A Parallel Multi-objective Cooperative Co-evolutionary Algorithm with Changing Variables

Biao Xu
School of Information and Control Engineering,
China University of Mining and Technology, China
Xuzhou, Jiangsu
xubiao512@163.com

Yong Zhang
School of Information and Control Engineering,
China University of Mining and Technology, China
Xuzhou, Jiangsu
yongzh401@126.com

Dun-wei Gong*
School of Information and Control Engineering,
China University of Mining and Technology, China
Xuzhou, Jiangsu
dwgong@vip.163.com

Ling Wang
Department of Automation, Tsinghua University,
China
Beijing
wangling@tsinghua.edu.cn

ABSTRACT

Multi-objective optimization problems with changing variables are very common in real-world applications. This kind of problems often has a changing Pareto-optimal set and a complex relation among decision variables. In order to rapidly track the time-dependent Pareto-optimal front, we propose a framework of parallel cooperative co-evolution based on dynamically grouping decision variables. Decision variables are first divided into a number of groups using the Spearman rank correlation analysis, with different groups having a weak correlation. Then, a sub-population is employed to optimize decision variables in each group using a traditional multi-objective evolutionary algorithm. The evaluation of a complete solution is fulfilled through the cooperation among sub-populations. We compare the proposed algorithm with three state-of-the-art algorithms by applying them to two modified benchmark optimization problems. Empirical results show that the proposed algorithm is superior to the compared ones.

CCS CONCEPTS

•**Artificial Intelligence** → **Problem Solving** ; *Control Methods ;*

KEYWORDS

Dynamic multi-objective optimization, parallel, changing variables, co-evolutionary

*Corresponding author.

GECCO '17 Companion, Berlin, Germany
© 2017 ACM. 978-1-4503-4939-0/17/07...$15.00
DOI: http://dx.doi.org/10.1145/3067695.3084222

ACM Reference format:
Biao Xu, Yong Zhang, Dun-wei Gong, and Ling Wang. 2017. A Parallel Multi-objective Cooperative Co-evolutionary Algorithm with Changing Variables. In *Proceedings of GECCO '17 Companion, Berlin, Germany, July 15-19, 2017,* 6 pages.
DOI: http://dx.doi.org/10.1145/3067695.3084222

1 INTRODUCTION

A dynamic multi-objective optimization problem (DMOP) involves simultaneously optimizing more than one conflicting objective, and the objectives, constraints, and/or parameters change over time [10]. DMOPs are very common in real-world applications, such as dynamic job-shop scheduling [1, 5, 17, 21], green-house control [27], dynamic airspace resectorization [22], vehicle path planning [11, 24], to say a few. For a DMOP, if the number of decision variables changes over time, it is called as a multi-objective optimization problem (MOP) with changing variables. For example, in production scheduling formulated with a multi-objective optimization problem, the production process, the type of parts in equipment, and raw materials may vary due to machine failure, the change of customer needs and other factors. For logistics scheduling, transport routes and vehicles are dynamically adjusted because of weather and other factors. Due to changing variables, a DMOP involves not only a changing Pareto-optimal set, but also a complex relation among decision variables. Accordingly, it is challenging to tackle the problem.

In this paper, a minimization problem is considered here. The MOP with changing variables can be formulated as follows:

$$\begin{aligned} &\min F(X,t) = (F_1(X,t), F_2(X,t), \cdots, F_M(X,t)) \\ &\text{s.t.} \begin{cases} g_i(X,t) \le 0, i = 1,2,\cdots,q; \\ h_j(X,t) = 0, j = 1,2,\cdots,s; \\ X \in [X_{\min}, X_{\max}] \end{cases} \end{aligned} \quad (1)$$

where F represents a set of M objectives to minimize, $X = (\omega_1(t)x_1, \omega_2(t)x_2, \cdots, \omega_D(t)x_D)$ is the decision variable, and $\omega_k(t)$ is a control parameter with its value of 0 or 1, and $\omega_k(t)$=0 means that the k-th component, x_k, is excluded from the decision variable at time scale t; otherwise, the decision variable includes the component. Besides, $g_i \leq 0$ and $h_j = 0$ represent the i-th and the j-th inequality and equality constraints, respectively.

