

# Using selfish gene theory to construct mutual information and entropy based clusters for bivariate optimizations

Feng Wang · Zhiyi Lin · Cheng Yang ·  
Yuanxiang Li

Published online: 12 February 2010  
© Springer-Verlag 2010

**Abstract** This paper proposes a new approach named SGMIEC in the field of estimation of distribution algorithm (EDA). While the current EDAs require much time in the statistical learning process as the relationships among the variables are too complicated, the selfish gene theory (SG) is deployed in this approach and a mutual information and entropy based cluster (MIEC) model with an incremental learning and resample scheme is also set to optimize the probability distribution of the virtual population. Experimental results on several benchmark problems demonstrate that, compared with BMDA, COMIT and MIMIC, SGMIEC often performs better in convergent reliability, convergent velocity and convergent process.

**Keywords** Estimation of distribution algorithm · Selfish gene theory · Mutual information · Incremental learning

## 1 Introduction

Estimation of distribution algorithms (EDAs) are a novel class of evolutionary optimization algorithms which have attracted a growing interest during the last few years (Larranaga and Lozano 2002; Zhou and Sun 2007). The principal advantages of EDAs over genetic algorithms are the absence of multiple parameters to be tuned (e.g. crossover and mutation probabilities) and the expressiveness and transparency of the probabilistic model that guides the search process. In addition, EDAs have been

proven to be better suited to some applications than GAs, while achieving competitive and robust results in the majority of tackled problems.

A wide variety of EDAs using different techniques to estimate and sample the probability distribution have been proposed for solving different kinds of optimization problems (Baluja 1994; Muhlenbein and Paass 1996; Bonet et al. 1997; Baluja and Davies 1997; Pelikan and Muhlenbein 1998, 1999; Harik et al. 1999, 2006; Yang et al. 2007; Ahn and Ramakrishna 2008; Yang and Yao 2008; Hong et al. 2008; Zhang et al. 2008; Zhou et al. 2008). In these existing algorithm, a class of EDAs which focus on the bivariate dependency have been applied in the optimization of discrete problems. Mutual information maximization for input clustering (MIMIC) proposed by de Bonet et al. (1997) uses a chain model of probability distribution to estimate the pairwise conditional probabilities and sample them to get next set of solutions. Combining optimizers with mutual information trees (COMIT) proposed by Baluja and Davies (1997, 1998) also uses pairwise interaction among variables. This model which uses the cluster structure to represent the relationships of the variables is more general than the chain model used by MIMIC as two or more variables can have a common parent. The bivariate marginal distribution algorithm (BMDA) proposed by Pelikan and Muhlenbein (1999) can be seen as an extension to the COMIT model which uses Pearson's chi-square statistics to detect the interaction between two variables. These algorithms perform well in some problems, where pairwise interaction among variables exists.

However, due to the complexity of the relationships among the variables, the above EDAs require much time in the statistic learning process of the pairwise interaction of variables. It definitely reduces the performances of the algorithms and becomes worse while the problem size

---

F. Wang (✉) · Z. Lin · C. Yang · Y. Li  
State Key Laboratory of Software Engineering,  
Wuhan University, Wuhan, China  
e-mail: fengwang@whu.edu.cn

increases. In order to improve the performance of the bivariate EDAs, we exploit the selfish gene theory (SG) (Dawkins 1989) and propose a new approach named SGMIEC to solve optimization problems. It samples in the virtual population where the genes (variables) are constructed as a cluster model by their mutual information and uses the reward and punishment scheme in SG to revise the probabilities of each locus of the genes. As there are only two individuals to be chosen in each iteration in the virtual population, this approach can significantly decrease the time to construct a mutual information and entropy based cluster and get better results than the previous approaches while employing an incremental learning scheme with resample.

The rest of this paper is organized as follows. Section 2 introduces the biological SG. Section 3 goes into details of describing the SGMIEC approach. Experimental results and their analysis are given in Sect. 4. Section 5 concludes the paper.

## 2 The selfish gene theory

In the selfish gene theory (SG; Dawkins 1989), the population can be simply seen as a pool of genes and the individual genes strive for their appearances in the genotype of vehicles. The survival of the fittest is a battle fought by genes, not individuals. Only genes can survive in the evolution process. Fulvio Corno and his cooperators followed this idea and proposed a new evolutionary optimization strategy called SG (Corno et al. 1998a, b). Unlike the traditional genetic algorithm, the individual in SG is not so important to the evolution and it is just carrier of genes. And therefore, the population here is like a storeroom of genes, which is named as virtual population. Individuals would be generated when necessary and be dumped after the statistical analysis of genes. As the individuals are not explicitly represented in the virtual population, during the evolution process, SG reproduces through its effects on the statistical parameters of virtual population.

