Solving Dynamic Multi-objective Optimization Problems Using Incremental Support Vector Machine

Weizhen HU, Min JIANG*, *Senior Member, IEEE,* Xing Gao,
Kay Chen TAN, *Fellow, IEEE,* and Yiu-ming Cheung, *Fellow, IEEE*

Abstract—The main feature of the Dynamic Multi-objective Optimization Problems (DMOPs) is that optimization objective functions will change with times or environments. One of the promising approaches for solving the DMOPs is reusing the obtained Pareto optimal set (POS) to train prediction models via machine learning approaches. In this paper, we train an Incremental Support Vector Machine (ISVM) classifier with the past POS, and then the solutions of the DMOP we want to solve at the next moment are filtered through the trained ISVM classifier. A high-quality initial population will be generated by the ISVM classifier, and a variety of different types of population-based dynamic multi-objective optimization algorithms can benefit from the population. To verify this idea, we incorporate the proposed approach into three evolutionary algorithms, the multi-objective particle swarm optimization(MOPSO), Nondominated Sorting Genetic Algorithm II (NSGA-II), and the Regularity Model-based multi-objective estimation of distribution algorithm(RE-MEDA). We employ experimentS to test these algorithms, and experimental results show the effectiveness.

Index Terms—Dynamic Multi-objective Optimization Problems; Incremental Support Vector Machine; Pareto Optimal Set

I. INTRODUCTION

One of the essential characteristics of Dynamic Multi-objective Optimization Problems (DMOPs) [1] is that the optimization functions will change with environments or times, and it is of great significance for many practical applications [2]. Therefore, efficiently solving DMOPs has become an important direction in the field of evolutionary computation community [3]. For example, when designing a missile, the defense department should consider its range, precision, weight and fuel consumption. This is a four-objective optimization problem, and these optimization objectives will vary depending on environments and times. However, Most of the existing methods do not perform well when dealing with such problems. The main difficulty is how to quickly track the changing Pareto optimal front (POF).

W.Hu and M. JIANG are with the Department of Cognitive Science, Xiamen University, China, Fujian, 361005. X. Gao is with the Software School of Xiamen University. Min JIANG is the corresponding author and email: minjiang@xmu.edu.cn.

KC TAN is with the Department of Computer Science, City University of Hong Kong.

Yiu-ming Cheung is with the Department of Computer Science, Hong Kong Baptist University.

In recent years, great progress has been made in this field, and many different algorithms have been proposed. Among those algorithms, the methods based on prediction have attracted special attention, and the basic idea of the method is to solve DMOPs by reusing "experience" effectively. For example, Muruganantham *et al.* [4] proposed a Kalman Filter prediction based DMOEA(MOEA/D-KF) that utilizes a prediction model based on Kalman Filter based on the multiobjective EA with Decomposition.

In this paper, we argue that Incremental Support Vector Machine (ISVM) [5], [6] can be used to train a prediction model, and the model helps any kind of population-based DMOPs algorithms by generating a high-quality initial population. This method has unique advantages in dealing with DMOPs, because it directly establishes connections between different POS, without requiring more computational resources to repeatedly train SVM classifiers.

The contribution of this research is to propose an algorithm which combining evolutionary multi-objective optimization algorithm with the ISVM technique. This combination has the following two advantages. First, the proposed design can improve the search accuracy in the way of reusing past experience. Second, this method trains the SVM classifier in an online manner, and then obtains the initial population for the next time, which can make more efficient use of computing resources.

The rest of this paper is organized as follows: In Section II, we will introduce some basic concepts of dynamic optimization problems, Incremental Support Vector Machine (ISVM) and related work. In Section III, we will propose the Incremental Support Vector Machine based Dynamic Multi-Objective Evolutionary optimization Algorithm, ISVM-DMOEA. In Section IV we firstly introduce the evaluation criteria, test examples and comparative methods, and then experimental results are analyzed. In Section V, we conclude the main work of this research and the future research direction are discussed.

II. PRELIMINARIES AND RELATED WORKS

A. Concepts of Dynamic Multi-objective Optimization

For dynamic multi-objective optimization problems, its optimization functions often vary with times or environments.

