A Generalized Differential Evolution Combined with EDA for Multi-objective Optimization Problems

Wang Chen, Yan-jun Shi, and Hong-fei Teng

School of Mechanical Engineering,
Dalian University of Technology, Dalian, P.R. China 116024
shiyj@dlut.edu.cn

Abstract. This paper proposed a multi-objective evolutionary algorithm (called by GDE-EDA hereinafter). The proposed algorithm combined a generalized differential evolution (DE) with an estimation of distribution algorithm (EDA). This combination can simultaneously use global information of population extracted by EDA and differential information by DE. Thus, GDE-EDA can obtain a better distribution of the solutions by EDA while keeping the fast convergence exhibited by DE. The experimental results of the proposed GDE-EDA algorithm were reported on a suit of widely used test functions, and compared with GDE and NSGA-II in the literature.

Keywords: Generalized differential evolution; estimation of distribution algorithm; multi-objective optimization.

1 Introduction

Many real-world problems in the engineering are multi-objective in nature, e.g., layout optimization of satellite module, mechanical component design, etc. Over the past decades, a large amount of studies had been focused on multi-objective optimization problems (MOOPs) and had obtained a lot of achievement. However, the complexity of some multi-objective optimization problems (e.g., very large search spaces, uncertainty, noises, disjoint Pareto curves, etc.) call for new or alternative approaches.

Differential Evolution (DE) is a stochastic, population-based evolutionary algorithm for global optimization proposed by Kenneth Price and Rainer Storn[1]. DE is capable of converging to the optimal by adopt the distance and direction information from the current population of solutions. Therefore, there have been a lot of the proposed extensions of DE for multi-objective optimization. Chang et al[2] gave the first reported attempt to extend differential evolution for multi-objective problems. Bergey[3] also reported a multi-objective evolutionary algorithm based on differential evolution (called by Pareto Differential Evolution, or PDE) at about the same time as Chang et al. Ref [4] introduced a new version of PDE (called by Self-Adaptive Pareto Differential Evolution, or SPDE).

Besides, there have been many literatures on multi-objective optimization problems with constraints. Iorio and Li[5] firstly proposed the Non-dominated Sorting Differential Evolution (NSDE) with a simple modification of the NSGA-II[6]. Then

D.-S. Huang et al. (Eds.): ICIC 2008, LNAI 5227, pp. 140–147, 2008.

they [7] proposed a variation of NSDE. Santana-Quintero and Coello Coello[8] proposed the ϵ-MyDE. Additionally, Ref [9] provided a new generalized differential evolution (GDE) algorithm. To sum up, DE is a very powerful to act as search engine of multi-objective optimization. However, some issues still was raised for multi-objective version of DE. For example, multi-objective DE seemed to have a high convergence rate, but has difficulties to reach the true Pareto front (see for example [8]). The main reason is that DE has no mechanism to extract global information to guide the search for exploring promising areas and to maintain diversity.

To overcome the aforementioned shortcomings, we herein combined Estimation of distribution algorithms (EDA)[10] to spread solutions along the front. EDAs maintained and successively improved a population of candidate solutions and a probability model for promising solutions until some stopping condition was met. Sun et al. [11] had combined DE with EDA to solve some test problems. Zhou et al.[12] proposed a M-MOEA and MOEA can be regarded as a combination of EDA and NSGA-II. The preliminary experimental results show that M-MOEA performs better than NSGA-II. Peter et al.[13] proposed a new algorithm(MIDEA) for evolutionary multi-objective optimization by learning and using probabilistic mixture distributions.

In this study, we combine the DE with EDAs for MOOPs, that is, Generalized Differential Evolution with Estimation Distribution Algorithm (called by GDE-EDA hereinafter). The following sections provide the details of GDE-EDA.

2 The Proposed GDE-EDA Algorithm

Mathematically, constrained multi-objective optimization problem (MOOP) can be presented in the following form without loss of generality:

$$\begin{aligned} \min\ & F(\vec{X})=\left(f_1(\vec{X}),f_2(\vec{X}),...,f_k(\vec{X})\right)^T \\ s.t.\ & H(\vec{X})=\left(h_1(\vec{X}),h_2(\vec{X}),...,h_p(\vec{X})\right)^T=0 \\ & G(\vec{X})=\left(g_1(\vec{X}),g_2(\vec{X}),...,g_m(\vec{X})\right)^T\leq 0 \end{aligned} \tag{1}$$

where k is the number of objectives to be optimized, p and m is the number of equality constraints and inequality constraint functions respectively. In Eq.1, $\vec{X}=(x_1,x_2,...,x_D)$ is decision vector and $F(\vec{x})$ is vector of objective functions. Unlike single-objective problem, there exists a set of Pareto-optimal solutions for $F(\vec{x})$. To provide an alternative solution, we proposed a new algorithm (GDE-EDA) detailed as follows.