There are four types of DMOPs according to the changes of the Pareto-optimal set (PS) and the Pareto front (PF), i.e., Type I-Type IV [8]. For Type I, the PS changes, whereas the PF remains unchanged. With respect to Type II, both the PS and the PF change. Regarding Type III, the PF changes, whereas the PS remains unchanged. For the last type, Type IV, both the PS and the PF remain unchanged as an optimization problem changes. For a MOP with changing variables, there are only two types, i.e., Type I and Type II. The reason is that the PS always changes with the number of decision variables. More precisely, the PS has two ways of change. One is that the value of each original decision variable does not change, and the PS changes due to the additional decision variable, denoted as Type I(A) or Type II(A), the other is that the value of at least one original decision variable changes as the problem changes, denoted as Type I(B) or Type II(B).

A method of effectively solving a DMOP is required to overcome difficulties raised by the change of a problem, such as tracking the time-dependent PF and providing solutions with a good diversity. According to the manners of tackling the change, there have been three techniques for a DMOP whose number of decision variables is not increased [8], that is $\omega_k(t)$=1, $k = 1, 2, \cdots, D$ in formula (1), such as memory-based mechanism [18, 20], prediction-based strategies [9, 15, 16, 28, 29], and multi-population approaches [2, 3, 14, 20]. However, they have a difficulty in handling a MOP with changing variables.

A variety of strategies incorporating into cooperative co-evolutionary algorithms (CCEAs) [19] have been proposed to tackle a single-objective optimization problem with a large number of decision variables [13, 23]. However, they have been rarely applied to MOPs, particularly to DMOPs. As stated by Potter [19], cooperative co-evolution has good potential for parallelism, only limited studies have, however, studied the capability. Recently, Dorronsoro et al. proposed three novel parallel synchronous cooperative co-evolutionary multi-objective algorithms [7], which divides a problem into a number of sub-problems by splitting the solution of the original problem. They demonstrated the efficiency of the proposed algorithms by solving continuous or combinatorial multi-objective optimization problems [6, 7].

In this paper, we present a framework of parallel CCEAs for a MOP with changing variables to speed up the algorithms in convergence. In the proposed framework, a number of initial groups are first obtained according to the relation among decision variables. When the number of decision variables increases, the Spearman rank correlation between a new decision variable and each group is calculated based on information provided by the population evolution to adjust decision variable groups. Finally, a strategy is employed to initialize each of the groups when the problem changes, in order to timely responding the change.

The remainder of this paper is organized as follows. Section 2 describes the proposed framework of parallel CCEAs based on dynamically grouping decision variables in detail. Experimental results are reported and analyzed in Section 3. Finally, Section 4 draws conclusions.

2 THE PROPOSED FRAMEWORK

In this section, we propose a framework of CCEAs based on dynamically grouping decision variables for a DMOP with changing decision variables. We first obtain a number of non-dominated solutions by solving the original problem with a traditional evolutionary algorithm (e.g., NSGA-II [5]). Then, the correlation coefficient between decision variables is calculated by the Spearman rank correlation analysis based on the above solutions. Following that, all the decision variables are divided into a number of groups by the proposed method. Finally, a strategy of dynamically grouping decision variables is adopted when the problem changes.

2.1 Forming the Initial Groups of Decision Variables Based on the Spearman Rank Correlation

In this subsection, we present a method of initially grouping decision variables based on the Spearman rank correlation [12].

A population for solving the original problem is first evolved for a number of generations, until λ solutions are obtained. Then, the Spearman correlation coefficient between the first decision variable, $X(1)$, and each of the others is calculated by taking the obtained solutions as samples. Finally, decision variables having a high correlation with the first one are saved in the same group, and removed from the set of decision variables. Repeat the process, until each decision variable has its group.

