Contents lists available at ScienceDirect

Neurocomputing

journal homepage: www.elsevier.com/locate/neucom

Hybrid sampling on mutual information entropy-based clustering ensembles for optimizations

Feng Wang [a,*], Cheng Yang [a], Zhiyi Lin [b], Yuanxiang Li [a], Yuan Yuan [c]

[a] State Key Lab of Software Engineering, Wuhan University, Wuhan, China
[b] Guangdong University of Technology, Guangzhou, China
[c] School of Engineering and Applied Science, Aston University, Birmingham B4 7ET, UK

ARTICLE INFO

Article history:
Received 24 May 2009
Received in revised form
1 November 2009
Accepted 20 November 2009
Communicated by D. Tao
Available online 16 December 2009

Keywords:
Estimation of distribution algorithm
Selfish gene theory
Mutual information entropy
Clustering ensembles

ABSTRACT

In this paper, we focus on the design of bivariate EDAs for discrete optimization problems and propose a new approach named HSMIEC. While the current EDAs require much time in the statistical learning process as the relationships among the variables are too complicated, we employ the Selfish gene theory (SG) in this approach, as well as a Mutual Information and Entropy based Cluster (MIEC) model is also set to optimize the probability distribution of the virtual population. This model uses a hybrid sampling method by considering both the clustering accuracy and clustering diversity and an incremental learning and resample scheme is also set to optimize the parameters of the correlations of the variables. Compared with several benchmark problems, our experimental results demonstrate that HSMIEC often performs better than some other EDAs, such as BMDA, COMIT, MIMIC and ECGA.

1. Introduction

Estimation of Distribution Algorithm (EDA) is a branch of evolutionary algorithms where classical genetic operators are replaced by the estimation of a probabilistic model and its simulation in order to generate the next population. Unlike GAs, there are neither crossover nor mutation operators in EDAs. Instead, the new population of individuals is sampled from a probability distribution which is estimated from a database formed by individuals of the former generations. 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. Recently, EDAs have attracted more and more researchers' attention, and a wide variety of EDAs using different techniques to estimate and sample the probability distribution have been proposed to solve different kinds of optimization problems [4,17,8,5,22,15,24,26,1].

In these existing algorithms, 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) [8] uses a chain model of probability distribution to estimate the pair wise conditional probabilities and sample them to generate next set of solutions. Since MIMIC deploys a greedy algorithm to search the best pair wise, the probability density function is approximately not accurate. Combining Optimizers with Mutual Information Trees (COMIT) proposed by Baluja & Davies (1997, 1998) [5,6] also uses pair-wise interaction among variables. This model which uses the tree 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. But COMIT also has some limitations, since it requires all the nodes (variables) should have father nodes except the root node in the construction of the tree process. The Bivariate Marginal Distribution Algorithm (BMDA) proposed by (Pelikan & Muhlenbein, 1999) [22] can be seen as an extension to the COMIT model which uses Pearson's chi-square statistics to detect the interaction between two variables. As BMDA uses a forest structure to denote the relationships of the variables, the space complexity of BMDA is much higher than the aforementioned algorithms. For some benchmark problems, these algorithms perform well where pairwise interaction among variable exists.

However, there are two problems in the above algorithms. On the one hand, due to the complexity of the variables, it is too difficult and time-consuming to find an appropriate probability structure model, and the above EDAs require much time in the statistic learning process of the pairwise interaction of variables. It definitely reduces the performances of the algorithm and becomes worse while the problem size is increasing. On the other hand, the current probability structure model usually concentrates on the parameter learning, but ignoring the relationships of the variables. The probability structure model is often fixed while

E-mail addresses: fengwang@whu.edu.cn (F. Wang).
yuany1@aston.ac.uk (Y. Yuan).

0925-2312/$ - see front matter © 2009 Elsevier B.V. All rights reserved.
doi:10.1016/j.neucom.2009.11.011

the parameters evolving. As a result of this, it requires more time to get a better solution for this one-side evolution.

In order to improve the performance of the bivariate EDAs, we firstly exploit the Selfish gene theory and propose a Mutual Information Entropy based approach to evaluate the relationships among the variables and construct a basic probability cluster. Then, we propose a hybrid sampling method to construct the sampling clusters by taking both the accuracy and diversity of the solutions into account. We named this method HSMIEC, and it samples in the virtual population where the genes (variables) are constructed as a tree model by their mutual information entropy. Further more, we deploy an incremental learning scheme to train the parameters which can help to decrease the algorithm complexity effectively. As a result of this, this approach can obviously decrease the time to construct a mutual information tree and get better results than the previous approaches.

The remainder of this paper is organized as follows. Section 2 introduces the Mutual Information Entropy-based clustering construction with Selfish gene theory, and gives a description of the incremental learning scheme for the parameter learning in HSMIEC. Section 3 goes into details of describing the hybrid sampling approach. Experimental results and their analysis are given in Section 4. Section 5 concludes the paper.

2. Mutual information and entropy based cluster construction

The Selfish gene theory is proposed by R. Dawkins which gives a different view on the evolution [23]. In this theory, the population can be regarded 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 good genes can survive in the evolution process. Fulvio Corno and his cooperators followed this idea and proposed a new evolutionary optimization strategy called Selfish gene theory (SG) [13].

In SG, the population is like a storeroom of genes which named 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.