Mathematically, a DMOP can be described as:

minimize $f(x,t) = \langle f_1(x,t), f_2(x,t), ..., f_m(x,t) \rangle$

$$s.t. \ x \in \Omega,$$

where t represents time or environment index and $x = \langle x_1, x_2, \ldots, x_n \rangle$ is the decision vector. m is the number of objectives, $f_i(x,t) : \Omega \to \mathbb{R} \ (i = 1, \ldots, M)$ which is the objective space.

Definition 1. [Dynamic Decision Vector Domination] *At a particular moment t, a decision vector x_b is Pareto dominated by another vector x_a, it can be expressed as $x_a \succ_t x_b$, if and only if*:

$$\begin{cases} \forall i = 1, \ldots, m, & f_i(x_a, t) \leq f_i(x_b, t) \\ \exists i = 1, \ldots, m, & f_i(x_a, t) < f_i(x_b, t) \end{cases}. \quad (1)$$

Definition 2. [Dynamic Pareto-optimal Set] *At time t, if and only if there is no other decision vector x can Pareto dominate decision vector x^*, the x^* is called Pareto optimal solution. All Pareto optimal solutions make up Dynamic Pareto-optimal Set(DPOS) at time t, that is*:

$$DPOS = \{x^* | \nexists x, \ x \succ_t x^*\}.$$

Definition 3. [Dynamic Pareto-optimal Front] *The Dynamic Pareto-Optimal Front (DPOF) is the set of corresponding objective vectors of the Pareto optimal solution at time t.*

$$DPOF = \{f(x^*, t) | x^* \in DPOS\}.$$

B. Incremental Support Vector Machines

Support Vector Machine (SVM) [7] was proposed in 1964 , developed rapidly after the 1990s and spawned a series of improved and extended algorithms. The SVM becomes a well-known learning method used for classification problems [8], and it is a generalized linear classifier for binary classification of data in supervised learning. Its decision boundary is the maximum-margin hyperplane that is solved for learning samples. Incremental Support Vector Machine (ISVM) is the combination of Online learning technology and the SVM. Incremental technology has been developed to facilitate bulk SVM learning [9] on very large data sets and has been widely used in the SVM community.

Given the training data $\{x_1, \cdots, x_N\}$ and learning objective $\{y_1, \cdots, y_N\}$ in the classification problem, where each sample of training data contains multiple features and thus constitutes a feature space. The learning objective is a binary variable $y_i \in \{1, -1\}$ representing negative class and positive class. A Support Vector Machine (SVM) is a discriminative classifier formally defined by a separating hyperplane, which can be described as:

$$\mathbf{w^T} \times x + b = 0,$$

where $\mathbf{w} \in \mathbb{R}^t$ and $b \in \mathbb{R}$.

In linear inseparable problems, the use of SVM with hard margins will generate classification errors, so a new optimization problem can be constructed by introducing loss function

on the basis of maximizing margins.The optimization problem of soft margin SVM is shown as follows [10]:

$$\textbf{Minmize}_{(\mathbf{w},\mathbf{b})} \ \frac{1}{2}||\mathbf{w}||^2 + l \cdot \sum_{i=1}^{N} \varepsilon_i$$
$$\textbf{subj. to}: \ y_i(\mathbf{w} \times x_i + b) \geq 1 - \varepsilon_i, i = i \cdots N. \quad (2)$$

Where l is a constant and the second term of Equation (2) provides an upper bound for the error in the training data, and the first term makes maximum margin of separation between classes.

when learning nonlinear SVMs, to simplify matters and then this quadratic program is typically expressed in its dual form:

$$\min_{0 \leq \alpha_i \leq l} W = \frac{1}{2} \sum_{i,j=1}^{N} \alpha_i Q_{ij} \alpha_j - \sum_{i=1}^{N} \alpha_i + b \sum_{i=1}^{N} y_i \alpha_i$$

Where $Q_{ij} = y_i y_j K(\mathbf{x}_i, \mathbf{x}_j), K(\mathbf{x}_i, \mathbf{x}_j) = \varphi(x_i) \cdot \varphi(x_j)$ is the given kernel function which to implicitly map into a higher (possibly infinite) dimensional feature space.