For the optimal solution set currently obtained, $P = (p_1, p_2, \cdots, p_\lambda)$, and the set of decision variables, $X = (x_1, x_2, \cdots, x_D)$, $X(j)$, is the j-th decision variable in X, and $size(X)$ means the size of X. δ is a threshold and $\delta \geq 0$. The steps of the proposed grouping method are provided as follows.

Step 1: Set the initial value of the number of groups as k=1;
Step 2: $X^k = \{X(1)\}$;
- Step 2.1: Set $j = 2$;
- Step 2.2: Calculate the Spearman correlation coefficient between $X(1)$ and $X(j)$, denoted as r_{1j}. If $|r_{1j}| \geq \delta$, let $X^k = X^k \cup \{X(j)\}$;
- Step 2.3 If $j < size(X)$, let $j = j + 1$, go to Step 2.2; otherwise, let X=$X \backslash X^k$;

Step 3: If $size(X) > 1$, let $k = k + 1$, go to Step 2; otherwise, output the decision variable groups, $X^1, X^2, \cdots, X^K$.

2.2 Dynamically Grouping based on the Spearman Rank Correlation Coefficient

When a new decision variable, y, occurs in the optimization problem, we should dynamically grouping decision variables according to the following manner. First, a new subpopulation with its individual including the new decision variable, y, is randomly initialized; then,for the new sub-population, together with the other sub-populations, execute a CCEA several generations until λ solutions are achieved; Store the Pareto-optimal solutions in the archive, A , with each being a complete solution (please see subsection 2.4 for details). at last, a representative individual from each of the original groups is randomly selected , and the Spearman rank correlation coefficient, r^k, is computed between the new decision variable, y, and the representative individual, by taking the solutions in A as samples. If $r^u = max\{|r^k|\} \geq \delta$, combine y into the corresponding group, X^u; otherwise, form a separate group.

2.3 Responding the change of the optimization problem

The PF may change to a large degree when a problem changes. In this case, how to make full use of information provided by the evolutionary population to speed up an algorithm in convergence and to rapidly seek the PF become very important.

In this subsection, we propose a strategy of responding the change of a problem, which initializes sub-populations using half of the number of individuals in the archive. Besides, we employ a Gaussian perturbation operator with the following expression to improve the diversity of each sub-population:

$$x_i(t+1) = a_i(t) + N(\mu, \sigma^2) \qquad (2)$$

where $x_i\,(t+1)$ is the i-th individual at time scale t+1. $a_i\,(t)$ means the i-th non-dominated solution randomly selected from the archive set. $N(\mu, \sigma^2)$ is a random value obeying the Gaussian distribution, μ and σ refer to the location and the scale parameters. In this paper, we set $\mu = 0$ and $\sigma = 1$. The other individuals in each of these sub-populations are randomly initialized.

2.4 Evaluating the Individuals in each Sub-population

In this subsection, we provide a simple strategy of efficiently selecting the representative, i.e., selecting the representative based on the distance between the candidate and each individual in the archive in this paper. In this strategy, the individual with the shortest distance to the candidate is first selected from the archive as the representative. Then, a complete solution is formed by replacing the values of the corresponding decision variables of the representative with those of the candidate.

Fig. 1 display the process of selecting the representative for the case of three sub-populations. In Fig. 1, The decision variables are divided into the following three groups: $X^1 = (x_1^1, x_2^1, \cdots, x_{n_1}^1)$, $X^2 = (x_1^2, x_2^2, \cdots, x_{n_2}^2)$, $X^3 = (x_1^3, x_2^3, \cdots, x_{n_3}^3)$. The three subpopulations, P^1, P^2 *and* P^3 are employed to optimize the above three groups, respectively. A complete solution in the archive, A, closest to the candidate is first selected as the representative. Then, a new complete solution is formed by replacing the corresponding value of the representative with that of the candidate. The objective value of the complete solution is assigned with the fitness of the candidate.