In SG, the list of genes in an individual is called genome and each position in the list of genes is termed as a locus. Each potential solution is encoded as a genotype, where each locus can be occupied by one of the several possible genes, named alleles. And therefore different alleles is represented by its frequency in the virtual population which is related to its goodness, but not represent the fitness. Fitness are calculated in the phenotype level by considering the full genome. Each solution is implicitly generated by changing the frequencies or the probabilities of the alleles. It proceeds as choosing two individuals randomly by the frequencies of alleles and compares the fitness of these two individuals. The individuals with higher fitness would be kept. Because the frequency of each allele is determined by the other alleles'

frequencies in a same gene group, SG evolves to get the best solution by changing the frequencies of the alleles according to the corresponding fitness of individuals.

## 3 SGMIEC: selfish gene on mutual information and entropy based clusters

The representation method is very important to the performance of an algorithm, which can directly affects the efficiency of the algorithm. Some researchers have proposed the dependency structure matrix in GA for optimizations (Yu and Goldberg 2004; Yu et al. 2007). As we mentioned above, the existing algorithms on bivariate optimization problems are difficult to get the best results when the problem size is increasing. In order to improve the performance of the optimization, here we propose a new SGMIEC algorithm, which is based on the SG and mutual information cluster.

---

### Algorithm 1 Selfish Gene on Mutual Information and Entropy based Cluster

---

1. INITIALIZATION
    - Initialize the frequencies of each locus in the genome;
    - Generate a dependency tree as the original cluster;
  2. MAIN LOOP: Repeat until Termination Condition is Met;
    - (a) Select two individuals according to Selfish Gene-based Selection on Virtual Population;
    - (b) Execute the Incremental Learning on Mutual Information Cluster Construction algorithm, then goto (a).
- 

### 3.1 Selfish gene-based selection on virtual population

In EDAs, new candidate solutions are generated through sampling from a density of promising solutions. It does not rely on crossover nor mutation operation in the whole evolution process. Only the selection operator is directly executed on the whole population.

In the SG, the individuals are stored with genes in a virtual population and can be selected after sampling by the density function  $P$ . As the SG states that, each variation of a gene, an allele, is in a constant battle against other alleles for the same spot on a chromosome, and any alleles more successful at increasing its presence over others have a much better chance at winning this battle over altruistic or passive genes. We believe that these successful alleles are the key genes for the individual survival and if the evolution proceeds by focusing on these key genes, it would have a good improvement on the performance. The success of an allele is often measured by the frequency with which it appears in the

virtual population, so the alleles with high frequencies are the successful genes.

Suppose the individual is identified by its  $N$  genes and the number of loci is  $m$ . Each locus  $l$  ( $l < m$ ) can be occupied by several different genes. Let  $p_{li}$  be the marginal probability of allele  $a_{li}$ , which represents the statistical frequency of  $a_{li}$  in locus  $l$  in the whole virtual population. And for each locus  $l$ , there might have  $n_l$  different alleles. The alleles that can occupy the locus  $l$  can be represented as a vector  $A_l = (a_{l1}, a_{l2}, \dots, a_{ln_l})$  and the marginal probabilities can be represented as  $P_l = (p_{l1}, p_{l2}, \dots, p_{ln_l})$ . Therefore, the virtual population can be statistically characterized as  $P = (P_1, P_2, \dots, P_m)$ . Because there is no explicit definition of the individuals in the virtual population, here we select two individuals randomly for every allele with the corresponding frequency in  $P$ .

### 3.2 Reward and punishment scheme

After we selected two individuals selected from the virtual population, we compared them by their fitness and the one with the higher fitness is regarded as the winner. In most GAs and EDAs, it only keeps the winner and ignores the loser while the loser can not generate any offspring. In the SG, the information of the candidate solution is explicitly represented by the successful genes. But if we only consider these successful genes, we might lose information in the rest of the individuals, including the losers. This is very important for the probability density function as the frequency of each allele depends on the frequencies of other alleles from different locus at the same time. As a result of this, we employ a reward and punishment scheme to update the information of the selected individuals.

The reward and punishment scheme goes as follows. Suppose the winner is  $G_{\text{best}}$  and the loser is  $G_{\text{worst}}$ . Firstly, for each pair of locus  $l1$  and  $l2$  ( $l1, l2$ ), calculate the number of the pair values of  $x_{l1}$  and  $x_{l2}$  (such as 00, 01, 10, 11) in  $G_{\text{best}}$  and  $G_{\text{worst}}$  respectively. Then, for each pair in  $G_{\text{best}}$ , increase the corresponding frequency by giving some rewards like  $\varepsilon$ , as well as decrease the frequency by giving the same punishment  $\varepsilon$  to the pairs in  $G_{\text{worst}}$ . After the change the frequencies of each pairs in the individuals, the unconditional probabilities  $p(x_{l1})$  and  $p(x_{l2})$  and conditional probability  $p(x_{l1}, x_{l2})$  can also be updated by the current value of the pair ( $l1, l2$ ).

$$\begin{aligned} \text{freq}(G_{\text{best}}(l1, l2)) &= \text{freq}(G_{\text{best}}(l1, l2)) + \varepsilon; \\ \text{freq}(G_{\text{worst}}(l1, l2)) &= \text{freq}(G_{\text{worst}}(l1, l2)) - \varepsilon; \end{aligned}$$

Herein, the value of  $\varepsilon$  determines the feedback, and therefore the balance between a fast convergence towards a local optimum and a broader exploration of the search space. Usually, it sets to  $0.01 \times N$ .