The Karush-Kuhn-Tucker (KKT) condition uniquely defines the solution of dual parameters$\{\alpha, b\}$, and minimizes the form (2):

$$g_i = \frac{\partial W}{\partial \alpha_i} = \sum_{j=1}^{N} Q_{ij}\alpha_j + y_i b - 1 = \begin{cases} > 0 & \alpha_i = 0 \\ = 0 & 0 < \alpha_i < l \\ < 0 & \alpha_i = l \end{cases}$$

$$h = \frac{\partial W}{\partial b} = \sum_{j=1}^{N} y_j \alpha_j = 0$$

The KKT conditions partition the training data into three parts:

$g_c = 0$, the set S of margin support vectors;

$g_c \leq 0$, the set E of error support vectors;

$g_c > 0$, the set R of the remaining vectors.

When we increment the unlearned examples into the solution, our goal will be to keep all previously seen training data in KKT conditions simultaneously [5].

C. Related Works

The DMOPs field has made great progress, and existing algorithms can be generally divided into the following categories: Diversity based Approaches, parallel approaches, Memory based Approaches and Change prediction based Approaches .

An important work of diversity based approaches is the Dynamic NSGA-II (DNSGA-II) proposed by [11] in 2006. Deb et al. extended NSGA-II to deal with DMOPs by introducing diversity in each change detection. If the target or constraint violation value has changed, the problem is considered to have changed. Then, all outdated solutions (that is, reassessment). This process allows for the use of changing objectives and constraint functions to evaluate both offspring and parent solutions. In [12], Chen proposed to maintain genetic diversity by regard it as an additional objective when solving multi-objective optimization. They presented the Individual Diversity Evolutionary Method (IDEM) to add a useful selection pressure for optimizing PS and maintaining diversity

[12]. The results show that the algorithm can converge to the optimal solution effectively and track the change of PFs while maintaining the diversity of solution sets.

Parallel EAs utilize several subpopulations that evolve simultaneously under different processors and then communicate some informations in a structured network [13]. In [14], Camara et al. proposed a method to apply parallel single pregenetic algorithm (PSFGA) to dynamic environment. PSFGA is a master-slave architecture algorithm, which divides the population into subpopulations which performs a certain amount of generations and preserve only non-dominant solutions, and then the main process joins all the solutions to a new population.

The memory-based approach uses additional memory to implicitly or explicitly store useful information from past generations to guide future searches. It has been proved that when the optimal solution returns to the previous position repeatedly or the environment changes periodically, this algorithm will help save computing time and bias search process, thus becoming very efficient. In [15], Azzouz *et al.* proposed an adaptive hybrid population management strategy, which is based on a technology that can measure the severity of environmental changes, then according to the technology it can adjust memory, local search (LS) and the number of random solutions.

When the behavior of dynamic problems follows a certain trend, a prediction model can be used. In 2014, the author defined a new prediction model [16] to solve DMOPs with Translational optimal POS (DMOP-tps). Dmop-tps is a specific type of DMOP in which POS is converted periodically over time. When the environment changes, the strategy proposed by Deb *et al.* [11] is used to detect the changes. Then, the population is re-initialized according to the properties of DMOP. Muruganantham *et al.* [4] proposed a Kalman Filter prediction based DMOEA(MOEA/D-KF) based on the multiobjective EA with Decomposition. When the change of environment is detected, kalman filter is applied to the whole population, which leads the search to the new pareto optimal solution in the decision space. The influence of the severity and frequency of change on the performance of the this algorithms is studied.

III. INCREMENTAL SUPPORT VECTOR MACHINE BASED DYNAMIC MULTI-OBJECTIVE OPTIMIZATION ALGORITHM

A dynamic multi-objective optimization problem is based on the fact that different environments follow different possible distributions, and these are not independent, but interrelated. For the dynamic multi-objective optimization problem, it is easy to get the solution. However, it is difficult for us to judge whether this solution is good or bad, but the solution already obtained should contain useful information, which can be used to predict POS at the next moment. So in this article, we turn the decision problem into a classification problem. In other words, we are constantly training the ISVM classifier through the information we have obtained, that is, the POS of past moments.

