

# Efficient Identification of the Pareto Optimal Set

Ingrida Steponavičė<sup>1</sup>, Rob J. Hyndman<sup>2</sup>, Kate Smith-Miles<sup>1</sup>  
and Laura Villanova<sup>3</sup>

<sup>1</sup> School of Mathematical Sciences, Monash University, Clayton, Australia

<sup>2</sup> Department of Econometrics & Business Statistics, Monash University, Clayton,  
Australia

<sup>3</sup> Ceramic Fuel Cells Limited, Noble Park, Australia

**Abstract.** In this paper, we focus on expensive multiobjective optimization problems and propose a method to predict an approximation of the Pareto optimal set using classification of sampled decision vectors as dominated or nondominated. The performance of our method, called EPIC, is demonstrated on a set of benchmark problems used in the multiobjective optimization literature and compared with state-of-the-art methods, ParEGO and PAL. The initial results are promising and encourage further research in this direction.

**Keywords:** multiobjective optimization, classification, expensive black-box function

## 1 Introduction

Many real-world optimization applications in engineering involve problems where analytical expression of the objective function is unavailable. Such problems usually require either an underlying numerical model or expensive experiments to be conducted. In an optimization setting, where objective functions are evaluated repeatedly, evaluation demands may result in unaffordably high cost for obtaining solutions. Therefore, the number of function evaluations is limited by available resources. Consequently, the solution of such global optimization problems is challenging because many global optimization methods require a large number of function evaluations.

This task becomes even more difficult in the case of multiple conflicting objectives, where there is no single optimal solution optimizing all objective functions simultaneously. Rather there exists a set of solutions representing the best possible trade-offs among the objectives — the so-called Pareto optimal solutions forming a Pareto optimal set. Unreasonably high evaluation costs could also prevent designers from comprehensively exploring the decision space and learning about possible trade-offs. In these cases, it is essential to find reliable and efficient methods for estimating the Pareto optimal set within a limited number of objective function evaluations.

Recently, researchers have developed methods to solve expensive problems by exploiting knowledge acquired during the solution process [1]. Knowledge of past

evaluations can be used to build an empirical model, called a surrogate model, that approximates the objective function. This approximation can then be used to predict promising new solutions at a smaller evaluation cost than that of the original problem [2]. Commonly used surrogate models include Gaussian process (or Kriging) models [3, 4], polynomial response surface models [5], radial basis functions [6], support vector regression [7], and local polynomial [8].

One of the state-of-art methods for expensive multiobjective optimization problems, named ParEGO, was developed by Knowles [9]. It is essentially a multiobjective translation of the efficient global optimization (EGO) method [1], where multiple objectives are converted to a single objective using a scalarization function with different parameter values at each step.

The Pareto active learning (PAL) method for predicting the Pareto optimal set at low cost has been proposed in [10]. Like ParEGO, it employs a Gaussian process (GP) model to predict objective function values. PAL classifies all sampled decision vectors as Pareto optimal or not based on the predicted objective function values. The classification accuracy is controlled by a parameter defined by the user which enables a trade-off between evaluation cost and predictive accuracy.

The main disadvantage of employing a GP is that model construction can be a very time-consuming process [11], where the time increases with the number of evaluated vectors used to model the GP. To overcome this issue, ParEGO uses a subset of the evaluated vectors to build the GP model, thus attempting to balance model accuracy and computation time. Moreover, using a GP becomes increasingly problematic in high dimensional spaces [7], so these methods do not scale well as the dimension of the problem increases.

In this paper, we propose an efficient Pareto iterative classification (EPIC) method which approximates the Pareto optimal set with a limited number of objective function evaluations. At the core of the proposed method is the classification of decision vectors to be either dominated or nondominated without requiring any knowledge about the objective space. The method iteratively classifies sampled decision vectors to one of the two classes until a stopping criterion is met. It returns the set of decision vectors predicted to be nondominated. A major advantage of this method is that it does not use any statistical model of the objective function, such as GP, and so it involves more modest computational requirements, and scales easily to handle high dimensional spaces.

The paper is organized as follows. Section 2 introduces the main concepts involved in multiobjective optimization. The EPIC method is described in Section 3. In Section 4, we outline an experiment setup, demonstrate the performance of EPIC on benchmark problems, and compare our experimental results with that of the ParEGO and PAL methods. Section 5 draws some conclusions and briefly discusses some future research directions.

