'}(x) = f(x')|_{x_i=a+\Delta} -$

$f(x')|_{x_i=a}$. If x_i is a separable variable, then $D_{x_i}(x')$ equals to $D_{x_i}(x)$. In the algorithm, a predefined threshold ϵ_1 is used as an allowable error for the equality. If $|D_{x_i}(x') - D_{x_i}(x)| \leq \epsilon_1$, x_i is viewed as a separable variable for the objective $f(x)$. The application to MOPs examines x_i for each objective. Only when x_i is separable for each objective, this decision variable is viewed as a separable variable of MOP and divided into a single group.

The second stage is identification of the interdependency information of nonseparable variables. Firstly, choose a nonseparable variable and perturb it to produce a perturbed solution x'. Secondly, put the variables that satisfy $|D_{x_i}(x') - D_{x_i}(x)| > \epsilon_2$ for any objective into the subcomponent group. Thirdly, the variables newly captured are perturbed together to find more variables belonging to this group. When no variables can be added to this group, another group is formed. This process is repeated and finally the subcomponents are established. Each group in the first stage can also be viewed as a subcomponent and it contains only one decision variable. The procedure of grouping decision variables is shown in Algorithm 1.

[9] shows that as an approach that covers both of separability and nonseparability forms, this approach's time complexity is $O(N + kn)$, which is more efficient than similar approaches like [20, 22], where N is the dimension of the decision variables, n is the number of nonseparable variables and k is the number of nonseparable subcomponents.

Algorithm 1: Fast Interdependency Identification

Input : Objectives Fs,
predefined threshold ϵ_1, ϵ_2;
Output: seps, groups;

```
seps ←[];
nonseps ←[];
allDimens ← [1, 2, ..., N];
randomly generate a decision vector x ;
D_1s ← calDiff(Fs,x,allDimens);
for each dimen ∈ allDimens do
    x' ← perturb(x);
    D_2s ← calDiff(Fs,x',dimen);
    if |D_2s(i) − D_1s(i)| ≤ ε_1 for all Fs then
        seps ← seps ∪ i;
    else
        nonseps ← nonseps ∪ i;
    end
end
groups = {};
j = 1;
while not empty nonseps do
    pdim ← nonseps(1);
    groups{j} ← nonseps(1);
    while both pdim and groups not empty do
        x' ← perturb(pdim,x);
        D_3s ← calDiff(Fs,x',nonseps);
        pdim ← [];
        for ns ∈ nonseps do
            if |D_2s(i) − D_1s(i)| > ε_2 exists in Fs then
                groups{j} ← groups{j} ∪ i;
                pdim ← pdim ∪ i;
            end
        end
        nonseps ← nonseps − groups{j};
    end
    if length(groups{j} == 1) then
        seps ← seps ∪ groups{j};
    else
        j = j + 1;
    end
end
```

3.2 Cooperative Coevolution for LSMOPs

As demonstrated above, CC firstly need to use a decomposition method to divide the N-dimensional decision vector into groups of variables, and this is done in Algorithm 1. Then it is the cooperative step, that is to say a representative individual should be picked from each sub-group. In return, these representative individuals are used to evaluate the individuals of each sub-group. At the beginning, the representative individuals are picked randomly, then they are picked from the best solutions of each sub-group.

In this paper, we present a novel cooperative coevolution algorithm and a simple example with four decision variables for showing the procedure is shown in Figure 1. This cooperative coevolution algorithm can be integrated into an MOEA and the MOEA is used to evolve individuals in the subgroup. The initial population are firstly non-dominated sorted and we randomly pick one individual from the first PF as the representative. With the representative individual, we calculate each individual's objective values. In each sub-group, then use an MOEA to evolve these individuals and the representative individual is updated too. The process is repeated until the end of cycles. The main loop of the proposed cooperative coevolution algorithm (CCLSM) is shown in Algorithm 2.

4 EXPERIMENT DESIGN

To examine the performance of the proposed algorithm CCLSM, we integrate CCLSM into NSGA-II [6] and basic IBEA [31], which are named CCL-NSII and CCL-IBEA. We adopt WFG [10], UF [28] and LSMOP [3] test problems. The LSMOP test problems are designed to have large number decision variables, which is exactly suitable for our experiments.

