第 30 卷第 7 期
2013 年 7 月

控 制 理 论 与 应 用
Control Theory & Applications

Vol. 30 No. 7
Jul. 2013

DOI: 10.7641/CTA.2013.12016

解决多目标优化问题的差分进化算法研究进展

叶洪涛[1†], 罗 飞[2], 许玉格[2]

(1. 广西工学院 电气与信息工程学院, 广西 柳州 545006;

2. 华南理工大学 自动化科学与工程学院, 广东 广州 510640)

摘要: 差分进化(differential evolution, DE)是一种简单但功能强大的进化优化算法. 由于其优秀的性能, 其诞生之日起就吸引了各国研究人员的关注. 作为一种基于群体的全局性启发式搜索算法, 差分进化算法在科学和工程中有许多成功的应用. 本文对解决多目标优化问题的差分进化算法研究进行了综述, 对差分进化的基本概念进行了详细的描述, 给出了几种解决多目标优化问题的差分进化算法变体, 并且给出了差分进化算法解决多目标优化问题的理论分析, 最后, 给出了差分进化算法解决多目标优化问题的工程应用, 并指出了未来具有挑战性的研究领域.

关键词: 多目标优化; 差分进化; 进化算法; 启发式; Pareto优化

中图分类号: TP181　　　**文献标识码**: A

Differential evolution for solving multi-objective optimization problems: a survey of the state-of-the-art

YE Hong-tao[1†], LUO Fei[2], XU Yu-ge[2]

(1. School of Electrical and Information Engineering, Guangxi University of Technology, Liuzhou Guangxi 545006, China;

2. College of Automation Science and Technology, South China University of Technology, Guangzhou Guangdong 510640, China)

Abstract: Differential evolution (DE) is a simple but powerful evolutionary optimization algorithm. It has drawn the attention of researchers all round the globe with its perfect performance since its inception. As a global search of metaheuristics based on population, DE has many successful scientific and engineering applications. A survey of DE for solving multi-objective optimization problems (MOPs) is presented. A detailed review of the basic concepts of DE is provided. Several important variants of DE for solving MOPs are presented. Moreover, the theoretical analyses on DE for solving MOPs are provided. Finally, the engineering applications of DE for solving MOPs and its future challenging field are also pointed out in the remainder of this paper.

Key words: multiobjective optimization; differential evolution; evolutionary algorithms; metaheuristics; Pareto optimality

1 Introduction

A multiobjective optimization problem is a simultaneous search process for optimal or near optimal trade-off solutions, given some conflicting objective functions[1]. The multi-objective optimization problems (MOPs) can be generally expressed as

$$\begin{cases} \min\ F(x) = (f_1(x), f_2(x), \cdots, f_m(x)), \\ \text{s.t. } G(x) = (g_1(x), g_2(x), \cdots, g_m(x)), \end{cases} \tag{1}$$

where x is a decision vector $(x_1, \cdots, x_n)$, $F(x)$ is an objective vector, and $G(x)$ represents the constraints.

There are two basic goals in multiobjective optimization: (a) to discover solutions as close to the Pareto front as possible; (b) to find solutions as diverse as possible in the obtained non-dominated front. Satisfying these two goals is a challenging task for any algorithms.

As a global search of metaheuristics based on population, differential evolution (DE) has received special attention. The first written article on DE appeared as a technical report by Price and Storn[2] in 1995. DE finished 3rd at the first international contest on evolutionary computation (1st ICEO), which was held in Nagoya, May 1996. DE is a branch of evolutionary algorithms (EAs) for optimization problems over continuous domains. However, unlike traditional EAs, DE employs difference of the parameter vectors to explore the objective function landscape. Over the years, the main advantages of the DE can be summarized as follows:

1) DE is relatively more immune to differences in initial populations than one-point optimizers. Because it is a direct search method, DE is versatile enough to solve problems whose objective functions lack the analytical de-

Received 3 February 2012; revised 31 March 2013.
†Corresponding author. E-mail: yehongtao@126.com; Tel.: +86 15177741648.
This work was supported by the Key Project of Chinese Ministry of Education (No.212135), Guangxi Natural Science Foundation (Nos.2012GXNSFB A053165), the Project of Education Department of Guangxi Autonomous Region (Nos. 201203YB131, 201202ZD071), Doctoral Initiating Project of Guangxi University of Technology (No. 11Z09), and the Fundamental Research Funds for the Central Universities (No. 20112M0126).

scription needed to compute gradients[3].

2) The number of control parameters in DE is very few. The classic DE has three parameters that need to be adjusted: a) the population size NP; b) the mutation scale factor F; c) the crossover rate C_{r}. And how these parameters affect the performance of the algorithm is well studied in [4–6].