## 2 Background

A multiobjective optimization problem has the form [12]

$$\begin{aligned} \min \mathbf{f}(\mathbf{x}) &= (f_1(\mathbf{x}), \dots, f_k(\mathbf{x}))^T \\ \text{subject to } \mathbf{x} &\in S, \end{aligned} \tag{1}$$

where  $S \subset \mathbb{R}^n$  is the feasible set and  $f_i : \mathbb{R}^n \rightarrow \mathbb{R}$ ,  $i = 1, \dots, k$  ( $k \geq 2$ ), are objective functions that are to be minimized simultaneously. All objective functions are represented by the vector-valued function  $\mathbf{f} : S \rightarrow \mathbb{R}^k$ . A vector  $\mathbf{x} \in S$  is called a *decision vector* and a vector  $\mathbf{z} = \mathbf{f}(\mathbf{x}) \in \mathbb{R}^k$  an *objective vector*.

In multiobjective optimization, the objective functions  $f_1, \dots, f_k$  in (1) are typically conflicting. In that case, there does not exist a decision vector  $\bar{\mathbf{x}} \in S$  such that  $\bar{\mathbf{x}}$  minimizes  $f_i$  in  $S$  for all  $i = 1, \dots, k$ , but there exists a number (possibly infinite) of Pareto optimal solutions. In mathematical terms, a decision vector  $\bar{\mathbf{x}} \in S$  and its image  $\bar{\mathbf{z}} = \mathbf{f}(\bar{\mathbf{x}})$  are said to be *Pareto optimal* or *nondominated* if there does not exist a decision vector  $\mathbf{x} \in S$  such that  $f_i(\mathbf{x}) \leq f_i(\bar{\mathbf{x}})$  for all  $i = 1, \dots, k$  and  $f_j(\mathbf{x}) < f_j(\bar{\mathbf{x}})$  for some  $j = 1, \dots, k$ . If such a decision vector  $\mathbf{x} \in S$  does exist,  $\bar{\mathbf{x}}$  and  $\bar{\mathbf{z}}$  are said to be *dominated* by and its image  $\mathbf{z} = \mathbf{f}(\mathbf{x})$ , respectively.

Sometimes in the literature, Pareto optimal and nondominated solutions are regarded as synonyms. However, in this paper, nondominated solutions refer to solutions which are not *dominated* by any other solution in the set of evaluated solutions. If an objective vector  $\mathbf{z}_1 = \mathbf{f}(\mathbf{x})$  does not dominate an objective vector  $\mathbf{z}_2 = \mathbf{f}(\bar{\mathbf{x}})$ , this does not imply that  $\mathbf{z}_2$  dominates  $\mathbf{z}_1$  (e.g., they can be both nondominated in a given set). Moreover, this does not imply Pareto optimality as nondominance is defined subject to the set of available objective vectors.

## 3 The EPIC method

The EPIC method is designed for expensive multiobjective optimization problems where the number of possible objective function evaluations is limited due to time, financial, or other costs. The main idea behind the proposed method is the classification of a set of sampled decision vectors into two classes: dominated and nondominated. The method comprises five steps: (i) evaluation of an initial set; (ii) training a classifier; (iii) predicting labels for unevaluated decision vectors; (iv) checking stopping conditions; and (v) selection of a decision vector to evaluate next. Steps (ii)–(v) are repeated until stopping conditions are satisfied. The pseudocode of EPIC is given in Algorithm 1.

Given a set of decision vectors  $S$  which can be generated using any sampling technique (e.g., Hammersley sequence or Latin hypercube sampling), the EPIC method selects and evaluates expensive objective functions for a small number of decision vectors called an initial set  $X_{\text{init}}$ . Then the evaluated decision vectors, which comprise the set of all evaluated decision vectors  $E$ , are checked for nondominance and divided into dominated and nondominated classes. The decision

---

**Algorithm 1** EPIC pseudocode

---

**Input:** sampled decision space  $S$ ,  $p_{\text{next}}$ ,  $p_{\text{nond}}$ **Output:** predicted Pareto optimal set  $P$ 

```

 $E = X_{\text{init}}, X_U = S \setminus E, i = 0$ 
while  $i \leq n$  do
   $i = i + 1, P = \emptyset$ 
  obtain labels for all  $\mathbf{x} \in E$ 
  train a classifier
  calculate the probability  $p$  of each  $\mathbf{x} \in X_U$ 
  for all  $\mathbf{x} \in X_U$  do
    if  $p \geq p_{\text{nond}}$  then  $P = P \cup \{x\}$ 
    end if
  end for
   $\mathbf{x}_e = \arg \min_{\mathbf{x} \in X_U} |p - p_{\text{next}}|$ 
   $E = E \cup \{\mathbf{x}_e\}, X_U = X_U \setminus \{\mathbf{x}_e\}$ 
end while