In order to examine the performance of CCLSM, we compare it with some popular MOEAs for MOPs with large-scale decision variables. They are NSGA-II [6], basic IBEA [31] where the fitness scaling factor is set to 0.05, and NSGA-III [5]. We use the platform PlatEMO [24] to conduct our experiments.

In the fast interdependency identification step, based on the empirical study of [9], ϵ_1 and ϵ_2 are generally set to the same and the performance of the grouping method is not sensitive to the parameters. Here we set both ϵ_1 and ϵ_2 to 0.01; The perturbing value of decision variable is also set to 0.01. In evolutionary process, the

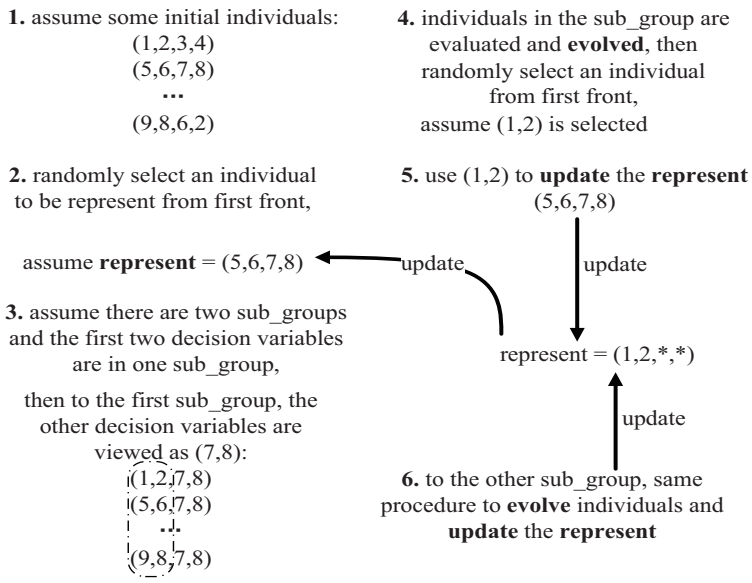

Figure 1: A CC example with four decision variables.

Algorithm 2: Main Loop of CCLSM

Input : Objectives Fs,
predefined threshold ϵ_1, ϵ_2;
Output : SolutionSet

```
[seps, groups] ← fastIdIdentif(Fs,ε1,ε2);
sub-groups ← combineGroup(seps, groups);
Pops ← initialPop();
Represent ← Pops(randomselect);
while not termination do
    for sub-group ∈ sub-groups do
        subPop ← getSubPop(Pops,sub-group);
        subPop ← MOEA(subPop, Represent);
        Represent ←update(Represent, subPop);
    end
end
```

population size is 100; the crossover probability is 0.9; the mutation probability is 1/n (where n is the number of decision variables). The maximum number of evaluations is set to 50000 in all experiments.

5 RESULTS

We adopt the Inverted Generational Distance (IGD) that is a variant of GD [25] for our experiments. The smaller the IGD value is, the better the result is, and results are shown below.

5.1 IGD Results on Test Problems

Test problems' average results for 15 times experiments are shown in Table 1.

As shown in Table 1, for example, the UF5 problems with 2 objectives and 100 decision variables, NSGA-II's result is 0.7123, IBEA's result is 0.9959, NSGA-III's result is 0.4815. However, after integrating the proposed algorithm, CCL-NSII's result is 0.2285 and CCL-IBEA's result is 0.2187, which greatly improve other popular MOEAs' performance. Other results also prove the proposed algorithm CCLSM's effectiveness.

5.2 The Pareto Front of the MOEAs

We also compare the Pareto front of these algorithms. For LSMOP9 test problems with 2 objectives and 100 decision variables, the Pareto front comparisons are shown in Figure 2. For WFG3 test problems with 10 objectives and 200 decision variables, the Pareto front comparisons are shown in Figure 3. The number of evaluations is 50000 in all experiments.