2.1 Generalized Differential Evolution

The multi-objective version of DE algorithm used in this study was referenced as Generalized Differential Evolution (GDE)[9]. GDE was an extension of Differential Evolution (DE) for global optimization with an arbitrary number of objectives and constraints. Like DE, GDE also has the mutation, crossover and selection operations. The mutation and crossover operators of GDE are in the same manner as basic DE, while GDE extends the selection operation of DE for constrained MOOP. In each generation, GDE uses the crossover and mutation operation of DE to generate offspring, and then uses the extension of select operation of DE to form a new population (the next generation).

However, GDE only made improvement on the selection scheme rather than improve the algorithm on the generate scheme of offspring which is the most important operation in EAs. Therefore, we improved this GDE using Estimation of Distribution Algorithm.

2.2 Estimation of Distribution Algorithm

EDA explicitly extract global statistical information from the selected solutions (often called parents) and build a posterior probability distribution model of promising solutions, based on the extracted information. New solutions are sampled from the model thus built and fully or in part replace the old population. To employ EDA, we have to deal with two important issues: (1) how to select the parent solution. We employ the 2-tournament selection for it is a typical selection method used in many applications; (2) how to build a probability distribution model $p(x)$. Since it is impractical to calculate the actual posterior distribution of the promising solutions, most of the existing EDA-like algorithms model the distribution functions by probabilistic graph models or Bayesian networks. We employ Gaussian model with diagonal covariance matrix (GM/DCM)[14] here because it considers the correlation between random variables, where the joint density function of the G -th generation is written as

$$p_G(x) = \prod_{i=1}^{n} N(x_i; \mu_i^G, \sigma_i^G), \tag{2}$$

where

$$N\left(x_i; \mu_i^G, \sigma_i^G\right) = \frac{1}{\sqrt{2\pi\sigma_i}} \exp\left(-\frac{1}{2}\left(\frac{x_i - \mu_i}{\sigma_i}\right)^2\right). \tag{3}$$

In Formula (2), the n -dimensional joint probability distribution is factorized as a product of n univariate and independent normal distributions. There are two parameters for each variable required to be estimated in the G -th generation: the mean μ_i^G, and the standard deviation σ_i^G. They can be estimated as

$$\hat{\mu}_i^G = \overline{x}_i^G = \frac{1}{M}\sum_{j=1}^{M} x_{ji}^G, \tag{4}$$

$$\hat{\sigma}_i^G = \sqrt{\frac{1}{M}\sum_{j=1}^{M}\left(x_{ji}^G - \overline{x}_i^G\right)^2}, \tag{5}$$

where $\left(x_{1,i}^G, x_{2,i}^G, \ldots, x_{M,i}^G\right)$ are values of the i -th variable of the selected M parent solutions in the G -th generation.