```

---

vectors and their labels “nondominated” and “dominated” are used to train a classifier, for example, support vector machines or a Naive Bayes classifier. After a classifier is trained, unevaluated decision vectors comprising a set  $X_U$  are given as an input to the classifier to predict their labels. The classifier provides probability  $p$  for each decision vector  $\mathbf{x}$  that it belongs to the nondominated class. If the probability of nondominance  $p$  is not lower than a predefined probability  $p_{\text{nond}}$ , the decision vector is included in a predicted Pareto optimal set  $P$ . Then a new decision vector  $\mathbf{x}_e$  is selected for evaluation. The implemented selection strategy in this paper is simply to select a decision vector  $\mathbf{x}_e$  whose probability of nondominance  $p$  is closest to a predefined value  $p_{\text{next}}$ . Next, the evaluated set  $E$  and its labels are updated and used to retrain the classifier. The method continues until the number of evaluations reaches a predefined limit  $n$ .

If the sampling of the design space is sufficiently dense, and the number of evaluations  $n$  is sufficiently large, then EPIC should provide a good approximation of the Pareto optimal set. The efficiency of EPIC is determined by the classification quality and the strategy for determining which decision vector should be evaluated next.

It is clear that the next vector to be evaluated should be selected based on the maximal information gain or uncertainty reduction which would help to improve classification. We assume that evaluating a decision vector near to the boundary that separates the two classes is most likely to provide more information and improve classifier performance. An obvious choice is to set  $p_{\text{next}} = 0.5$ . However, further thought suggests that alternative values of  $p_{\text{next}}$  may be preferred. The initial set used to train the classifier is a small subset of the sampled decision space and the vectors labelled “nondominated” might be mis-classified since they are assigned with respect to the evaluated vectors. Therefore it may be better to first evaluate decision vectors that are most likely nondominated (i.e.,  $p_{\text{next}} > 0.5$ ) in order to get a better representation of the nondominated class. Later, it

may be better to concentrate on uncertainty related to the boundary separating the two classes. Alternatively, selecting several decision vectors based on different  $p_{\text{next}}$  values and evaluating them simultaneously could improve classification resulting in better overall method performance.

One more parameter that has to be chosen is the probability  $p_{\text{nond}}$ , used to determine which decision vectors will be included in the predicted Pareto optimal set. The prediction of the Pareto optimal set is made at each iteration, with the predicted set obtained in the previous iteration replaced by the newly predicted set. A higher value of  $p_{\text{nond}}$  will result in a smaller set of points classified to be nondominated. Again, the value of this parameter needs to be selected carefully in order to control the efficiency of the algorithm in determining the Pareto optimal set.

We leave the optimal selection of  $p_{\text{next}}$  and  $p_{\text{nond}}$  to future research. In this paper, we have selected  $p_{\text{nond}} = 0.6$ , and have considered several values of  $p_{\text{next}} = \{0.6, 0.7, 0.8, 0.9\}$ .

The main advantages of the EPIC method are: i) simplicity to implement, ii) computational speed; iii) multiple decision vectors can be selected at each iteration; and iv) has no limitations on high dimensional problems.

## 4 Experimental Results

### 4.1 Experimental setup

To assess the performance of EPIC, we compare it to PAL and ParEGO. For that we need to have some performance measures. One of the measures is based on a hypervolume (HV) metric [13] that describes the spread of the solutions over the Pareto optimal set in the objective space as well as the closeness of the solutions to it. Moreover, HV is the only measure that reflects the Pareto dominance [14]. That is, if one set entirely dominates another, the HV of the former will be greater. As PAL and EPIC are based on classification, the quality of prediction is also measured by the percentage of correctly classified decision vectors. We calculated other metrics as well such as set coverage. However, they are not so informative and we do not include them in this work.

### 4.2 Methods comparison

EPIC and PAL differ from ParEGO as they do not generate new decision vectors but rather select them from the sampled decision space. We used Latin hypercube sampling to sample the decision space for both EPIC and PAL. The sampling size selected for all problems is 500 decision vectors. ParEGO and EPIC were allowed to evaluate 200 or 250 decision vectors, while PAL was running until all the decision vectors were assigned to one of the classes. As the method performance depends on the initial set, we ran the methods with 100 different initial sets and calculated the average values of the performance metrics. All the methods were run with the same initial sets consisting of  $11d - 1$  decision vectors, where  $d$  is the dimension of the decision space, as proposed in [1].

