Binary Nearest Neighbor Classification of Predicting Pareto Dominance in Multi-objective Optimization

Guanqi Guo, Cheng Yin, Taishan Yan, and Wenbin Li

Hunan Institute of Science and Technology, Yueyang, Hunan, 414006, China
{gq.guo,yincheng886337,yantaishan163,wenbin_lii}@163.com

Abstract. A method of predicting Pareto dominance in multi-objective optimization using binary nearest neighbor classification (BNNC) is proposed. It encodes real value feature variables into binary bit strings with the same length. The similarity of two feature variables is directly measured by weighted sum of the binary bits. The analysis shows that when the orders of magnitude for various feature variables differ from each other, the similarity measured by scaled feature variables is able to more uniformly reflect the contribution of each feature variable to Pareto dominance relationship, and BNNC has computational complexity of $O(N)$. Experiments results show that, in addition to remarkably increasing classification accuracy rate, it is more efficient and robust than the canonical nearest neighbor rule and Bayesian classification when used to classify those problems with unbalanced class proportions and feature vectors no less than 2 dimensions.

Keywords: multi-objective optimization, Pareto dominance, pattern classification, binary nearest neighbor classification.

1 Introduction

In recent years, there is an increasing interest in applying evolutionary algorithms to solve multi-objective optimization problems [1], [2]. A number of well-known multi-objective optimization evolutionary algorithms (MOEAs) have been proposed, such as NSGA-II [3], SPEA-II [4], PAES [5], MOPSO [6]. The common characteristic is that these algorithms determine the Pareto dominance of two candidate solutions by computing and comparing their objective vectors, and then iteratively identifies all the non-dominated solutions in the evolutionary populations. As the number of evolutionary generation is increasing, the frequently updated non-dominated set asymptotically converges to the Pareto front.

However, for the complicated structure design optimization problems [7], [8], evaluation of the objective vectors or constraint functions is no more than computing values of simple functions. It is a time consuming procedure often taking hours or days. It is simply referred to the curse of computation cost. For the overwhelming expense evaluating objective vectors, the current popular MOEAs are almost not competent to large scale complicated multi-objective optimization tasks.

To reduce the computation expense evaluating objective vectors, GUO [9] proposed a method of predicting Pareto dominance using pattern recognition. For a

Y. Tan, Y. Shi, and Z. Ji (Eds.): ICSI 2012, Part I, LNCS 7331, pp. 537–545, 2012.

multi-objective optimization problem, the method combines decision vectors of two candidate solutions into a feature vector determining their Pareto dominance. The Pareto dominance relation non-dominated, dominated and incomparable of two candidate solutions is predicted by pattern classification algorithm. No need of modeling fitness estimation and fitness inherit for objective functions of a MO problem, the predicted Pareto dominance can be used in any MO algorithms based on Pareto dominance concept, thus providing an efficient approach for relieving the curse of computation cost in solving complicated MO problems.

Based on the assumption that class-conditional probability follows normal distributions, GUO preliminarily implemented a kind of Bayesian classifiers by minimizing the classification error rate and minimizing the average risk respectively. The classifier obtained acceptable prediction accuracy on SCH [10] function. However, as the complexity of the test problems increases, the shortcoming of Bayesian classifier is evident. In most complex MO problems, the distribution of class-conditional probability is unknown a prior, the assumption of normal distribution consequently causes a number of predicting errors.

This paper proposes a method of using binary nearest neighbor classification to predict Pareto dominance. The method is used in test problems with multi-dimensional objective space and unbalanced class proportion. The rest of the paper is organized as follows. Section 2 gives the Pareto optimality concept briefly. Section 3 proposes and analyzes the binary nearest neighbor classification in detail. Section 4 presents and compares the experimental data. And the conclusions are summarized in Section 5.

2 Outlines of Pareto Optimality

A general multi-objective optimization problem can be described as a vector function **F** that maps a space of n-dimensional decision vectors to a space of m-dimensional objective vectors. Formally:

$$\begin{aligned} &\min\ \mathbf{y} = \mathbf{F}(\mathbf{x}) = (f_1(\mathbf{x}), f_2(\mathbf{x}), \cdots, f_m(\mathbf{x}))\ , \\ &s.t.\ \ \mathbf{x} = (x_1, x_2, \cdots, x_n) \in X\ \ , \\ &\qquad \mathbf{y} = (y_1, y_2, \cdots, y_m) \in Y\ \ . \end{aligned} \tag{1}$$

Where, **x** denotes decision vector, X decision space, **y** objective vector, and Y objective space.