### 3.3 Incremental learning on mutual information and entropy based cluster construction

After the marginal probabilities of alleles are updated by the reward and punishment scheme, the whole virtual population should be changed consequently by sampling by the new probability distribution function. In information theory, entropy which represents the energy of the individuals (variables) is usually used to evaluate the state of an evolving system (Cover and Thomas 2006). Here we use a mutual information and entropy based cluster to represent the virtual population. Compared with ECGA which proposed by Harik (1999), we employed the clusters instead of the groups to represent the relationships of the variables. The entropy here is used to test the correlations of the variables which can help to improve the convergence performance. And each node in the information cluster stands for a locus in the genome. While mutual information is measured by the mutual dependence of two variables, the mutual information cluster is reconstructed by the new marginal probabilities after each sampling.

Since a virtual population can be represented with a probability distribution  $P(x_1, x_2, \dots, x_N) = (P_1, P_2, \dots, P_m)$  over individuals of length  $N$ , where  $x_1, \dots, x_N$  are variables corresponding to the values of the bits, now we want to establish a model which satisfies,

$$P(x_1, x_2, \dots, x_N) = \prod_{i=1}^n P(x_i | x_j, j \neq i)$$

where  $P(x_i | x_j)$  is the conditional probability of variable  $x_i$  in the population. From the above equation, we can see that the conditional probability distribution for any one bit depends on the value of at most one other bit. Suppose the variables are all identically distributed, then the mutual information of the variables is defined as

$$I(x_i, x_j) = \sum_{a,b} P(x_i = a, x_j = b) \cdot \log \frac{P(x_i = a, x_j = b)}{P(x_i = a) \cdot P(x_j = b)}$$

We set a correlation measurement parameter  $\rho$  in the cluster generation process to identify the relationships of the variables.

**Theorem 1** (Correlation Measurement) *Let  $I(x_i, x_j)$  denotes the mutual information between  $x_i$  and  $x_j$ , and  $H(x_i)$  denotes the entropy of the variable  $x_i$ , the correlation measurement can be set as,*

$$\rho = \frac{I(x_i, x_j)}{H(x_i)}$$

In order to accelerate the construction of the cluster, our SGMIEC algorithm deploy an incremental learning scheme in the mutual information cluster construction. In Baluja

and Davies's work on COMIT (1997), they chose the best  $M$  individuals generated from the dependency tree in each iteration and calculated the statistics of probability distributions in the  $M$  individuals which can be used to regenerate a new dependency tree. Since the mutual information calculation is very time-consuming and the dependency tree regeneration is also complex, it costs too much time in the tree-growing operation with  $O(n)$  in COMIT. In SGMIEC, as we use a selfish gene-based selection strategy and only two individuals are chosen every time, it costs much less time to generate the cluster in the mutual information calculating process. But the cluster regeneration subprocess still costs too much time. Here, we use the incremental learning scheme which can help to decrease the time cost and improve the performance finally. As we mentioned before, we employ a reward and punishment scheme in the individual evaluation, and the bit values of the selected two individuals  $G_{\text{best}}$  and  $G_{\text{worst}}$  are updated as a trivial revision of  $\varepsilon$ .

Since there are only two individuals chosen in the population each time, the new probability distribution is only changed by the variances of these two new individuals. As a result of this, the incremental learning on mutual information cluster construction scheme is set as Algorithm 2 shows.

---

**Algorithm 2** Incremental Learning on Mutual Information and Entropy based Cluster Construction

---

1. Compare the values of the each bit in the same locus of two individuals, if the individuals are too close (most values of the same locus are same), repeat resample;
  2. If the values in the same locus are different, employ a reward and punishment scheme to update unconditional and conditional probabilities, then calculate the mutual information  $I$ ;
  3. Calculate the new mutual information  $I(t)$  by  $I(t) = I * (1 - \alpha) + \alpha * I(t - 1)$ , where  $\alpha$  is Incremental Learning factor;
  4. Calculate the correlation measurement parameter  $\rho$  between every two variables;
  5. Generate new cluster by the new correlation measurement parameter;
  6. Generate new virtual population according to the new cluster.
- 

The incremental learning scheme can speed up the convergence velocity as well as the resample scheme can improve the diversity of the population and help to escape local optima.