3) DE is much simpler and straightforward to implement and modify than most other EAs. Compared to some of the most competitive real parameter optimizers like covariance matrix adaptation evolution strategy[7], the space complexity of DE is low.

Perhaps these advantages triggered the popularity of DE among researchers all around the world in a short period of time. Next, we provide the basic concepts of the DE algorithm.

DE generates new candidate solutions by combining the parent individual and several other individuals of the same population. A candidate replaces its parent only if it has better fitness. It is a rather greedy selection scheme that often outperforms traditional EAs[8]. More specifically DE's basic strategy can be summarized as follows:

Initialization. DE is a parallel direct search method. It begins with a randomly initiated population of NP D–dimensional parameter vectors $x_{i,G}$, $i = 1, 2, \cdots, NP$ as a population for each generation G. The initial population $(G = 0)$ of the jth parameter of the ith vector is

$$x_{j,i,0} = x_{j,\min} + \mathrm{rand}_{i,j}[0, 1] \cdot (x_{j,\max} - x_{j,\min}), \tag{2}$$

where $x_{j,\min}$, $x_{j,\max}$ indicate the lower and upper bounds, respectively. $\mathrm{rand}_{i,j}[0, 1]$ is a uniformly distributed random number lying between 0 and 1.

Mutation. DE mutates and recombines the population to produce a population of NP trial vectors. Specifically, for each individual $x_{i,G}$ a mutant vector $v_{i,G}$, is generated according to

$$v_{i,G} = x_{r_1^i,G} + F \cdot (x_{r_2^i,G} - x_{r_3^i,G}), \tag{3}$$

where F, commonly known as scale factor, is a positive real number. Three other random individuals $x_{r_1^i,G}$, $x_{r_2^i,G}$ and $x_{r_3^i,G}$ are sampled randomly from the current population such that

$$r_1^i, r_2^i, r_3^i \in \{1, 2, \cdots, NP\},\ i \neq r_1^i \neq r_2^i \neq r_3^i.$$

Crossover. To complement the differential mutation search strategy, DE adopts a crossover operation, often referred to as discrete recombination. In particular, DE crosses each vector with a mutant vector.

$$u_{j,i,G} = \begin{cases} v_{j,i,G}, & \text{if } \mathrm{rand}_{i,j}[0, 1] \leqslant C_{\mathrm{r}}, \text{ or } j = j_{\mathrm{rand}}, \\ x_{j,i,G}, & \text{otherwise}, \end{cases} \tag{4}$$

where C_{r} is called the crossover rate.

Selection. To decide whether or not it should become a member of generation $G+1$, the trial vector $v_{i,G}$, is compared to the target vector $x_{i,G}$, using the greedy criterion. The selection operation is described as

$$x_{i,G+1} = \begin{cases} v_{i,G}, & \text{if } f(u_{i,G}) \leqslant f(x_{i,G}), \\ x_{i,G}, & \text{otherwise}, \end{cases} \tag{5}$$

where $f(x)$ is the objective function to be minimized.

Variants of DE. There are several variants of DE which can be classified using the notation DE/$x/y/z$. where x represents the vector to be mutated, y is the number of difference vectors considered for perturbation of x, and z denotes the crossover scheme. The different mutation schemes, suggested by Storn and Price[3], are summarized as follows:

1) DE/rand/1/bin:

$$v_{i,G} = x_{r_1^i,G} + F(x_{r_2^i,G} - x_{r_3^i,G}).$$

2) DE/rand/2/bin:

$$v_{i,G} = x_{r_1^i,G} + F(x_{r_2^i,G} - x_{r_3^i,G}) + F(x_{r_4^i,G} - x_{r_5^i,G}).$$

3) DE/best/1/bin:

$$v_{i,G} = x_{\mathrm{best},G} + F(x_{r_1^i,G} - x_{r_2^i,G}).$$

4) DE/best/2/bin:

$$v_{i,G} = x_{\mathrm{best},G} + F(x_{r_1^i,G} - x_{r_2^i,G}) + F(x_{r_3^i,G} - x_{r_4^i,G}).$$

5) DE/current-to-best/1/bin:

$$v_{i,G} = x_{i,G} + F(x_{\mathrm{best},G} - x_{i,G}) + F(x_{r_1^i,G} - x_{r_2^i,G}).$$

Since DE algorithms can tackle a group of candidate solutions, it seems natural to use them in MOPs to search a group of optimal solutions. MOPs involve multiple objectives, which should be optimized simultaneously and that often are in conflict with each other. So in MOPs, the decision is not so straightforward. Then the concept of dominance is used. A solution is said to dominate another solution if it is as good as the other and better in at least one objective. That is x^* dominates x, if and only if

$$\begin{cases} \forall i \in \{1, \cdots, m\}, & f_i(x^*) \leqslant f_i(x), \\ \wedge \exists j \in \{1, \cdots, m\}, & f_j(x^*) < f_j(x). \end{cases} \tag{6}$$

The outline of DE algorithm for solving MOPs is presented in Algorithm 1. The candidate replaces the parent if it dominates the parent. Many variants of creation of a candidate are possible. The DE scheme DE/rand/1/bin is described in Algorithm 2.