Since the objectives in (1) are often conflicting with each other and incommensurable, it is impossible to optimize all objectives simultaneously. Therefore, there only exists a set of Pareto-optimal solutions.

Pareto Dominance: For every vector $\mathbf{u}=(u_1, u_2,\ldots,u_{\mathrm{m}})\in Y$, $\mathbf{v}=(v_1, v_2,\ldots,v_{\mathrm{m}})\in Y$, iff $\forall i \in \{1,2,...,m\}$: $u_{\mathrm{i}}\leq v_{\mathrm{i}} \wedge \exists j \in \{1,2,...,m\}$: $u_{\mathrm{j}} < v_{\mathrm{j}}$, **u** is called to dominate **v**, denoted as $\mathbf{u} \prec \mathbf{v}$; or **v** dominated by **u**, denoted as $\mathbf{u} \prec \mathbf{v}$. Otherwise, **u** and **v** are incomparable, denoted as **u**~**v**.

Pareto Optimum: $\mathbf{x}\in X$ is referred as to a Pareto optimal solution (namely Pareto non-dominated solution or non-inferior solution), iff $\neg\exists \mathbf{x}' \in X,\ \mathbf{v} = \mathbf{F}(\mathbf{x}') \prec \mathbf{u} = \mathbf{F}(\mathbf{x})$.

Pareto Optimal Set: The set of all Pareto optimum in the decision space X is referred as to Pareto optimal set, and the set of the corresponding objective vectors is referred as to Pareto front or Pareto optimal surface.

3 Binary Nearest Neighbor Classification of Pareto Dominance

3.1 Prediction Model of Pareto Dominance

According to Pareto-optimality concept, Pareto dominance relation of two candidate solutions ($\mathbf{u}$, $\mathbf{v}$) is classified into three categories: $\mathbf{u} \prec \mathbf{v}$, $\mathbf{u} \succ \mathbf{v}$, $\mathbf{u} \sim \mathbf{v}$. Where, for the sake of convenience, $\mathbf{u}$ and $\mathbf{v}$ respectively represent n-dimensional decision vectors of the candidate solutions. Regarding a couple ($\mathbf{u}$, $\mathbf{v}$) as a $2n$-dimensional feature vector of a pattern, the Pareto dominance relation can be regarded as the category of the pattern, namely Pareto dominance class. Each category is labeled with class ω_i, where i=1~3. ω_1, ω_2 and ω_3 respectively represent Pareto dominance relation $\mathbf{u} \prec \mathbf{v}$, $\mathbf{u} \succ \mathbf{v}$, $\mathbf{u} \sim \mathbf{v}$.

For a MO problem, assuming two candidate solution sets with size p, q respectively are generated randomly in the decision variable domain, each sample $s(\mathbf{x}_1, \mathbf{x}_2)$ of Pareto dominance class is constructed as the way that $\mathbf{x}_1$ is selected randomly from the p solutions and $\mathbf{x}_2$ from the q solutions. Then p+q candidate solutions can be be used to construct a Pareto dominance sample set with size $p \cdot q$. The class label ω_i^j of the sample $s_j(\mathbf{x}_1^j, \mathbf{x}_2^j)$ is determined by computing and comparing the objective vectors of the candidate solutions $\mathbf{x}_1^j$ and $\mathbf{x}_2^j$, where i=1~3, j=1~ $p \cdot q$. Given a sample set $S = \{ (s_j, \omega_i^j) \}$ with appropriate size, by using the classifier structure shown in Fig.1 and utilizing an efficient learning algorithm, it is possible to predict the Pareto dominance class of any decision variable $\mathbf{u}$ and $\mathbf{v}$.

