2007 International Conference on Computational Intelligence and Security

A Decision-tree-based Multi-objective Estimation of Distribution Algorithm

Xiaoping Zhong
School of Aeronautics,
Northwestern Polytechnical University
No. 127 West Youyi Road, Xi'an, China
zxp_proteus@mail.nwpu.edu.cn

Weiji Li
School of Aeronautics,
Northwestern Polytechnical University
No. 127 West Youyi Road, Xi'an, China
lwjnwpu@126.com

Abstract

A new decision-tree-based multi-objective estimation of distribution algorithm (DT-MEDA) for optimization problems with continuous variables is developed. Decision-tree-based probabilistic models are used to encode conditional dependencies among variables in DT-MEDA. By building and sampling the probabilistic models, the algorithm reproduces the genetic information of the next generation. Incorporating this reproduction mechanism together with the ranking method and the truncated selection, DT-MEDA can approximate the Pareto front. Polynomial mutation operator is used to enhance exploration and maintain diversities in the populations. Furthermore, DT-MEDA adopts a procedure to eliminate a solution with smallest crowding distance at a time in the truncated selection, so that it can obtain a well spread solution set. The performance of the proposed algorithm is evaluated on four biobjective test problems and metrics from literature. Simulation results show that the proposed approach is competitive with NSGA-II and DT-MEDA is a general and effective method for multi-objective optimization.

1. Introduction

The estimation of distribution algorithms (EDAs) [1,2,3] have received an increasing attention over recently years. Many EDAs have been proposed to address discrete and continuous multi-objective optimization problems [4,5,6,7,8]. For continuous variables, the naive MIDEA [8] uses clustering technique to divide the promising solutions into linear clusters and for each cluster one Gaussian network is used. But clustering does not allow proper mixing of building blocks between clusters [9]. Očenášek presented AMBOA in which decision trees are used to encode conditional dependencies among continuous variables for single objective optimization [10]. The lack for mixing of building blocks is overcome. AMBOA performs comparably to an effective optimization algorithm ES-CMA [12]. Therefore we extend AMBOA and suggest a decision-tree-based multi-objective estimation of distribution algorithm for continuous problems.

2. Decision-tree-based multi-objective estimation of distribution Algorithm

The decision-tree-based probabilistic model [9,10,11] is used to encode conditional dependencies among variables in DT-MEDA. The solutions selected to build such model are based on the fitness that is a combination of the rank and crowding distance of each solution. There are two ways for the combination. One is to add the crowding distance to the rank the other is a hierarchical way. However, the rank and the crowding distance are two different traits for a solution, so the addition will blur the concept and leads to inaccurate result. As for the hierarchical way, the comparison is based on rank first. If a tie occurs, it will be broken by considering the crowding distance. The one with greater crowding distance wins. Here, the hierarchical way is used.

Along with non-dominated sorting and crowding distance estimation, truncated selection is performed based on the fitness. After the proper number of solutions is chosen, those solutions are used to construct a probabilistic model in which a decision tree is built for each variable. And a Gaussian kernel model is employed to capture the local distribution of each tree leaf. Offspring are generated from these models by probabilistic logic sampling [11]. And polynomial mutation is performed on the new solutions because of the exploratory capability it could give to the algorithm. In order to overcome the deficiency in adapting the variance of the search distribution, a

0-7695-3072-9/07 $25.00 © 2007 IEEE
DOI 10.1109/CIS.2007.136

IEEE computer society

variance adaptation mechanism is used in DT-MEDA. For multi-objective optimization problems with constraints, constrained-dominance [13] is adopted as constraint handling approach during ranking.

2.1. Non-dominated sorting and truncated selection

The fast non-dominated sorting approach, crowding distance estimation and truncated selection are incorporated in DT-MEDA to pick out a set of best solutions. We assign rank 1 to solutions in the first front, rank 2 to those in the second front, and so on. The lower front a solution belongs to, the better it is. After identifying all the non-dominated solutions (those in the first front) in the population, they are copied to a mating pool. Crowding distance of each solution in the mating pool is estimated in objective space. Then the truncated selection is done based on crowding distances of solutions.

In case of non-dominated solutions exceeds the population size, some solutions with the smaller values of crowding distance are removed. In this context, only the most crowded solution is removed. This process continues until the number of left solutions is equal to the population size. Note that the crowding distance of the adjoining solutions should be re-estimated each time a solution is left out. For simplicity it can be re-estimated for all the remained solutions of this rank in the mating pool in the implementation.