Since 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 approach which based on the selfish gene theory and mutual information cluster.

As showen in Fig. 1, firstly, some initialization work should be done by initializing the frequencies of each locus in the genome. Then select N individuals according to selfish gene-based selection (more details are given in Section 2.1) on virtual population. For each individual, calculate the mutual information and entropy and construct some clusters. And an incremental learning scheme is employed to accelerate the construction of the mutual information clusters. Finally, regenerate new clusters according to the hybrid sampling scheme which can balance the accuracy and diversity of the solutions in a better way.

2.1. Selfish gene-based selection on virtual population

In SG, the individuals are stored with genes in a virtual population and can be selected after sampling by the density function P. As the Selfish gene theory 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 [13,10]. 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.

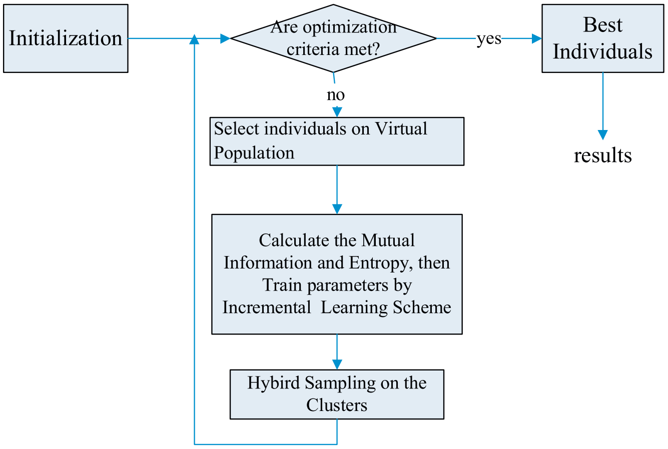
Initialization
Are optimization criteria met?
yes
Best Individuals
no
results
Select individuals on Virtual Population
Calculate the Mutual Information and Entropy, then Train parameters by Incremental Learning Scheme
Hybird Sampling on the Clusters

Fig. 1. Virtual population based incremental learning approach for optimization using Selfish gene theory.

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 $A_l=(a_{l1},a_{l2},\ldots,a_{ln_l})$ and the marginal probabilities can be represented as $P_l=(p_{l1},p_{l2},\ldots,p_{ln_l})$. Therefore, the virtual population can be statistically characterized as $P=(P_1,P_2,\ldots,P_m)$. 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 every time.

2.2. 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 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 [11]. Here we use a mutual information and entropy based cluster to represent the virtual population. We also employed the clusters to curve and the relationships of the variables, which is similar to ECGA [16]. 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 $P(x_1,x_2,\ldots,x_N)=(P_1,P_2,\ldots,P_m)$ over individuals of length N, where $x_1,\ldots,x_N$ are variables corresponding to the values of the bits, now we want to establish a model which satisfies,

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

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

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

We set a correlation measurement parameter ρ 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*

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

Proof. Since x_i and x_j are identically distributed and

$$\rho=1\text{-}\frac{H(x_2|x_1)}{H(x_1)}$$

then,

$$\rho=\frac{H(x_1)\text{-}H(x_2|x_1)}{H(x_1)}=\frac{H(x_2)\text{-}H(x_2|x_1)}{H(x_1)}(\text{since}H(x_1)=H(x_2))=\frac{I(x_1;x_2)}{H(x_1)}\qquad\square$$

2.3. Incremental learning on mutual information and entropy based cluster

In order to accelerate the construction of the mutual information cluster, HSMIEC deploys an incremental learning scheme in cluster construction. In Baluja and Davies's work on COMIT [5], they chose the best S individuals generated from the dependency tree in each iteration and calculated the statistics of probability distributions in the S 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 HSMIEC, as we use a selfish gene based selection strategy and a hybrid sampling method to choose appropriate cluster ensembles, 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 on its parameter training. Here we use the incremental learning scheme which can help to decrease the time cost and improve the performance finally.

As 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 entropy cluster construction scheme is set as follows.

Algorithm 1. 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 $I(t)=I*(1\text{-}\alpha)+\alpha*I(t\text{-}1)$, α is Incremental Learning factor.
4. Calculate the correlation measurement parameter ρ between every two variables.
5. Generate new cluster by the new correlation measurement parameter.
6. Generate new virtual population according to the new cluster.

Since learning models are always formulated as optimization problems, the learning parameter setting and training have become an important research issue in a learning model which can directly affect the solutions of the optimization problem [27,19,28]. In order to avoid premature convergence, the incremental learning factor α, which denotes how much history information will be used in the next iteration decided by the effects of the history information, usually sets to 0.1. 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.

3. Hybrid sampling on clustering ensembles

Recently, cluster ensembles have emerged as a technique for overcoming problems with clustering algorithms [29,30,7,14,3,2,9,12,20,21]. It is well known that off-the-shelf clustering methods may discover different patterns in a given set of data. This is because each clustering algorithm has its own bias resulting by the optimization of different criteria. It has been noted that the most crucial factor of clustering ensembles is to construct an accurate and diverse ensemble committee of the clustering ensembles [25,18]. Here we follow this idea and try to find an appropriate cluster which can represent the current probability structure of the variables.