From Figure 2, the results show that all the test algorithms search some local solutions. Solutions of NSGA-II, IBEA and NSGA-III deviate from the true Pareto front and are poorly distributed. However, CCL-NSII and CCL-IBEA are both close to the left part of LSMOP9's true Pareto front, and they search better solutions than NSGA-II, IBEA and NSGA-III on LSMOP9 for diversity.

From Figure 3, results obviously show that CCL-NSII and CCL-IBEA achieve better solutions on WFG3 than the other MOEAs, as their Pareto fronts are more approximate to the true Pareto front.

Figures show that after integrating CCLSM, MOEAs achieve better solutions. Results prove the effectiveness of CCLSM.

6 CONCLUSIONS

The goal of this paper is to investigate an algorithm LSMOPs. The goal is successfully achieved by introducing a fast interdependency identification algorithm and a cooperative coevolution algorithm for LSMOPs with the obtained sub-groups. In order to examine the effectiveness of the proposed algorithm, we integrate the proposed algorithm CCLSM into NSGA-II and IBEA to obtain two algorithm CCL-NSII and CCL-IBEA. Experiments are carried out

Table 1: Average IGD Results on test problems

Test Problems	Objective Number	Variable Numbrer	NSGA-II	IBEA	NSGA-III	CCL-NSII	CCL-IBEA
WFG2	5	100	0.8724	1.3963	0.8273	0.6664	0.7332
WFG2	5	200	0.9067	1.4905	0.8390	0.7948	0.7839
WFG3	10	200	3.3684	4.8088	2.3518	2.1588	1.3146
WFG3	7	300	1.7390	2.2983	1.4187	1.3246	0.9021
UF5	2	100	0.7123	0.9959	0.4815	0.2285	0.2187
UF5	2	200	0.8545	1.0721	0.7126	0.5912	0.5732
LSMOP1	5	100	4.5605	0.5227	1.1894	0.5270	0.2653
LSMOP1	3	200	1.7494	0.9006	0.9233	0.3582	0.3257
LSMOP5	5	100	12.6995	1.0821	3.7012	0.6463	0.7309
LSMOP5	4	200	13.9413	3.1324	5.2335	0.8181	1.0302
LSMOP9	5	100	22.0062	12.3691	21.8193	6.6637	4.3553
LSMOP9	5	200	24.8746	14.0627	19.8327	7.1205	4.7233