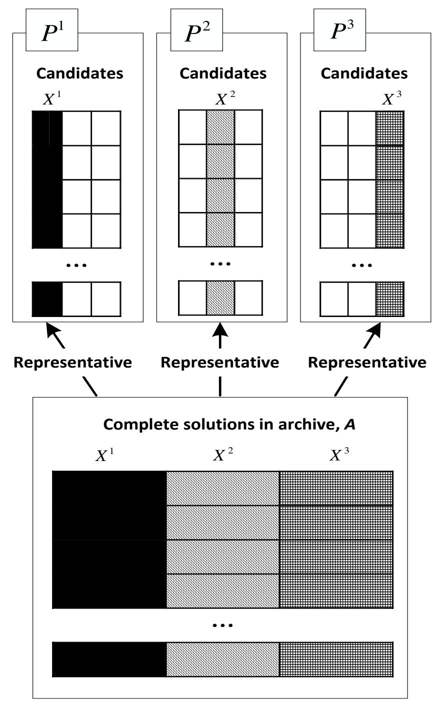

Figure 1: The strategy of selecting the representative

2.5 The Spearman Rank Correlation-based Parallel Cooperative Co-evolutionary Algorithm

We present a Spearman rank correlation-based parallel cooperative co-evolutionary algorithm by incorporating the proposed strategies into NSGA-II (SCC-NSGAII, for short) in this subsection. The pseudo code of the proposed parallel algorithm is displayed in Algorithm 1.

The decision variables are first divided into a number of groups with the strategy proposed in subsection 2.1. Then, each sub-population is initialized in line 3. This parallel

Algorithm 1 Parallel CCEA for DMOPs

1: grouping(X) //Grouping the decision variables utilizing the strategy proposed in subsection 2.1
2: g←0
3: $\exists i \in [1, K]$, initialize(P_0^i) //$\exists$ means a parallel run
4: sync() // Synchronization point
5: $\forall i \in [1, K]$, archive(P_0^i) //Construct a complete solution by randomly selecting a representative from each subpopulation and remain the non-dominated solutions; $\forall$means a sequential run
6: $\exists i \in [1, K]$, evaluate(P_0^i) //Evaluate solutions in each subpopulation
7: sync()
8: **while** not termination condition() **do**
9: $\exists i \in [1, K]$, evolve(P_g^i) // Evolve each sub-population one generation
10: sync()
11: $\forall i \in [1, K]$, archive(P_g^i)
12: **if** the problem changes() **then**
13: adjust grouping
14: conduct the responding strategy
15: **end if**
16: g←g+1
17: **end while**
18: output the archive, A

CCEA framework includes the first synchronization point (line 4) before the archive is obtained by randomly selecting a representative from each sub-population to construct complete solutions and remaining the non-dominated solutions. Following that, all the sub-populations evaluate their initial populations in parallel, as shown in line 6, followed by the next synchronization point (line 7). The loop for the cooperative evolution is begun until the termination criterion is met. For each iteration of the loop, all the sub-populations evolve one generation in parallel (line 9), and then synchronize (line 10) to achieve non-dominated solutions (line 11). If the problem changes, the groups of decision variables will be adjusted using the strategy proposed in subsection 2.2 (line 13), and the responding strategy presented in subsection 2.3 will start (line 14). Finally, the archive is output and the algorithm is ended.

3 EXPERIMENTAL STUDIES

In this section, we evaluate the proposed method by applying it to solve two benchmark optimization problems, and comparing it with three state-of-the-art methods. The implementation environment is as follows: Inter(R) Core(TM) i3-4170 CPU, 4.00GB RAM, windows 7 and Matlab R2012a.

3.1 Test Problems

We select the improved ZDT [4] and DTLZ [30], two well-defined test problem suites as the benchmark optimization problems, and denote them as DMOP1-DMOP2. ZDT and DTLZ are two continuous problem suites that can be scaled to any number of decision variables. To comprehensively evaluate the proposed method, we consider two- and three-objective optimization problems of these two suites with various characteristics, such as linear, non-concave, and multimodal, which are summarized in Table 1.

For these problems, the number of decision variable is $|X| = m+p$, where m means the number of decision variables of the initial problem, p is that of additional variables. In the experiments, m is set to 100 and p is an integer in the range of [0, 5]. That is to say, only one variable is added to the current optimization problem, and there are 5 changes. As a result, we consider six cases for each optimization problem.