If the number of non-dominated solutions is less than the population size, it is necessary to pick up all the dominated solutions in the second front and copy them to the mating pool. If it is not enough yet, those in the following front (the third front) are identified and added to the mating pool, too. This procedure is repeated until the number of solutions in mating pool is not less than the population size. Then the above elimination procedure is performed on the dominated solutions on the highest front in the mating pool until the appropriate number of solutions is left.

2.2. Variance adaptation

If the kernel width σ of Gaussian distributions decreases fast, offspring will be very similar to its parent, which leads to hard generation of better solutions. To prevent variance from premature shrinking, an overall scaling factor, η, is used to control the kernel width of the marginal distributions adaptively. We find that the variance update formula for single objective optimization in [10] is not good for multi-objective optimization problems. Based on experiments, the modifications of η becomes

$$\eta' = \eta^{(t)} \alpha^{0.4 N_{succ}} \alpha^{0.6 N_{fail} p/(p-1)}$$

$$\eta^{(t+1)} = \begin{cases} \eta' e^{0.7 pt/t_{max}} & \text{if } t < 0.5 t_{max} \\ \eta' e^{0.3 pt/t_{max}} & \text{otherwise} \end{cases}$$

where N_{succ} denotes the number of new solutions being selected into the next generation, N_{fail} represents the number left, t is the current generation, t_{max} is the maximum generation, $p = 0.05 + 0.3/\sqrt{m}$, m is the number of design variables, $\alpha = exp(4/N_{sam})$ and N_{sam} is the size of new solutions. Then each kernel width σ is set to its product with $\eta^{(t+1)}$.

2.3. Structure of DT-MEDA

The flow of DT-MEDA is described as follows:

(1) Set population size N_{pop}, the promising solutions size N_{par}, the offspring size N_{sam}, mutation probability P_m, the distribution index for mutation η_m, the maximum iteration number t_{max}, and generate the initial population $P(t)$ randomly, set t=0;

(2) Select N_{par} promising solutions from $P(t)$ and form a set $S(t)$;

(3) Build a decision-tree-based probabilistic model B using $S(t)$ by DT metrics [4,9,11];

(4) Sample N_{sam} new solutions from B, and the new solutions form $O(t)$;

(5) Perform polynomial mutation on solutions in $O(t)$ with probability P_m;

(6) Create a new population $P(t+1)$ by selecting N_{pop} solutions from the merge of $P(t)$ and $O(t)$ using ranking method and truncated selection described in section 2.1. And Modify the variance according to section 2.2, let t=t+1;

(7) If the termination criteria are not met, go to (2).

Note that if $S(t)$ and $P(t)$ have the same size, then step (2) is skipped and $S(t)$ is the same as $P(t)$. The algorithm was implemented in C++ and can be requested from the authors via email.

3. Numerical experiments and results

The performance of DT-MEDA is evaluated and compared to NSGA-II (real-coded) on four test problems using the convergence metric γ and diversity metric Δ [13]. 500 uniformly spaced solutions are chosen from the true Pareto front. The two algorithms are run for a maximum of 250 generations with a population size 100. In NSGA-II, the crossover probability is 0.9 and the mutation probability is 1/n (where n is the number of design variables). The distribution index for crossover operators is 20 and mutation operators 20. For DT-MEDA, 100 promising solutions are selected as parents used to build a probabilistic model and 100 solutions are sampled on each generation. The distribution index for mutation operators is the same as it is in NSGA-II, 20.

3.1. Test problems

Four benchmark biobjective problems ZDT4, ZDT6, FON and OSY from [13] are used to test the performance of DT-MEDA.

3.2. Simulation results and discussion

The results are from 30 independent runs of the two algorithms. Each experiment starts from a randomly generated population. Table 1 shows the mean and variance of the convergence metric γ obtained by DT-MEDA and NSGA-II on the four test problems. For brevity, γ(M) is used to denote this metric of the non-dominated sets obtained by DT-MEDA and γ(N) denotes that of NSGA-II. Δ (M) stands for the diversity metric of non-dominated sets obtained by DT-MEDA, Δ(N) denotes that of NSGA-II.

