Appl. Math. J. Chinese Univ.
2019, 34(1): 1-26

IBEA-SVM: An Indicator-based Evolutionary Algorithm Based on Pre-selection with Classification Guided by SVM

LI Hao-ran [1] HE Fa-zhi [*1,2] YAN Xiao-hu [1]

Abstract. Multi-objective optimization has many important applications and becomes a challenging issue in applied science. In typical multi-objective optimization algorithms, such as Indicator-based Evolutionary Algorithm (IBEA), all of parents and offspring need to be evaluated in every generation, and then the better solutions of them are selected as the next generation candidates. This leads to a large amount of calculation and slows down convergence rate for IBEA related applications. Our discovery is that the evaluation of evolutionary algorithm is a binary classification in nature and a meaningful preselection method will accelerate the convergence rate. Therefore this paper presents a novel preselection approach to improve the performance of the IBEA, in which a SVM (Support Vector Machine) classifier is adopted to sort the promising solutions from unpromising solutions and then the newly generated solutions are conversely added as train sample to increase the accuracy of the classifier. Firstly, we proposed an online and asynchronous training method for SVM model with empirical kernel. The initial population is randomly generated among population size, which is used as initial training. In the process of training, SVM classifier is modified and perfected to adapt to the evolutionary algorithm sample. Secondly, the classifier divides all the new generated solutions from the whole solution spaces into promising solutions and unpromising ones. And only the promising ones are forwarded for evaluation. In this way, the evaluation time can be greatly reduced and the solution quality can be obviously improved. Thirdly, the promising and unpromising solutions are labeled as new train samples in next generation to refine classifier model. A number of experiments on benchmark functions validates the proposed approach. The results show that IBEA-SVM can significantly outperform previous works.

Received: 2018-03-03. Revised: 2018-11-28.
MR Subject Classification: 35B35, 65L15, 60G40.
Keywords: multi-objective optimization, SVM, IBEA, training, classification.
Digital Object Identifier(DOI): https://doi.org/10.1007/s11766-019-3706-1.
Supported by the National Science Foundation of China (Grant No. 61472289) and Hubei Province Science Foundation (Grant No. 2015CFB254).
* Corresponding author.

§1 Introduction

Optimization algorithms have been widely applied to diverse fields, such as computer vision, image processing [7] [51], distribution of power system, collaborative supply chain [48] [49], industry [22] [32] [50] and manufacturing [15] [39]. They are great challenges in applied science [16] [27] [29].

Especially, the multi-objective optimization problem(MOP) [3] [9] is more challenging because it has synchronously to consider the two objectives in process of optimization. MOP can be formally described as follows:

$$\begin{aligned} Minimize \quad & F(x) = [f_1(x), f_2(x), ..., f_m(x)]^T \\ s.t \quad & x = (x_1, x_2, ..., x_n)^T \in \Omega \end{aligned} \tag{1}$$

where x is the decision variable vector and Ω is the search space, and R^m is the objective vector space.F(x) is the objective vector with m real value objective functions.

In MOP, there is a restricted relationship between two objective functions. It is difficult for a single point in Ω to minimize two objectives at the same time because, in most cases, a smaller fitness of one objective function will lead to a bigger fitness of another objective function. The key to MOP is to search out a number of compromise solutions, whose objective functions achieve the minimum value on balance.

Multi-objective optimization evolutionary algorithms(MOEAs) [8] rely on evolutionary process of population to search out these compromise solutions. Although this evolutionary process can find out various solutions which locate at different place of search space, it takes a large number of evaluations before convergence because MOEAs need to evaluate all newly generated solutions for measuring their quality at each iteration.

In some application fields, considering the computation cost of evaluations, it is not worth that all newly generated solutions are evaluated. Therefore, it is necessary to reduce the number of evaluations but not to reduce the quality of compromise solutions. Constructing an efficient preselection strategy to remove out these unpromising solutions is a feasible method.

However, as best as we know, previous literature rarely discusses how to combine preselection and IBEA. And how well the preselection can work with IBEA remains unknown. This paper makes an attempt to improve indicator-based evolutionary algorithm (IBEA) by preselection strategy based on an improved SVM.

The rest of the paper is organized as follows. In Section 2, the related work of IBEA and SVM is introduced, such as the development of IBEA, basic evaluations and advanced evaluations for IBEA. In Section 3, the details of the proposed IBEA with SVM classification are described, such as the overall framework IBEA-SVM, the procedures of IBEA-SVM, online and asynchronous training method for IBEA. In Section 4, the proposed method is tested on benchmark with a number of experiments. Section 5 concludes this paper with future work.

§2 Related Work

2.1 Multi-objective evolutionary algorithms

Among different approaches for MOP, MOEA is the most popular one, which can obtain a good approximation solution of MOP.

MOEA is a process of iterative calculation [1]. Before iteration, it initializes the population which includes a number of first generation solutions. Then it launches the iteration which consists of 3 major steps. In step 1, MOEA reproduces offspring population through evolutionary operation such as crossover and mutation. In step 2 of fitness computation, MOEA calculates the fitness of offspring population. In step 3 of evaluation process, the fitness of children solutions are compared with those of parent solutions, and then the better solutions are selected as next population.

In evaluation process of single-objective problem (SOP), the fitness of objective function is used to rank the priority of all possible solutions. However, this fitness does not work in MOP. There are more than two objective functions in MOP, which means that a uniform relationship of these objective functions should be built to compare the priority of solutions.

In MOP, this uniform relationship is defined as a dominant relationship.

Let x^a, x^b be two feasible solutions, x^a dominates x^b,denoted as $F(x^a) \prec F(x^b)$, if and only if $f_i(x^a) \leqslant f_i(x^b)$ for i=1,...,m and $F(x^a) \neq F(x^b)$. x^a is non-dominant solution.x^b is dominant solution.A solution $x \in \Omega$ is called a Pareto optimal solution if there does not exist a solution $x \in \Omega$ such that $F(x) \prec F(x)$.The set of all Pareto optimal solutions is called Pareto set (PS) in decision space. The set of PS's objective function is called Pareto front (PF) in objective space.Since the whole PS cannot be calculated in most situations, decision maker usually uses a sub-space of whole PS as the approximation solution of the PS.

Typical MOEA approximation methods include NSGA2 [13], MOEA/D [42] and IBEA [52], which are based on basic theory of dominant relationship.

NSGA2 (Non-dominated Sorting Genetic Algorithm 2) introduces crowding distance to maintain population's diversity which guarantees every solution can distribute over search space. At each generation, NSGA2 adopted an elite retention mechanism to keep good parent individuals, which will be compared with offspring population to select next population. NSGA2 presented a hierarchic non-dominant sorting approach to select better solutions to approximate the PS, which divides all solutions into several clusters, computes the crowding distance of solution in each cluster and orders all solutions by cluster level and crowding distance.

MOEA/D (Multiobjective Evolutionary Algorithm based on Decomposition) decomposes a multiobjective optimization problem into a number of scalar optimization subproblems. It optimizes a number of subproblems simultaneously by evolving a subproblem population of solutions. At each generation, the population is comprised of the best solutions known for each subproblem. In MOEA/D, a series of different scales local neighborhood relations from other subproblems are also used for diversity and centrality. The neighborhood relation is defined as the distance between their aggregation coefficient vectors. When a large number of weight vectors are evenly distributed, the neighborhood relation arrives at the best status, which can approximate the PF very well. MOEA/D is a general framework for solving MOPs, which can incorporate any decomposition approach into MOEAs, such as weighted sum approach, tchebycheff approach and boundary intersection approach.

IBEA presents an indicator concept to comprehensively evaluate the solution quality. A quality indicator is a general function that maps k Pareto set approximations to a real number, which is extension to dominant relationship. The process of calculation takes into account convergence, homogeneity and universality. It is believed that indicator is a specific and operable computation method derived from dominant relationship. The indicators are utilized to measure the quality of solutions at each generation, and to select the solution set with better quality value than that of previous generation. Because the indicator is comprehensive evaluation, IBEA does not require additional diversity maintenance mechanism. This is the advantage of IBEA over MOEA/D and NSGA2 and is the reason that IBEA is widely used in many applications, such as supply chain and urban sewage treatment.

Our work is closely related with IBEA, the details of which are discussed in the following section.

2.2 Indicator-based Evolutionary Algorithm

There are two typical IBEA indicators: hypervolume indicator and additive epsilon indicator.

The first one is hypervolume indicator [1], which is a comprehensive quality evaluation method of the solution set. It evaluates coverage, homogeneity and universality simultaneously, and then obtains comprehensive evaluation results. Hypervolume indicator is defined as the volume of hypercube constructed by solution and reference solution in objective zone. After comparing indicators of two solutions, the solution with larger indicator is better than the one with smaller indicator, which means that the solution with larger indicator dominates the one with smaller indicator. As shown in Fig 1, the solutions with the largest hypervolume indicators are the pareto solutions. Therefore, the solution with better indicators can approximate to PS, which makes IBEA to converge to compromise solutions. According to hypervolume indicator approach, although several solutions owe different coordinate locations at decision space, they may have similar indicator (volume in Fig 1(left)). In this situation, the similar solutions will be equally treated in a reasonable way. This is the advantage of IBEA over other methods, which do not owe this homogeneity and universality.