2.3 The GDE-EDA Algorithm

Borrowing from the aforementioned GDE and EDA, we proposed a new multi-objective algorithm, namely GDE-EDA. As mentioned above, GDE used the distance and direction information from the current population to guide its further search, but had no mechanism to extract global information for exploring promising areas and to maintain diversity in solving MOOPs. EDA explicitly extract global statistical information from the selected solutions and can avoid trapping in local optimum. Therefore, we combine GDE with EDAs, attempting to obtain a better algorithm. Figure 1 illuminates the pseudo code of GDE-EDA, where parameter $\delta(CR < \delta \leq 1)$ is used to balance contributions of the global information and the differential information. G, G_{MAX} denote the current generation and max generation of evolution process respectively. $r1, r2, r3$ is random integer.

```
initialize $G = 0, m = 0$
while $G < G_{MAX}$
    select $M$ solutions from $S_G = \{\vec{x}_{1,G}, \vec{x}_{2,G}, ..., \vec{x}_{N,G}\}$, construct $p_G(x) = \prod_{i=1}^{n} N(x_i; \mu_i^G, \sigma_i^G)$
    for each $\vec{x}_{i,G}, i = 1,2,...,N$
        for each $x_{ji,G}, j = 1,2,...,D$
            if $randb(j) < CR$ or $j = randr(i)$
                $u_{ji,G} = x_{jr1,G} + F(x_{jr2,G} - x_{jr3,G})$
            else if $CR \leq randb(j) < \delta$
                $u_{ji,G}$ is sampled according to $N(x_i; \mu_i^G, \sigma_i^G)$
            else
                $u_{ji,G} = x_{ji,G}$
            end
        end
        selection
            $\vec{x}_{i,G+1} = \begin{cases} \vec{u}_{i,G}, \text{if } \vec{u}_{i,G} \preceq_c \vec{x}_{i,G} \\ \vec{x}_{i,G}, \text{otherwise} \end{cases}$
            if $(\forall j : g_j(\vec{x}'_{i,G}) \leq 0)$ or $(\vec{x}_{i,G+1} == \vec{x}_{i,G})$ or $(\vec{x}_{i,G} \not\prec_c \vec{x}'_{i,G})$
                $m = m + 1$
                $\vec{x}_{N+m,G+1} = \vec{x}'_{i,G}$
            end
    end
    while $m > 0$
        for each $\vec{x} \in S_{G+1} = \{\vec{x}_{1,G+1}, \vec{x}_{2,G+1}, ..., \vec{x}_{N+m,G+1}\}$
            if$(\forall i \ \vec{x} \not\prec_c \vec{x}_{i,G+1})$ or $(\forall(\vec{x}_{i,G+1} : \vec{x}_{i,G+1} \not\prec_c \vec{x}) \quad CD(\vec{x}) \leq CD(\vec{x}_{i,G+1}))$
                remove $\vec{x}$ from $S_{G+1}$
                $m = m - 1$
            end
        end
    end
    $G = G + 1$
end
```

Fig. 1. The pseudo code of GDE-EDA

3 Experiments

We employed a set of widely used test functions available from the literature [6, 15, 16] to evaluate our proposed approach. The main objective of experiments was attempted to better understand the strengths and weaknesses of GDE-EDA for different types of test problems.

We used the following metrics to measure the performance[6]: (1) convergence metric Υ, which measure the convergence property of a algorithm by using distance to PF_{true} which is determined by approximating as a combination of piecewise linear segments with the average of these distances defining the metric value. (2) diversity metric Δ [6], which measures distributed extent of a obtained Pareto-optimal solutions over a non-dominated region. The smaller the value Δ, the better the spread extent, and $\Delta = 0$ stand for ideal solution set. The diversity metric was formulated as

$$\Delta = \frac{d_f + d_l + \sum_{i=1}^{N-1} |d_i - \bar{d}|}{d_f + d_l + (N-1)\bar{d}}, \tag{6}$$

where the parameter d_f and d_l are the Euclidean distance between the extreme solutions and the boundary solutions of the obtained non-dominated set. The parameter $\bar{d}$ is the average of all distances $d_i, i = 1,2,...,N-1$, assuming that there are N Pareto-optimal solutions.

The experimental results are compared with that of the state-of-the-art in the area, NSGA-II and GDE.

The size of the population of NSGA-II, GDE and GDE-EDA was 100 and all approaches are run for a maximum of 25000 function evaluations. We use the simulated binary crossover (SBX) operator and polynomial mutation for NSGA-II. Suitable control parameter values of three algorithms for each problem were found by trying out a list of different control parameter values.

3.1 Bi-objective Test Problems

Firstly, GDE-EDA was used to solve the widely used bi-objective optimization problem[6]. All these problems have two objective functions and none of these problems have any constraint. Control parameters for three algorithms are shown in Table 1.

Table 1. Control parameters for the three algorithms

problem	NSGA-II				GDE		GDE-EDA		
	p_c	p_m	η_c	η_m	CR	F	CR	δ	F
Schaffer	0.9	$0.1/n$	20	20	0.5	0.3	0.1	0.2	0.4
ZDT1	0.9	$0.1/n$	20	20	0.3	0.5	0.1	0.15	0.5
ZDT3	0.9	$0.1/n$	20	20	0.2	0.2	0.17	0.2	0.2
ZDT6	0.9	$0.1/n$	20	20	0.2	0.2	0.17	0.2	0.2

3.2 Constrained Bi-objective Test Problems

Then, some constrained bi-objective problems were used to test the constraint-handling ability of GDE-EDA. All these problems[15] have two objective functions and two constraint functions. Control parameters for three algorithms are shown in Table 2.