The performance of the methods was measured at every iteration to assess the progress obtained after each decision vector evaluation. We calculated the HV metric of evaluated decision vectors for all the methods. For ease of comparison, we considered the ratio between the HV obtained by each method and the HV of the true Pareto optimal set. It should be noted the HV value calculated using the evaluated vectors does not decrease with an increasing number of evaluations. For the average HV metric calculation, when for some runs the PAL method terminated earlier and did not use the maximum number of iterations, we used the same nondominated set evaluated at the last iteration for the rest of iterations.

The methods were tested on the following set of standard benchmark problems in multiobjective optimization with different features:

- OKA2** [15]. This problem has two objective functions and three decision variables. Its true Pareto optimal set in the objective space is a spiral shaped curve and the density of the Pareto optimal solutions in the objective space is low. (The reference point is  $R = (6.00, 6.0483)$ .)
- Kursawe** [16]. This problem has two objective functions and a scalable number of decision variables. In our experiment, three decision variables were used. Its Pareto optimal set in the decision space is disconnected and symmetric, and disconnected and concave in the objective space. (The reference point is  $R = (-3.8623, 25.5735)$ .)
- ZDT3** [17]. This has two objective functions and three decision variables. The Pareto optimal set in the objective space consists of several noncontiguous convex parts. However, there is no discontinuity in the decision space. (The reference point  $R = (2.0000, 2.6206)$ .)
- Viennet** [18]. This consists of three objective functions and two decision variables. This problem has not been solved with the PAL algorithm as we used an implementation provided by the authors which is suitable only for problems with two objective functions. (The reference point is  $R = (9.249, 62.68, 1.1964)$ .)
- DTLZ4** [19]. This problem is scalable and has  $M$  objective functions and  $k + M - 1$  of decision variables, where  $k = 10$  as recommended by the authors. We solved this problem consisting of 5, 6 and 7 objectives and, respectively, 14, 15 and 16 decision variables. (The reference points for the problem with 5, 6 and 7 objectives is  $R = (3.9324, 3.2452, 3.4945, 3.4114, 3.3022)$ ,  $R = (3.9943, 3.2319, 3.3666, 3.1851, 3.3236, 3.2196)$  and  $R = (3.7703, 3.3593, 3.3192, 3.3825, 3.4326, 3.2446, 3.3209)$ , respectively.)

To classify decision vectors as dominated and nondominated we applied a support vector machine (SVM) [20]. The basic idea of SVM classifiers is to choose the hyperplane that has the maximum distance between itself and the nearest example of each class [20, 21]. SVMs are computationally deficient classifiers and can deal with both linear and nonlinear as well as separable and nonseparable problems. We used SVM with a radial basis function kernel allowing to capture nonlinear relation between class labels and features by mapping data to a higher dimensional space. The drawback of using SVMs is that they

do not directly produce probability estimates. However, these can be calculated using different strategies, for example, Platt’s method [22] or isotonic regression [23]. For EPIC implementation we used LIBSVM, a library for SVMs [24], which provides probability estimates.

We experimented with different  $p_{\text{next}}$  values: 0.6, 0.7, 0.8, and 0.9. However, we did not find any appreciable performance differences as well as no single  $p_{\text{next}}$  value provided the best results for all considered problems.

### 4.3 Methods comparison

All of the experiments were repeated 100 times and the average metrics are presented in Figures 1–7. Looking at the plots of correct classification, it can be noted that at the very first iterations, EPIC classification is not as good as that of PAL, but its classification improves fast and within 20 iterations EPIC outperforms PAL for all three problems. This can be explained by the fact that when the training set (consisting of evaluated vectors) increases, the classifier learns about the boundary separating the two classes.

The HV metric plots show that the performance of the methods is problem dependent. For example, we cannot distinguish any method to be best when considering OKA2 and ZDT3 problems as the HV curves overlap. However, when solving DTLZ4 problem with 5, 6 and 7 objectives, the HV measure clearly indicates that EPIC outperforms ParEGO.