The second one is Additive epsilon indicator [45], which is a binary quality indicator and is defined as the minimum distance. As shown in Fig 2, general reference sets are Pareto front, which reflects approximate priority of dominant-level. I_ϵ(A,B) represents the minimal distance, which is needed for A to be translated to dominates B. Based on the distance in objective space, undominated solutions can weakly dominate other solutions. Additive epsilon indicator is easy to be used to compare the quality of two solutions relatively to each other because of its strong separating capacity.

There are two reasons why hypervolume is more popular than additive epsilon indicator.

Firstly, hypervolume indicator is suitable for the situation of unknown PF. Most of evaluation method need to acquire the PF of MOP in advance, but the PF is unknown in practical problem. So, the research to hypervolume indicator is more significant than additive epsilon indicator.

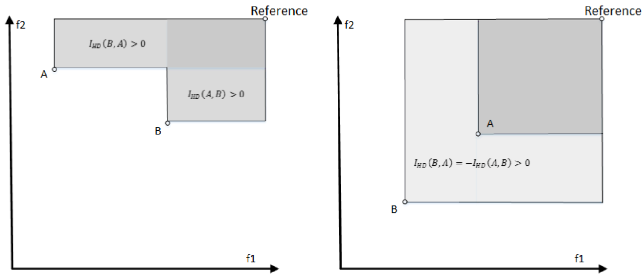

Figure 1: Illustration of I_{HD}(A,B) applied to two solutions. Left: no obvious dominant relationship between A and B.Right:A dominates B

Secondly, the evaluation method of hypervolume indicator is compliant with Pareto evaluation method [4]. It means that if the hypervolume indicator of A set is larger than that of B sets, the A set dominates B set. Thus, the hypervolume indicator is better in expandability and portability.

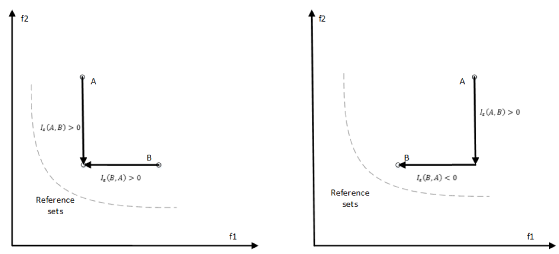

Figure 2: Illustration of I_{HD}(A,B) applied to two solutions. Left: no obvious dominant relationship between A and B.Right:A dominates B

Therefore, hypervolume indicators has more research significance than additive epsilon indicator.

2.3 Basic Evaluations and Advanced Evaluations

The early MOEAs directly evaluate all candidate solutions (children solutions plus parent solutions) by applying dominant relationship to sort candidate solutions. Dominant relationship is used for comparing the priority of two object functions among a pair of solutions in process of evaluation.

There are two typical computation models for dominant relationship.

The first one is Non-dominant selection(NDS), which uses dominant relationship and crowding distance to rank all candidate solutions. Firstly, candidate solutions are divided into several clusters by dominant relationship and solutions in the same cluster cannot dominant each other. Secondly, NDS calculates crowding distance of solutions in the same cluster and ranks them by crowding distance. Thirdly, NDS orders all candidate solutions by dominant relationship among clusters and ranking within the cluster.

The second one is Indicator-based selection(IBS) [53], which uses indicator to sort rank all candidate solutions. Indicator is a more accurate expression of dominant relationship. Indictor function is defined as a vector from solutions to reference solutions, whose value is the distance between two solutions in objective space. So, IBS uses fitness pointer function to quantitatively express dominant relationship. Firstly, IBS calculate fitness pointer function of each solution. Secondly, among these solutions, IBS chooses a solution with the smallest fitness value. Thirdly, IBS removes this solution and updates the fitness values of the remaining solutions. The first removed one is better than later one in dominant relationship. According to this regulation, IBS sorts whole candidate solutions.

The MOEAs directly based on dominant relationship are NSGA2, PESA2 [10] and SPEA2 [18].

However, in most cases, the run time in fitness computation and evaluation (especially evaluation) are very high. The reason is that MOEA must compute and evaluate all candidate solutions of huge numbers. At each generation, evolutionary operation creates children solutions from parent solutions. The candidate solutions of this iteration include both parent solutions and children solutions. If an evolutionary algorithm directly evaluates all candidate solutions, it will take a very long time for the operation.

Candidate solutions can be divided into three parts according to value of fitness: possible solutions, impossible solutions and uncertain solutions. The possible solutions are a group of solutions whose fitness is better than that of other solutions on all object functions. In contrast, the impossible solutions are a group of solutions whose fitness is worse than that of other solutions on all object functions. After excluding possible and impossible solutions, the remaining solutions are uncertain solutions, which are hard to be compared with fitness.Therefore, the uncertain solutions must be evaluated by dominant relationship to distinguish dominant-level.

The early MOEA methods ignored the obvious difference between possible solutions and impossible solutions and directly evaluated the dominant relationship based on all solutions [30]. The total cost of fitness computation and domination evaluation is very high. Therefore, in applied science and engineering, the function evaluation can be very expensive financially and computationally. There is a great demand in reducing the cost of function evaluation.

This is the motivation of preselection strategy for MOEA problems.

There are two situations of preselection strategy. The simple situation is to exclude the impossible solutions. In this situation, the impossible solutions can be easily removed with fitness computation and comparison [3]. Furthermore, the advanced situation is not only to remove these impossible solutions, but also to filter some from uncertain solutions. In second situation, preselection strategy divides offspring solutions into two class: promising solutions and unpromising solutions. The promising solutions consist of possible solutions and better uncertain solutions. The unpromising solutions consist of impossible solutions and worse uncertain solutions.

The second situation is a very challenging for preselection. For a given uncertain solutions set, it is difficult to distinguish better solution and worse solutions. So how to design a high-performance classifier to acquire an accurate preselection result is a critical step for preselection strategy.

Zhang and Zhou [41] put forward a classification based preselection (CPS) framework and use classification and regression tree and k-nearest neighbor to build classification model in NSGA2. The CPS strategy is implemented into a Pareto domination based multi-objective evolutionary algorithm framework. Although CPS do perform in Pareto domination based approaches, they do not ensure feasibility on indicator based approaches and the decomposition based approaches. Our work exactly fills the void of how to apply CPS to indicator based approaches.

Lin and Zhang [23] investigates how to use SVM classifier for preselection to improve the performance of the multi-objective evolutionary algorithm based on decomposition(MOEA/D) [43]. Their algorithm builds a classification model on search space to filter all new generated solutions, removes these unpromising solutions and mainly evaluates those promising solutions. Here we apply a similar strategy, but our method does not make that the predicting results affect evolutionary operations all the processing, we set a threshold generation after which the prediction of SVM starts to work. Because we consider that SVM model is immature at early stage.

Fan and Hu [14] make a survey and demonstrates the possibility of putting unsupervised learning into MOEA. They find that machine learning can extract the potential relation among solutions, which in return benefit the evolutionary process if utilized properly. In their review, they propose a model-based MOEAs, which integrates with k-mean and estimation of distribution(EDA) to assist evolutionary operation. As we should see, our method is consistent with this idea and supplement the range of application.

However, as best as we know, previous literatures rarely discuss how to combine preselection and IBEA. And how well the preselection can work with IBEA remains unknown. The problem is more difficult than it seems at first sight. Clearly, this problem cannot be solved by a simple integration of classifiers such as SVM because it is difficult to balance the time saved by preselection and the time consumed by the building of SVM model. On the contrary, it requires a more sophisticated approach.

Therefore, this paper proposes a novel preselection approach based IBEA, which adopt an improved SVM classification model to improve the performance of IBEA.

§3 The proposed IBEA-SVM algorithm

3.1 The overview of IBEA-SVM framework

The proposed IBEA algorithm is named as IBEA-SVM. In our approach, the preselection is a SVM classifier which distinguishes promising solutions from candidate solutions. The main challenge is how to carefully integrate the SVM and IBEA.

As shown in Fig.3 (a), there are 3 main operations in IBEA: the reproduction operation produces children solutions; the evaluation operation computes children solutions' indicators; the selection operation sorts the best performance solutions from children solutions and parent solutions to next generations.

Although the quality of children solutions become better in overall processing of evolution, the most efficient evolutionary operation still can't guarantee that all children solutions are better than parent solutions. In practice, the unpromising solutions occupy a certain quantity of whole children solutions. In traditional non-preselection IBEA, these unpromising solutions will be evaluated and compared with other solutions. IBEA will reduce the considerable amount of calculation if some parts of unpromising children solutions are removed before evaluation.

As shown in Fig.3 (b), the most difficult and urgent problem is how to build a classifier to recognize unpromising and promising solutions. As a classical and practical classification method, SVM is very appropriate for this problem. SVM is a supervised learning model with associated learning algorithms, which analyzes data for classification and regression analysis[5]. There are two main steps of SVM. In step 1 of training, a set of sample data is labeled as multiple tags and they are regarded as train data. SVM uses these train data to train its model. In order to acquire a higher accuracy, model need update its parameter to match these train data. In step 2 of classifying, another set of sample data from same aggregate without tags is regarded as test data. SVM uses updated model to give test data different tags. Through these steps, this set of test data is divided into different classes.