Algorithm 1 Outline of the DE for solving MOPs.

Step 1 Initialize and evaluate population

$$P_G = \{x_{1,G}, \cdots, x_{NP,G}\}.$$

Step 2 While stopping criterion is not satisfied, do:

Step 2.1 For each individual $x_{i,G}$ from P_G, repeat:

Step 2.1.1 Generate candidate $v_{i,G}$ from parent $x_{i,G}$.

Step 2.1.2 Evaluate the candidate.

Step 2.1.3 If the candidate dominates the parent, the candidate replaces the parent. Otherwise, the candidate is discarded.

Step 2.2 If the population exceeds NP, truncate it.

Step 3 Return non-dominated individuals from P_G.

Algorithm 2 Candidate generation in DE/rand/1/bin.

Candidate generation.

Input: Parent $x_{i,G}$.

Step 1 Randomly select individuals.

Step 2 Calculate candidate $v_{i,G}$ as

$$v_{i,G} = x_{r_1^i,G} + F \cdot (x_{r_2^i,G} - x_{r_3^i,G}),$$

where F is a scale factor.

Step 3 Modify the candidate by binary crossover with the parent using crossover rate C_{r}.

Output: Candidate $v_{i,G}$.

DE algorithms have been proposed in the literature to overcome the drawbacks of traditional approaches to MOPs. Indeed, DE algorithms have been proved very efficient in solving MOPs. Many DE algorithms were formulated by the researchers to tackle MOPs in the past years.

Abbass et al.[9] proposed a Pareto-frontier differential evolution (PDE) algorithm for solving MOPs. The PDE employed DE to creat new individuals and keep only the nondominated ones as the basis for the next generation. The PDE was found to outperform the strength Pareto evolutionary algorithm (SPEA)[10] on two test problems.

Xue et al.[11] proposed a multiobjective differential evolution (MODE) algorithm. In MODE, the fitness of an individual was firstly calculated using Pareto-based ranking and then reduced with respect to the individual's crowding distance value. The MODE was tested on five benchmark problems where it tended to be more effective in finding the Pareto front in the sense of accuracy and approximate representation of the real Pareto front with comparable efficiency.

Yao et al.[12] proposed a multiobjective DE algorithm, which took the selection by the non-dominated sorting and crowding distance. The results indicated that the algorithm was better than the non-dominated sorting genetic algorithms II (NSGA-II)[13] both in convergence and in diversity. Varadarajan[14] presented a DE algorithm to solve optimal power flow problem with multiple and competing objectives. The problem was formulated as a nonlinear constrained true multiobjective optimisation problem with competing objectives.

Some researchers proposed non-Pareto-based approaches for solving MOPs. Li and Zhang[15–16] presented a multiobjective DE algorithm based on decomposition (MOEA/D-DE) for continuous MOPs with variable linkages. The DE/rand/1/bin scheme is used to creat new trial solutions, and a neighborhood relationship among all the sub-problems generated is defined, such that they all have similar optimal solutions. Summation of normalized objective values with diversified selection approach was used by Qu and Suganthan[17] without the need for performing non-dominated sorting.

In this work, we focus on a review of the state-of-theart in MOPs with DE as a search engine. The remainder of this paper is organized as follows. Section 2 provides several prominent variants of DE for solving MOPs. Section 3 presents the theoretical analyses on DE for solving MOPs. Section 4 provides an overview of the most significant engineering applications. Section 5 highlights the potential future research directions. Section 6 concludes this paper.

2 Prominent variants of differential evolution for solving MOPs

The DE algorithm has attracted the attention of the researchers from different fields since its inception in 1995. It has resulted in a large number of variants of the basic DE algorithm. Some variants are designed to deal with specific applications, and others are generalized for numerical optimization. In this section, we undertake an in-depth discussion of the most important variants of DE for solving MOPs.

2.1 DE with adaptation strategy

DE algorithms have been successfully used in solving MOPs. However, there is need to choose the suitable parameters to ensure the success of the algorithms. The classical DE algorithms contain three control parameters C_{r}, F and NP.

Self-adaptation allows an evolution strategy to adapt itself without any user interaction[18]. Adaptive parameter control can enhance the robustness of the algorithm by dynamically adapting the parameters to the characteristic of different fitness landscapes. It avoids the user's prior knowledge of the relationship between the parameter settings and the characteristics of optimization problems[19]. Some researchers developed DE algorithms with adaptation strategy for sloving MOPs.