Table 2. Control parameters for the three algorithms

problem	NSGA-II				GDE		GDE-EDA		
	p_c	p_m	η_c	η_m	CR	F	CR	δ	F
CONSTR	0.9	$0.1/n$	20	20	0.8	0.5	0.7	0.8	0.5
SRINIVAS	0.9	$0.1/n$	20	20	0.8	0.5	0.7	0.85	0.8
TANAKA	0.9	$0.1/n$	20	20	0.2	0.2	0.7	0.85	0.8

3.3 Tri-objective Test Problems

GDE-EDA was also used to solve the widely used tri-objective test problems, i.e., DTLZ1, DTLZ4 and DTLZ7[16], and compared with GDE3 and NSGA-II. All these problems have three objectives and none of these problems have any constraint. Control parameters for three algorithms are shown in Table 3.

Table 3. Control parameters for the three algorithms

problem	NSGA-II				GDE		GDE-EDA		
	p_c	p_m	η_c	η_m	CR	F	CR	δ	F
DTLZ1	0.9	$0.1/n$	20	20	0.1	0.5	0.1	0.2	0.5
DTLZ4	0.9	$0.1/n$	20	20	0.1	0.3	0.1	0.2	0.4
DTLZ7	0.9	$0.1/n$	20	20	0.2	0.3	0.1	0.15	0.2

3.4 Results and Discussion

We made experiments on the aforementioned test functions, including bi-objective with constrain or not, tri-Objective test Problems. The experimental results of GDE-EDA were compared with that of GDE3 and that of EDA, and shown on Table 4 and Table 5 using two metrics measure the performance for comparative study. Table 4 showed the mean and variance of the convergence metric Υ obtained from NSGA-II, GDE and GDE-EDA. We can see that on most of problems, GDE-EDA is able to find better convergence solutions than other two algorithms on metric Υ. These results indicated that GDE-EDA kept the fast convergence exhibited by DE. Table 5 showed the mean and variance of the diversity metric Δ obtained from all three approaches. Considering the results from the metric Δ, we can believe that GDE-EDA was able to find better distribution solutions than other two algorithms except in ZDT3 problem.

Table 4. Statistic results of the convergence metric Υ

	NSGA-II		GDE		GDE-EDA	
	mean	variance	mean	variance	mean	variance
SCH	0.008022	0.000022	0.007966	0.000029	0.006521	0.000021
ZDT1	0.001816	0.000004	0.000344	0.000000	0.000261	0.000008
ZDT3	0.012311	0.000384	0.010493	0.000343	0.009781	0.000327
ZDT6	0.008695	0.000002	0.003626	0.000002	0.003122	0.000000
CONSTR	0.052367	0.085623	0.049658	0.005297	0.047253	0.000981
SRINIVAS	0.025254	0.000421	0.021833	0.000562	0.019237	0.000346
TANAKA	0.026427	0.000402	0.022866	0.000475	0.020895	0.000340
DTLZ1	0.008583	0.000084	0.007678	0.000063	0.007273	0.000062
DTLZ4	0.381828	0.044978	0.338800	0.134357	0.262161	0.038408
DTLZ7	0.003537	0.000006	0.002978	0.000004	0.002581	0.000004

Table 5. Statistic results of the diversity metric Δ

	NSGA-II		GDE		GDE-EDA	
	Mean	variance	mean	variance	mean	variance
SCH	0.466607	0.001654	0.399495	0.001301	0.333344	0.000934
ZDT1	0.407005	0.000065	0.342632	0.000043	0.297462	0.000035
ZDT3	0.555183	0.000753	0.507527	0.000736	0.495982	0.000747
ZDT6	0.437205	0.000044	0.361424	0.000035	0.301597	0.000027
CONSTR	0.794079	0.005043	0.633901	0.004201	0.568596	0.003269
SRINIVAS	0.405889	3.137228	0.318545	1.798468	0.316344	1.651079
TANAKA	0.795027	0.001342	0.748506	0.006955	0.692994	0.001208
DTLZ1	0.806938	0.005576	0.627404	0.003728	0.596731	0.003283
DTLZ4	0.698626	0.101996	0.638099	0.106633	0.597793	0.097992
DTLZ7	0.677810	0.297248	0.628380	0.309496	0.597150	0.294422

Therefore, it seemed that GDE-EDA outperformed GDE3 and EDA for most of experiments on two metrics measure from the results obtained so far. The reason may be that GDE-EDA combined the advantages of GDE and EDA, which can extract the global information of population for exploration and use the differential information by DE for exploitation, and control two parameters to balance the trade-off between exploration and exploitation.