As shown in Fig.3 (c), according to process of IBEA and structure of SVM, we propose a novel preselection framework to integrate IBEA with SVM. After last iteration of selection, a set of solutions is automatically labeled 2 tags. The selected solutions are positive sample data and the unselected solutions are negative sample data. All of them are regarded as train data. Before process of reproduction, these train data are sent to training process of SVM's model. And then model uses these train data to update its parameter. At the same time, IBEA uses reproduction operation to produce offspring solutions. These offspring solutions are naturally regarded as the test data without tags. And then updated model gives these offspring solutions different tags. The offspring with positive tag can be delivered to evaluation process. At each iteration, SVM's model need to update its parameter one time.

The proposed integration framework of IBEA and SVM is shown in Fig.3 (c).

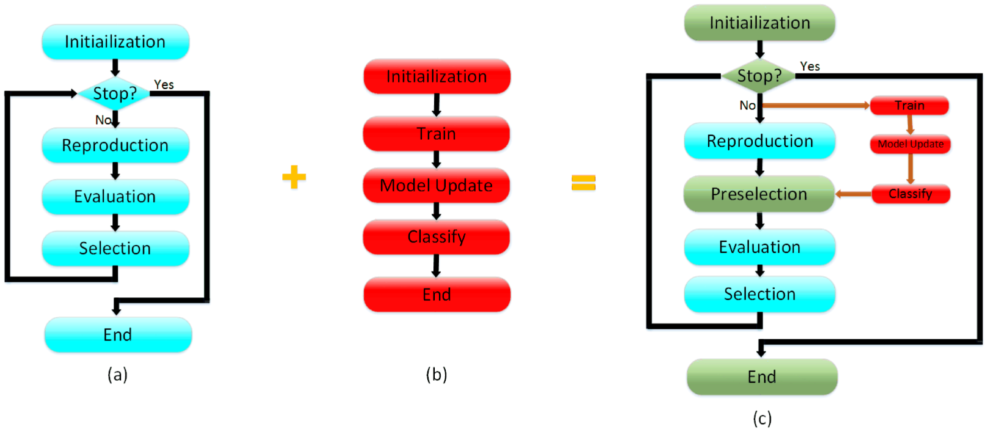

Figure 3: Integration of IBEA and SVM.(a)IBEA(b)SVM(c)IBEA-SVM

3.2 IBEA-SVM Algorithm

The proposed IBEA-SVM algorithm is shown in Algorithm 1, which has following 7 steps:

• Step1. Initialization: In line 1, the algorithm implements initializing the population, including assign random decision variable for N solutions and calculate the fitness value for each solution.

• Step2. Classifier training: In line 2-6, the algorithm divides population P into two parts P1 and P2, which consist of the train data. The classifier uses train data to build model.

• Step3. Reproduction: In line 7, the algorithm generates an offspring population Pc from parent population P by Differential Evolution(DE) or Genetic Algorithm(GA).

• Step4. Preselection: In line 8-16, the SVM model predicts for Pc and assign the label for solutions from P_c. If it satisfies reliability, the solutions whose label is 1 are selected into candidate population Q.

• Step5. Selection: In line 17, the algorithm uses indicator-based selection to pick out N solutions from old population P and candidate population Q. These N solutions constitute new population P for next iteration. This step is the same as the original IBEA's selection step.

• Step6. Update train data: In line 18-21, the SVM model needs to update its train data for building model in next iteration. The same operation divides Q into two parts: Q1 and Q2. It selects the top N/2 best of solutions from P1 and Q1, which make up of new P1. The same operation does in P2 and Q2 to produce new P2. P1 and P2 make use for next iteration.

• Step7. Stopping running and output: In line 23-25, when it arrives the termination condition, it exports approximation to the PS and approximation to the PF.

The inputs of IBEA-SVM algorithm are:

• Continuous Multi-Objective Problem

• N: the size of population

• Termination condition

- α: reliability that algorithm rely on predicting of SVM model

---

Algorithm 1 IBEA-SVM

Require: MOP: Multi-Objective Problem; N: the size of population; Termination condition; α: reliability that algorithm rely on predicting of SVM model

Initialize the population $P = \{x_1, x_2, ..., x_N\}$
Set $P1 = IBS(P,—N/2—)$ and P2=P-P1
while Termination condition is not satisfied **do**
    Train a SVM classification model based on P1(label=1) and P2(label=0)
    Model=SVMTrain(P1,P2)
    Generate a new candidate population P_c from P by DE operation
    Use model to predicate the labels of solutions of P_c
    labels=SVMPredict(Model,P_c)
    $P = \{y \in P_c | y'label = 1\}$
    Randomly choose y $\in$ Y
    Generate a random number r1=uniform[0,1]
    if r1¿α **then**
        set Q=Q∪y
    end if
    Q is the candidate population after prediction
    Update P=IBS(P∪Q,N)
    Set Q1 be the nondominated population in Q and Q2 be the dominated population in Q
    Update P1=IBS(P1∪Q1,|N/2|)
    Update P1=IBS(P2∪Q2,|N/2|)
end while

Ensure: Approximation to the PS: P Approximation to the PF: F(p)

3.3 An online and asynchronous training method for SVM model with empricail kernel

Different from the original IBEA, our approach adopt a classifier to preselect the promising solutions from candidate solutions. Therefore, the difficult of our algorithm is how to carefully integrate an SVM classifier into IBEA.

How to design a high-performance classifier to acquire an accurate preselection result is a critical step for preselection strategy. This is the key issue of the proposed IBEA-SVM Algorithm. Therefore, we propsoed three strategies to deal with the issue.

The first challenge is how to train a SVM classifier with high accuracy. The process of SVM classification includes SVM training, model updating and SVM classifying. For an accurate classification, the training process is the most important one. The reason comes from two aspects. The result of a poor training will be that the better solutions in uncertain solutions are selected as unpromising solutions. Also, the result of a poor training will be that the worse solutions in uncertain solutions are selected as promising solutions.

Typical training methods include offline training and online training.

In the offline training method, existing IBEA experiment data are firstly collected to produce the training data. And then, these data will be used to tune parameters of a SVM model with best fitness. Once the learning process ends, the parameter will stop updating. At last, this trained SVM model with fixed parameters is integrated into IBEA process to guide the preselection process at each iteration.For simple problems, the offline training is workable because SVM's parameters are fixed in IBEA process. However, the fixed parameters limit the generalization ability of SVM mode and the mode can not accurately learn and predicate the whole evolutionary development.

In the online training method, real-time IBEA experiments produce training data at each iteration of IBEA process. The real-time data are used to dynamically update the parameter of a SVM model. As the iteration developing, the parameters of SVM model are continuously adjusted to fit the latest development of solutions. Therefore, the online training method is more accurate than offline method for dynamic and complex much-objective problems.

In this paper, we present an online method for SVM model to accurately learn the trend of IBEA evaluation. In order to learn dynamically the lasted parameters, the N-1 th iteration solutions are collect to produce the training data for N th iteration in the IBEA process. Specifically, for N-1 th iteration solutions data, we divide the data into positive training data and negative training data. The positive training data is the selected solutions from candidate solutions at N-1 th iteration. The negative training data is the abandoned solutions from candidate solutions at N-1 th iteration. At the N th iteration, SVM model will use both the positive training data and the negative training data to confirm the fittest parameters.In this way, the SVM model parameters are continually updated and developed with training data at each iteration, and ultimately the SVM model can match the whole evolutionary development of IBEA.

The second challenge is when a SVM classification will be used to preselect the solution. In the previous preselection literatures, the training process and classifying process are synchronized in the whole evolutionary process. If IBEA needs N iterations to find out PS, it also needs N times to train and classify in the whole process. The existing methods are brute and poor efficiency. It ignores the obvious characteristic of change of model.

In this paper, we discover that at early time of IBEA-SVM, evolutionary operation implements explore search, whose randomness is high and step size is large. At the same time, the SVM model is trained by early sample data, which contributes to higher false alarm rate and lower accuracy.

According to these two reasons, the early use of SVM predicting is inappropriate. On the contrary, at the later stage of IBEA-SVM processing, the SVM classifier has been trained with a large number of sample data and is becoming mature and mature, which ultimately results in a convergent result.

Therefore, we proposed an asynchronous training method for SVM model, in which the training processing is conduced in early stage, while both training processing and predicting processing are conducted in later stage. Specifically, we adopt a model conversion factor to control the critical generation of algorithm. Before the critical generation, we only train the SVM model. After the critical generation, we not only train the SVM model, but also apply the SVM model to predicting. The model conversion factor is a random number of Gaussian distribution. According to experience of repeated experiments, we set that mean value is 0.5 and variance is 0.1. The conversion generation is the product results between conversion factor and total generation. Before conversion generation, IBEA-SVM only implements SVM training and model update. After conversion generation, IBEA-SVM implements SVM training, model update and classifying. This method not only ensures the quality of model, but also improves the speed of calculation.