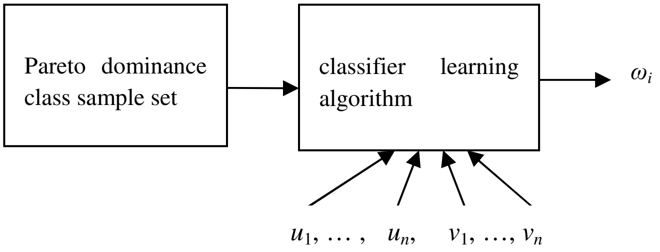
Pareto dominance class sample set
classifier learning algorithm
ω_i
$u_1, \dots, u_n, v_1, \dots, v_n$

Fig. 1. Pareto dominance classifier structure: each sample consists of a decision vector couple ($\mathbf{u}$, $\mathbf{v}$) and a known class label ω_i. $\mathbf{u}$, $\mathbf{v}$ are n-dimensional decision vectors of two candidate solutions respectively

3.2 Binary Nearest Neighbor Classification

For a MO problem, each dimension of the decision vector is encoded as a binary string with length l bits $x_i \in \{0,1\}^l$, where i=1~n. The value of the encoded binary string $\boldsymbol{x}_i$ is calculated by

$$x_i = a_i + \frac{(b_i - a_i) \times \sum_{k=1}^{l} x_{ik} \times 2^{k-1}}{2^l - 1} . \tag{2}$$

Where, $[a_i, b_i]$ is the domain of x_i.

Given two binary strings x_i and y_i with length l, the distance $d_i(x_i, y_i)$ between the strings x_i and y_i is calculated by

$$d_i(x_i, y_i) = \sum_{k=1}^{l} (x_{ik} \oplus y_{ik}) \times 2^{k-1} . \tag{3}$$

Where, $\oplus$ is the logical exclusive or operator. Then the similarity $\delta(\mathbf{x}, \mathbf{y})$ of two decision vectors **x** and **y** is measured by

$$\delta(\mathbf{x}, \mathbf{y}) = \sum_{i=1}^{n} d_i(x_i, y_i),\ i = 1 \sim n . \tag{4}$$

Given decision vector set $X = \{\mathbf{x}_1, \ldots, \mathbf{x}_N\}$, the Pareto dominance class label $C(\mathbf{u}, \mathbf{v})$ of arbitrary decision vector couple ($\mathbf{u}$, $\mathbf{v}$) is assigned by the binary nearest neighbor classification algorithm described as follows:

```
Step1. Find out $\mathbf{x}_i$, $i = \arg\min_{i=1\sim N} \delta(\mathbf{u}, \mathbf{x}_i)$;
Step2. Find out $\mathbf{x}_j$, $j = \arg\min_{j=1\sim N} \delta(\mathbf{v}, \mathbf{x}_j)$;
Step3. Let $C(\mathbf{u},\mathbf{v}) = C(\mathbf{x}_i,\mathbf{x}_j)$, where $C(\mathbf{x}_i,\mathbf{x}_j) \in \{\omega_1, \omega_2, \omega_3\}$.
```

The reason of adopting encoded binary string in BNNC is that the weighted sum of binary bits reflects the similarity of two feature variables more accurately than Euclidian distance of real value space, especially for problems in which the domains of feature variables differs in orders of magnitude.

Taking the following problem for consideration:

$$\begin{gathered} \min f_1(x_1, x_2) = x_1,\ f_2(x_1, x_2) = \frac{1 + x_2}{x_1}; \\ 0 \le x_1 \le 0.15,\ 0 \le x_2 \le 15 . \end{gathered} \tag{5}$$

The values of feature variables of two groups of samples (**a**, **b**, **c**) and (**x**, **y**, **z**) and the corresponding encoded binary strings are shown in Table 1. The Pareto dominance class label of the couple (**c**, **z**) will be determined by prediction.