As mentioned above, we have employed the mutual information and entropy based clusters to represent the relationships of the variables. In order to improve the robustness of putative clusters to sampling variability, a hybrid sampling method is proposed which evaluates the qualities of all obtained clustering results by considering both the accuracy and diversity of ensemble committees and then chooses part of promising clustering results to regenerate clusters and build the ensemble committees.

Suppose the cluster ensemble X_i has n ensemble committees, and the clustering accuracy can be defined as

$$Accuracy(X_i)=\frac{\sum_{j=1}^{n}I(X_{ij},X_{ir_j})/H(X_{ij})}{M}$$

where M is the number of correlations of the variables, and r_j represents the father of node j in mutual information tree.

Here the accuracy is a normalized parameter for information measurement. For bivariate EDAs, the goal is to minimize $\sum_{j=1}^{n}I(X_{ij,}X_{ir_j})/H(X_{ij})$. If the value of the above formula is smaller, the accuracy of the algorithm is better [8]. Since $\sum_{j=1}^{n}I(X_{ij,}X_{ir_j})/H(X_{ij})$ belongs to [0, 1], we need to do some normalization and make it divided by M. As mentioned above, M is the number of correlations of the variables, here it can also be regarded as the number of edges in the clusters.

The clustering diversity can also be set as follows:

$$Diversity(X_i)=\sum_{i,j=1;j\neq i}^{N}\sum_{m=1}^{L}\frac{f(X_{im},X_{jm})}{N\text{-}1}$$

$$f(X_{im},X_{jm})=\begin{cases}\dfrac{1.0}{L(L\text{-}1)} & \text{if } X_{im}!=X_{jm}\\ 0 & \text{else}\end{cases}$$

where L is the number of the variables, and N is the length of the cluster committee.

The diversity can be evaluated by the differences of the values in the same bit for different individuals. For example, X and Y are two individual in the population. Suppose X is represented as 11111 and Y is represented as 01011. For X and Y, the values of the first bit and the third bit are different, since the length of the chromosome is $L=5$, the contribution of these two individuals for the diversity of the population is 0.1.

For each cluster ensemble, the accuracy function and the diversity function are normalized in HSMIEC. So the process of constructing a good ensemble committee can be viewed as a two-objective optimization problem which can be set as,

$$Object(X_i)=\beta Accuracy(X_i)\text{-}(1\text{-}\beta)Diversity(X_i)$$

where β is the weight in a range of (0,1). Accuracy and diversity are two important issues for the design of an algorithm. From this formula, the optimization goal is to minimize the function $Object(X_i)$ so that if the diversity is minimum, the accuracy can get its maximum value.

After calculating the accuracies and diversities of the cluster ensembles, good clusters can be selected to construct suitable mutual information tree. As we mentioned above, it costs much time to get a better solution if the cluster sampling and parameter training are proceeded, respectively. Here, the process of cluster sampling is synchronous with the process of the parameter learning.

4. Experimental results and analysis

To evaluate the performance of HSMIEC, we test four benchmark problems for a range of problem size ($n=10$ to 200 with step 10 for three problems, and $n=9$ to 210 with step 18 for Deceptive-3 problem) and compare the results with what obtained by BMDA, COMIT MIMIC and ECGA. For ECGA, the population size is set to 1000, the crossover probability is set as 1 and the tournament size is set as 16. 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, BMDA.

As we mentioned above, the incremental learning factor α denotes how much history information will be used in the next iteration decided by the effects of the history information, if α is too large, then the differences of the individuals would be too small which would make the algorithm get trapped in premature convergence. β is a weight parameter that balances the diversity and accuracy of the clusters. It also directly affects also directly affects the performance of the algorithm. The parameter setting is a challengeable issue, in our approach, the incremental learning factor $\alpha=0.1$ and the weight parameter $\beta=0.5$ which are set by reiterative experiments.

All parameters hold constant for all the runs. For each problem and problem size, 20 successful independent runs are collected. Each run is terminated either when the global optimum is found or when the number of evaluations reaches the limit(200,000).

4.1. One-max function

This fitness function is actually a simple linear function over the single bits with all coefficients equal to 1, which means it is just the sum of all bits in a string.

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

where x_i is the value on the i-th position in string x. Onemax fitness function does not have a permutation as an input parameter because its value is the same for any permutation of bits in an input string.

Fig. 2 shows the convergent reliability of BMDA, MIMIC, COMIT, ECGA and HSMIEC under different numbers of variables, as well as Fig. 3 shows the convergent velocity. Fig. 4 shows the convergent process of the above algorithms.

4.2. Quadratic function

The quadratic fitness function used for comparisons in this section is defined as

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

where

$$f(u,v)=0.9\text{-}0.9(u+v)+1.9uv$$

With both arguments equal to zero we get $f_2(0,0)=0.9$. With different arguments we get $f_2(0,1)=f_2(1,0)=0$. With both

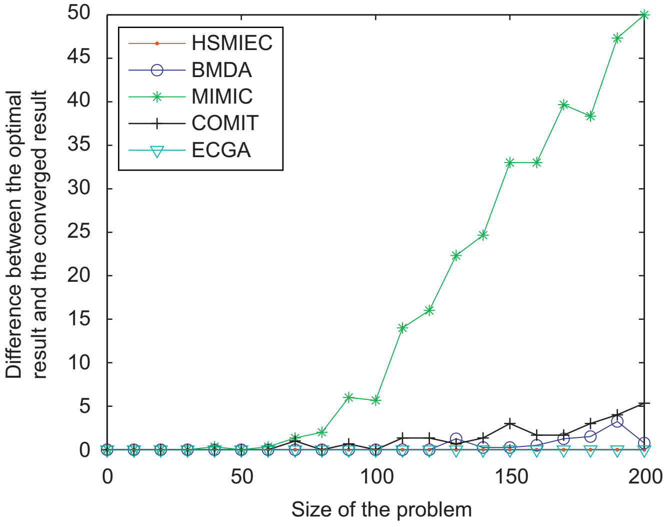

Fig. 2. Convergent reliability on Onemax function.