At the same time, a good initial population is the key to solving the dynamic multi-objective problem. A good initial population not only speeds up the solution, but also improves the quality and accuracy of the solution. By training the SVM classifier in an online way, we can use this continuously improved classifier to filter out the solutions , and then generate the initial population at the next moment.

In this article, we propose a incremental support vector machine (ISVM) based dynamic optimization algorithm. The details of the proposed algorithm is descried in Alg. 1.

This algorithm is mainly divided into two modules. At the initial moment, we combine the solution (positive example) in POS obtained by using multi-objective evolutionary algorithm and the randomly generate some solutions of the problem (non-POS, negative example) into training samples, and then obtain a SVM classifier SC_S. When the environment changes, we regard the changes as the incremental learning process of SVM classifier for the change process of dynamic multiobjective optimization problems. At each change moment, we consider the obtained POS and non-POS as samples to update the parameters of SC_S classifier. At the same time, using SC_S classifier, we can classify the solutions in the next moment into two categories, "good" and "bad". And the good solutions is reserved as the initial population of the next moment. Finally, we can get the SC_S classification model with the best performance.

Remark 1. *In this research, we only used the NSGA-II, the MOPSO and the RM-MEDA algorithms to obtain POS at next time, but in fact, any population-based algorithm can use our proposed method to achieve performance improvements.*

Remark 2. *The $PSAMPLES_t$ at particular moment are obtained by POS_t while $NSAMPLES_t$ are randomly generate [17],and they have the same number.*

Remark 3. *At the initial moment, we train a SVM classifier SC_S by using $Pg \in POS_0$ and $Ng \notin POS_0$, and when t change from 1 to n, we still train the same SC_S using $Pg \in PSAMPLES_t$ and $Ng \in NSAMPLES_t$.*

IV. EMPIRICAL STUDY

In general, our proposed algorithm is applicable to any kind of population-based optimization algorithm. In our experiment, we chose three representative experiments to prove our method. The first multiobjective optimization algorithm is based on particle swarm optimization [18], it was called MOPSO for short. The second one is the NSGA-II and it is a multiobjective genetic algorithm that applies nondominated sorting and crowding distance. The third one is a distribution estimation algorithm [19] based on global statistical information to construct a probability model, We simply refer to it as the RM-MEDA. Obviously, these three algorithms belong to different categories, and they were not originally designed for dynamic optimization. But they are well developed, so we can

Algorithm 1: ISVM-DMOEA:Incremental Support Vector Machines based Dynamic Multi-objective Evolutionary Algorithm

---

Input: The Dynamic Multi-objective Optimaztion Function $F(X)$;
Output: $POSs$: the POSs of $F(X)$;

1 Randomly initiate a Population the Pop_0;
2 POS_0 =DMOEA(Pop_0);
3 POS_s =POS_0;
4 Train a SVM classifier SC_S by using $Pg \in POS_0$ and $Ng \notin POS_0$;
5 Randomly generate solutions $\{xy_1, \cdots, xy_p\}$ of the function $F(X)_1$;
6 **if** xy_i *pass the recognition of the SVM* SC_S **then**
7 $\quad$ Put xy_i into Pop_1
8 **end**
9 POS_1=DMOEA(Pop_1);
10 POS_t =POS_1;
11 **for** $t = 1$ *to* n **do**
12 $\quad$ $PSAMPLES_t$=POS_t;
13 $\quad$ Train SC_S by using $Pg \in PSAMPLES_t$ and $Ng \in NSAMPLES_t$;
14 $\quad$ Randomly generate solutions $\{xy_1, \cdots, xy_p\}$ of the function $F(X)_{t+1}$;
15 $\quad$ **if** xy_i *pass the recognition of the SVM* SC_S **then**
16 $\quad\quad$ Put xy_i into Pop_{t+1}
17 $\quad$ **end**
18 $\quad$ POS_{t+1} = DMOEA(Pop_{t+1});
19 $\quad$ $POSs$ = $POSs \cup POS_{t+1}$;
20 **end**
21 **return** $POSs$;

---

improve our level of persuasion and confidence in the technologies we propose. The three corresponding algorithms with incremental SVM are called ISVM-MOPSO, ISVM-NSGA-II, and ISVM-RMMEDA respectively. It is worth noting that parameters, such as population size, iteration number are all the same. In other words, for the three groups of algorithms, we did not deliberately adjust the experimental parameters for getting better performances.