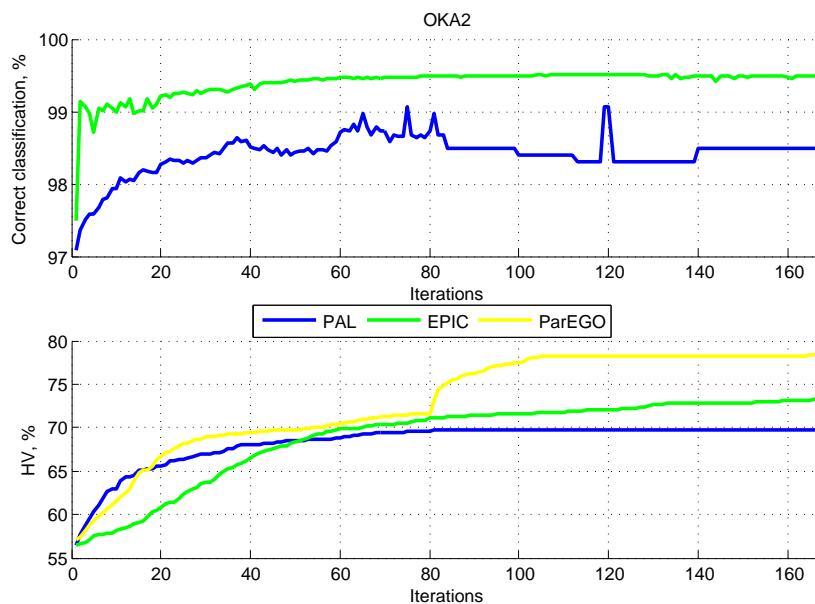
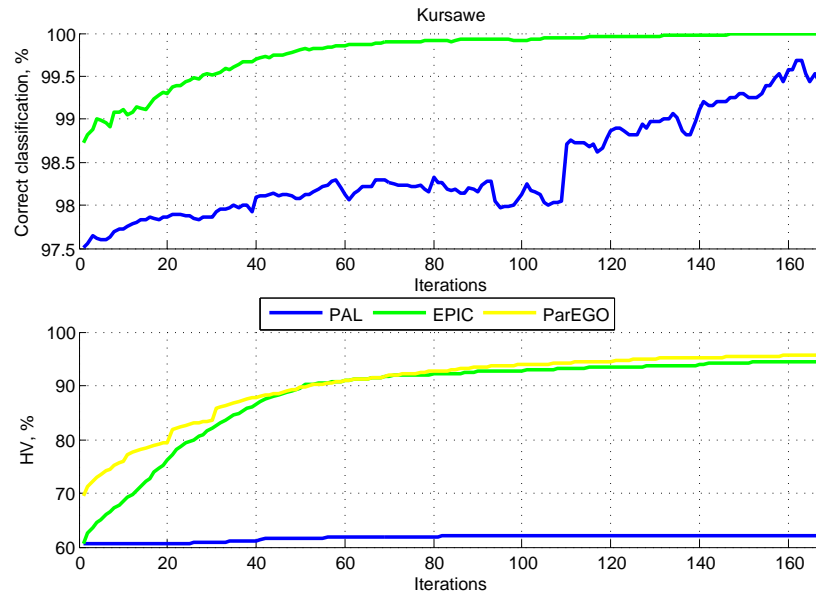
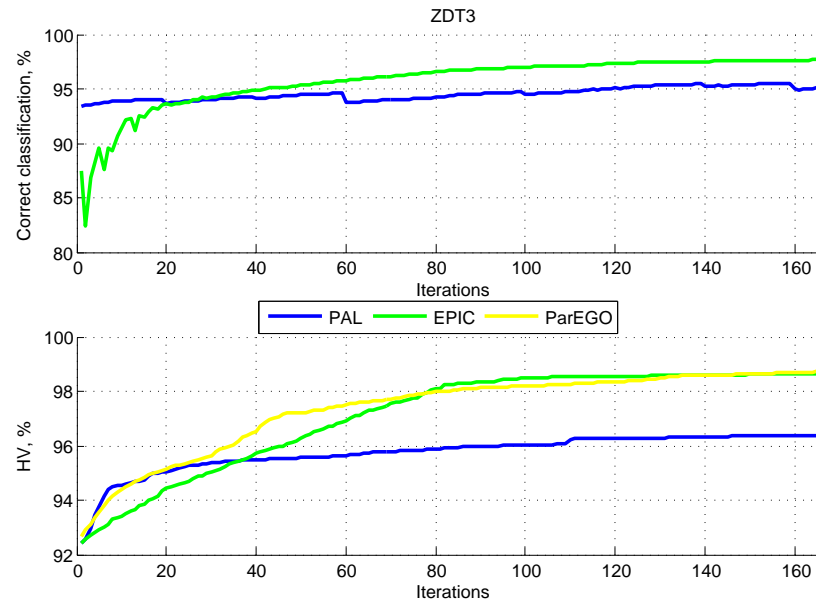