The third challenge is how to improve the computation efficiency at later stage of evolutionary process. The asynchrony of training and classifying contributes to improving efficiency. However, after critical generation, the training situation changes with the increasing of generation. As generation increases, the amount of repeated train data also increase. If the SVM model is trained with the repeated data, the train processing will take a very long time. In order to speedup the training processing at later stage, we propose a strategy to filtrate out the repeated data before SVM training. Thus, our method is faster than previous online training method.

After addressing the three challenging issues, the proposed training method for SVM model has following steps.

• Step1. The initialized population P is divided into two parts: P1 and P2. We use P1=IBS(P,|N/2|) to denote the indicator-based selection. P1 set is the top 50% of P solutions in this selection. P2 set consists of other solutions from P. The solutions of P1 set is considered to be promising, so their label is 1. The solutions of P2 set is considered to be unpromising, so their label is 0. The P1 and P2 constitute the train data. The SVM classifier use these train data to build its model and determine model's parameter.

• Step2. After the population P reproduces offspring population P_c by GA [17] or DE [11] operation, SVM model can estimate which solutions of Pc belong to promising clustering. It rejects the probable unpromising solutions and reserve these promising solutions. Q set is the candidate population after SVM predicting. The number of promising solutions is due to the accuracy of predicting. In fact, at the earlier stage of running, the scale of train data is not big enough for modeling, so there is a reliability parameter that decide whether it uses SVM model to predict.

• Step3. SVM model needs to improve at each iterative, as well as the train data. The candidate population Q is divided into two parts: Q1 and Q2. Q1 set is the promising solutions which is the same as P1. Q2 set is the unpromising solutions which is the same as P2. The algorithm uses indicator-based selection to pick out the N/2% solutions from P1 and Q1 then collects them as new P1. The same operator does in P2 and Q2, which leads to new P2. The new P1 and P2 are used for training model in next iterative. In a word, SVM model is built at every iteration by updated train data. As the number of iteration is increasing, the performance of SVM model is improving.

Finally, we use experiments to adopt a suitable kernel and fine-tuned parameters for MOP in our IBEA-SVM framework besides the online training and asynchronous classification.

• How to explore an empirical SVM model with suitable kernel will be discussed in following section 4.2, in which we firstly select 2 practical kernel functions from 4 usual kernel functions by theoretical analysis, and then compare these 2 kernel functions by experiment to decide the most suitable kernel function of IBEA-SVM.

• Furthermore, in order to match the above SVM kernel. we also use experiments to select a suitable evolutionary operation. How to explore an fast-convergence and widely-search evolutionary operations will be discussed in following 4.3, in which we compare 2 typical evolutionary operations with experiment to decide the best one to match the evolutionary process of IBEA-SVM.

In short summary of this section 3, we present an online and asynchronous SVM model with empirical kernel.

§4 Experimental research and empirical model

4.1 Experiment methodology: steps, data and parameters setting

Our experiment methodology is inspired from previous literatures, such as [2].Therefore, the proposed experiment has following steps.

In section 4.2, after theoretical analysis, we initially choose RBF kernel and Sigmoid kernel from 4 commonly-used SVM kernels. Then, we use experiments to evaluate influence of the two initial kernel functions, and select the Sigmoid kernel as the best one.

In section 4.3, based on IBEA's evolutionary framework with Sigmoid kernel, we compare two widely-used GA operation and DE operation to evaluate influence of reproduction operation, and select the DE as the best one.

In section 4.4, the proposed IBEA-SVM algorithm is compared with previous IBEA algorithms on benchmark data set. Different combinations of SVM kernel and reproduction operation are tested in proposed IBEA-SVM framework. The results show that our IBEA-SVM outperforms state-of-the-art IBEA algorithms on benchmark data set.

Our data set comes from 11 benchmark functions [51], which are ZDT1, ZDT2, ZDT3, ZDT4 ZDT6, DTLZ1, DTLZ2, UF1 UF2 UF3 and UF4. These benchmark functions are classical and universal. Most of them contain appropriate decision variable and 2 or 3 objective functions. ZDT1, ZDT2, ZDT3, ZDT4 and ZDT6 all have 2 objective functions. Among them, ZDT1, ZDT2 and ZDT3 have 30 decision variables. ZDT4 and ZDT6 have 10 decision variables. DTLZ1 and DTLZ2 both have 3 objective functions. They have 10 decision variables. UF1, UF2, UF3 and UF4 all have 2 objective functions and 30 decision variables.

Our parameters setting also follows the standard method [45] [54], which is states as follows:

• Population Size N is 200

• The probability of relying on the SVM model to update candidate solutions: 1-α= 0.9

• The probability of mutation in GA is 0.1

• The probability of crossover in GA is 0.9

• Maximal number of function evaluations: 10,000 for ZDT and DTLZ test instances, 20,000 for UF test instances

• The type of indicator is hypervolume indicator

• The model conversion factor is 0.5

4.2 Influence of SVM Kernels

Before experiments, we theoretically analyze the influence of 4 SVM's kernels within our IBEA-SVM framework, and select 2 suitable kernels for following experiments.

In building SVM classification model, the SVM kernel is a significant factor to affect the performance of classification. SVM kernel is a method to simplify the complicated calculation in high dimensional space. Kernel function is defined as K(x,y)=<f(x),f(y)> ,where x and y are the input variable in n dimensional space. f is the map from n dimensional space to m dimensional space. <x, y> is the mathematical operators of dot product.

SVM kernel function has 4 specific functions: Linear, Polynomial, Radial Basis Function and Sigmoid.

• Linear function's expression is K(x,y)=x*y. When the number of features is very large, linear function is always chosen.

• Polynomial function's expression is K(x,y)=$[\alpha$*(x*y)+$\beta]^d$, and this kernel is rarely used because of the complexity of model and excessive number of parameters.

• Radial Basis Function(RBF)'s expression is K(x,y)=$e^{-\alpha*|x-y|^2}$.Because RBF kernel can handle non-linear situation and is convenient of computing. It is the first choice for general users.

• Sigmoid function's expression is K(x,y)=tanh[α*(x*y)+β], which takes two parameters: α and β. When $\alpha > 0$, we can view α as a scaling parameter of the input data, and β as a shifting parameter that controls the threshold of mapping. When $\alpha > 0$ and $\beta < 0$, it is more suitable for the sigmoid kernel.

The preselection is a non-linear classification process and has a small number of features. Two of above four kernels, the RBF kernel and Sigmoid kernel, match well the characteristic of preselection. Therefore, we choose the RBF kernel and Sigmoid kernel as candidate kernels in next step of experiments, in which the best one will be selected.

For a fair comparison, the experiments adopt GA reproduction operation, which is widely used in previous methodology, such as literatures [12].

In following tables, for illustration, IBEA-SVM with RBF is denoted as RBF and IBEA-SVM with Sigmoid is denoted as Sigmoid.

Table 1 shows the mean IGD value of IBEA-SVM with RBF and Sigmoid after 20%, 40%, 60%, 80% and 100% FES on 11 benchmark functions over 30 runs of experiment. Figure 4 plots the trend of IGD between RBF and Sigmoid. RBF is red line. Sigmoid is blue line.

According to Table 1, we can conclude that Sigmoid is better than RBF on these 11 benchmark functions. Actually, Sigmoid takes 34 better performance on 55 comparisons in the

Table 1: The mean IGD value of IBEA-SVM with RBF and Sigmoid after 20%,40%,60%,80%, and 100% FES over 30 runs on 11 benchmark functions.'+','-',and'=' denotes that IBEA-SVM with RBF performs better, worse, or similar to IBEA-SVM with Sigmoid. According to the Wilcoxon test with the 5% significance level.