## 4 Experimental results and analysis

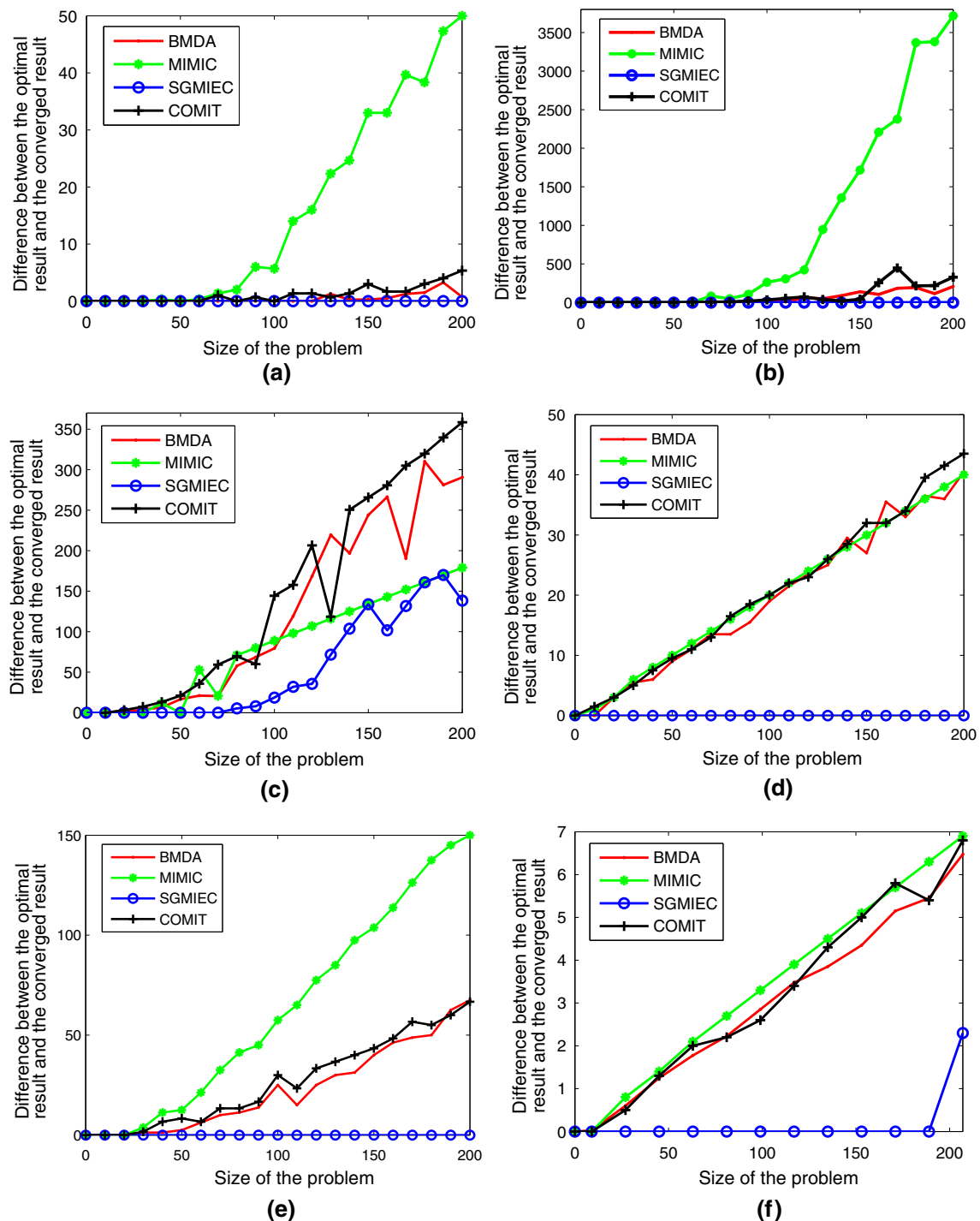
We test six benchmark problems (Bonet et al. 1997; Hong et al. 2008; Pelikan and Muhlenbein 1999; Pelikan et al. 1999) (see Appendix) for a range of problem sizes

( $n = 10\text{--}200$  with step 10 for five problems, and  $n = 9\text{--}210$  with step 18 for deceptive-3 problem) evaluate the performance of SGMIEC and compare the results with what obtained by BMDA, COMIT and MIMIC. For convenience of comparison, the optimal individual is reserved at each generation for MIMIC. Parameter settings in our experiments are given as follows, the number of samples per iteration is fixed to 200 and a fixed selection method (truncation selection with 5%) is used for COMIT, MIMIC and BMDA. For SGMIEC, the incremental learning factor  $\alpha = 0.1$  and the value of  $\varepsilon$  in reward and punishment scheme sets to default value. All parameters are held constant through all the runs. For each problem and problem size, 20 successful independent runs are performed. Each run is terminated either when the global optimum is found or when the evaluations reach the limit (200,000).