The true PFs of DMOP1 is are shown in Fig.2, where $p = 0$ means the PF of the problem without any additional variable, $p = 1$ represents the PF of the problem with one additional variable. We can similarly understand the meanings of the other values of p. For the other problem, it has the same true PF before and after the optimization problem changes. Please refer to [4, 30] for details.

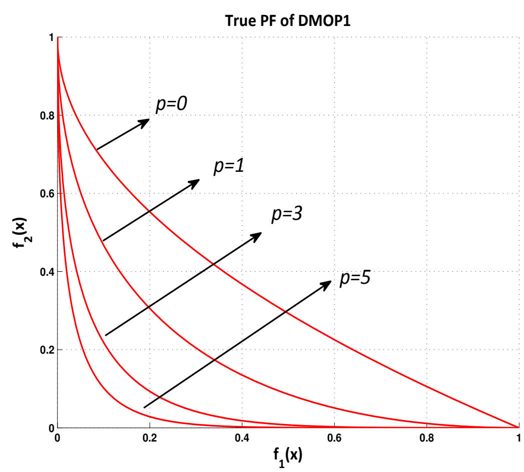

Figure 2: The true PFs of DMOP1

3.2 The Comparative Algorithms and Parameter Settings

We compare the proposed method, denoted as SCC-NSGAII, with NSGA-II, the random grouping-basedcooperative co-evolution NSGA-II (RCC-NSGAII), and the uniform grouping-based cooperative co-evolutionary NSGA-II (UCC-NSGAII).

The termination criterion is that the number of evaluations reaches to a predefined one, 600,000, for all the algorithms. The population size is set to 50 for SCC-NSGAII, RCC-NSGAII, and UCC-NSGAII, and 100 for NSGA-II. The complete solution is formed via the strategy proposed in subsection 2.4 to evaluate a candidate in a sub-population for all these algorithms. Besides, δ and λ are set to 0.8 and 20, respectively.

Table 1: Benchmark of MOPs with changing variables

Function	Definition	Characteristics
$DMOP1$ $(ZDT1)$	$f_1(X_I) = x_1,$ $f_2(X_{II}, \omega) = g(1 - \sqrt{(f_1/g)})^{(1+\sum_{i=m+1}^{\|X\|} \omega_i)}$ $g(X_{II}, \omega) = 1 + 9 \sum_{i=2}^{m} (x_i - \frac{1}{2+p})^2 \Big/ (\|X\| - 1)$ $+9 \sum_{i=m+1}^{\|X\|} \omega_i (x_i - \frac{1}{2+p})^2 \Big/ (\|X\| - 1)$ s.t.$X = (X_I, X_{II}), \|X\| = m + p, m = 100,$ $p = \|\{i \| \omega_i = 1\}\|, \omega_i \in \{0, 1\}, i = 1, 2, \ldots$ $0 \leq x_i \leq 1, i = 1, 2, \ldots, m.$	$TypeII(B), Non-concave$ $PS: x_1 \in [0,1], x_i = \frac{1}{2+p},$ $i = 2, 3, \ldots, \|X\|.$ $PF:$ $f_1 = x_1,$ $f_2 = (1 - \sqrt{x_1})^{(1+\sum_{i=m+1}^{\|X\|} \omega_j)}$
$DMOP2$ $(DTLZ1)$	$f_1(X) = 0.5 x_1 x_2 (1 + g(X_M));$ $f_2(X) = 0.5 x_1 (1 - x_2)(1 + g(X_M));$ $f_3(X) = 0.5(1 - x_1)(1 + g(X_M))$ $g(X_M) = ((\|X\| - 2) + \sum_{i=3}^{m} ((x_i - \frac{1}{2+p})^2 - \cos(20\pi(x_i - \frac{1}{2+p})))$ $+ \sum_{i=m+1}^{\|X\|} \omega_i ((x_i - \frac{1}{2+p})^2 - \cos(20\pi(x_i - \frac{1}{2+p}))))$ s.t.$0 \leq x_i \leq 1, i = 1, 2, \ldots, \|X\|, \|X\| = m + p, m = 100,$ $p = \|\{i \| \omega_i = 1\}\|, \omega_i \in \{0, 1\}$	$TypeI(B), Linear, Multimodal$ $PS: x_1, x_2 \in [0,1], x_i = \frac{1}{2+p},$ $i = 3, 4, \ldots, \|X\|;$ $PF:$ $f_1 = 0.5 x_1 x_2;$ $f_2 = 0.5 x_1 (1 - x_2);$ $f_3 = 0.5(1 - x_1).$