Benchmark	Kernel	20%	40%	60%	80%	100%	Wilcoxon
zdt1	RBF	0.19157	0.01767	0.00553	0.00434	0.00417	0/5/0
	SIGMOID	0.14241	0.01479	0.00525	0.0042	0.00402	
zdt2	RBF	0.93593	0.14624	0.05134	0.02139	0.01341	0/5/0
	SIGMOID	0.78583	0.04312	0.03731	0.02046	0.01138	
zdt3	RBF	0.11449	0.01631	0.01391	0.01267	0.01212	4/1/0
	SIGMOID	0.08911	0.01727	0.01526	0.01488	0.01473	
zdt4	RBF	1.20654	1.28859	1.39949	1.40918	1.26812	0/5/0
	SIGMOID	0.87197	0.66652	0.52608	0.52578	0.52577	
zdt6	RBF	2.27802	0.49855	0.0964	0.02487	0.01158	0/5/0
	SIGMOID	1.98142	0.37931	0.07705	0.02125	0.01026	
dtlz1	RBF	2.62201	0.90206	0.21746	0.17479	0.16331	4/1/0
	SIGMOID	3.44551	0.38019	0.22307	0.19474	0.16506	
dtlz2	RBF	0.00607	0.00488	0.00484	0.00482	0.00478	5/0/0
	SIGMOID	0.00703	0.00543	0.00534	0.00518	0.00514	
UF1	RBF	0.17185	0.12355	0.11953	0.11822	0.12472	1/4/0
	SIGMOID	0.13925	0.1228	0.12174	0.11466	0.11991	
UF2	RBF	0.08664	0.06496	0.07079	0.06069	0.06367	1/4/0
	SIGMOID	0.08882	0.0639	0.05604	0.05331	0.05386	
UF3	RBF	0.48587	0.31818	0.27907	0.31537	0.29439	2/3/0
	SIGMOID	0.47977	0.38767	0.28563	0.25595	0.25355	
UF4	RBF	0.09862	0.07959	0.07153	0.0688	0.06668	4/1/0
	SIGMOID	0.09858	0.08329	0.07774	0.0725	0.0701	
sum		3/8/0	4/7/0	6/5/0	4/7/0	4/7/0	21/34/0

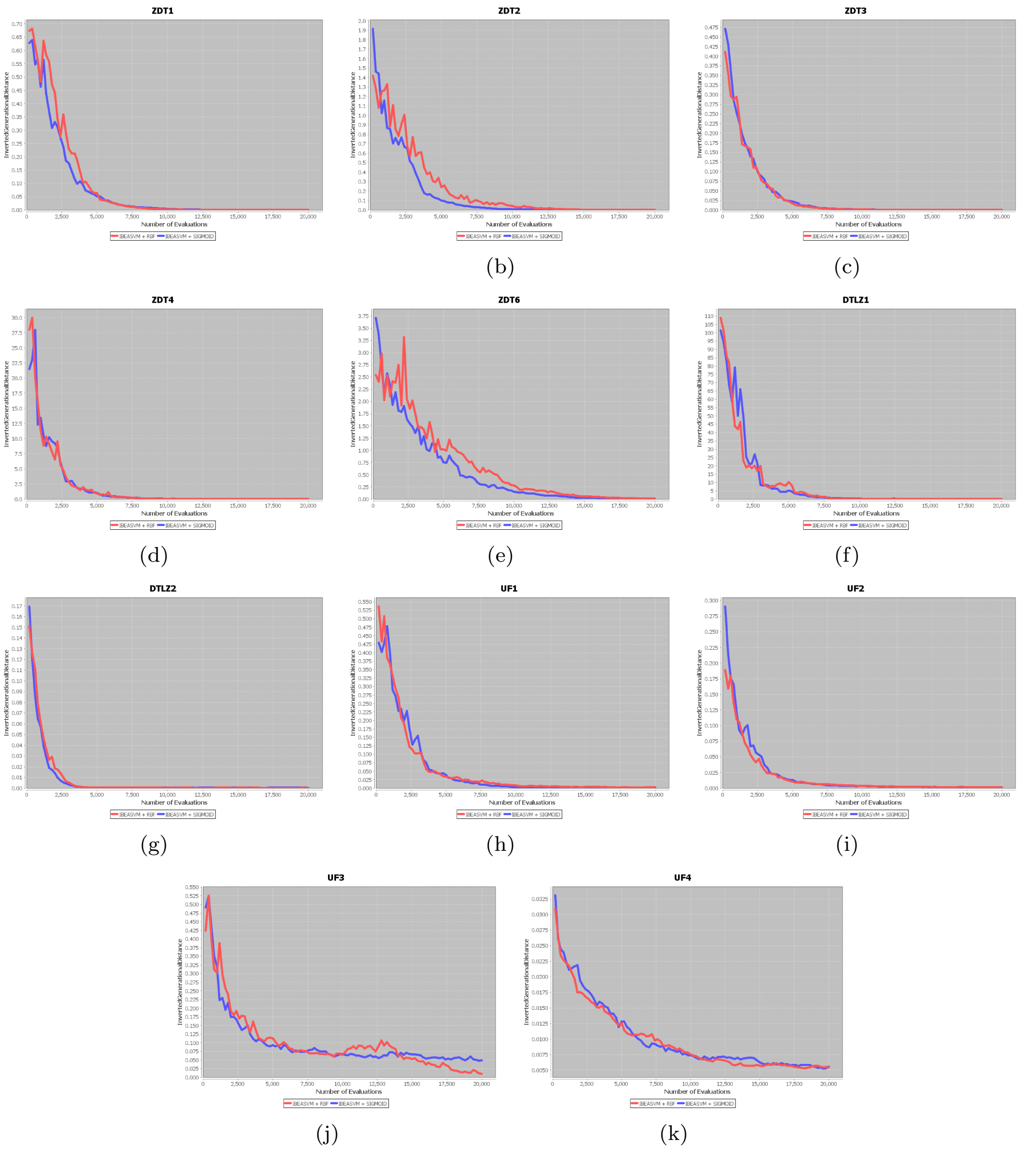

Figure 4: The mean IGD values versus the number of function evaluations obtained by IBEA-SVM with RBF and Sigmoid

Wilcoxon test.

In ZDT family, except zdt3, Sigmoid is better than RBF. The zdt3 is a benchmark function known with discontinuous PF, while others all have continuous PF. It can be concluded that RBF is better at handling the discontinuous PF benchmark functions.

In DTLZ family, RBF is better than Sigmoid in these 2 DTLZ benchmark functions. It means that RBF is better at handling benchmark function with more than 3 objectives.

In UF family, except UF4, Sigmoid is better than RBF. UF4 has more complex function forms than other 3 UF functions. It is derived that Sigmoid is better at handling benchmark functions of common function form.

In summary, overall speaking, the sigmoid kernel is the best one for our framework of IBEA-SVM.

4.3 Influence of Reproduction Operator

According to the experiment methodology, this section addresses the influence of reproduction operation. The proposed IBEA-SVM with sigmoid kernel is tested with GA operation and DE operation. For simply, we use GA and DE to denote IBEA-SVM with GA and IBEA-SVM with DE respectively.

Inverted General Distance (IGD) is defined as the computed distance between output PF and true PF, which is widely used for measuring the quality of output PF in mainstream literatures. So, in our manuscript, IGD is used for the measurement index to evaluate quality of MOEAs.

Table 2 reveals that the mean IGD value of IBEA-SVM with GA and DE after 20%, 40%, 60%, 80% and 100% FES on 11 benchmark functions over 30 runs of experiment. Figure 5 plots the trend of IGD between GA and DE. GA is red line.DE is blue line.We can see that on all the 55 comparisons, IBEA-SVM with DE performs better than GA on 31 comparisons, and worse than GA on 24 comparisons.

In ZDT family, DE operation is better than GA operation on whole results. Especially at the early stage of convergence, DE operation completely defeats GA operation. The reason is the different selection strategy between GA and DE. In GA's selection, the candidate solutions simply replace the parent solutions with a certain probability. In DE's selection, DE will compare the fitness between a pair of parent solution and candidate solution, and select the better one to realize the evolution of population. So, DE has a higher search efficiency than GA.

In DTLZ family, GA operation is better than DE operation in function with more than 3 objectives. The reason is the different number of selected candidate solutions. In DE operation, each offspring solution is only compared with its parent solution. This one-to-one selection strategy with limited candidate number may lead to low-quality solutions. In GA operation, the selected strategy is based on the population. The large number of candidate solutions will contribute to high-quality solutions.

In UF family, except UF3, DE operation is better than GA operation. The difference in

Table 2: The mean IGD value of IBEA-SVM with GA and DE after 20%,40%,60%,80%, and 100% FES over 30 runs on 11 benchmark functions.'+','-',and'=' denotes that IBEA-SVM with RBF performs better, worse, or similar to IBEA-SVM with Sigmoid. According to the Wilcoxon test with the 5% significance level.

Benchmark	Operator	20%	40%	60%	80%	100%	Wilcoxon
zdt1	GA	0.11758	0.01826	0.0055	0.00433	0.00422	0/5/0
	DE	0.07097	0.00682	0.0041	0.00401	0.00395	
zdt2	GA	0.85865	0.12839	0.04863	0.02363	0.01063	3/2/0
	DE	0.79235	0.06268	0.05186	0.02473	0.01241	
zdt3	GA	0.09923	0.01946	0.01268	0.01299	0.01224	3/2/0
	DE	0.04629	0.01809	0.01618	0.01576	0.01573	
zdt4	GA	1.83131	0.58145	0.41357	0.21495	0.20414	3/2/0
	DE	0.69603	0.59085	0.36182	0.22608	0.20561	
zdt6	GA	2.07314	0.37145	0.09714	0.02623	0.01251	1/4/0
	DE	1.81634	0.38419	0.08254	0.01724	0.01052	
dtlz1	GA	0.66273	0.45439	0.23948	0.18856	0.16113	3/2/0
	DE	0.47732	0.85313	0.57125	0.31526	0.14701	
dtlz2	GA	0.00614	0.0049	0.00485	0.00487	0.00489	5/0/0
	DE	0.00696	0.00523	0.00534	0.00526	0.00533	
UF1	GA	0.12746	0.11899	0.1219	0.12475	0.11868	2/3/0
	DE	0.15072	0.12185	0.12039	0.11456	0.11155	
UF2	GA	0.09339	0.06936	0.06781	0.06194	0.05154	0/5/0
	DE	0.09284	0.05792	0.05252	0.05037	0.04609	
UF3	GA	0.47912	0.32027	0.28692	0.2995	0.31098	4/1/0
	DE	0.49407	0.39301	0.31472	0.31889	0.27828	
UF4	GA	0.09869	0.08143	0.07441	0.07001	0.06702	0/5/0
	DE	0.08748	0.06966	0.06216	0.05822	0.05689	
sum		3/8/0	6/5/0	5/6/0	6/5/0	4/7/0	24/31/0