A. Performance Metrics, Testing Functions and Settings

In this study, when comparing with other competitive algorithms, we use inversed generational distance (IGD) and its variants as performance indicators to evaluate the quality of solutions obtained by different algorithms.

1)The inverted generational distance (IGD) [21] is an index to measure the distance between the real Pareto Optimal Front expressed by Q^*, and the approximate Pareto Optimal Front obtained by the algorithm, which we use Q to represent.Then the definition of the IGD can be described as

$$\text{IGD}(Q^*, Q, C) = \frac{\sum_{p^* \in Q^*} \min_{p \in Q} \|p^* - p\|}{|Q^*|}. \quad (3)$$

It is worth noting that the definition of IGD in this paper is slightly different from the original definition. The main difference is the parameter C in Equation (3), which we call the configuration of the benchmark functions. The configurations used in this experiment are shown in Table I. IGD compares the ideal POF with the POF obtained by other algorithms. If the distance between P and P* is closer, the smaller the value of IGD, the higher the performance of the algorithm is to some extent. One variant of the IGD, called MIGD, can also be used to evaluate dynamic multiobjective optimization algorithms , and it takes the average of the IGD values in some time steps over a run as the performance metric, given by: [4], [22]. MIGD is the average value of IGD values at a certain time step in each run,described as:

$$\text{MIGD}(Q^*, Q, C) = \frac{1}{|T|} \sum_{t \in T} \text{IGD}(Q_t^*, Q_t, C), \quad (4)$$

Where T represents the set of all moments. At time t, Q_t^* represents the point set of the ideal POF, and Q^t represents the approximate POF obtained by the algorithm. In addition, we also hope to evaluate these algorithms in a dynamic environment, so we have defined a new indicator DMIGD based on MIGD. The definition of DMIGD is as follows:

$$\text{DMIGD}(Q^*, Q, C) = \frac{1}{|E|} \sum_{C \in E} \text{MIGD}(Q_t^*, Q_t, C), \quad (5)$$

Where $|E|$ is the number of different environments experienced. We used eight different environmental configurations to conduct our experiments. It is worth noting that DMIGD allows us to evaluate the dynamic multi-objective optimization algorithm from a high-level perspective, and MIGD only considers the dynamics in an environment, so it is obviously different from MIGD.

B. Test Instances and Experimental Settings

In this research, we take the IEEE CEC 2015 benchmark problems set as test functions and the problem set has eleven testing functions. Details of the functions definitions are given in [23]. In the definitions, the decision variables are $x = (x_1, \ldots, x_n)$ and $t = \frac{1}{n_t} \left\lfloor \frac{\tau_T}{\tau_t} \right\rfloor$, where n_t, τ_T, and τ_t are the severity of change, maximum number of iterations, and frequency of change respectively. Table I describes the different combinations of n_t, τ_t, and τ_T used in our experiments. Please note that, for each n_t-τ_T combination, there will be $\frac{\tau_T}{\tau_t}$ environment changes. In other words, in all of our experiments, every test function requires 20 environmental transformations for each pair of n_t-τ_T combination.

The POFs of the testing functions have different shapes and each function belongs to a certain DMOPs type. For example, the POF of dMOP3 and DIMP2 is convex while the POF of HE2 is discontinuous. and for FDA5,the spread of POF solutions changes over time. Table II describes the types of the testing functions. Type I means POS changes, but POF does not; Type II implies that when POS changes, POF changes

TABLE I. Environment Settings

	n_t	T_t	T_T
C1	10	5	100
C2	10	10	200
C3	10	25	500
C4	10	50	1000
C5	1	10	200
C6	1	50	1000
C7	20	10	200
C8	20	50	1000

accordingly; Type III indicates that POF has changed, but POS has not.