Fig. 1. Comparative performance on the OKA2 problem

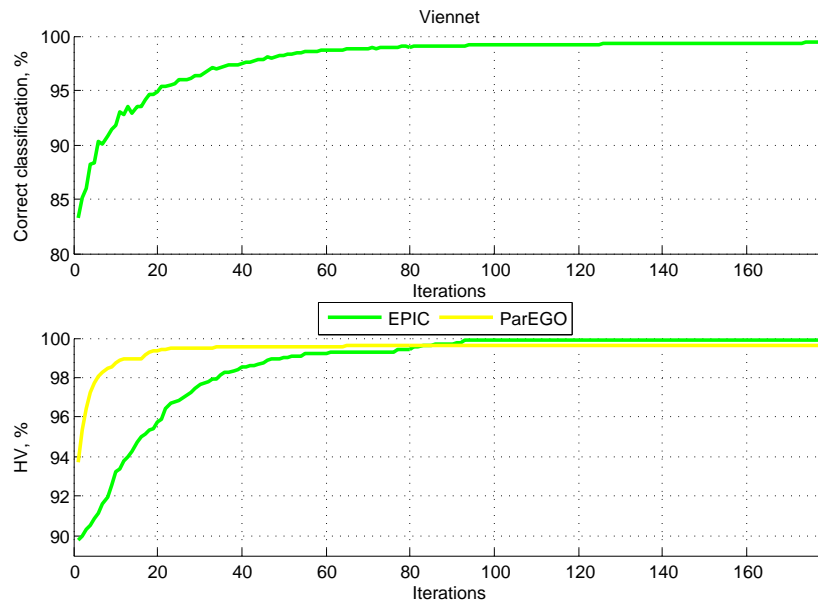


**Fig. 2.** Comparative performance on the Kursawe problem

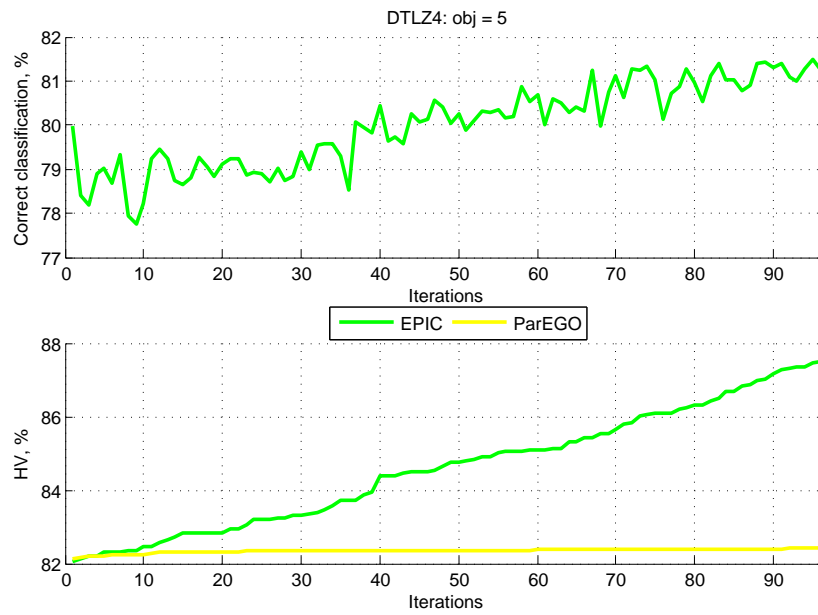


**Fig. 3.** Comparative performance on the ZDT3 problem

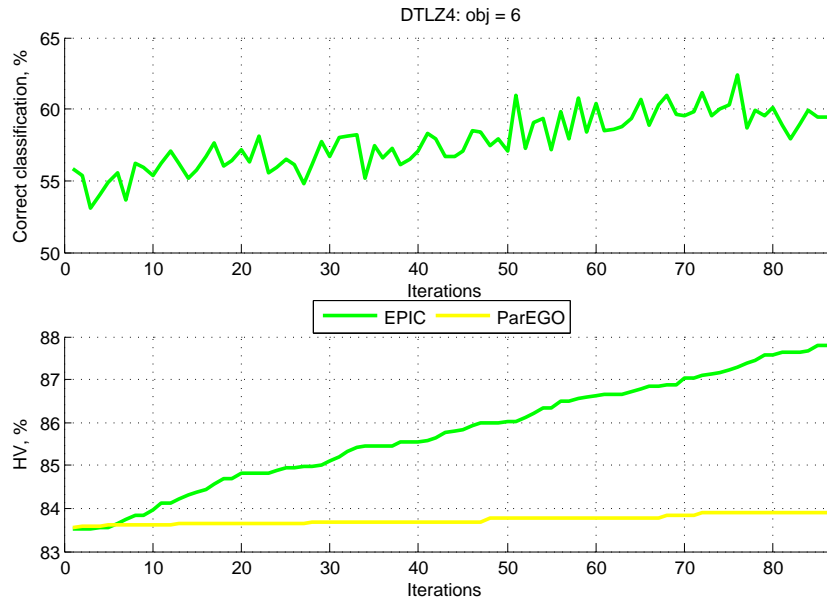




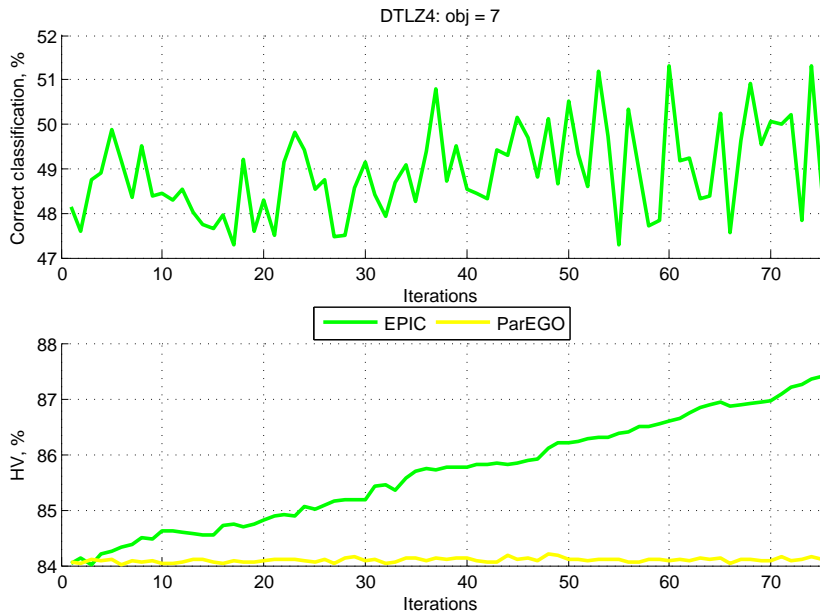
**Fig. 4.** Comparative performance on the Viennet problem



**Fig. 5.** Comparative performance on the DTLZ4 problem with 5 objectives



**Fig. 6.** Comparative performance on the DTLZ4 problem with 6 objectives



**Fig. 7.** Comparative performance on the DTLZ4 problem with 7 objectives. (Note: here, HV is calculated using an approximation method not exactly.)

## 5 Conclusions and Future Works

### 5.1 Conclusions

Multiobjective black-box optimization problems appear in many engineering applications whose function evaluations are very expensive with respect to time or money and therefore are limited by available or affordable resources. To deal with such problems, we propose a simple method called EPIC to approximate the Pareto optimal set using a limited number of function evaluations. The key idea is to use a classifier to classify decision vectors into two classes — nondominated and dominated — and then predict the Pareto optimal set using only information about the decision space and without evaluating the rest of sampled decision vectors. The initial experimental results demonstrate that our new method, EPIC, is competitive with existing methods PAL and ParEGO, and even outperforms it on some problems. The results of all methods seem quite dependent on the characteristics of the problem though, and this dependence is worthy of further investigation to understand the strengths and weaknesses of the different approaches.

### 5.2 Future works

Possible future research includes developing a strategy for selecting more than one decision vector at each iteration for evaluation. It might consider some clustering approach to choose more diverse vectors in the objective space in order to ensure a better approximation of the Pareto optimal set in the sense of uniform distribution of the vectors in the objective space. Also, we need to explore the influence of the probability values  $p_{\text{next}}$  and  $p_{\text{nond}}$  used to select and classify decision vectors, and to investigate how these can be chosen in an automatic way based on knowledge about the problem at hand.