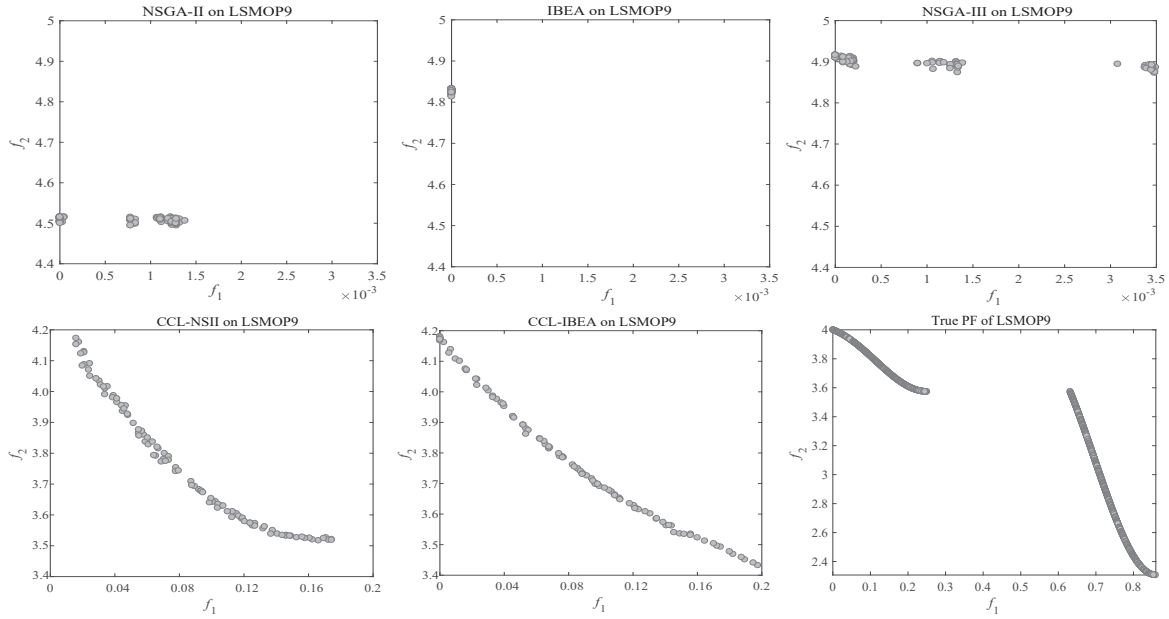

Figure 2: The Pareto Front comparisons for LSMOP9.

to compare the performance of CCL-NSII, CCL-IBEA and other popular MOEAs for MOPs with large-scale decision variables. From the experimental results, the following conclusions can be drawn: the proposed algorithm CCLSM successfully solves LSMOPs, the results of MOEAs after integrating CCLSM are better than several popular MOEAs'.

There are several relevant directions to pursue in the future. First, it is desirable to study in more detail the variable grouping algorithm. Second, we also consider developing new cooperative coevolution mechanism. Finally, it would be interesting to propose other MOEAs to get better results.

ACKNOWLEDGMENTS

This work is supported by the national science foundation of China (jj0500120310).

REFERENCES

[1] Luis Miguel Antonio and Carlos A Coello Coello. 2013. Use of cooperative coevolution for solving large scale multiobjective optimization problems. In *Evolutionary Computation (CEC), 2013 IEEE Congress on*. IEEE, 2758–2765.

[2] Wenxiang Chen, Thomas Weise, Zhenyu Yang, and Ke Tang. 2010. Large-scale global optimization using cooperative coevolution with variable interaction learning. In *International Conference on Parallel Problem Solving from Nature*. Springer, 300–309.

[3] Ran Cheng, Yaochu Jin, Markus Olhofer, et al. 2017. Test problems for large-scale multiobjective and many-objective optimization. *IEEE transactions on cybernetics* 47, 12 (2017), 4108–4121.

[4] Kalyanmoy Deb. 2001. *Multi-objective optimization using evolutionary algorithms*. Vol. 16. John Wiley & Sons.

[5] Kalyanmoy Deb and Himanshu Jain. 2014. An evolutionary many-objective optimization algorithm using reference-point-based nondominated sorting approach, part I: Solving problems with box constraints. *IEEE Trans. Evolutionary Computation* 18, 4 (2014), 577–601.

[6] Kalyanmoy Deb, Amrit Pratap, Sameer Agarwal, and TAMT Meyarivan. 2002. A fast and elitist multiobjective genetic algorithm: NSGA-II. *IEEE transactions on evolutionary computation* 6, 2 (2002), 182–197.