3.3 Performance Metrics

The inverted generational distance (IGD) [26] is widely used in multi-objective problems. To adapt the IGD metric to dynamic multi-objective optimization, its modified versions, denoted as $MIGD$ [25, 28], is taken as the performance metrics.

The $MIGD$ value of a Pareto-optimal set is calculated by averaging the IGD values of the set in a number of time scales over a run,

$$MIGD = \frac{1}{|T|} \sum_{t=1}^{T} IGD \tag{3}$$

where T is a set of time scales in a run, and $|T|$ means its cardinality. The $MIGD$ metric can measure both diversity and convergence, which assists in evaluating the tracking ability of an algorithm. To have a small $MIGD$ value, the obtained PF must be very close to the true PF.

IGD requires a reference set of optimal solutions, which are uniformly distributed on the PFs of a problem. We set the number of reference points to 5,000 for all the problems.

For each problem, 20 independent runs are conducted. The mean and the standard deviation of each indicator over 20 runs are recorded before the problem changes. Besides, the Mann-Whitney U test at the significant level of 0.05 is utilized to test the significance of different algorithms in terms of a metric.

3.4 Experimental Results and Analysis

Table 2 lists the mean and the standard deviation of $MIGD$ obtained by different algorithms on DMOP1 and DMOP2, where the boldface data are the best among these methods, and those labeled by '*' mean data obtained by the proposed algorithm significantly different from those obtained by the comparative one.

Table 2 reports that, for DMOP1, with the help of the proposed strategy of grouping, SCC-NSGAII performs the best, followed by NSGA-II, RCC-NSGAII and UCC-NSGAII. Additionally, for DMOP1, NSGA-II significantly outperforms RCC-NSGAII and UCC-NSGAII in terms of $MIGD$. This means that an inappropriate grouping strategy may deteriorate an algorithm.

DMOP2 has a linear and simple Pareto front, but it is a multi-modal problem with a great number of local optima. Table 2 shows that, for DMOP2, SCC-NSGAII archives the best performance on the $MIGD$ metric. Accordingly, the proposed method has a good capability in tackling DMOP2.

4 CONCLUSIONS

Aiming at a MOP with changing variables, we have proposed a parallel cooperative co-evolutionary algorithm based on dynamically grouping decision variables, termed SCC-NSGAII, in which the decision variables are grouped according to the Spearman rank correlation. In SCC-NSGAII, a complex DMOP is decomposed into a number of simple subproblems via grouping decision variables. A number of subpopulations are utilized to optimize the groups in parallel. Besides, SCC-NSGAII contains a method of selecting the representative based on the distance between a candidate and each of individuals in the archive. To demonstrate the effectiveness of SCC-NSGAII, we apply it to two benchmark optimization problems, DMOP1 and DMOP2, in comparison with three state-of-the-art algorithms. The experimental results demonstrate that SCC-NSGAII is very competitive among the compared algorithms.