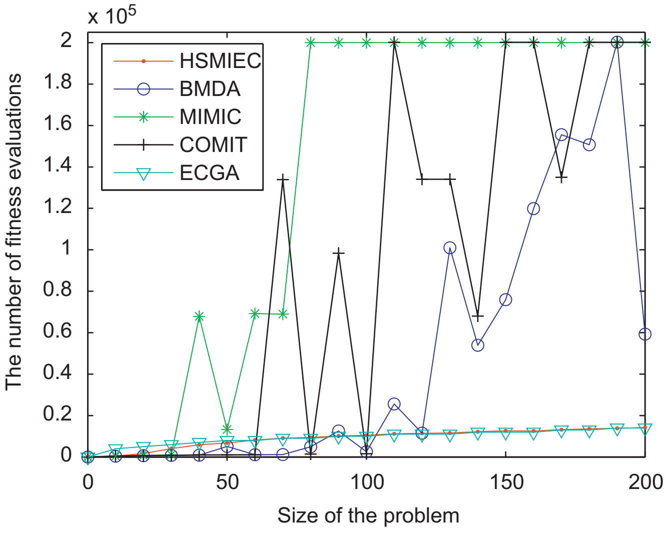

Fig. 3. Convergent velocity on Onemax Function.

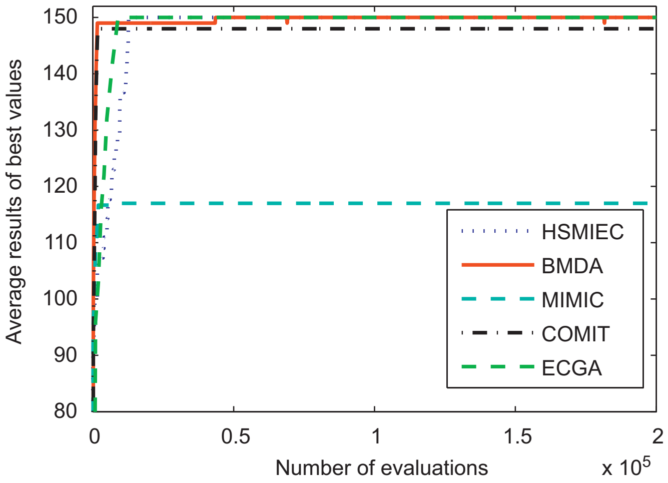

Fig. 4. Convergent process on Onemax Function.

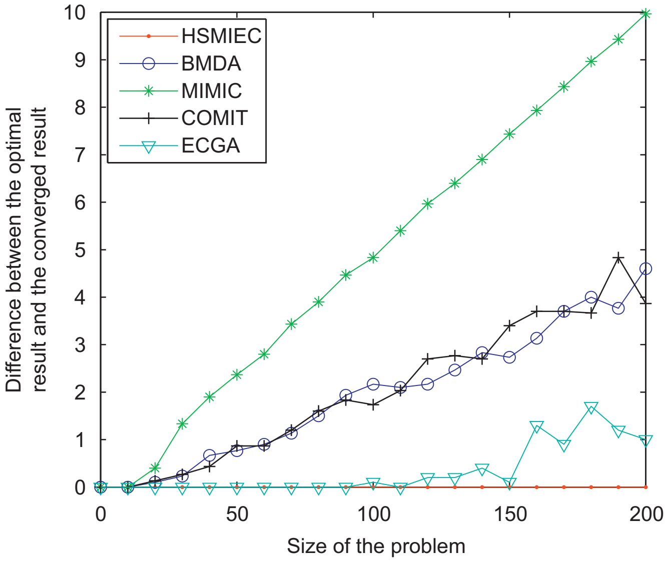

Fig. 5. Convergent reliability on Quadratic Function.

arguments equal to one we get $f_2(1,1) = 1$. The optimum is clearly in the string with ones on all positions.

The experimental results on Quadratic function are shown in Figs. 5, 6 and 7, respectively.