Table 1. Mean and variance of

Convergence metric γ		ZDT4	ZDT6
γ(M)	Mean	0.002013	0.019145
	Variance	0.000004	0.000069
γ(N)	Mean	0.004711	0.008966
	Variance	0.000006	0.000001
		FON	OSY
γ(M)	Mean	0.002108	0.454643
	Variance	0	0.032137
γ(N)	Mean	0.003780	0.344955
	Variance	0	0.016811

Table 2 shows the mean and variance of the diversity metric Δ obtained by DT-MEDA and NSGA-II on the four test problems

Table 2. Mean and variance of Δ

Diversity metric Δ		ZDT4	ZDT6
Δ (M)	Mean	0.103513	0.191394
	Variance	0.013463	0.012254
Δ (N)	Mean	0.013809	0.022902
	Variance	0.000061	0.000015
		FON	OSY
Δ (M)	Mean	0.021759	0.058320
	Variance	0.000046	0.004323
Δ (N)	Mean	0.007672	0.090751
	Variance	0.000025	0.003517

From table 1 and table 2, we can see that DT-MEDA performs better on all ZDT4 and FON while NSGA-II shows better on ZDT6 and OSY in terms of convergence metric γ. With regard to diversity metric Δ, DT-MEDA gets better spread of non-dominated solutions only on OSY.

We perform additional experiments by increasing the number of maximum generation to 500 with other parameters fixed. Table 3 and table 4 show the convergence and diversity metric respectively.

Table 3. Mean and variance of

Convergence metric γ		ZDT4	ZDT6
γ(M)	Mean	0.001126	0.003974
	Variance	0	0
γ(N)	Mean	0.001658	0.004200
	Variance	0	0
		FON	OSY
γ(M)	Mean	0.002175	0.414480
	Variance	0	0.029617
γ(N)	Mean	0.003735	0.282066
	Variance	0	0.019050

Table 4. Mean and variance of Δ

Diversity metric Δ		ZDT4	ZDT6
Δ (M)	Mean	0.001881	0.002076
	Variance	0.000002	0.000001
Δ (N)	Mean	0.005072	0.008363
	Variance	0.000006	0.000013
		FON	OSY
Δ (M)	Mean	0.015502	0.042904
	Variance	0.000014	0.002455
Δ (N)	Mean	0.005014	0.070678
	Variance	0.000017	0.002632

It can be seen that DT-MEDA converge better in ZDT4, ZDT6 and FON after 500 generations. The variance in 30 runs is very small (less than 1e-6) except in OSY. DT-MEDA gets better spread of non-dominated solutions on ZDT4, ZDT6 and OSY. The results reveal that DT-MEDA converges slower than NSGA-II. But it can approximate the true Pareto-optimal front very closely after a relative more generation.

The non-dominated solutions obtained in a certain run on ZDT4 and ZDT6 are shown in figures 1 and 2. There are axial translations for clarity in these two figures. It can be seen that the four obtained sets are very close to the known Pareto front. The two obtained by DT-MEDA are more evenly scattered.

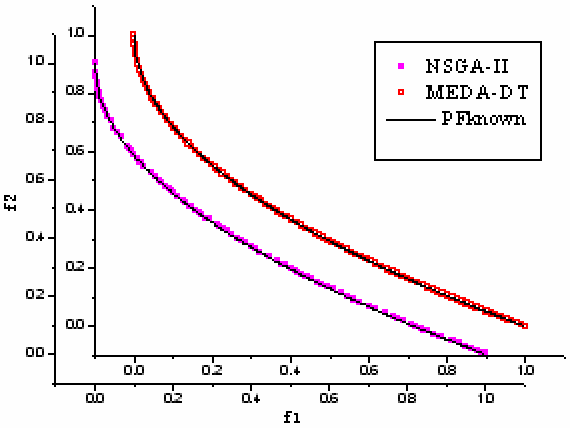

Figure 1. Non-dominated solutions on ZDT4

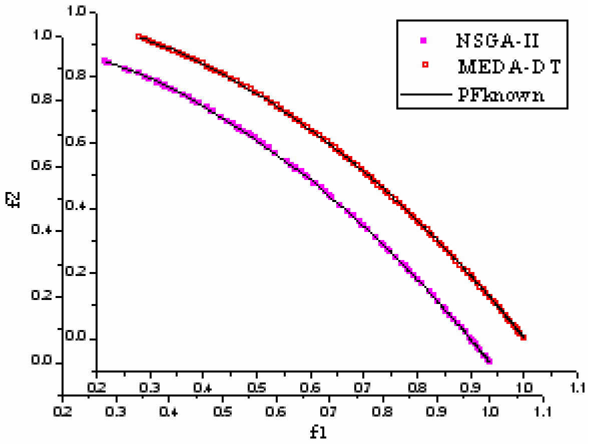

Figure 2. Non-dominated solutions on ZDT6

DT-MEDA needs more CPU-time cost than NSGA-II. For a single run on Pentium-4-2.66GHz PC with 512M memories, the time cost is given in the following table for ZDT4 and OSY