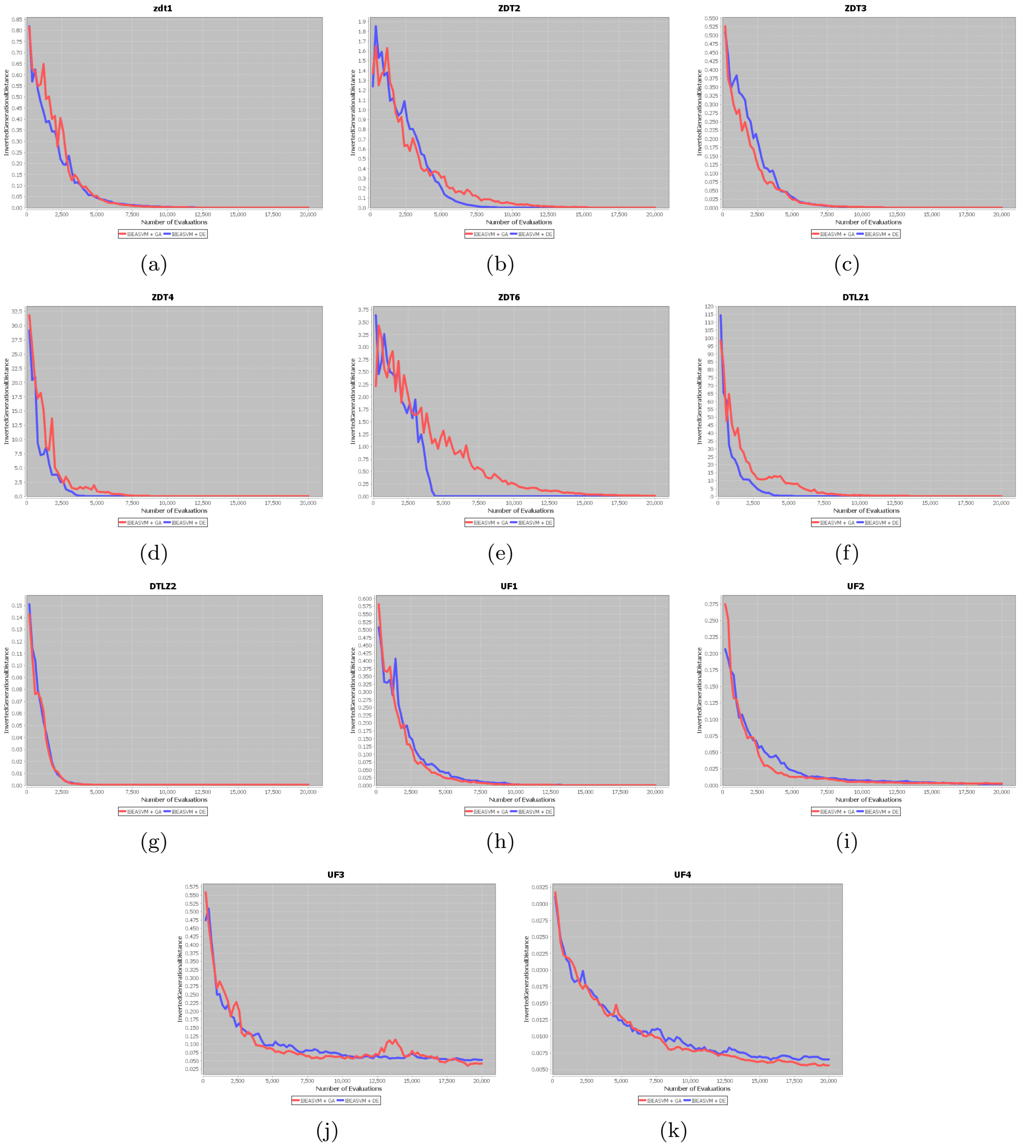

Figure 5: The mean IGD values versus the number of function evaluations obtained by IBEA-SVM with GA and DE

Table 3: The mean IGD value of IBEA-SVM and IBEA after 20%,40%,60%,80%, and 100% FES over 30 runs on 11 benchmark functions.'+','-',and'=' denotes that IBEA-SVM with RBF performs better, worse, or similar to IBEA-SVM with Sigmoid. According to the Wilcoxon test with the 5% significance level.

Benchmark	Algorithm	20%	40%	60%	80%	100%	Wilcoxon
zdt1	IBEA-SVM	0.08689	0.0067	0.00421	0.00418	0.00417	4/1/0
	IBEA	0.20546	0.01715	0.00543	0.00435	0.00415	
zdt2	IBEA-SVM	0.76314	0.07315	0.03971	0.02015	0.01094	3/2/0
	IBEA	0.83341	0.09314	0.04178	0.01908	0.00919	
zdt3	IBEA-SVM	0.0483	0.01644	0.01447	0.01516	0.00714	4/1/0
	IBEA	0.10883	0.01744	0.01494	0.01156	0.00931	
zdt4	IBEA-SVM	0.88154	0.73131	0.54141	0.39131	0.20414	2/3/0
	IBEA	1.16317	0.93141	0.51371	0.36189	0.19314	
zdt6	IBEA-SVM	1.98142	0.38141	0.13452	0.02241	0.01411	3/2/0
	IBEA	1.77802	0.34855	0.14142	0.02641	0.01513	
dtlz1	IBEA-SVM	0.49429	0.82372	0.23994	0.16521	0.16269	2/3/0
	IBEA	3.06551	0.70128	0.22656	0.17457	0.15958	
dtlz2	IBEA-SVM	0.00594	0.00483	0.00474	0.00467	0.00465	3/2/0
	IBEA	0.00612	0.00491	0.00469	0.0046	0.00487	
UF1	IBEA-SVM	0.14788	0.11976	0.11668	0.12243	0.11923	4/1/0
	IBEA	0.17562	0.13504	0.14368	0.12175	0.12173	
UF2	IBEA-SVM	0.09231	0.05948	0.05681	0.05003	0.04555	4/1/0
	IBEA	0.09702	0.07152	0.05369	0.05732	0.05586	
UF3	IBEA-SVM	0.48287	0.33876	0.29314	0.28833	0.27496	3/2/0
	IBEA	0.47279	0.31467	0.30206	0.30587	0.31784	
UF4	IBEA-SVM	0.08534	0.07987	0.05932	0.05532	0.05224	4/1/0
	IBEA	0.09787	0.07518	0.07209	0.0692	0.05313	
Sum		9/2/0	7/4/0	5/6/0	6/5/0	7/4/0	36/19/0

performance is mainly due to crossover operation and selection operation. DE uses difference vector from a three parent solutions to produce candidate solutions, which leads to diversity results and more reasonable search directions. GA usually uses two parent solutions to produce candidate solutions, which leads a less diversity than DE.

In summary, on the whole, the DE operation is the best one for the proposed IBEA-SVM framework.

4.4 Comparison with other state-of-the MOEAs

In this section, we compare proposed IBEA-SVM algorithm with other state-of-the MOEAs.

The comparison results with classical IBEA algorithms are shown in Table 3 and Fig. 6. Table 3 shows the mean IGD value of IBEA-SVM and IBEA after 20%,40%,60%,80%, and 100% FES on 11 benchmark functions over 30 runs of experiment. Figure 6 plots the trend of IGD among IBEA-SVM(Sigmoid+DE), IBEA-SVM(RBF+GA) and IBEA. IBEA-SVM(Sigmoid+DE) is green line. IBEA-SVM(RBF+GA) is blue line. IBEA is red line.