ACKNOWLEDGMENTS

This work was jointly supported by National Natural Science foundation of China (No. 61375067, 61473299, 61573361), National Basic Research Program of China (973 Program)

Table 2: The mean and standard deviation of $MIGD$ for each algorithm over 20 runs

		$MIGD$			
Function	Stats.	NSGA-II	RCC-NSGAII	UCC-NSGAII	SCC-NSGAII
DMOP1	Mean	6.88E-03	7.68E-03	7.62E-03	**6.68E-03**
	Std	(-5.60E-05)	(9.8E-05)*	(7.1E-05)*	**(-9.10E-05)**
DMOP2	Mean	4.02E-01	2.58E-01	1.25E-01	**1.12E-01**
	Std	(1.3E-02)*	(1.8E-02)*	(3.1E-03)*	**(-1.00E-02)**

(No. 2014CB046306-2), Outstanding innovation team of China University of Mining and Technology (No. 2015QN003), the Six Talent Peaks Project in Jiangsu Province (No. DZXX-053), Natural Science foundation of Anhui Province (No.1608 085QG169), and Natural Science Foundation of Anhui Higher Education Institutions (No. KJ2015A035).

REFERENCES

[1] Manuel Blanco Abello, Lam Thu Bui, and Zbignew Michalewicz. 2011. An adaptive approach for solving dynamic scheduling with time-varying number of tasks-Part II. In *Evolutionary Computation (CEC), 2011 IEEE Congress on*. 1711–1718.

[2] Jürgen Branke, Thomas Kaußler, Christian Smidt, and Hartmut Schmeck. 2000. A multi-population approach to dynamic optimization problems. In *Evolutionary Design and Manufacture*. Springer, 299–307.

[3] Swagatam Das, Ankush Mandal, and Rohan Mukherjee. 2014. An adaptive differential evolution algorithm for global optimization in dynamic environments. *IEEE Transactions on Cybernetics* 44, 6 (2014), 966–978.

[4] Kalyanmoy Deb. 1999. Multi-objective genetic algorithms: Problem difficulties and construction of test problems. *Evolutionary Computation* 7, 3 (1999), 205–230.

[5] Kalyanmoy Deb, Bhaskara Rao N Udaya, and S. Karthik. 2007. Dynamic multi-objective optimization and decision-making using modified NSGA-II: a case study on hydro-thermal power scheduling. 4403 (2007), 803–817.

[6] Bernabé Dorronsoro, GréGoire Danoy, Pascal Bouvry, and Antonio J. Nebro. 2011. *Multi-objective Cooperative Coevolutionary Evolutionary Algorithms for Continuous and Combinatorial Optimization*. Springer Berlin Heidelberg. 49–74 pages.

[7] Bernabé Dorronsoro, GréGoire Danoy, Antonio J Nebro, and Pascal Bouvry. 2013. Achieving super-linear performance in parallel multi-objective evolutionary algorithms by means of cooperative coevolution. *Computers & Operations Research* 40, 6 (2013), 1552–1563.

[8] Marco Farina, Kalyanmoy Deb, and Paolo Amato. 2004. Dynamic multiobjective optimization problems: test cases, approximations, and applications. *Evolutionary Computation, IEEE Transactions on* 8, 5 (2004), 425–442.

[9] Iason Hatzakis and David Wallace. 2006. Dynamic multi-objective optimization with evolutionary algorithms: a forward-looking approach. In *Proceedings of the 8th annual conference on Genetic and evolutionary computation*. ACM, 1201–1208.

[10] Yaochu Jin and Jürgen Branke. 2005. Evolutionary optimization in uncertain environments-a survey. *Evolutionary Computation IEEE Transactions on* 9, 3 (2005), 303–317.

[11] Omprakash Kaiwartya, Sushil Kumar, DK Lobiyal, Pawan Kumar Tiwari, Abdul Hanan Abdullah, and Ahmed Nazar Hassan. 2015. Multiobjective dynamic vehicle routing problem and time seed based solution using particle swarm optimization. *Journal of Sensors* 2015 (2015).

[12] Ann Lehman, Norm O'Rourke, Larry Hatcher, and Ed Stepanski. 2005. JMP for Basic univariate and multivariate statistics: Methods for researchers and social scientists second edition. (2005).

[13] Xiaodong Li and Xin Yao. 2012. Cooperatively Coevolving Particle Swarms for Large Scale Optimization. *IEEE Transactions on Evolutionary Computation* 16, 2 (2012), 210–224.