## References

1. D. R. Jones, M. Schonlau, W. J. Welch, Efficient global optimization of expensive black-box functions, *Journal of Global Optimization* 13 (4) (1998) 455–492.
2. L. Santana-Quintero, A. Montao, C. C. Coello, A review of techniques for handling expensive functions in evolutionary multi-objective optimization, in: Y. Tenne, C.-K. Goh (Eds.), *Computational Intelligence in Expensive Optimization Problems*, Vol. 2, Springer Berlin Heidelberg, 2010, pp. 29–59.
3. J. Sacks, W. J. Welch, T. J. Mitchell, H. P. Wynn, Design and analysis of computer experiments, *Statistical Science* 4 (4) (1989) 409–423.
4. J. D. Martin, T. W. Simpson, Use of kriging models to approximate deterministic computer models, *AIAA journal* 43 (4) (2005) 853–863.
5. G. E. Box, N. R. Draper, *Empirical model-building and response surfaces*, John Wiley & Sons, 1987.
6. H. Fang, M. F. Horstemeyer, Global response approximation with radial basis functions, *Engineering Optimization* 38 (4) (2006) 407–424.

7. A. I. Forrester, A. J. Keane, Recent advances in surrogate-based optimization, *Progress in Aerospace Sciences* 45 (1–3) (2009) 50–79.
8. P. Lancaster, K. Salkauskas, Surfaces generated by moving least squares methods, *Mathematics of computation* 37 (155) (1981) 141–158.
9. J. Knowles, Parego: A hybrid algorithm with on-line landscape approximation for expensive multiobjective optimization problems, *IEEE Transactions on Evolutionary Computation* 10 (1) (2006) 50–66.
10. M. Zuluaga, A. Krause, G. Sergent, M. Püschel, Active learning for multi-objective optimization, in: *Proceedings of the 30th International Conference on Machine Learning*, 2013.
11. R. Jin, W. Chen, T. Simpson, Comparative studies of metamodelling techniques under multiple modelling criteria, *Structural and Multidisciplinary Optimization* 23 (1) (2001) 1–13.
12. A. Chinchuluun, P. M. Pardalos, A. Migdalas, L. Pitsoulis, *Pareto Optimality, Game Theory and Equilibria*, 2nd Edition, Springer, 2008.
13. E. Zitzler, L. Thiele, Multiobjective optimization using evolutionary algorithms – a comparative case study, in: *Parallel Problem Solving from Nature - PPSN-V*, Springer, 1998, pp. 292–301.
14. C. Azevedo, A. Araujo, Correlation between diversity and hypervolume in evolutionary multiobjective optimization, in: *IEEE Congress on Evolutionary Computation (CEC)*, 2011, pp. 2743–2750.
15. T. Okabe, Y. Jin, M. O. B. Sendhoff, On test functions for evolutionary multi-objective optimization, in: X. Yao, E. Burke, J. Lozano, J. Smith, J. Merelo-Guervs, J. Bullinaria, J. Rowe, P. Tio, A. Kabn, H.-P. Schwefel (Eds.), *Parallel Problem Solving from Nature – PPSN VIII*, Vol. 3242, Springer Berlin Heidelberg, 2011, pp. 792–802.
16. F. Kursawe, A variant of evolution strategies for vector optimization, in: H.-P. Schwefel, R. Manner (Eds.), *Parallel Problem Solving from Nature*, Vol. 496, Springer Berlin Heidelberg, 1991, pp. 193–197.
17. E. Zitzler, K. Deb, L. Thiele, Comparison of multiobjective evolutionary algorithms: Empirical results, *Evolutionary Computation* 8 (2) (2000) 173–195.
18. R. Viennet, C. Fonteix, I. Marc, New multicriteria optimization method based on the use of a diploid genetic algorithm: Example of an industrial problem, in: *Selected Papers from the European conference on Artificial Evolution*, Springer-Verlag, London, UK, 1996, pp. 120–127.
19. K. Deb, L. Thiele, M. Laumanns, E. Zitzler, Scalable multi-objective optimization test problems, in: *Congress on Evolutionary Computation (CEC 2002)*, IEEE Press, 2002, pp. 825–830.
20. V. Vapnik, *The Nature of Statistical Learning Theory*, 2nd Edition, Springer, 1999.
21. K. P. Bennett, E. J. Bredensteiner, Duality and geometry in SVM classifiers, in: *Proceedings of 17th International Conference on Machine Learning*, Morgan Kaufmann, 2000, pp. 57–64.
22. J. C. Platt, Probabilistic outputs for support vector machines and comparisons to regularized likelihood methods, in: *Advances in Large Margin Classifiers*, MIT Press, 1999, pp. 61–74.
23. B. Zadrozny, C. Elkan, Transforming classifier scores into accurate multiclass probability estimates, in: *Proceedings of the International Conference on Knowledge Discovery and Data Mining*, 2002, pp. 694–699.
24. C.-C. Chang, C.-J. Lin, LIBSVM: A library for support vector machines, *ACM Transactions on Intelligent Systems and Technology* 2 (2011) 27:1–27:27.