Abbass[20] proposed a self-adaptive Pareto differential evolution (SPDE) algorithm for multiobjective optimization. The SPDE algorithm self-adapted the crossover rate C_{r} for solving MOPs. The mutation scale factor F was generated from the normal distribution $U(0,1)$ for each variable. The experiments reported by Abbass showed that the SPDE algorithm is very competitive to other evolutionary multiobjective optimization algorithms. Bi and Xiao[21] proposed a self-adaptive differential evolution multi-objective optimization (SDEMO). The elitist selection strategy and the crowding distance calculation in the model of SDEMO were improved to achieve better convergence performance based on the model of NSGA-II.

The concept of self-adaptive DE has been extended to handle MOPs by some researchers in the past years. Wu et al.[22] proposed a multiobjective self-adaptive differential evolution (MOSADE) algorithm for the simultaneous optimization of component sizing and control strategy in parallel hybrid electric vehicles. The MOSADE algorithm adopted an external elitist archive to retain nondominated solutions that were found during the evolutionary process. And the MOSADE algorithm employed a progressive comparison truncation operator based on the normalized nearest neighbor distance to preserve the diversity of Pareto optimal solutions. The results indicated the capability of the proposed algorithm to generate well-distributed Pareto optimal solutions. Huang et al.[23–24] proposed a multiobjective self-adaptive differential evolution with objective-wise learning strategies to solve numerical optimization problems with multiple conflicting objectives. Zamuda et al.[25] proposed a differential evolution for multiobjective optimization with self-adaptation (DEMOwSA) algorithm. The DEMOwSA uses the adaptation mechanism from evolution strategies to adapt F and C_{r} parameters of the candidate creation in DE.

Xue et al.[26] presented a fuzzy logic controller to adjust the parameters of the multiobjective DE algorithm dynamically. The fuzzy logic controlled multiobjctive DE

(FLCMODE) was applied to a suite of benchmark functions proposed by Zitzler et al.[27]. Compared with those results obtained by using MODE with constant parameter settings, the results showed that the FLC-MODE obtained were better in 80% of the testing examples. Qian and Li[28] proposed a new adaptive differential evolution algorithm (ADEA) for solving MOPs. In the ADEA, the parameter F based on the number of the current Pareto front and the diversity of the current solutions is given for adjusting search size in every generation to find Pareto solutions in mutation operator.

2.2 DE based on opposite operation

The opposite operation used in DE for solving MOPs has been demonstrated effectivièly[29]. Dong and Wang[30] proposed a multiobjective DE algorithm based on opposite operation. The opposite number is given as follows:

Definition 1 (Opposite number.) Let $x \in [a, b]$ be a real number. The opposite number $\tilde{x}$ is defined by

$$\tilde{x} = a + b - x. \tag{7}$$

The multiobjective DE based on opposite operation is presented in Algorithm 3.

Algorithm 3 ODE based on opposite operation.

Step 1 Initialize the population P using opposite operation and choose the non-dominated set E.

Step 2 While stopping criterion is not met, do:

Step 2.1 Perform mutation using DE scheme.

Step 2.2 Perform crossover.

Step 2.3 Repair the offspring which is out of the decision space. P' consists of the offspring.

Step 2.4 Generate opposite points of offspring according to the number of new non-dominated solutions. The opposite population is denoted by OP'.

Step 2.5 In set $P \cup P' \cup OP'$, select the next generation and update the external non-dominated set.

Step 3 Return the external non-dominated set E.

2.3 Hybrid DE algorithms

Hybridisation primarily refers to the process of combining the best features of more algorithms together, to form a new algorithm that is expected to outperform its ancestors over application-specific or general benchmark problems[31]. Researchers have hybridized DE with other algorithms. Deb et al.[32] proposed a hybrid methodology evolutionary and local search approaches. Local search approaches primarily explore a small neighborhood of a candidate solution in the search space until a locally optimal point is found. In [33–34] the authors combine DE with chaotic theory. These approaches aim to aggregate the advantages of both methods efficiently tackle the MOPs.

Chang and Wu[35] investigated the optimal multiobjective planning of large-scale passive harmonic filters using the hybrid DE (HDE) algorithm. The migrant and accelerating operating embedded in HDE are used to handle local optimal solutions and time consumption problems. Gujarathi and Babu[36] presented a hybrid strategy of multiobjective DE (hybrid-mode). The hybrid-MODE is consisted of an EA for global search and a deterministic algorithm for local search.

2.4 DE based on multi-populations

Santana-Quintero and Coello Coello[37] proposed the ε-MyDE algorithm. The algorithm adopts a secondary population in order to retain the non-dominated solutions found during the evolutionary process. Additionally, the algorithm also incorporates the concept of ε-dominance to get a good distribution of the solutions retained.