Firstly, we test and compare the convergent reliability of BMDA, MIMIC, SGMIEC and COMIT under different numbers of variables through calculating the difference between the converged solutions and the optimal solutions. The results are shown in Fig. 1. From Fig. 1, we can observe that the performance of BMDA, MIMIC, and COMIT are not satisfied for all tested six benchmark problems. For example, the difference between the converged value of the weighed One-Max Problem and the global optimal value of the Weighed One-Max Problem of MIMIC is as large as 500, when the number of variables increases up to 120. Furthermore, the curves of BMDA, MIMIC, and COMIT show great fluctuation, which means inferior stability. The results in Fig. 1 demonstrate clearly the stability of SGMIEC because the quality of its solutions is averagely superior to that of the others. Moreover, SGMIEC is able to identify high-quality solutions for all tested problems.

Secondly, we test and compare the convergent velocity of BMDA, MIMIC, SGMIEC and COMIT. The results are shown in Fig. 2. From Fig. 2, we can see that SGMIEC often requires much less fitness evaluations to converge when compared with BMDA, MIMIC and COMIT. And this phenomenon is more obvious in high dimensions. BMDA, MIMIC and COMIT show them a poor performer in deceptive problems (Trap problem and deceptive-3 problem). Experimental results for Trap function of order 5 show that SGMIEC is able to find the global optima for this problem, while BMDA, MIMIC and COMIT cannot find the global optima if the size of problem is more than 20. Additionally, these results also indicate that the needed evaluations do not increase significantly as problem size increases expect four peaks problem. SGMIEC is able to find the best solution in 40-dimensional four peaks problem requiring only about 26,380 fitness evaluations, as well as 109,733 fitness evaluations for 70-dimensional four peaks problem.

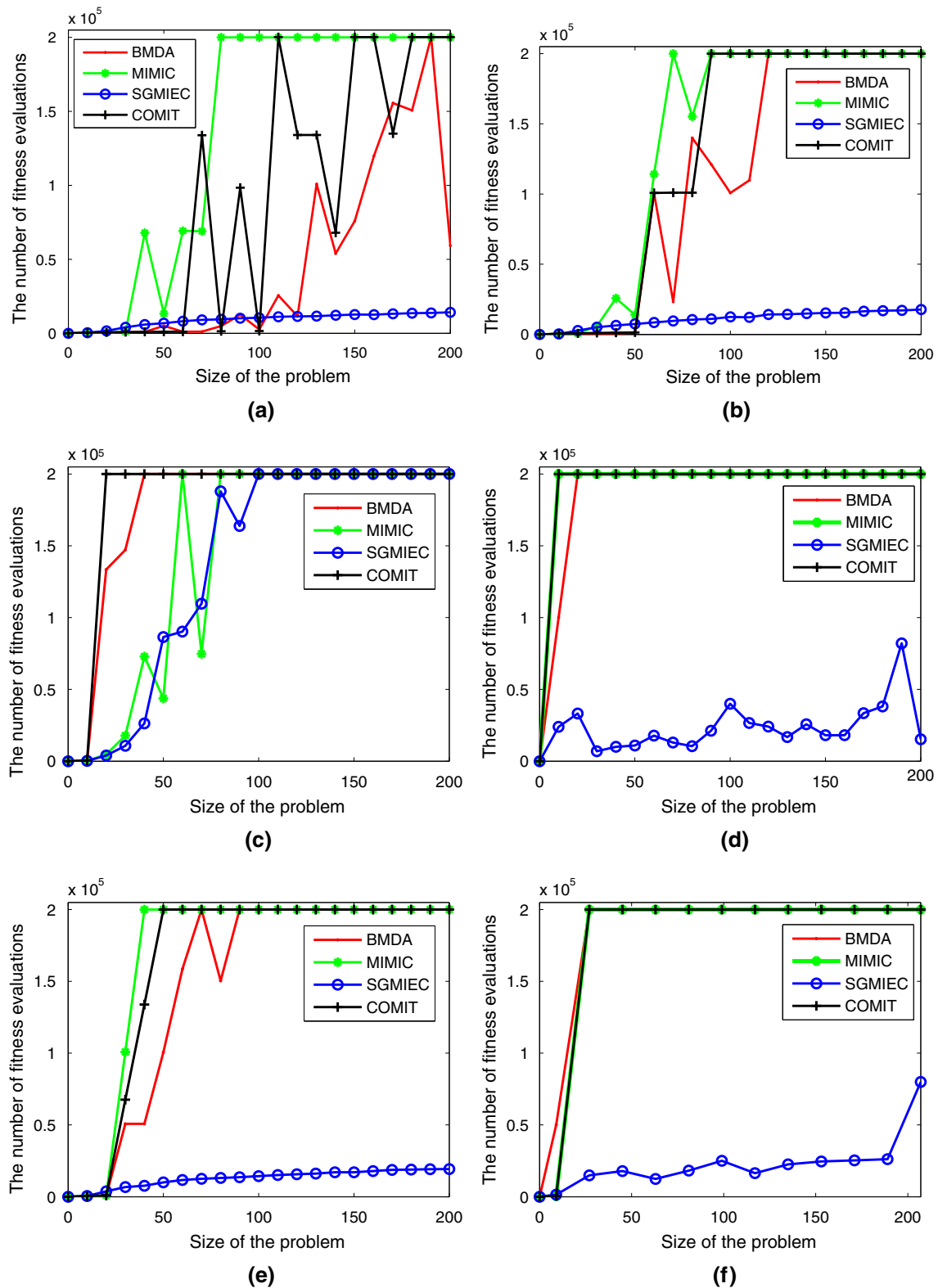
Finally, the convergence processes of BMDA, MIMIC, SGMIEC and COMIT are also tested and the results are