4.3. Deceptive-3 function

Deceptive function is often used for comparisons of different optimization methods for its being deceptive. With deceptive problems, the average fitness of low order schemata present in optimum is lower than the average fitness of alternative ones. The fitness function is defined as,

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

where

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

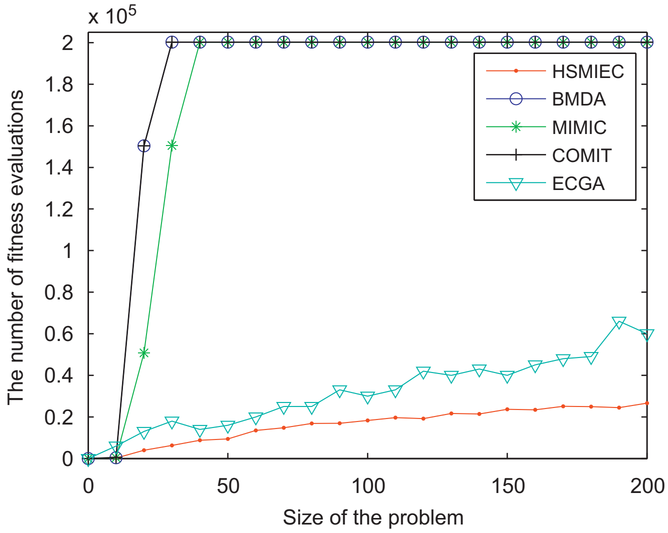

Fig. 6. Convergent velocity on Quadratic Function.

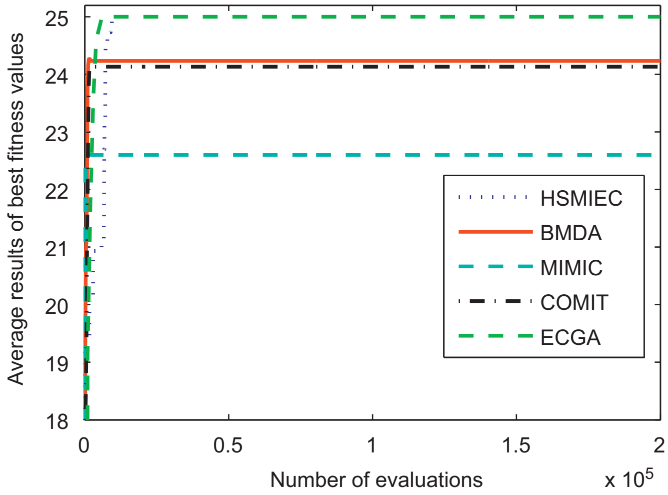

Fig. 7. Convergent process on Quadratic Function.

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. The input string is first partitioned into independent groups of 3 bits each. And this partitioning is unknown to the algorithm and it does not change during the run. The deceptive-3 function has one global optimum for all ones and a deceptive attractive to all zeros. The corresponding experimental results on this problem are also displayed in Figs. 8, 9 and 10.

4.4. Trap function

The general k-bit trap functions are defined as

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

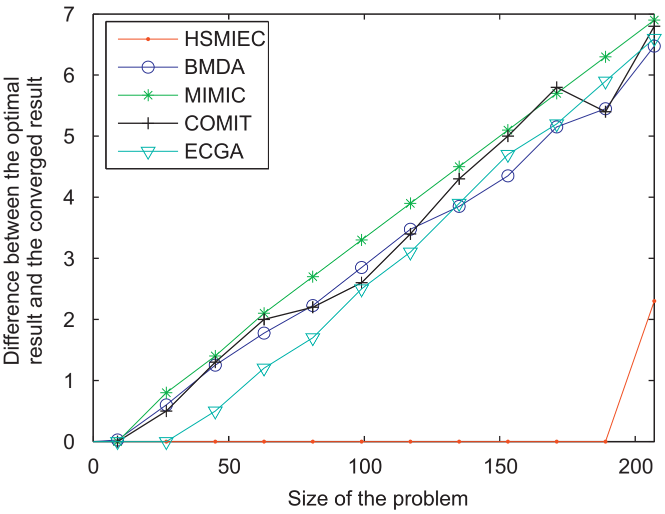

Fig. 8. Convergent reliability on Deceptive Function.

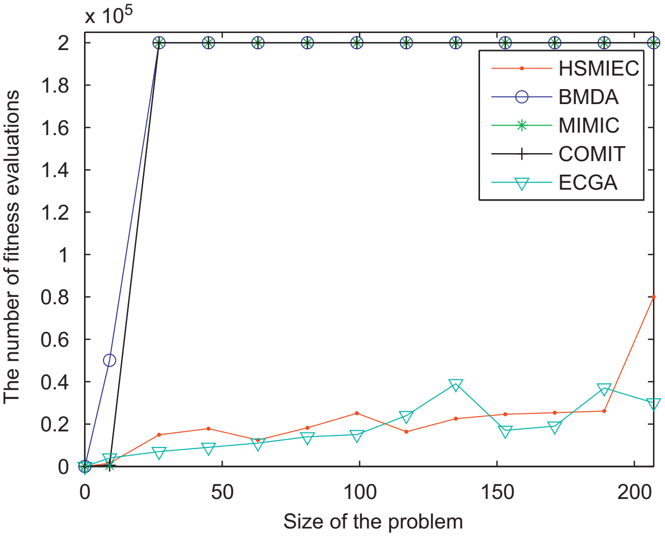

Fig. 9. Convergent velocity on Deceptive Function.