Yao et al.[38] presented a DE algorithm based on multi-swarm and sub-objective optimization to solve the difficulty in selecting the weighting coefficients in processing the objective function of hot strip mills. Each sub-swarm optimizes a sub-objective and evolves independently. This not only solves the issue of weighting coefficients, but also increases the convergence speed and accuracy. Song and Zhang[39] proposed a multi-population mechanism for DE to make the Pareto fronts more evenly distributed. Compared with NSGA-II, the proposed method is more efficient.

3 Theoretical analyses on DE for solving MOPs

The theoretical analysis of MOPs is more difficult than its single objective counterpart since it involves issues such as the size of Pareto set, diversity of the obtained solutions and convergence to the Pareto front[40]. Consequently, there are few results on theoretical analysis of DE for solving MOPs.

Convergence analyses on DE for solving MOPs are very important to understand their search behaviors. Some convergence analyses about multiobjective extensions of DE have been done. Xue et al.[41] performed a mathematical modeling and convergence analysis of a continuous multiobjective DE (C-MODE) algorithm. The authors study the C-MODE operators and their effects on the convergence properties of the algorithm by examining the evolving probability distribution of the population. To facilitate the mathematical analysis, the authors assume the population is initialized by sampling from a Gaussian distribution with a mean μ^0 and a covariance matrix Σ^0. The authors prove that the initial population P_0 is Gaussian distribution and contains the Pareto optimal set Λ^*, the subsequent populations generated by the C-MODE without selection are also Gaussian distribution and the population mean converges to the center of the Pareto optimal set Λ^*, i.e., if X_t be a solution vector belonging to the population P_t at the generation t, then

$$\lim_{t \to \infty} E(X^t) = E(X^*), \tag{8}$$

where X^* denotes a uniformly distributed random solution with probability support of Λ^*. The convergence properties of C-MODE were studied in a similar manner to the work presented by Hanne in [42].

Running time analysis of DE for solving MOPs is a critical issue by its own right. Meng et al[38] compared the time complexity of DE based on double populations for constrained MOPs with NSGA-II and SPEA. The authors considered the population size influence on time complexity only. The runtime complexity is

$$\mathrm{O}((N1+N3)^3)+\mathrm{O}((N1+N3)^2)+\mathrm{O}((N1+N3)\log(N1+N3)),$$

where N_1 is the population size of feasible solutions, N_3 is the maximum population size of the best infeasible solutions $\mathrm{O}(N_3)$. The authors point out that the runtime complexity is less than N_3, where $N=N_1+N_2+N_3$, and N_2 is the population size of infeasible solutions.

4 Engineering applications of DE for solving MOPs

Due to the multicriteria nature of most real-world problems, the literature on engineering applications of DE for solving MOPs is huge and multifaceted. There are more than thousands of application papers in diverse areas. For the sake of space economy, the major applications are summarized in Table 1.

Table 1 Engineering applications of DE for solving MOPs

	Subareas and details	Types of DE applied
Signal processing	Digital filter design	Hybrid DE[35]
	Microwave filter design	Generalized DE[43]
	Micro-Array data analysis	Multiobjective DE[44]
Chemical engineering	Optimization of adiabatic styrene reactor	Hybrid-MODE[36]
	Optimization of chemical process	Improved DE[45]
Control system	PID regulator design	DE based on double populations[46]
	Multiobjective robust PID controller	Multiobjective DE[47]
Power system	Reactive power optimization	Opposition-based DE[48]
Economics	Economic environmental dispatch	Multiobjective DE[49]
	Portfolio optimization	DEMPO[50]
Others	Engineering design	DE with hybrid selection mechanism[51]
	Product development	Multiobjective DE[52]
	Hybrid electric vehicles optimal design	Self-adaptive DE[53]

5 Potential future research directions with DE for solving MOPs

Like all other metaheuristics to solve MOPs such as particle swarm optimization (PSO)[54], DE also has some disadvantages. Most multiobjective versions of DE seem to converge very fast to the vicinity of the true Pareto front, but present problems to actually reach it and to spread solutions along the front. There are still many problems and new application areas are continually emerging for the algorithm. Below, we pointed out some potential future research directions with DE for solving MOPs.

Unlike the significant advancement made in the theoretical understanding of GA, the theoretical analysis on DE has still not made a considerable progress. Not much research has so far been devoted to theoretically analyze the search mechanism. The timing complexity analysis of DE for solving MOPs has been reported scarcely. Convergence properties analysis is still a challenging field of future search.

Many MOPs typically deal with more than three objective functions. Many conventional MOEAs applying Pareto optimality as a ranking metric may perform poorly over a large number of objective functions. Extending the multiobjective variants of DE to solve many-objective problems remains open as an active and challenging field of future research so far.