4 Conclusion and Future Research

In this study we proposed a multi-objective evolutionary algorithm (GDE-EDA) to tackle both unconstrained and constrained multi-objective problems. The proposed GDE-EDA algorithm employed EDA to extract the global information of search for producing better distribution solutions, and employed DE to obtain the fast convergence. The experimental results indicated that GDE-EDA can generate Pareto-optimal solution set with better convergence and better distribution than EDA and GDE3. Further studies on employing different variations of EDA and DE for better performance will be summarized in our next work.

Acknowledgements

This work was supported by National High-tech R&D program of P R China (Grant No. 2006AA04Z109), National Natural Science Foundation of P R China (Grant Nos. 50575031) and National Defense Basic Scientific Research Project of P R China (Grant No. B0920060901).

References

1. Storn, R., Price, K.: Differential Evolution—A Simple and Efficient Heuristic for Global Optimization over Continuous Spaces. Journal of Global Optimization 11(4), 341–359 (1997)
2. Chang, C.S., Xu, D.Y., Quek, H.B.: Pareto-Optimal Set Based Multiobjective Tuning of Fuzzy Automatictrain Operation for Mass Transit System. Electric Power Applications, IEE Proceedings 146(5), 577–583 (1999)

3. Bergey, P.K.: An Agent Enhanced Intelligent Spreadsheet Solver for Multi-Criteria Decision Making. In: Proceddings of the Fifth Americas Conference on Information Systems, Milwaukee, Wisconsin, USA, pp. 966–968 (1999)
4. Abbass, H.A.: The Self-Adaptive Pareto Differential Evolution Algorithm. In: Proceedings of the 2002 Congress on Evolutionary Computation, CEC 2002 (2002)
5. Iorio, A.W., Li, X.: Solving Rotated Multi-Objective Optimization Problems Using Differential Evolution. In: Webb, G.I., Yu, X. (eds.) AI 2004. LNCS (LNAI), vol. 3339, pp. 861–872. Springer, Heidelberg (2004)
6. Deb, K., et al.: A Fast and Elitist Multiobjective Genetic Algorithm: NSGA-II. IEEE Transactions on Evolutionary Computation 6(2), 182–197 (2002)
7. Iorio, A.W., Li, X.: Incorporating Directional Information within a Differential Evolution Algorithm for Multi-Objective Optimization. In: Proceedings of the 8th annual conference on Genetic and evolutionary computation (2006)
8. Santana-Quintero, L.V., Coello, C.A.C.: An Algorithm Based on Differential Evolution for Multi-Objective Problems. International Journal of Computational Intelligence Research 1(2), 151–169 (2005)
9. Kukkonen, S., Lampinen, J.: GDE3: The Third Evolution Step of Generalized Differential Evolution. In: 2005 IEEE Congress on Evolutionary Computation (2005)
10. Mühlenbein, H., Paaβ, G.: From Recombination of Genes to the Estimation of Distribution Algorithms I. LNCS, vol. 1411, pp. 178–187 (1996)
11. Sun, J., Zhang, Q., Tsang, E.P.K.: DE/EDA: A New Evolutionary Algorithm for Global Optimization. Information Sciences 169(3-4), 249–262 (2005)
12. Zhou, A., Zhang, Q., Jin, Y., et al.: A Model-based Evolutionary Algorithm for Bi-Objective Optimization. In: Proceedings of the IEEE Congress on Evolutionary Computation, pp. 2568–2575 (2005)
13. Bosman, P.A.N., Thierens, D.: Multi-Objective Optimization with Diversity Preserving Mixture-Based Iterated Density Estimation Evolutionary Algorithms. International Journal of Approximate Reasoning 31(3), 259–289 (2002)
14. Larrañaga, P., Lozano, J.A.: Estimation of Distribution Algorithms: A New Tool for Evolutionary Computation. Kluwer Academic Publishers, Dordrecht (2002)
15. Nebro, A.J., et al.: A Cellular Genetic Algorithm for Multiobjective Optimization. NICSO, pp. 25–36 (2006)
16. Deb, K., et al.: Scalable Test Problems for Evolutionary Multi-Objective Optimization. Evolutionary Multiobjective Optimization, pp. 105–145 (2005)