[14] Ruochen Liu, Yangyang Chen, Wenping Ma, Caihong Mu, and Licheng Jiao. 2014. A novel cooperative coevolutionary dynamic multi-objective optimization algorithm using a new predictive model. *Soft Computing* 18, 10 (2014), 1913–1929.

[15] Ruochen Liu, Jing Fan, and Licheng Jiao. 2015. Integration of improved predictive model and adaptive differential evolution based dynamic multi-objective evolutionary optimization algorithm. *Applied Intelligence* 43, 1 (2015), 192–207.

[16] Arrchana Muruganantham, Kay Chen Tan, and Prahlad Vadakkepat. 2016. Evolutionary dynamic multiobjective optimization via Kalman filter prediction. *IEEE transactions on cybernetics* 46, 12 (2016), 2862–2873.

[17] Su Nguyen, Mengjie Zhang, Mark Johnston, and Kay Chen Tan. 2012. Automatic Design of Scheduling Policies for Dynamic Multi-objective Job Shop Scheduling via Cooperative Coevolution Genetic Programming. *IEEE Transactions on Evolutionary Computation* 18, 2 (2012), 193–208.

[18] Mohammad Nabi Omidvar, Xiaodong Li, and Xin Yao. 2011. Smart use of computational resources based on contribution for cooperative co-evolutionary algorithms. In *Genetic and Evolutionary Computation Conference, GECCO 2011, Proceedings, Dublin, Ireland, July*. 1115–1122.

[19] Mitchell A Potter. 1997. *The design and analysis of a computational model of cooperative coevolution*. George Mason University.

[20] Christopher D Rosin and Richard K Belew. 1997. New methods for competitive coevolution. *Evolutionary Computation* 5, 1 (1997), 1–29.

[21] Provas Kumar Roy and Sudipta Bhui. 2015. A multi-objective hybrid evolutionary algorithm for dynamic economic emission load dispatch. *International Transactions on Electrical Energy Systems* 26, 1 (2015), 49–78.

[22] Jiangjun Tang, Sameer Alam, Chris Lokan, and Hussein A Abbass. 2012. A multi-objective evolutionary method for dynamic airspace re-sectorization using sectors clipping and similarities. In *Evolutionary Computation (CEC), 2012 IEEE Congress on*. IEEE, 1–8.

[23] Frans Van den Bergh and Andries Petrus Engelbrecht. 2004. A cooperative approach to particle swarm optimization. *IEEE transactions on evolutionary computation* 8, 3 (2004), 225–239.

[24] Paul PY Wu, Duncan Campbell, and Torsten Merz. 2011. Multiobjective four-dimensional vehicle motion planning in large dynamic environments. *Systems, Man, and Cybernetics, Part B: Cybernetics, IEEE Transactions on* 41, 3 (2011), 621–634.

[25] Biao. Xu, Yong Zhang, Dunwei Gong, Yinan Guo, and Miao Rong. 2017. Environment Sensitivity-based Cooperative Coevolutionary Algorithms for Dynamic Multi-objective Optimization. *IEEE/ACM Transactions on Computational Biology and Bioinformatics* DOI: 10.1109/TCBB.2017.2652453 (2017).

[26] 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* (2008).

[27] Zhuhong Zhang. 2008. Multiobjective optimization immune algorithm in dynamic environments and its application to greenhouse control. *Applied Soft Computing* 8, 2 (2008), 959–971.

[28] Aimin Zhou, Yaochu Jin, and Qingfu Zhang. 2014. A population prediction strategy for evolutionary dynamic multiobjective optimization. *Cybernetics, IEEE Transactions on* 44, 1 (2014), 40–53.

[29] Aimin Zhou, Yaochu Jin, Qingfu Zhang, Bernhard Sendhoff, and Edward Tsang. 2007. Prediction-based population reinitialization for evolutionary dynamic multi-objective optimization. In *Evolutionary Multi-Criterion Optimization*. Springer, 832–846.

[30] Eckart Zitzler, Kalyanmoy Deb, and Lothar Thiele. 2000. Comparison of Multiobjective Evolutionary Algorithms: Empirical Results. *Evolutionary Computation* 8, 2 (2000), 173.