TABLE II. Characteristic of the test functions

Name	Decision Variable Dimension	Objectives	DMOP Type
FDA4	12	3	TYPE I
FDA5	12	3	TYPE II
DIMP2	10	2	TYPE I
dMOP2	10	2	TYPE II
HE7	10	2	TYPE III
HE9	10	2	TYPE III

As depicted in Table II, the dimensions of the decision variables are 10 and 12, In all experiments, the population size was set to 200. As mentioned earlier, for each environment configuration, we changed each benchmark function 20 times, and in every change, we let the entire population perform 50 iterations in the evolutionary algorithm. For the incremental Support Vector Machines, we set the kernel type is Gaussian kernel and kernel scale is obtained by the Grid Search method [24].

C. Experimental Results

In this study, we conducted two different kinds of experiments. In the first kind of experiment, we incorporate the proposed approach into the development of three well-known evolutionary algorithms, multiojective particle swarm optimization (MOPSO), nondominated sorting genetic algorithm II (NSGA-II), and the regularity model-based multi-objective estimation of distribution algorithm (RM-MEDA). We employ six benchmark functions to test these algorithms. The experimental results are recored in Table. III. In almost all experiments, the proposed method has been significantly improved compared with the original method.

In the second kind of experiment, we compared the ISVM-RM-MEDA with some chosen algorithms, and the results are recorded in Table. IV. The MBN-EDA [25] is an estimation algorithm, in which the dependence between decision variables and target variables is obtained by using a multidimensional Bayesian network. MOEA\D-KF was presented in [4] and this idea for solving DMOPs is to predict the decision space by Kalman filter. SVM-NSGA-II was proposed in [26] and it is a support vector machine based dynamic MOPs algorithm.

For incremental SVM, a major iteration corresponds to contain a new example. Therefore, the computational complexity estimates must be multiplied by the number of training instances learned so far. The actual runtime depends on the balance between arithmetic operations and memory access in a small iteration. In our proposed method, when adding new samples incrementally, we can use the SVM classifier that has been obtained in the past moment to simplify the computation cost of searching the solution of the next quadratic programming. By the way, for samples that are not partitioned into the set s of margin support vectors, there is no need to carry out new iterations and update core parameters, which is the key module for computation cost in incremental SVM usage.

V. CONCLUSION

Evolutionary algorithms can solve dynamic multi-objective optimization problems more effectively through good initial populations, but getting a good initial population is a difficult and resource-consuming problem. In order to avoid wasting valuable computing resources on repeated computing, our basic idea is that incremental training support vector machine classifiers can be trained by using the POS have been obtained at different times, and when changes occur, the parameter of SVM classifier will be updated in an online manner, then the trained classifier will filter the solution at the next moment, and the good individuals will be selected as the initial population. This initial population can help any kind of population-based algorithms to solve the dynamic multi-objective optimization problem quickly and more accurately.

This research is just a new starting point, after which we will learn how to combine incremental support vector machines with transfer learning [27] or imbalanced learning techniques [28]. On the other hand, we also want to study use deep learning methods [20] to automatically generate better samples to solve the real world problems, such as motion generation of Multi-Legged Robot [29] and dynamic path planning [30].

REFERENCES

[1] M. Farina, K. Deb, and P. Amato, "Dynamic multiobjective optimization problems: test cases approximations, and applications," *IEEE Transactions on Evolutionary Computation*, vol. 8, no. 5, pp. 425–442, oct 2004.

[2] C. Cruz, J. R. González, and D. A. Pelta, "Optimization in dynamic environments: a survey on problems methods and measures," *Soft Comput*, vol. 15, no. 7, pp. 1427–1448, dec 2010.

[3] T. T. Nguyen, S. Yang, and J. Branke, "Evolutionary dynamic optimization: A survey of the state of the art," *Swarm and Evolutionary Computation*, vol. 6, pp. 1–24, oct 2012.