6 Conclusions

This paper attempted to provide an overall picture of the state-of-the-art research on DE for solving MOPs. Starting with a comprehensive introduction to the basic strategy of DE, it provided several prominent variants of DE for solving MOPs. Moreover, it provided the theoretical analyses on DE for solving MOPs. Next it provided an overview of the most significant engineering applications. Finally, it pointed out the potential future research directions. The content of this paper indicates that DE for solving MOPs will continue to remain an active and challenging research field in the years to come.

References:

[1] CASTRO P A D, ZUBEN F J V. Multi-objective Bayesian artificial immune system: empirical evaluation and comparative analyses [J]. *Journal of Mathematical Modelling and Algorithms*, 2009, 8(2): 151 – 173.

[2] STORN R, PRICE K V. *Differential evolution-A simple and efficient adaptive scheme for global optimization over continuous spaces* [R]. USA: International Computer Science Institute, 1995.

[3] PRICE K V, STORN R, LANPINEN J. *Differential Evolution: A Practical Approach to Global Optimization* [M]. Berin: Springer, 2005.

[4] DEB K. *Multi-Objective Optimization using Evolutionary Algorithms* [M]. Chichester: Wiley, 2001.

[5] GAMPERLE R, MULLER S, KOUMOUTSAKOS P. A parameter study for differential evolution [M] //*Advance in Intelligent System, Fuzzy System, Evolutionary Computation*. New York: WSEAS Press, 2002.

[6] WANG Y, CAI Z X, ZHANG Q F. Differential evolution with composite trial vector generation strategies and control parameters [J]. *IEEE Transactions on Evolutionary Computation*, 2011, 15(1): 55 – 66.

[7] HANSEN N, OSTERMEIER A. Completely derandomized self-adaptation in evolution strategies [J]. *Evolutionary Computation*, 2011, 9(2): 159 – 195.

[8] ROBIC T, FILIPIC B. DEMO: Differential evolution for multiobjective optimization [M] //*Lecture Notes in Computer Science*. Berlin: Springer, 2005.

[9] ABBASS H A, SARKER R, NEWTON C. PDE: A Pareto-frontier differential evolution approach for multi-objective optimization problems [C] //*Proceedings of the Congress on Evolutionary Computation*. Piscataway, NJ: IEEE, 2002: 831 – 836.

[10] ZITZLER E, THIELE L. Multiobjective evolutionary algorithms: A comparative case study and the strength Pareto approach [J]. *IEEE Transactions on Evolutionary Computation*, 1999, 3(4): 257 – 271.

[11] XUE F, SANDERSON A C, GRAVES R J. Pareto-based multiobjective differential evolution [C] //*Proceedings of the 2003 Congress on Evolutionary Computation*. Piscataway, NJ: IEEE, 2003: 862 – 869.

[12] YAO Feng, YANG Weidong, ZHANG Ming. Multi-objective differential evolution used for load distribution of hot strip mills [J]. *Control Theory & Application*, 2010, 27(7): 897 – 902
(姚峰, 杨卫东, 张明. 多目标差分进化在热连轧负荷分配中的应用 [J]. 控制理论与应用, 2010, 27(7): 897 – 902.)

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

[14] VARADARAJAN M, SWARUP K S. Solving multi-objective optimal power flow using differential evolution [J]. *IET Generation, Transmission and Distribution*, 2008, 2(5): 720 – 730.

[15] LI H, ZHANG Q F. Multiobjective optimization problems with complicated Pareto sets, MOEA/D and NSGA-II [J]. *IEEE Transactions on Evolutionary Computation*, 2009, 13(2): 284 – 302.

[16] LI H, ZHANG Q F. A multiobjective differential evolution based on decomposition for multiobjective optimization with variable linkages [M] //*Lecture Notes in Computer Science*. Berlin: Springer–Verlag, 2006.

[17] QU B Y, SUGANTHAN P N. Multiobjective evolutionary algorithms based on the summation of normalized objectives and diversified selection [J]. *Information Science*, 2010, 180 (17): 3170 – 3181.

[18] EIBEN A E, SMITH J E. *Introduction to Evolutionary Computing (Natural Computing Series)* [M]. Berlin: Springer, 2003.

[19] ZHANG J Q, SANDERSON A C. JADE: Adaptive differential evolution with optional external archive [J]. *IEEE Transactions on Evolutionary Computation*, 2009, 13(5): 945 – 958.

[20] ABBASS H A. The self-adaptive Pareto differential evolution algorithm [C] //*Proceedings of IEEE Congress on Evolutionary Computation*. Piscataway, NJ: IEEE, 2002, 1: 831 – 836.