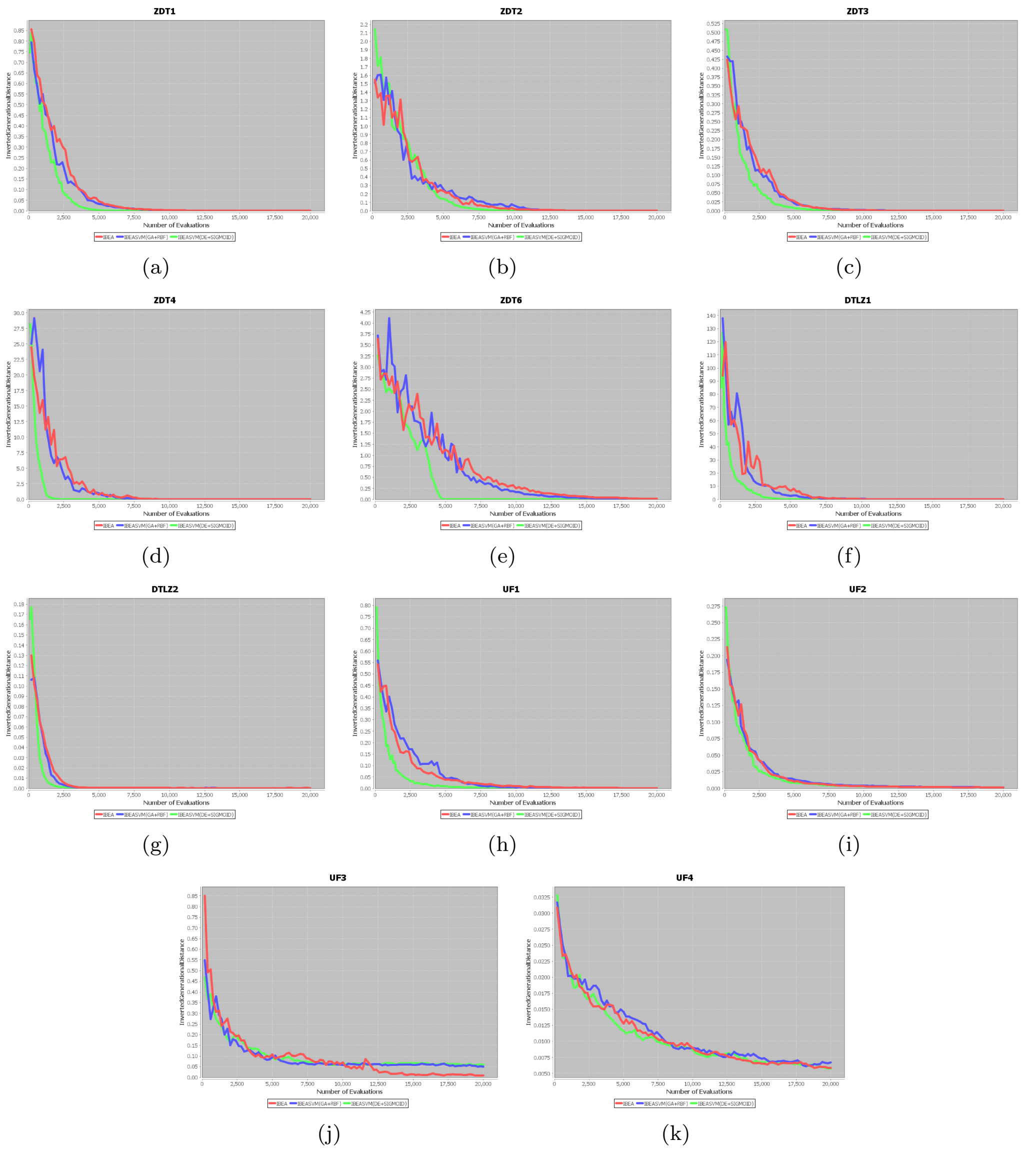

Figure 6: The mean IGD values versus the number of function evaluations obtained by IBEA-SVM and IBEA

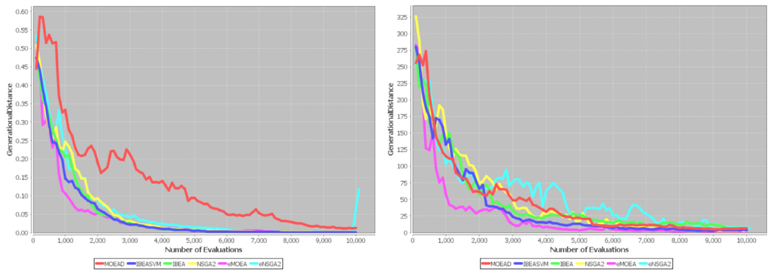

Figure 7: The mean IGD values versus the number of function evaluations obtained by MOEAD, IBEA-SVM, IBEA, NSGA2, eMOEA and eNSGA2 on ZDT3 function and DTLZ3_2 function.

According to above statistical results in Table 3 and Figure 6, IBEA-SVM acquires the better performance on 34 comparisons, worse performance on 21 comparisons. It can be concluded that he proposed preselection approach to remove unpromising solutions before evaluation in IBEA-SVM significantly improve the performance of classical IBEA algorithms.

The proposed IBEA-SVM is also compared with state-of-the-art MOE algorithms, and it is shown in Fig 7.

The difference between IBEA-SVM and SVM is that IBEA-SVM uses preselection approach to remove unpromising solutions before evaluation. This approach leads that IBEA-SVM acquires the better performance on 34 comparisons, worse performance on 21 comparisons. The statistical results in Table 3 denote that the preselection approach can significantly improve the performance of IBEA.

Figure 7 plots the trend of IGD among MOEAD, IBEA-SVM, IBEA, NSGA2, eMOEA and eNSGA2 in ZDT3 function and DTLZ3_2 function.

In ZDT3 function, all of above algorithms, except MOEAD, acquire relatively fast convergence at the early stage of evolutionary process. That is, all of the algorithms besides MOEAD tend to a stable value at the later stage of evolutionary process. MOEAD was designed for many-objective problem with more than 3 objectives. It is not suitable for the benchmark functions with 2 objectives, such as ZDT3 function. On the whole, the result of MOEAD is relatively poor on ZDT3 function.NSGA2, eNSGA2 and IBEA get the similar performance as each other. Our IBEA-SVM performs better than the typical 3 algorithms because of the proposed preselection strategy. Our IBEA-SVM is also completive with eMOEA which gets the fast convergence speed on ZDT3. Therefore, IBEA-SVM achieve the best group position in ZDT3 function.

In DTLZ3_2 function, the algorithms perform can be classified as follows, IBEA, NSGA2 and eNSGA2 appear a wide range of fluctuations at the early stage of evolutionary process. Their final result are relatively poor. The proposed IBEA-SVM, MOEAD and eMOEA get the similar performance as each other. This group algorithms get the better performance than above 3 algorithms. They experiences a relatively stable convergence at the early stage and remains a low final IGD result at the later stage.Therefore, IBEA-SVM achieve the best group position in DTLZ3_2 function.

In summary, IBEA-SVM outperforms the classical IBEA algorithms and also achieve the best group position when compared with other state-of-the-art MOE algorithm.

§5 Conclusions

A novel IBEA-SVM algorithm is presented with following contributions. The first one is a IBEA-SVM framework to integrate IBEA and SVM. The proposed integration approach includes procedures of initialization, model training, reproduction, preselection, selection, update of train data and output results.

The second contribution is to design a high-performance classifier to acquire an accurate preselection result, which includes strategies. That is, online training strategy ensures SVM model to acquire the real-time evolutionary trend, asynchronous training strategy ensures SVM to focus on training at early time and predicting at later time and filtering strategy to delete a repeated train data at later stage to improve the SVM training efficiency.

The third contribution is that, after experiments, IBEA-SVM outperforms the classical IBEA algorithms and also achieves the best group position when compared with other state-of-the-art MOE algorithm.

In future research, we try to explore following directions but not limited. Firstly, we will try to accelerate the proposed IBEA-SVM algorithma with multi-core CPU/many-core GPU computing techniques [31] [45] [47]. Secondly, we will continue to enhance the IBEA-SVM with new evolutionary mechanism [33] [34] [46], new evaluation methods, new learning methods [20] [35] [37] and different combinations of them. Thirdly, we will extend and apply the IBEA-SVM in other scientific and engineering applications in CAD/Graphics/Images/Video[6] [19] [21] [24] [25] [26] [28] [36] [38] [40] [44].

References

[1] Bader J, Zitzler E. *HypE: An algorithm for fast hypervolume-based many-objective optimization*, Evolutionary Computation, 2011, 19(1): 45-76.

[2] Bandaru S, Ng A, Deb K. *On the performance of classification algorithms for learning pareto-dominance relations*, Evolutionary Computation (CEC), 2014, pp 1139-1146, IEEE.

[3] Barnhart C, Johnson E, Jin Y. *Branch-and-Price: Column Generation for Solving Huge Integer Programs*, Operations Research, 1998, 46(3): 316-329.

[4] Brockhoff D, Zitzler E. *Improving hypervolume-based multiobjective evolutionary algorithms by using objective reduction methods*, Evolutionary Computation (CEC), 2007, pp 2086-2093, IEEE.

[5] Chang C, Lin C. *LIBSVM: a library for support vector machines*, ACM Transactions on Intelligent Systems and Technology (TIST), 2(3), 27.

[6] Chen Y, He F, Wu Y, et al. *A local start search algorithm to compute exact Hausdorff Distance for arbitrary point sets*, Pattern Recognition, 2017, 67: 139-148.

[7] Chen X, He F, Yu H. *A Matting Method Based on Full Feature Coverage*, Multimedia Tools and Applications, DOI:10.1007/s11042-018-6690-1.

[8] Coello C, Lamont G, Veldhuizen D. *Evolutionary algorithms for solving multi-objective problems*, 2007, New York: Springer.

[9] Coello C. *Twenty Years of Evolutionary Multi-Objective Optimization: A Historical View of the Field*, IEEE Computational Intelligence Magazine, 2006, 1(1): 28-36.