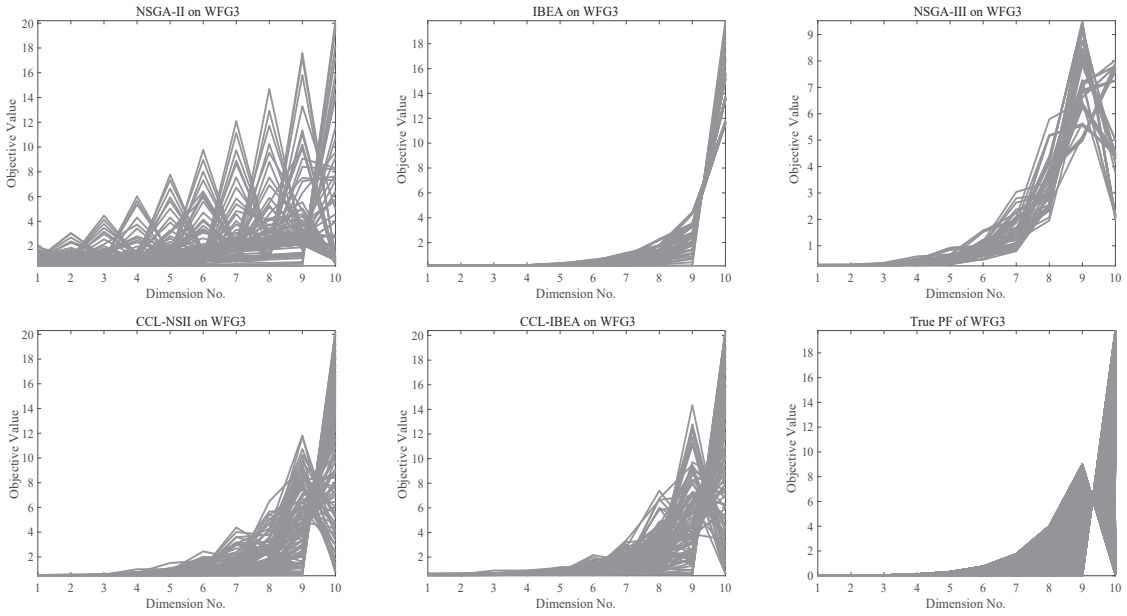

Figure 3: The Pareto Front comparisons for WFG3.

[7] Juan J Durillo, Antonio J Nebro, Carlos A Coello Coello, José García-Nieto, Francisco Luna, and Enrique Alba. 2010. A study of multiobjective metaheuristics when solving parameter scalable problems. *IEEE Transactions on Evolutionary Computation* 14, 4 (2010), 618–635.

[8] Raquel Hernández Gómez and Carlos A Coello Coello. 2013. MOMBI: A new metaheuristic for many-objective optimization based on the R2 indicator. In *Evolutionary Computation (CEC), 2013 IEEE Congress on*. IEEE, 2488–2495.

[9] Xiao-Min Hu, Fei-Long He, Wei-Neng Chen, and Jun Zhang. 2017. Cooperation coevolution with fast interdependency identification for large scale optimization. *Information Sciences* 381 (2017), 142–160.

[10] Simon Huband, Philip Hingston, Luigi Barone, and Lyndon While. 2006. A review of multiobjective test problems and a scalable test problem toolkit. *IEEE Transactions on Evolutionary Computation* 10, 5 (2006), 477–506.

[11] Kousuke Izumiya and Masaharu Munetomo. 2017. Multi-objective Evolutionary optimization based on Decomposition with Linkage Identification considering monotonicity. In *Evolutionary Computation (CEC), 2017 IEEE Congress on*. IEEE, 905–912.

[12] Miqing Li, Shengxiang Yang, and Xiaohui Liu. 2014. Shift-based density estimation for Pareto-based algorithms in many-objective optimization. *IEEE Transactions on Evolutionary Computation* 18, 3 (2014), 348–365.

[13] Wenwen Li, Ender Özcan, Robert John, John H Drake, Aneta Neumann, and Markus Wagner. 2017. A modified indicator-based evolutionary algorithm (mIBEA). In *Evolutionary Computation (CEC), 2017 IEEE Congress on*. IEEE, 1047–1054.

[14] Xiaodong Li and Xin Yao. 2012. Cooperatively coevolving particle swarms for large scale optimization. *IEEE Transactions on Evolutionary Computation* 16, 2 (2012), 210–224.

[15] Xi Lin, Qingfu Zhang, and Sam Kwong. 2017. An efficient batch expensive multi-objective evolutionary algorithm based on Decomposition. In *Evolutionary Computation (CEC), 2017 IEEE Congress on*. IEEE, 1343–1349.

[16] Jinpeng Liu and Ke Tang. 2013. Scaling up covariance matrix adaptation evolution strategy using cooperative coevolution. In *International Conference on Intelligent Data Engineering and Automated Learning*. Springer, 350–357.

[17] Ruochen Liu, Xiaolin Song, Lingfen Fang, and Licheng Jiao. 2017. An r-dominance-based preference multi-objective optimization for many-objective optimization. *Soft Computing* 21, 17 (2017), 5003–5024.