[4] A. Muruganantham, K. Tan, and P. Vadakkepat, "Evolutionary Dynamic Multiobjective Optimization Via Kalman Filter Prediction," *IEEE Transactions on Cybernetics*, vol. 46, no. 12, pp. 2862–2873, Dec 2016.

[5] P. Laskov, C. Gehl, S. Krger, and K. R. Mller, "Incremental support vector learning: analysis, implementation and applications," *Journal of Machine Learning Research*, vol. 7, no. 3, p. 2006, 2006.

[6] T. Tušar and B. Filipič, "Differential evolution versus genetic algorithms in multiobjective optimization," in *International Conference on Evolutionary Multi-Criterion Optimization*. Springer, 2007, pp. 257–271.

TABLE III. The MIGD Values between NSGA-II, ISVM- NSGA-II, MOPSO, ISVM-MOPSO, RM-MEDA and ISVM-RM-MEDA

DMIGD	NSGA-II	ISVM-NSGAII	MOPSO	ISVM-MOPSO	RM-MEDA	ISVM-RM-MEDA
FDA4	0.2783	**0.1839**	0.0812	**0.0731**	0.0684	**0.0639**
FDA5	0.3580	**0.2604**	0.2510	**0.1586**	0.2345	**0.1157**
DIMP2	3.9818	**2.5154**	2.6386	**2.3410**	4.9631	5.2002
DMOP2	0.6590	**0.0922**	0.2984	**0.0881**	4.5942	4.5228
HE7	0.0946	**0.0534**	0.0636	**0.0561**	0.0428	**0.0342**
HE9	0.2945	**0.2473**	0.2502	**0.2450**	0.2581	**0.2337**

TABLE IV. The MIGD Values between MBN-EDA, MOE/D-KF,SVM-NSGA-II,ISVM-RM-MEDA

DMIGD	MBN-EDA	MOE/D-KF	SVM-NSGA-II	ISVM-RMMEDA
FDA4	0.4300	0.1913	0.2166	**0.0639**
FDA5	0.5100	0.4963	0.2910	**0.1157**
DIMP2	6.9700	22.9536	**2.8463**	5.2002
DMOP2	1.4000	3.0619	**0.1356**	4.5228
HE7	0.2100	0.2365	0.0728	**0.0342**
HE9	0.3600	0.4108	0.2497	**0.2337**

[7] B. E. Boser, I. M. Guyon, and V. N. Vapnik, "A training algorithm for optimal margin classifiers," in *Proceedings of the fifth annual workshop on Computational learning theory*. ACM, 1992, pp. 144–152.

[8] E. Osuna, R. Freund, and F. Girosi, "An improved training algorithm for support vector machines," in *Neural Networks for Signal Processing VII-Proceedings of the 1997 IEEE Workshop*, 1997, pp. 276–285.

[9] G. Cauwenberghs and T. Poggio, "Incremental and decremental support vector machine learning," in *International Conference on Neural Information Processing Systems*, 2000, pp. 388–394.

[10] C. P. Diehl and G. Cauwenberghs, "Svm incremental learning, adaptation and optimization," in *International Joint Conference on Neural Networks*, 2003, pp. 2685–2690 vol.4.

[11] K. Deb, S. Karthik *et al.*, "Dynamic multi-objective optimization and decision-making using modified nsga-ii: a case study on hydro-thermal power scheduling," in *International Conference on Evolutionary Multi-Criterion Optimization*. Springer, 2007, pp. 803–817.

[12] H. Chen, M. Li, and X. Chen, "Using diversity as an additional-objective in dynamic multi-objective optimization algorithms," 2009.

[13] E. Alba, "Parallel evolutionary algorithms can achieve super-linear performance," *Information Processing Letters*, vol. 82, no. 1, pp. 7–13, 2002.

[14] M. Cmara, J. Ortega, and F. D. Toro, "A single front genetic algorithm for parallel multi-objective optimization in dynamic environments," *Neurocomputing*, vol. 72, no. 16, pp. 3570–3579, 2009.