**Fig. 1** Convergent reliability. **a** OneMax problem, **b** weighed One-Max problem, **c** four peaks problem, **d** trap problem, **e** satisfaction problem, **f** deceptive-3 problem

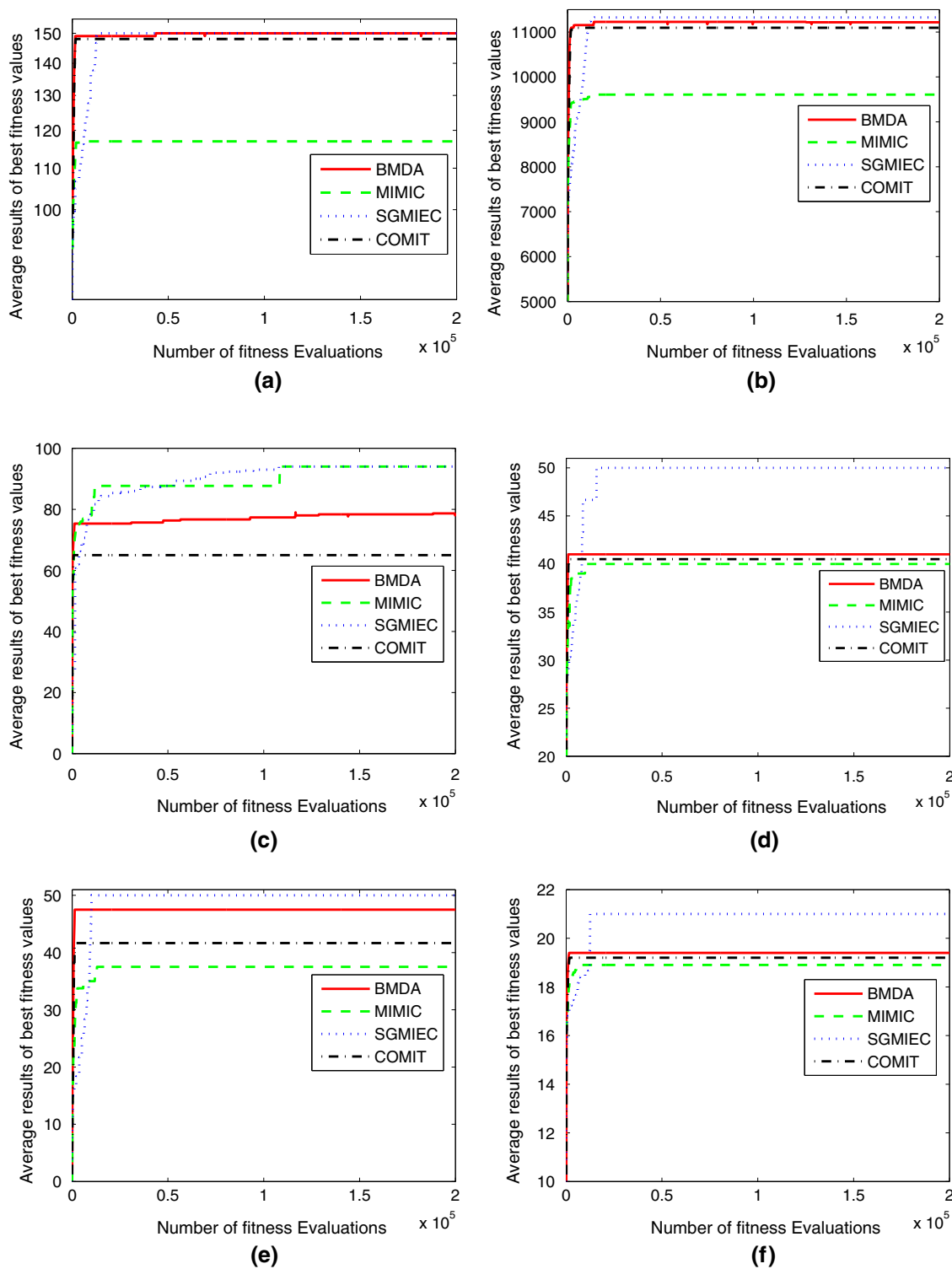
shown in Fig. 3. Although the convergence speed of SGMIEC is not always faster than BMDA, MIMIC and COMIT, it avoids getting trapped in local optima since it keeps the balance between a fast convergence and population diversity successfully. BMDA, MIMIC and COMIT often fail to solve the problems. After a quick

convergence, they plateau at a suboptimal level. A possible reason for this is that, the population becomes more and more homogeneous over time and the lack of diversity leads to models that are unable to break out from the sampling of more and more identical solutions.