Table 5. Time cost of NSGA-II and DT-MEDA

Algorithm	ZDT4		OSY	
	250	500	250	500
NSGA-II	1.1	2.0	1.0	1.8
DT-MEDA	13.0	24.8	8.5	16.4

In the above table, 250 and 500 are the generations, and the other digits are the time cost in seconds. Although DT-MEDA is more time consuming, this shortcoming can be negligible in engineering applications where the CPU time is mainly consumed by function evaluations.

4. Summary and conclusions

Here we have presented am estimation of distribution algorithm for multi-objective optimization problems in continuous domain. This algorithm uses decision-tree-based probabilistic model to encode conditional dependencies among variables. New solutions are sampled from the probabilistic model on each generation.

The proposed algorithm is applied to four biobjective test problems. The simulation results reveal that DT-MEDA converges slower than NSGA-II. But it can approximate the true Pareto-optimal front very closely and get a uniformly scattered non-dominated solution set. And these results show that the DT-MEDA is competitive with NSGA-II. It is an effective and robust multi-objective optimization algorithm.

References

[1] M. Pelikan, D. E. Goldberg, and Erick Cantú-Paz, “BOA: The Bayesian optimization algorithm”, Illinois Genetic Algorithms Laboratory, Tech. Rep. 99003, 1999.

[2] P. Larrañaga, J. A. Lozano, *Estimation of distribution algorithms: A new tool for evolutionary optimization*, Kluwer Academic Publishers, Boston, 2001.

[3] Q. Zhang, J. Sun, E. Tsang and J. Ford, "Hybrid estimation of distribution algorithm for global optimsation", *Engineering Computations*, vol. 21, no. 1, 2004, pp. 91-107.

[4] M. Laumanns, and J. Očenášek, “Bayesian Optimization Algorithms for Multi-objective Optimization”, Lecture Notes in Computer Science: Parallel Problem Solving from Nature - PPSN VII, Granada, Spain, 2002, pp. 298-307.

[5] J. Schwarz, and J. Očenášek, “Pareto Bayesian optimization algorithm for the multiobjective 0/1 knapsack problem”, Proceedings of the 7th International Mendel Conference on Soft Computing, Brno University of Technology, Brno, 2001, pp. 131-136.

[6] N. Khan, D. E. Goldberg, and M. Pelikan, “Multi-objective Bayesian optimization algorithm”, Illinois Genetic Algorithms Laboratory, Tech. Rep. 2002009, 2002.

[7] Q. Zhang, A. Zhou and Y. Jin, “RM-MEDA: A regularity model based multiobjective estimation of distribution algorithm”, *IEEE Transaction on Evolutionary Computation*, in press, 2007.

[8] D. Thierens, and P. A. N. Bosman, “Multi-objective mixture-based iterated density estimation evolutionary algorithms”, Proceedings of the GECCO-2001 Genetic and Evolutionary Computation Conference, San Francisco, California, 2001, pp. 663-670.

[9] J. Očenášek, and J. Schwarz, “Estimation distribution algorithm for mixed continuous-discrete optimization problems”, 2nd Euro-International symposium on computational intelligence, Kosice, Slovakia, 2002, pp. 227-232.

[10] J. Očenášek, S. Kern, N. Hansen, and P. Koumoutsakos “A mixed Bayesian optimization algorithm with variance adaptation”, Parallel Problem Solving from Nature - PPSN VIII, Berlin, 2004, pp. 352-361.

[11] J. Očenášek, “Parallel estimation of distribution algorithms”, PhD dissertation, Brno University of Technology, 2002.

[12] N. Hansen, S. D. Muller, and P. Koumoutsakos, “Reducing the time complexity of the derandomized evolution strategy with covariance matrix adaptation (CMA-

ES)", *Evolutionary Computation*, vol. 11, no. 1, pp.1-18, 2003.

[13] K. Deb, A. Pratap, S. Agarwal, and T. Meyarivan, "A fast and elistist multi-objective genetic algorithm: NSGA-II" *IEEE Transaction on Evolutionary Computation*, vol. 6, no. 2, pp. 182-197, 2002.