Table 1. The samples and their encoded binary strings

Selected samples	Feature vectors		The encoded binary strings	
	x_1	x_2	x_1	x_2
a	0.01	10	0001	1010
b	0.12	3	1100	0011
c	0.15	7	1111	0111
x	0.01	11	0001	1011
y	0.11	4	1011	0100
z	0.07	11	0111	1011

By real-value nearest neighbor classification (RNNC) rule, the similarity δ_{r} depends on the Euclidean distances between samples. A few simple computation gives $\delta_r(\mathbf{a},\mathbf{c}) = 3.003$, $\delta_r(\mathbf{b},\mathbf{c}) = 4.000$, $\delta_r(\mathbf{x},\mathbf{z}) = 0.060$ and $\delta_r(\mathbf{y},\mathbf{z}) = 7.000$. It shows that **c** is closer to **a**, and **z** is closer to **x**. Then $\mathbf{c} \prec \mathbf{z}$ is obtained from *C*(**c**,**z**)= *C*(**a**,**x**), where *C*(**a**,**x**) is obtained by evaluating and comparing the objective vectors of **a** and **x** . In fact, the evaluated Pareto dominance relation is **c**~**z**.

However, the similarity measured by the weighted sum of binary bits are $\delta_b(\mathbf{a},\mathbf{c}) = 27$, $\delta_b(\mathbf{b},\mathbf{c}) = 7$, $\delta_b(\mathbf{x},\mathbf{z}) = 6$ and $\delta_b(\mathbf{y},\mathbf{z}) = 27$. According to BNNC algorithm, the actual Pareto dominance relationship **c**~**z** can be derived from *C*(**c**,**z**)= *C*(**b**,**x**).

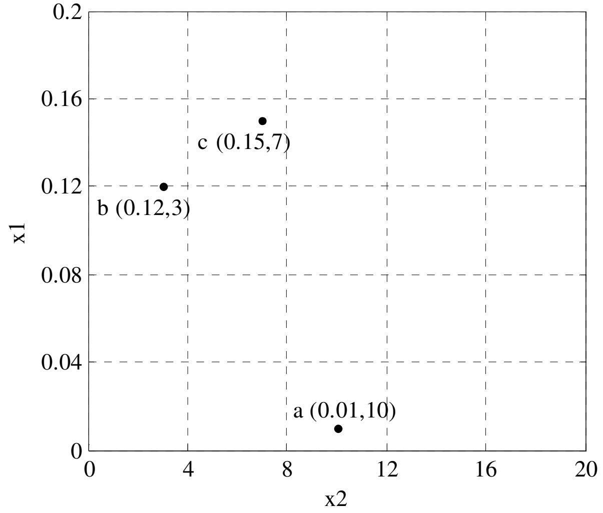

Fig. 2. The locations of sample **a**, **b** and **c** in the transformed coordinate system, where the height of domain of x_1 is enlarged to the same as the width of x_2. It is the similar case as x_1 and x_2 are encoded to binary strings with the same length.

In RNNC, Euclidean distance is calculated on original coordinate scale, the value is inclined to be dominated by the feature variable x_2 with greater interval [0,15], the contribution of the feature variable x_1 with smaller interval [0,0.15] is almost neglected on extreme cases. Consequently, the similarity measured by Euclidean distance gives the result that **a** is closer to **c** than **b**.

However, regarding the similarity of Pareto dominance, the contribution of each feature variables to the similarity computation must be identically considered. The intention can be implemented by using transformed coordinate scales as adopted in BNNC. In the transformed coordinate system, the fact **b** is closer to **c** than **a** is evident as visualized in Fig.2. Therefore, the weighted sum of binary string used in BNNC more appropriately represents the similarity of feature space.

3.3 Time Complexity

For Pareto dominance prediction of MO problems, when the canonical nearest neighbor rule[11] is used, the candidate solution sample set of size N is constructed to candidate solution couple sample set of size N^2. Therefore, the time complexity of finding out the nearest neighbor for any observed sample ($\mathbf{x}_1$, $\mathbf{x}_2$) is $O(N^2)$. But it is no need to construct candidate solution couple sample set to implement BNNC. For any observed sample $\mathbf{x}_1$ and $\mathbf{x}_2$, BNNC directly finds out the nearest neighbors from the candidate solution sample set of size N respectively. It is clear that the total time complexity is $O(N)$.

4 Experimental Results

We used BNNC and RNNC to predict the Pareto dominance for the problem in (5). The only difference is that RNNC uses real-value representation of feature variables but BNNC binary bit strings. In experiments, x_1 takes the fixed domain [0,0.1], but x_2 takes three kinds of domains [0,5], [0,50] and [0, 500].