**Fig. 2** Convergent velocity. **a** OneMax problem, **b** weighed One-Max problem, **c** four peaks problem, **d** trap problem, **e** satisfaction problem, **f** deceptive-3 problem





**Fig. 3** Convergent process. **a** The 150-dimensional OneMax problem, **b** the 150-dimensional weighed One-Max problem, **c** the 50-dimensional four peaks problem, **d** the 50-dimensional trap problem, **e** the 50-dimensional satisfaction problem, **f** the 63-dimensional deceptive-3 problem

## 5 Conclusion

In this paper, we have proposed a selfish gene-based approach called SGMIEC to solve the discrete optimization

problems. Based on the SG, we notice that the performance of individual is only decided by some key genes. That means if these key genes are decided, the performance of the individual can be identified. While the current EDAs

require much time in the statistic learning process as the relationships among the variables are too complicated, here we employ a mutual information and entropy based cluster model to test the impacts of the genes and an incremental learning method with resample scheme is also used in the mutual information cluster construction.

To evaluate the performance of SGMIEC, we test six benchmark problems for a range of problem size ( $n = 10$ – $200$  with step 10) and compare the results with what obtained by BMDA, COMIT and MIMIC. The experimental results presented here indicate that our SGMIEC is obviously superior than the other three methods in convergent reliability, convergent velocity as well as convergent results (solutions).

**Acknowledgments** This work was supported by the Research Project of Wuhan University under Grant 6082018.

## Appendix

The first benchmark problem is the One-Max:

$$\max \left( \sum_{i=1}^n x_i \right) \quad x_i \in \{0, 1\}$$

The second benchmark problem is the Weighed One-Max:

$$\max \left( \sum_{i=1}^n ix_i \right) \quad x_i \in \{0, 1\}$$

The third benchmark problem is the four peaks:

Given an  $N$ -dimensional input vector  $\mathbf{X}$ , the four peaks evaluation function is defined as:

$$f(\mathbf{X}, T) = \max[\text{tail}(0, \mathbf{X}), \text{head}(1, \mathbf{X})] + R(\mathbf{X}, T)$$

where

$\text{tail}(b, \mathbf{X})$  = number of trailing b's in  $\mathbf{X}$

$\text{head}(b, \mathbf{X})$  = number of trailing b's in  $\mathbf{X}$

$$R(\mathbf{X}, T) = \begin{cases} N & \text{if } \text{tail}(0, \mathbf{X}) > T \text{ and } \text{head}(1, \mathbf{X}) > T \\ 0 & \text{otherwise} \end{cases}$$

In all trails,  $T$  was set to be 10% of  $N$ , the size of the problem.

The fourth benchmark problem is the trap problem: The general  $k$ -bit trap functions are defined as

$$F_k(b_1, \dots, b_k) = \begin{cases} f_{\text{high}} & \text{if } u = k \\ f_{\text{low}} - (u \times f_{\text{low}}) / (k - 1) & \text{otherwise} \end{cases}$$

where  $b_i$  is in  $0, 1$ ,  $u = \sum_{i=1}^k b_i$  and  $f_{\text{high}} > f_{\text{low}}$ . Usually,  $f_{\text{high}}$  is set at  $k$  and  $f_{\text{low}}$  is set at  $k - 1$ . The trap functions denoted by  $F_{m \times k}$  are defined as

$$F_{m \times k}(k_1, \dots, k_m) = \sum_{i=1}^m F_k(k_i), k_i \in \{0, 1\}^k$$

The  $m$  and  $k$  are varied to produce a number of test functions. In all trails,  $k$  was set to be 5.

The fifth benchmark problem is the Satisfaction problem:

$$\max \left( \sum_{i=1}^n f(x_{5i-4}, x_{5i-3}, x_{5i-2}, x_{5i-1}, x_{5i}) \right) \quad x_i \in \{0, 1\}$$

where  $f(x_{5i-4}, x_{5i-3}, x_{5i-2}, x_{5i-1}, x_{5i})$  equals to 5 if and only if all variables equal to 1. Otherwise it equals to 0.