[10] Corne D W, Jerram N R, Knowles J D, et al. *PESA-II: region-based selection in evolutionary multiobjective optimization*, Conference on Genetic and Evolutionary Computation, Morgan Kaufmann Publishers Inc, 2001: 283-290.

[11] Das S, Suganthan P. *Differential evolution: A survey of the state-of-the-art*, IEEE transactions on evolutionary computation, 2011, 15(1), 4-31.

[12] Deb K, Thiele L, Laumanns M, et al. *Scalable multi-objective optimization test problems*, Evolutionary Computation, 2002, CEC'02, Proceedings of the 2002 Congress on (Vol 1, pp 825-830), IEEE.

[13] Deb K, Pratap A, Agarwal S, et al. *A fast and elitist multiobjective genetic algorithm: NSGA-II*, IEEE Transactions on Evolutionary Computation, 2002, 6(2): 182-197.

[14] Fan Z, Hu K, Li F. *Multi-objective evolutionary algorithms embedded with machine learning A survey*, Evolutionary Computation (CEC), 2016, pp 1262-1266, IEEE.

[15] Gui W, Zhang H. *Asymptotic properties and expectation-maximization algorithm for maximum likelihood estimates of the parameters from Weibull-Logarithmic model*, Applied Mathematics-A Journal of Chinese Universities, 2016, 31(4): 425-438.

[16] Kim W, Xiong S, Liang Z. *Effect of Loading Symbol of Online Video on Perception of Waiting Time*, International Journal of Human-Computer Interaction, 2017(5).

[17] Konak A, Coit W. *Multi-objective optimization using genetic algorithms: A tutorial*, Reliability Engineering System Safety, 2006, 91(9): 992-1007.

[18] Laumanns M. *SPEA2 : Improving the Strength Pareto Evolutionary Algorithm*, Technical Report Gloriastrasse, 2001.

[19] Li K, He F, Yu H, et al. *A parallel and robust object tracking approach synthesizing adaptive bayesian learning and improved incremental subspace learning*, Frontiers Comput Sci, DOI: 10.1007/s11704-018-6442-4.

[20] Li K, He F Z, Yu H P. *Robust Visual Tracking based on convolutional features with illumination and occlusion handing*, Journal of Computer Science and Technology, 2018, 33(1): 223-236.

[21] Li K, He F, Yu H, et al. *A correlative classifiers approach based on particle filter and sample set for tracking occluded target*, Applied Mathematics-A Journal of Chinese Universities, 2017, 32(3): 294-312.

[22] Li W, Mcmahon C. *A Simulated Annealing-based Optimization Approach for Integrated Process Planning and Scheduling*, International Journal of Computer Integrated Manufacturing, 2007, 20(1): 80-95.

[23] Lin X, Zhang Q, Kwongs S. *A decomposition based multiobjective evolutionary algorithm with classification*, IEEE Congress on Evolutionary Computation, 2016, pp 3292-3299, IEEE.

[24] Lv X, He F, Cai W, et al. *A string-wise CRDT algorithm for smart and large-scale collaborative editing systems*, Advanced Engineering Informatics, 2017, 33: 397-409.

[25] Lv X, He F, Cai W, et al. *Supporting selective undo of string-wise operations for collaborative editing systems*, Future Generation Computer Systems, 2018, 82: 41-62.

[26] Lv X, He F, Cheng Y et al. *A novel CRDT-based synchronization method for real-time collaborative CAD Systems*, Advanced Engineering Informatics, 2018, 38: 381-391.

[27] Ni B, He F, Pan Y, et al. *Using shapes correlation for active contour segmentation of uterine fibroid ultrasound images in computer-aided therapy*, Applied Mathematics-A Journal of Chinese Universities, 2016, 31(1): 37-52.

[28] Sun J, He F, Chen Y, Chen X et al. *A multiple template approach for robust tracking of fast motion target*, Applied Mathematics-A Journal of Chinese Universities, 2016, 31(2): 177-197.

[29] Tao Q, Zhang M. *Mathematical theory of signal analysis vs complex analysis method of harmonic analysis*, Applied Mathematics-A Journal of Chinese Universities, 2013, 28(4): 505-530.

[30] Trivedi A, Srinivasan D, Sanyal K, Ghosh A. *A Survey of Multiobjective Evolutionary Algorithms based on Decomposition*, IEEE Transactions on Evolutionary Computation, 2017, 21(3): 440-462.

[31] Van L, Melab N, Talbi E. *Gpu computing for parallel local search metaheuristic algorithms*, Journal of Supercomputing, 2016, 72(6): 2394-2416.

[32] Wang S, Lu X, Li X, Li W. *A systematic approach of process planning and scheduling optimization for sustainable machining*, Journal of Cleaner Production, 2015, 87(1): 914-929.

[33] Wu Y, He F, Zhang D, et al. *Service-oriented feature-based data exchange for cloud-based design and manufacturing*, IEEE Transactions on Services Computing, 2018,11: 341-353.

[34] Xiong S, Zhao J, et al. *A computer-aided design system for foot-feature-based shoe last customization*, International Journal of Advanced Manufacturing Technology, 2010, 46(1-4): 11-19.

[35] Yan X, He F, Chen Y. *A novel hardware/software partitioning method based on position disturbed particle swarm optimization with invasive weed optimization*, Journal of Computer Science and Technology, 2017, 32(2): 340-355.

[36] Yan X, He F, Hou N, et al. *An efficient particle swarm optimization for large-scale hardware/software co-design system*, International Journal of Cooperative Information Systems, 2018, 27(1): 1741001.

[37] Yu H, He F, Pan Y, et al. *A novel region-based active-contour model via local patch similarity measure for image segmentation*, Multimedia Tools and Appliations, 2018, 77(18), 24097-24119.

[38] Yu H, He F, Pan Y. *A Novel Segmentation Model for Medical Images with Intensity Inhomogeneity Based on Adaptive Perturbation*, Multimedia Tools and Applications, DOI: 10.1007/s11042-018-6735-5.

[39] Zhang D J, He F Z, Han S H, et al. *Quantitative optimization of interoperability during feature-based data exchange*, Integrated Computer-Aided Engineering, 2016, 23(1): 31-50.

[40] Zhang D, He F, Han S, et al. *An efficient approach to directly compute the exact Hausdorff distance for 3D point sets*, Integrated Computer-Aided Engineering, 2017, 24(3): 261-277.

[41] Zhang J, Zhou A, Zhang G. *A classification and Pareto domination based multiobjective evolutionary algorithm*, Evolutionary Computation, 2015, pp 2883-2890, IEEE.

[42] Zhang Q, Zhou A, Zhao S. *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, 2008, 264.

[43] Zhang Q, Li H. *MOEA/D:A Multiobjective Evolutionary Algorithm Based on Decomposition*, IEEE transactions on evolutionary computation, 2007, 11(6): 712-731.

[44] Zhang S, He F, Ren W, Yao J. *Joint learning of image detail and transmission map for single image dehazing*, The Visual Computer, DOI 10.1007/s00371-018-1612-9.

[45] Zhou Y, He F, Qiu Y. *Optimization of parallel iterated local search algorithms on graphics processing unit*, Journal of Supercomputing, 2016, 72(6): 2394-2416.

[46] Zhou Y, He F, Qiu Y. *Dynamic Strategy based Parallel Ant Colony Optimization on GPUs for TSPs*, Science China Information Sciences, 2017, 60(6): 068-102.

[47] Zhou Y, He F, Qiu Y. *Parallel ant colony optimization on multi-core simd cpus*, Future Generation Computer Systems, 2018, 79(2): 473-487.

[48] Zhu H. *Avoiding Conflicts by Group Role Assignment*, IEEE Trans on Systems, Man, and Cybernetics: Systems, 2016, 46(4): 535-547.

[49] Zhu H. *Role-Based Collaboration and the E-CARGO: Revisiting the Developments of the Last Decade*, IEEE Systems, Man, and Cybernetics Magazine, 2015, 1(3): 27-35.

[50] Zhu H, Zhou M. *Role-Based Collaboration and its Kernel Mechanisms*, IEEE Trans on Systems, Man, and Cybernetics, Part C: Applications and Reviews, 2006, 36(4): 578-589.

[51] Zitzler E, Kunzli S. *Indicator-Based Selection in Multiobjective Search*, Lecture Notes in Computer Science, 2004, 3242: 832-842.

[52] Zitzler E, Thiele L. *Multiobjective optimization using evolutionary algorithms A comparative case study*, Int Conf Parallel Problem Solving from Nature (PPSN-V), 1998, 1498(3): 292-301.

[53] Zitzler E, Thiele L, Laumanns M. *Performance assessment of multiobjective optimizers: An analysis and review*, IEEE Transactions on Evolutionary Computation, 2003, 7(2): 117-132.

[54] Zitzler E, Knzli S. *Indicator-Based Selection in Multiobjective Search*, 2004 Parallel Problem Solving from Nature.

[1] School of Computer Science and Technology, Wuhan University, Wuhan, China

[2] State Key Laboratory of Digital Manufacturing Equipment and Technology,Huazhong University, Wuhan, 430074, China

Email: fzhe@whu.edu.cn