Table 2. The average accuracy rates of BNNC and RNNC classifying the 1600 observed data over 100 sample sets. Each sample set consists of 10000 randomly generate candidate solutions. The rows correspond to the three different domain combination of feature variable x_1 and x_2.

Average class proportion in sample data (%)			Average accuracy of BNNC predicting observed data (%)			Average accuracy of RNNC predicting observed data (%)		
ω_1	ω_2	ω_3	ω_1	ω_2	ω_3	ω_1	ω_2	ω_3
11.12	11.39	77.49	71.51	70.84	93.40	61.30	70.16	90.15
13.85	14.33	71.82	76.07	78.76	91.53	50.63	42.54	78.80
14.07	14.96	70.97	80.51	78.26	89.46	38.41	39.08	73.92

For each group domain of $x1$ and $x2$, 10000 sample candidate solutions are randomly generated. The Pareto dominance relationships between samples are determined by evaluating and comparing the objective vectors. Each sample set is used to classifying the same set of randomly generated 1600 observed data. The average class proportion over 100 random sample sets and the average accuracy rate classifying the 1600 observed data are listed in Table 2.

It shows that the classification accuracy of RNNC decreases as the difference between domains of feature variables increases. But BNNC is not sensitive to the intervals of feature variables besides the overall stronger competence in classification accuracy.

For testing the robustness of BNNC, we performed additional classification tests on problems as listed in (6)~(8), which are the popular benchmark [10], [12], [13] in testing multi-objective optimization algorithms.

$$\min\ f_1(x) = x^2,\ f_2(x) = (x-2)^2; \quad -5 \le x \le 5\ . \tag{6}$$

$$\begin{aligned}\min\ & f_1(x_1,x_2) = [1+(A_1-B_1)^2+(A_2-B_2)^2],\\ & f_2(x_1,x_2) = [(x_1+3)^2+(x_2+1)^2];\\ & A_1 = 0.5\sin 1 - 2\cos 1 + \sin 2 - 1.5\cos 2,\\ & A_2 = 1.5\sin 1 - 2\cos 1 + 2\sin 2 - 0.5\cos 2,\\ & B_1 = 0.5\sin x_1 - 2\cos x_1 + \sin x_2 - 1.5\cos x_2,\\ & B_2 = 1.5\sin x_1 - \cos x_1 + 2\sin x_2 - 0.5\cos x_2;\\ & -\pi \le x_1, x_2 \le \pi\ .\end{aligned} \tag{7}$$

$$\begin{aligned}\min\ & f_1(x_1,x_2) = x_1^{\ 2} + (x_2-1)^2,\\ & f_2(x_1,x_2) = x_1^{\ 2} + (x_2+1)^2 + 10,\\ & f_3(x_1,x_2) = (x_1-1)^2 + x_2^{\ 2} + 2;\\ & -2 \le x_1, x_2 \le 2\ .\end{aligned} \tag{8}$$

In the experiments, BNNC and Bayesian classifier based on the criterion minimizing the error rate and the assumption that class condition probability density follows normal distribution, are compared. As to the size of the sample set and the observed data, we adopted the same testing conditions as described above. For each algorithm and problem, the classifying trial is performed 100 runs on varied sample set.

The average class proportion of the sample sets and the average classification accuracy rate of the two methods are shown in Table 3. The resulting data in first to third row respectively corresponds to problem (6)~(8). It indicates that the classification accuracy of Bayesian classifier evidently depends on the dimension of the feature vectors, especially the unbalanced degree of class proportion in samples. For the samples of problem (6) with 2 objectives has relatively balanced class distribution, both methods are able to get satisfying classification result, but BNNC averagely obtains higher prediction accuracy rate approximate to 98% for each class.

For the class proportions in the samples of the problem (7) and (8) with 2 feature variables are extremely unbalanced, the Bayesian classifier are almost unable to recognize the classes with miner proportion, no matter how we adjust the algorithm parameters and the size of samples. But BNNC is able to obtain acceptable classification accuracy rate about 66~75% for miner classes.

Table 3. The average class proportions over 100 sample sets, and the average classification accuracy rates of BNNC and Bayesian classifier. The rows correspond to problems (6)~(8) respectively.