The six benchmark problem is the deceptive-3 problem:

$$\max \sum_{i=1}^{\frac{n}{3}} f(x_{3i-2}, x_{3i-1}, x_{3i}) \quad x_i \in \{0, 1\}$$

where

$$u = x_{3i-2} + x_{3i-1} + x_{3i}$$

and

$$f(u) = \begin{cases} 0.9 & \text{if } u = 0; \\ 0.8 & \text{if } u = 1; \\ 0.0 & \text{if } u = 2; \\ 1.0 & \text{otherwise;} \end{cases}$$

This problem is a hard deceptive problem which has a large number of local optimal solutions.

## References

- Ahn CW, Ramakrishna RS (2008) On the scalability of real-coded bayesian optimization algorithm. *IEEE Trans Evol Comput* 12(3):307–322
- Baluja S (1994) Population-based incremental learning: a method for integrating genetic search based function optimization and competitive learning. Technical Report CMU-CS-94-163, Carnegie Mellon University, Pittsburgh
- Baluja S, Davies S (1997) Using optimal dependency-trees for combinational optimization. In: *ICML '97: Proceedings of the fourteenth international conference on machine learning*, San Francisco, pp 30–38
- Baluja S, Davies S (1998) Fast probabilistic modeling for combinatorial optimization. In: *Proceedings of 15th national conference on artificial intelligence (AAAI)*, pp 469–476
- Bonet J, Isbell CL, Viola P (1997) Mimic: finding optima by estimating probability densities. In: *Advances in neural information processing systems*, vol 9. MIT Press, Cambridge, pp 424–430
- Corno F, Reorda MS, Squillero G (1998a) A new evolutionary algorithm inspired by the selfish gene theory. In: *ICEC'98: IEEE international conference on evolutionary computation*, pp 575–580
- Corno F, Reorda M, Squillero G (1998b) The selfish gene algorithm: a new evolutionary optimization strategy. In: *SAC98: 13th annual ACM symposium on applied computing*, Atlanta, pp 349–355



- Cover TM, Thomas JA (2006) Elements of information theory, 2nd edn. Wiley series in telecommunications and signal processing. Wiley, New York
- Dawkins R (1989) The selfish gene—new edition. Oxford University Press, Oxford
- Harik G (1999) Linkage learning via probabilistic modeling in the ecga. Technical report, University of Illinois at Urbana-Champaign
- Harik GR, Lobo FG, Goldberg DE (1999) The compact genetic algorithm. *IEEE Trans Evol Comput* 3(4):287–297
- Harik GR, Lobo FG, Sastry K (2006) Linkage learning via probabilistic modeling in the extended compact genetic algorithm(ecga). In: Scalable optimization via probabilistic modeling, pp 39–61
- Hong Y, Kwong S, Wang H, Xie ZH, Ren Q (2008) Svpcca: Selection on virtual population based compact genetic algorithm. In: IEEE Congress on evolutionary computation, pp 265–272
- Larranaga P, Lozano J (2002) Estimation of distribution algorithms: a new tool for evolutionary computation. Kluwer, Boston
- Muhlenbein H, Paass G (1996) From recombination of genes to the estimation of distributions i. binary parameters. In PPSN IV: Proceedings of the 4th international conference on parallel problem solving from nature, London, pp 178–187
- Pelikan M, Muhlenbein H (1998) Marginal distribution in evolutionary algorithms. In In Proceedings of the international conference on genetic algorithms Mendel'98, pp 90–95
- Pelikan M, Muhlenbein H (1999) The bivariate marginal distribution algorithm. In: Advances in soft computing: engineering design and manufacturing. Springer, London, pp 521–535
- Pelikan M, Goldberg DE, Cantu-Paz E (1999) Boa: the bayesian optimization algorithm. In Proceedings of the genetic and evolutionary computation conference (GECCO-99). Morgan Kaufmann, pp 525–532
- Yang S, Yao X (2008) Population-based incremental learning with associative memory for dynamic environments. *IEEE Trans Evol Comput* 12(5):542–561
- Yang SY, Ho SL, Ni GZ, Machado JM, Wong KF (2007) A new implementation of population based incremental learning method for optimizations in electromagnetics. *IEEE Trans Mag* 43(4):1601–1604
- Yu T-L, Goldberg DE (2004) Dependency structure matrix analysis: offline utility of the dependency structure matrix genetic algorithm. In GECCO (2), pp 355–366
- Yu T-L, Sastry K, Goldberg DE, Pelikan M (2007) Population sizing for entropy-based model building in discrete estimation of distribution algorithms. In GECCO, pp 601–608
- Zhang Q, Zhou A, Jin Y (2008) Rm-meda: A regularity model-based multiobjective estimation of distribution algorithm. *IEEE Trans Evol Comput* 12(1):41–63
- Zhou SD, Sun ZQ (2007) A survey on estimation of distribution algorithms. *Acta Autom Sin* 33(2):113–124
- Zhou A, Zhang Q, Jin Y, Sendhoff B (2008) Combination of eda and de for continuous biobjective optimization. In: IEEE congress on evolutionary computation, pp 1447–1454