[21] BI Xiaojun, XIAO Jing. Multi-objective evolutionary algorithm based on self-adaptive differential evolution [J]. *Computer Integrated Manufacturing Systems*, 2011, 17(12): 2660 – 2665.
(毕晓君, 肖婧. 基于自适应差分进化的多目标进化算法 [J]. 计算机集成制造系统, 2011, 17(12): 2660 – 2665.)

[22] WU L H, WANG Y N, YUAN X F, et al. Multiobjective optimization of HEV fuel economy and emissions using the self-adaptive differential evolution algorithm [J]. *IEEE Transactions on Vehicular Technology*, 2011, 60(6): 2458 – 2470.

[23] HUANG V L, QIN A K, SUGANTHAN P N, et al. Multiobjective optimization based on self-adaptive differential evolution algorithm [C] //*Proceedings of the Evolutionary Computation*. Berlin: Springer, 2007: 3601 – 3608.

[24] HUANG V L, ZHAO S Z, MALLIPEDDI R, et al. Multi-objective optimization using self-adaptive differential evolution algorithm [C] //*Proceedings of the Eleventh conference on Congress on Evolutionary Computation*. Piscataway, NJ: IEEE, 2009: 190 – 194.

[25] ZAMUDA A, BREST J, BOSKOVIC B, et al. Differential evolution for multiobjective optimization with self adaptation [C] //*IEEE Congress on Evolutionary Computation*. Piscataway, NJ: IEEE, 2007: 3617 – 3624.

[26] XUE F, SANDERSON A C, BONISSONE P P, et al. Fuzzy logic controlled multiobjective differential evolution [C] //*Proceedings of the 12th IEEE International Conference on Fuzzy Systems*. Piscataway, NJ: IEEE, 2005: 720 – 725.

[27] ZITZLER E, DEB K, THIELE L. Comparison of multiobjective evolutionary algorithms: empirical results [J]. *Evolutionary Computation*, 2000, 8(2): 173 – 195.

[28] QIAN W Y, LI A J. Adaptive differential evolution algorithm for multiobjective optimization problems [J]. *Applied Mathematics and Computation*, 2008, 201(1/2): 431 – 440.

[29] RAHNAMAYAN S, TIZHOOSH H R, SALAMA M M A. Opposition-based differential evolution [J]. *IEEE Transactions on Evolutionary Computation*, 2008, 12(1): 64 – 79.

[30] DONG N, WANG Y P. Multiobjective differential evolution based on opposite operation [C] //*International Conference on Computational Intelligence and Security*. Beijing: IEEE Computer Society, 2009, 1: 123 – 127.

[31] DAS S, SUGANTHAN P N. Differential evolution: A survey of the stateof–the-art [J]. *IEEE Transactions on Evolutionary Computation*, 2011, 15(1): 4 – 31.

[32] DEB K, MIETTINEN K, CHAUDHURI S. Toward an estimation of nadir objective vector using a hybrid of evolutionary and local search approaches [J]. *IEEE Transactions on Evolutionary Computation*, 2010, 14(6): 821 – 841.

[33] NIU Dapeng, WANG Fuli, HE Dakuo, et al. Chaotic differential evolution for multiobjective optimization [J]. *Control and Decision*, 2009, 24(3): 361 – 364.
(牛大鹏, 王福利, 何大阔, 等. 多目标混沌差分进化算法 [J]. 控制与决策, 2009, 24(3): 361 – 364.)

[34] WANG Xiuzhen, LI Peng, YU Guoyan. Multi-objective chaotic differential evolution algorithm with grading second mutation [J]. *Control and Decision*, 2011, 26(3): 457 – 463.
(王筱珍, 李鹏, 俞国燕, 等. 分阶段二次变异的多目标混沌差分进化算法 [J]. 控制与决策, 2011, 26(3): 457 – 463.)

[35] CHANG Y P, WU C J. Optimal multiobjective planning of large-scale passive harmonic filters using hybrid differential evolution method considering parameter and loading uncertainty [J]. *IEEE Transactions on Power Delivery*, 2005, 20(1): 408 – 416.

[36] GUJARATHI A M, BABU B V. Optimization of adiabatic styrene reactor: a hybrid multiobjective differential evolution (H-MODE) approach [J]. *Industrial and Engineering Chemistry Research*, 2009, 48(24): 11115 – 11132.

[37] SANTANA-QUINTERO L V, COELLO COELLO C A. An algorithm based on differential evolution for multiobjective problems [J]. *International Joumal of Computational Intelligence Research*, 2005, 1(2): 151 – 169.

[38] YAO Feng, YANG Weidong, ZHANG Ming．Multi-objective load distribution of hot strip mills based on multi-swarmand sub-objective differential evolution [J]. textitJournal of University of Science and Technology Beijing, 2010, 32(11): 1501 – 1512.
(姚峰, 杨卫东, 张明. 基于多子群目标分段差分进化的多目标热连轧负荷分配 [J]. 北京科技大学学报, 2010, 32(11): 1501 – 1512.)