Sample classes (%)			BNNC (%)			Bayesian (%)		
ω_1	ω_2	ω_3	ω_1	ω_2	ω_3	ω_1	ω_2	ω_3
34.06	33.52	32.42	97.48	97.67	97.91	82.31	86.75	91.95
14.76	15.22	70.03	66.96	70.46	87.45	20.67	22.22	94.76
15.05	15.37	69.58	74.55	72.46	90.42	33.95	30.40	98.79

5 Conclusions

Multi-objective optimization based on Pareto optimality needs to identify Pareto dominance among the candidate solutions. For extremely complicated optimization problems, population-based evolutionary algorithm may be confronted with the curse of computation cost for evaluating and comparing a large number of objective vectors. This study proposes a kind of method predicting Pareto dominance among the candidates by binary nearest neighbor classification. The algorithm encodes each feature variable into a binary string with fixed length. The similarity of two candidate solutions is measured by weighted sum of binary bits. The analysis and experiments shows that the method is able to predict Pareto dominance efficiently. In addition to get the higher classification accuracy, it is more robust than real value nearest neighbor and Bayesian classification when used in MO problems, in which the orders of magnitude of feature variable domains differ from each other and the class proportions of the samples are unbalanced.

Acknowledgments. The authors gratefully acknowledge the support from the National Natural Science Foundation of China under grant No.60975049 and 30971570, and the Natural Science Foundation of Hunan Province under grant key project No.11JJ2037.

References

1. Deb, K.: Multi-objective optimization using evolutionary algorithms: an introduction. KanGAL Report Number 2011003, Indian Institute of Technology Kanpur (2011)
2. Zhou, A., et al.: Multiobjective evolutionary algorithm: A survey of the state of the art. In: Swarm and Evolutionary Computation, vol. 1, pp. 32–49 (2011)

3. Deb, K., et al.: A fast and elitist multiobjective genetic algorithm: NSGA-II. IEEE Transactions on Evolutionary Computation 6(2), 182–197 (2002)
4. Zitzler, E., Laumanns, M., Thiele, L.: SPEA2: Improving the strength pareto evolutionary algorithm. In: Giannakoglou, K., Tsahalis, D.T., Périaux, J., Papailiou, K.D., Fogarty, T. (eds.) Evolutionary Methods for Design, Optimization and Control with Applications to Industrial Problems, pp. 95–100. Springer, Berlin (2002)
5. Knowles, J.D., Corne, D.W.: The Pareto archived evolutionary strategy: A new baseline algorithm for Pareto multiobjective optimization. In: Proc. of CEC 1999, vol. 1, pp. 98–105. IEEE Press, Piscataway (1999)
6. Coello, A.C.C., Pulido, G.T., Lechuga, M.S.: Handling multiple objectives with particle swarm optimization. IEEE Transactions on Evolutionary Computation 8(3), 256–279 (2004)
7. Li, X., Li, W.: Problems in complex engineering system optimization design and alternative solution. Chinese Journal of Mechanical Engineering 42(6), 156–160 (2006) (in Chinese)
8. Nain, P.K.S., Deb, K.: A multi-objective search and optimization procedure with successive approximate models. KanGAL Report 2004012, Indian Institute of Technology Kanpur (2004)
9. Guo, G., Li, W., Yang, B., Li, W., Yin, C.: Predicting Pareto Dominance in Multi-objective Optimization Using Pattern Recognition. In: 2012 International Conference on Intelligent Systems Design and Engineering Applications, Sanya, Hainan, China, January 6-7 (2012)
10. Schaffer, J.D.: Multiple objective optimization with vector evaluated genetic algorithms. In: Grefensttete, J.J. (ed.) Proceeding of the First International Conference on Genetic Algorithms, pp. 93–100. Lawrence Erlbaum, Hillsdale (1987)
11. Theodoridis, S., Koutroumbas, K.: Pattern Recognition, 4th edn. China Machine Press (2010)
12. Poloni, C.: Hybrid GA for multi-objective acrodynamic shape optimization. In: Winter, G., Periaux, J., Galan, M., Cuesta, P. (eds.) Genetic Algorithms in Engineering and Computer Science, pp. 397–414. Wiley, New York (1997)
13. Viennet, R.: Multi-criteria optimization using a genetic algorithm for determining a Pareto set. International Journal of Systems Science 27(2), 255–260 (1996)