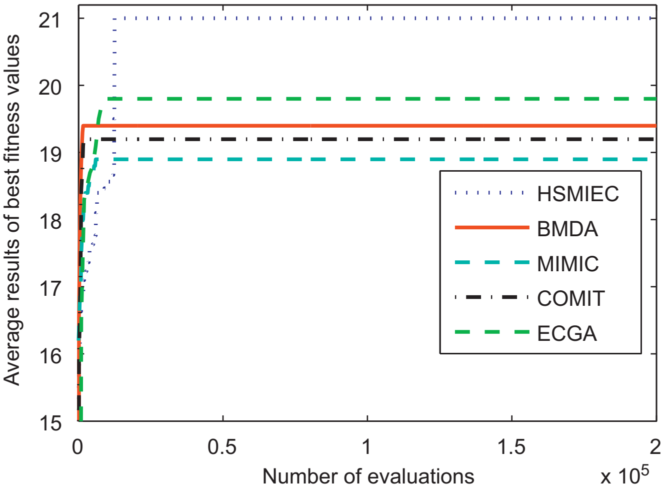

Fig. 10. Convergent process on Deceptive Function.

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

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

The m and k are varied to produce a number of test functions. In all trails, k was set to 5. The Trap function fool the gradient-based optimizers to favor zeros, but the optimal solution is composed of all ones. We also do the same experiments by using four different algorithm to test the solutions on the Trap function. The following Figs. 11, 12 and 13 shows the experiment results.

4.5. Results analysis

From the above figures, we can observe that the performance of BMDA, MIMIC, COMIT and ECGA are not satisfied for all tested four benchmark functions. For example, for the Onemax function, the convergent curves of BMDA, MIMIC, and COMIT show great fluctuation, which means inferior stability.

The results also demonstrate clearly the stability of HSMIEC because the quality of its solutions is averagely superior to that of

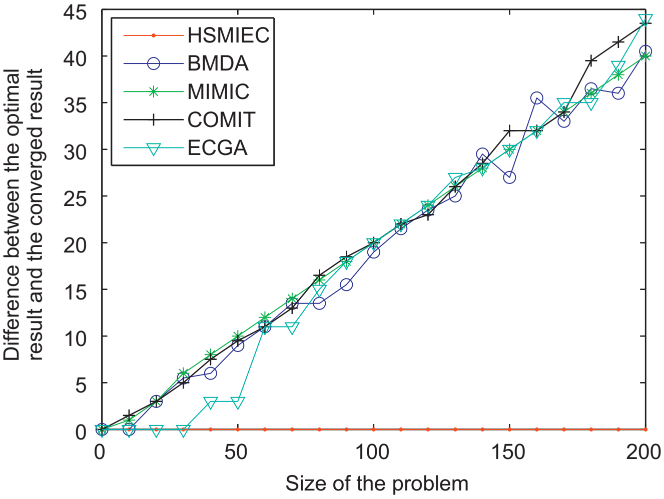

Fig. 11. Convergent reliability on Trap Function.

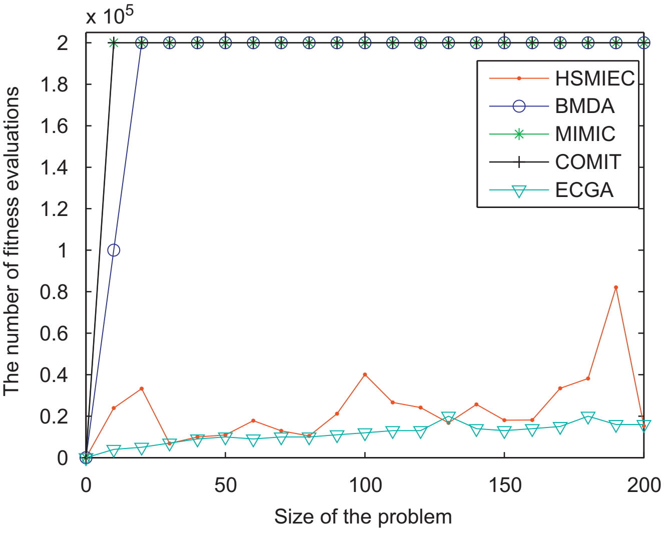

Fig. 12. Convergent velocity on Trap Function.

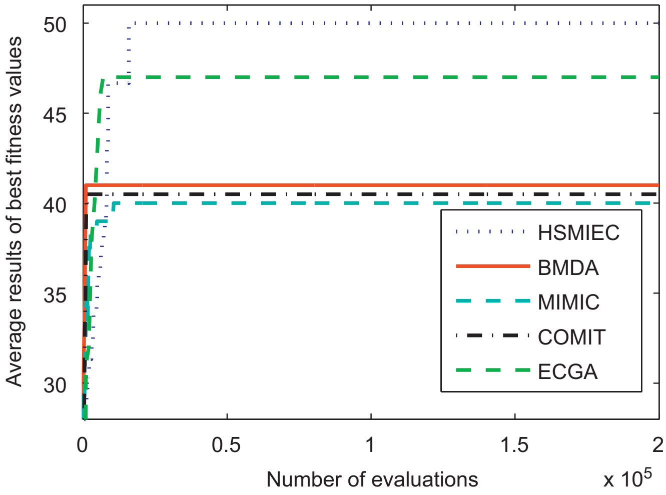

Fig. 13. Convergent process on Trap Function.

the others. For the Onemax function, both ECGA and HSMIEC can all find good solutions with different problem sizes. But for the other three benchmark functions, HSMIEC is much better than ECGA. For Quadratic function, HSMIEC performs much better while the problem size increases. Experimental results for Trap function of order 5 and Deceptive-3 function show that HSMIEC is able to find the global optima for this problem, while BMDA, MIMIC, COMIT and ECGA cannot find the global optima if the size of problem is more than 20.