[18] Xiaoliang Ma, Fang Liu, Yutao Qi, Xiaodong Wang, Lingling Li, Licheng Jiao, Minglei Yin, and Maoguo Gong. 2016. A multiobjective evolutionary algorithm based on decision variable analyses for multiobjective optimization problems with large-scale variables. *IEEE Transactions on Evolutionary Computation* 20, 2 (2016), 275–298.

[19] Antonio J Nebro, Juan José Durillo, Jose Garcia-Nieto, CA Coello Coello, Francisco Luna, and Enrique Alba. 2009. Smpso: A new pso-based metaheuristic for multi-objective optimization. In *Computational intelligence in miulti-criteria decision-making, 2009. mcdm'09. ieee symposium on*. IEEE, 66–73.

[20] Mohammad Nabi Omidvar, Xiaodong Li, Yi Mei, and Xin Yao. 2014. Cooperative co-evolution with differential grouping for large scale optimization. *IEEE Transactions on evolutionary computation* 18, 3 (2014), 378–393.

[21] Mitchell A Potter and Kenneth A De Jong. 1994. A cooperative coevolutionary approach to function optimization. In *International Conference on Parallel Problem Solving from Nature*. Springer, 249–257.

[22] Yuan Sun, Michael Kirley, and Saman Kumara Halgamuge. 2015. Extended differential grouping for large scale global optimization with direct and indirect variable interactions. In *Proceedings of the 2015 Annual Conference on Genetic and Evolutionary Computation*. ACM, 313–320.

[23] Yan-Yan Tan, Yong-Chang Jiao, Hong Li, and Xin-Kuan Wang. 2013. MOEA/D+ uniform design: A new version of MOEA/D for optimization problems with many objectives. *Computers & Operations Research* 40, 6 (2013), 1648–1660.

[24] Ye Tian, Ran Cheng, Xingyi Zhang, and Yaochu Jin. 2017. PlatEMO: A MATLAB Platform for Evolutionary Multi-Objective Optimization [Educational Forum]. *IEEE Computational Intelligence Magazine* 12, 4 (2017), 73–87.

[25] David A Van Veldhuizen and Gary B Lamont. 2000. On measuring multiobjective evolutionary algorithm performance. In *Evolutionary Computation, 2000. Proceedings of the 2000 Congress on*, Vol. 1. IEEE, 204–211.

[26] Zhenyu Yang, Ke Tang, and Xin Yao. 2008. Large scale evolutionary optimization using cooperative coevolution. *Information Sciences* 178, 15 (2008), 2985–2999.

[27] Qingfu Zhang and Hui Li. 2007. MOEA/D: A multiobjective evolutionary algorithm based on decomposition. *IEEE Transactions on evolutionary computation* 11, 6 (2007), 712–731.

[28] Qingfu Zhang, Aimin Zhou, Shizheng Zhao, Ponnuthurai Nagaratnam Suganthan, Wudong Liu, and Santosh Tiwari. 2008. Multiobjective optimization test instances for the CEC 2009 special session and competition. *University of Essex, Colchester, UK and Nanyang technological University, Singapore, special session on performance assessment of multi-objective optimization algorithms, technical report* 264 (2008).

[29] Xingyi Zhang, Ye Tian, Ran Cheng, and Yaochu Jin. 2016. A decision variable clustering-based evolutionary algorithm for large-scale many-objective optimization. *IEEE Transactions on Evolutionary Computation* (2016).

[30] Heiner Zille, Hisao Ishibuchi, Sanaz Mostaghim, and Yusuke Nojima. 2017. A Framework for Large-scale Multi-objective Optimization based on Problem Transformation. *IEEE Transactions on Evolutionary Computation* (2017).

[31] Eckart Zitzler and Simon Künzli. 2004. Indicator-based selection in multiobjective search. In *International Conference on Parallel Problem Solving from Nature*. Springer, 832–842.

[32] Eckart Zitzler, Marco Laumanns, and Lothar Thiele. 2001. SPEA2: Improving the strength Pareto evolutionary algorithm. *TIK-report* 103 (2001).