[39] SONG Tong, ZHUANG Yi. A kind of multi-objective optimization algorithm based on differential evolution with multi-population mechanism [J]. *Computer Science*, 2012, 39(8): 205 – 209.
(宋通, 庄毅. 基于多种群差分进化的多目标优化算法 [J]. 计算机科学, 2012, 39(8): 205 – 209.)

[40] DEB K. *Multiobjective Optimization Using Evolutionary Algorithms* [M]. Chichester, UK: Wiley, 2001.

[41] XUE F, SANDERSON A C, GRAVES R J. Modeling and convergence analysis of a continuous multiobjective differential evolution algorithm [C] //*Proceedings of IEEE Congress on Evolutionary Computation*. Piscataway, NJ: IEEE, 2005: 228 – 235.

[42] HANNE T. On the convergence of multiobjective evolutionary algorithms [J]. *European Journal of Operational Research*, 1999, 117(3): 553 – 564.

[43] GOUDOS S K, SAHALOS J N. Pareto optimal microwave filter design using multiobjective differential evolution [J]. *IEEE Transactions on Antennas and Propagation*, 2010, 58(1): 132 – 144.

[44] SURESH K, KUNDU D, GHOSH S, et al. Multi-objective differential evolution for automatic clustering with application to micro-array data analysis [J]. *Sensors*, 2009, 9(5): 3981 – 4004.

[45] NIU Dapeng, WANG Fuli, HE Dakuo, et al. Optimization of nosiheptide fermentation process based on the improved differential evolution algorithm for multi-objective optimization [J]. *Control Theory & Applications*, 2010, 27(4): 505 – 508.
(牛大鹏, 王福利, 何大阔, 等. 基于改进多目标差分进化算法的诺西肽发酵过程优化 [J]. 控制理论与应用, 2010, 27(4): 505 – 508.)

[46] CHEN Yi, BO Yuming, ZOU Weijun, et al. Satisfactory optimization for PID Regulator with constraints on exceeding tolerance characteristic indices [J]. *Information and Control*, 2010, 39(5): 581 – 587.
(陈益, 薄煜明, 邹卫军, 等. 超差特征量指标集约束下PID调节器的满意优化 [J]. 信息与控制, 2010, 39(5): 581 – 587.)

[47] ZHAO S Z, QU B Y, SUGANTHAN P N, et al. Multi-objective robust PID controller tuning using multi-objective differential evolution [C] //*International Conference on Control, Automation, Robotics and Vision*. Piscataway, NJ: IEEE, 2011: 2398 – 2403.

[48] MA Lixin, WANG Shouzheng, LÜ Xinhui, et al. Opposition-based differential evolution for reactive power optimization [J]. *Control Engineering of China*, 2010, 17(6): 803 – 806.
(马立新, 王守征, 吕新慧, 等. 电力系统无功优化的反向优化差分进化算法 [J]. 控制工程, 2010, 17(6): 803 – 806.)

[49] BASU M. Economic environmental dispatch using multi-objective differential evolution [J]. *Applied Soft Computing*, 2011, 11(2): 2845 – 2853.

[50] KRINK T, PATERLINI S. Multiobjective optimization using differential evolution for real-world portfolio optimization [J]. *Computational Management Science*, 2011, 8(1/2): 157 – 179.

[51] GONG W Y, CAI Z H, ZHU L. An efficient multiobjective differential evolution algorithm for engineering design [J]. *Structural and Multidisciplinary Optimization*, 2009, 38(2): 137 – 157.

[52] XUE F, SANDERSON A C, GRAVES R J. Multiobjective evolutionary decision support for design-supplier-manufacturing planning [J]. *IEEE Transactions on Systems, Man and Cybernetics*, 2009, 39(2): 309 – 320.

[53] WU L H, WANG Y N, YUAN X F, et al. Multiobjective optimization of HEV fuel economy and emissions using the self-adaptive differential evolution algorithm [J]. *IEEE Transactions on Vehicular Technology*, 2011, 60(6): 2458 – 2470.

[54] REYES-SIERRA M, COELLO COELLO C A. Multi-objective particle swarm optimizers: A Survey of the state-of-the-art [J]. *International Journal of Computational Intelligence Research*, 2006, 2(3): 287 – 308.

作者简介:

叶洪涛 (1979–), 男, 副教授, 目前研究方向为智能控制及优化, E-mail: yehongtao@126.com;

罗　飞 (1957–), 男, 教授, 目前研究方向为智能控制理论及应用, E-mail: aufeiluo@scut.edu.cn;

许玉格 (1978–), 女, 副教授, 目前研究方向为智能控制, E-mail: xuyuge@scut.edu.cn.