Further more, we can also see that HSMIEC often requires much less fitness evaluations to converge when compared to BMDA, MIMIC and COMIT. 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). But compared with ECGA, the results revealed that the convergence velocity of HSMIEC is not always the fast. For Onemax function, the needed evaluations for HSMIEC is mostly same as what for ECGA. But for Quadratic function, HSMIEC requires much less evaluations than ECGA with the problem size increases. For Trap function, the evaluation times for ECGA is less than HSMIEC. It is because that ECGA used greedy algorithm to search the MDL (minimum description model). For some separable computing problems, since there is no overlap among the sub-clusters, ECGA converged faster than other EDAs.

In addition, these results also indicate that HSMIEC can avoid getting trapped in local optima since it keeps the effectively balance between a fast convergence and population diversity. For Onemax and Quadratic function, both HSMIEC and ECGA can escape from local optima while for Deceptive-3 and Trap function, only HSMIEC can do that. 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.

In general, the experimental results presented here indicate that HSMIEC is obviously superior than the other methods. The reasons of the good performance of HSMIEC could be states as follows. First of all, the Selfish gene theory guarantees that all the information we used in the virtual population construction is effective for individuals, and the incremental learning scheme can speed up the cluster construction process which help to improve the efficiency of this algorithm. Secondly, the hybrid sampling scheme ensure the diversity of individuals which are very helpful to avoid local optima. But the cluster construction process is still time consuming, if the problem size and the variable numbers increase, the evaluation times of HSMIEC will increase and the performance will decrease quickly.

5. Conclusion

The main feature of EDAs is the approximation of the probability density function of the decision variables. In this paper, a selfish gene based approach HSMIEC was proposed to solve the discrete optimization problems. Based on the Selfish gene theory, we noticed 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 by considering both the accuracy and diversity of the cluster ensembles, we deploy a hybrid sampling method to choose the most suitable cluster in the probability model. And an incremental learning method with resample scheme is also used in the mutual information cluster construction.

Experimental results show that our HSMIEC performs better than some current bivariate EDAs, such as MIMIC, COMIT and BMDA. And compared with some multi-variate EDAs, such as ECGA, HSMIEC can also obtain better solutions than ECGA in convergent reliability. However, the advantage of HSMIEC is not so obvious as in the convergence velocity comparison. With the problem size increases, the convergence velocity of HSMIEC is significantly faster than MIMIC, COMIT and BMDA, but for some separable computing problems, if there is no overlap among the sub-clusters, ECGA performs a little better than HSMIEC. Furthermore, if the number of variables increases and the relationships among them become more complex, the current HSMIEC might not be suitable to solve those kind of problems. And the parameter setting problem for α and β is also a very important and challengeable issue. In this paper, we set the values according to multi-experiment comparison which is not convincing enough. In the next step, we will devote to exploit if there exists some theoretical basis on the choices of the values. And how to improve the current method and make it suitable for multi-variate optimizations is another issue that we will study in our future work.

References

[1] C.W. Ahn, R.S. Ramakrishna, On the scalability of real-coded bayesian optimization algorithm, IEEE Transactions on Evolutionary Computation 12 (3) (2008) 307–322.

[2] S. Alexander, G. Joydeep, Cluster ensembles—a knowledge reuse framework for combining multiple partitions, Journal of Machine Learning Research 3 (2002) 583–617.

[3] S. Alexander, Relationship-based clustering and cluster ensembles for high-dimensional data mining, Ph.D. Thesis, The University of Texas at Austin, 2002.

[4] S. Baluja, 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, PA, USA, 1994.

[5] S. Baluja, S. Davies, Using optimal dependency-trees for combinational optimization, in: ICML '97: Proceedings of the 14th International Conference on Machine Learning, San Francisco, CA, USA, 1997, pp. 30–38.

[6] S. Baluja, S. Davies, Fast probabilistic modeling for combinatorial optimization, in: Proceedings of 15th National Conference on Artificial Intelligence (AAAI), 1998, pp. 469–476.

[7] F. Bernd, M. Buhmann Joachim, Bagging for path-based clustering, IEEE Transaction on Pattern Analysis and Machine Intelligence 25 (2003) 1411–1415.

[8] J. Bonet, C.L. Isbell, P. Viola, Mimic: finding optima by estimating probability densities, in: Advances in Neural Information Processing Systems, vol. 9, MIT Press, Cambridge, MA, 1997, pp. 424–430.

[9] R. Cai, Z. Hao, X. Yang, W. Wen, An efficient gene selection algorithm based on mutual information, Neurocomputing 72 (4–6) (2009) 991–999.

[10] F. Corno, M.S. Reorda, G. Squillero, A new evolutionary algorithm inspired by the Selfish gene theory, in: ICEC'98: IEEE International Conference on Evolutionary Computation, 1998, pp. 575–580.

[11] T.M. Cover, J.A. Thomas, Elements of Information Theory, in: Wiley Series in Telecommunications and Signal Processing, second ed., Wiley-Interscience, New York, 2006.

[12] K. Faceli, C.P. Marcilio deSouto, D. Jo, A.C.P.L.F. de Carvalho, Multi-objective clustering ensemble for gene expression data analysis, Neurocomputing 72 (2009) 2763–2774.