[15] R. Azzouz, S. Bechikh, and L. B. Said, "A dynamic multi-objective evolutionary algorithm using a change severity-based adaptive population management strategy," *Soft Computing*, vol. 21, no. 4, pp. 1–22, 2015.

[16] Z. Li, H. Chen, Z. Xie, C. Chen, and A. Sallam, "Dynamic multiobjective optimization algorithm based on average distance linear prediction model," *The Scientific World Journal*, vol. 2014, 2014.

[17] H. Guo, Y. Li, J. Shang, M. Gu, Y. Huang, and B. Gong, "Learning from class-imbalanced data: Review of methods and applications," *Expert Systems with Applications*, vol. 73, pp. 220–239, 2016.

[18] C. C. Coello and M. S. Lechuga, "Mopso: A proposal for multiple objective particle swarm optimization," in *Evolutionary Computation, 2002. CEC'02. Proceedings of the 2002 Congress on*, vol. 2. IEEE, 2002, pp. 1051–1056.

[19] Q. Zhang, A. Zhou, and Y. Jin, "Rm-meda: A regularity model-based multiobjective estimation of distribution algorithm," *IEEE Transactions on Evolutionary Computation*, vol. 12, no. 1, pp. 41–63, 2008.

[20] M. Jiang, Y. Ding, B. Goertzel, Z. Huang, C. Zhou, and F. Chao, "Improving machine vision via incorporating expectation-maximization into deep spatio-temporal learning," in *2014 International Joint Conference on Neural Networks (IJCNN)*. IEEE, 2014, pp. 1804–1811.

[21] M. R. Sierra and C. A. C. Coello, "Improving pso-based multi-objective optimization using crowding, mutation and ϵ-dominance," in *International Conference on Evolutionary Multi-Criterion Optimization*. Springer, 2005, pp. 505–519.

[22] A. Muruganantham, K. C. Tan, and P. Vadakkepat, "Solving the IEEE CEC 2015 dynamic benchmark problems using Kalman filter based dynamic multiobjective evolutionary algorithm," in *Proceedings in Adaptation Learning and Optimization*. Springer Science mathplus Business Media, nov 2015, pp. 239–252.

[23] M. Helbig and A. Engelbrecht, "Benchmark functions for cec 2015 special session and competition on dynamic multi-objective optimization," Tech. Rep., 2015.

[24] S. Fine and K. Scheinberg, "Incremental learning and selective sampling via parametric optimization framework for svm," in *Advances in neural information processing systems*, 2002, pp. 705–711.

[25] H. Karshenas, R. Santana, C. Bielza, and P. Larranaga, "Multiobjective estimation of distribution algorithm based on joint modeling of objectives and variables," *IEEE Transactions on Evolutionary Computation*, vol. 18, no. 4, pp. 519–542, 2014.

[26] M. Jiang, W. Hu, L. Qiu, M. Shi, and K. C. Tan, "Solving dynamic multi-objective optimization problems via support vector machine," in *2018 Tenth International Conference on Advanced Computational Intelligence (ICACI)*, March 2018, pp. 819–824.

[27] M. Jiang, W. Huang, Z. Huang, and G. G. Yen, "Integration of global and local metrics for domain adaptation learning via dimensionality reduction," *IEEE Transactions on Cybernetics*, vol. 47, no. 1, pp. 38–51, Jan 2017.

[28] M. Jiang, Z. Huang, L. Qiu, W. Huang, and G. G. Yen, "Transfer learning-based dynamic multiobjective optimization algorithms," *IEEE Transactions on Evolutionary Computation*, vol. 22, no. 4, pp. 501–514, Aug 2018.

[29] M. Jiang, Z. Huang, G. Jiang, M. Shi, and X. Zeng, "Motion generation of multi-legged robot in complex terrains by using estimation of distribution algorithm," in *2017 IEEE Symposium Series on Computational Intelligence (SSCI)*. IEEE, 2017, pp. 1–6.

[30] M. Jiang, Y. Yu, X. Liu, F. Zhang, and Q. Hong, "Fuzzy neural network based dynamic path planning," in *2012 International Conference on Machine Learning and Cybernetics*, vol. 1. IEEE, 2012, pp. 326–330.