[13] F. Corno, M. Reorda, G. Squillero, The selfish gene algorithm: a new evolutionary optimization strategy, in: SAC98: 13th Annual ACM Symposium on Applied Computing, Atlanta, Georgia, USA, 1998, pp. 349–355.

[14] A.L. Fred, A.K. Jain, Combining multiple clusterings using evidence accumulation, IEEE Transaction on Pattern Analysis and Machine Intelligence 27 (2005) 835–850.

[15] G.R. Harik, F.G. Lobo, D.E. Goldberg, The compact genetic algorithm, IEEE Transactions on Evolutionary Computation 3 (4) (1999) 287–297.

[16] G. Harik, Linkage learning via probabilistic modeling in the ecga, Technical Report, University of Illinois at Urbana-Champaign, 1999.

[17] H. Muhlenbein, G. Paass, From recombination of genes to the estimation of distributions i. binary parameters, in: PPSN IV: Proceedings of the Fourth International Conference on Parallel Problem Solving from Nature, London, UK, 1996, pp. 178–187.

[18] Y. Hong, S. Kwong, H. Wang, Q. Ren, Resampling-based selective clustering ensembles, Pattern Recognition Letters 30 (2009) 298–305.

[19] J. Li, N.M. Allinson, D. Tao, X. Li, Multitraining support vector machine for image retrieval, IEEE Transactions on Image Processing 15 (11) (2006) 3597–3601.

[20] V. Moschou, D. Ververidis, C. Kotropoulos, Assessment of self-organizing map variants for clustering with application to redistribution of emotional speech patterns, Neurocomputing 71 (1–3) (2007) 147–156.

[21] I. Partalas, G. Tsoumakas, I.P. Vlahavas, Pruning an ensemble of classifiers via reinforcement learning, Neurocomputing 72 (7–9) (2009) 1900–1909.

[22] M. Pelikan, H. Muhlenbein, The bivariate marginal distribution algorithm, in: Advances in Soft Computing: Engineering Design and Manufacturing, Springer, London, 1999, pp. 521–535.

[23] R. Dawkins, The Selfish Gene—New Edition, Oxford University Press, Oxford, 1989.

[24] G.R. Harik, F.G. Lobo, K. Sastry, Linkage learning via probabilistic modeling in the extended compact genetic algorithm (ecga), Scalable Optimization via Probabilistic Modeling, 2006, pp. 39–61.

[25] M. Stefano, T. Pablo, M. Jill, G. Todo, Consensus clustering: a resampling-based method for class discovery and visualization of gene expression moscovery data, Machine Learning 52 (2003) 91–118.

[26] S.Y. Yang, S.L. Ho, G.Z. Ni, J.M. Machado, K.F. Wong, A new implementation of population based incremental learning method for optimizations in electromagnetics, IEEE Transactions on Magnetics 43 (4) (2007) 1601–1604.

[27] D. Tao, X. Li, X. Wu, W. Hu, S.J. Maybank, Supervised tensor learning, Knowledge and Information Systems 13 (1) (2007) 1–42.

[28] D. Tao, X. Li, X. Wu, S.J. Maybank, Geometric mean for subspace selection, IEEE Transactions on Pattern Analysis and Machine Intelligence 31 (2) (2009) 260–274.

[29] F.X. Zhang, B.C.E. Random projection for high dimensional data clustering, in: Proceedings of International Conference on Machine Learning, 2003, pp. 186–193.

[30] F.X. Zhang, B.C.E. Clustering ensembles for high dimensional clustering: an empirical study, Technical Report, Oregon State University, 2006.

Feng Wang received the M.S. and Ph.D. degree in Computer Science in 2005 and 2008, respectively, both from Wuhan University, China. She is a faculty member of the State Key Lab of Software Engineering of Wuhan University since 2008. Her research interests include evolutionary computation, intelligent information retrieval, evolvable hardware and complex system. She serves as a reviewer for several IEEE transactions, other international journals and conferences. She is a member of IEEE and ACM.

Cheng Yang was born in Wuhan, China, on February 04, 1985. He received the M.S. degree in Computer Science and Technology from Wuhan University of Technology, Wuhan, China, in 2007 and the B.S. degree in Computer Software and Theory from State Key Laboratory of Software Engineering, Wuhan University, Wuhan, China, in 2009. Currently, he is an Engineer at Alibaba Cloud Computing Research Institute. His research interests include evolutionary computation, pattern recognition, data mining, and search engine.

Zhiyi Lin was born in Fujian, China, on November 10, 1979. He received the B.S. degree in Computer Science and Technology from Wuhan University of Technology, Wuhan, China, in 2006 and the Ph.D. degree in Computer Software and theory from State Key Laboratory of Software Engineering, Wuhan University, Wuhan, China, in 2009. Currently, he is a Teacher at Faculty of Computer, Guangdong University of Technology, Guangzhou, China. His research interests include evolutionary computation, Pattern recognition, and data mining.

Yuanxiang Li is currently a Professor with the State Key Lab of Software Engineering, Wuhan University, China. He received his Ph.D. degree from Wuhan University. His research interests include evolutionary computation, parallel computing, and evolvable hardware.

Yuan Yuan is currently a Lecturer with the School of Engineering and Applied Science, Aston University, United Kingdom. She received her B.Eng. degree from the University of Science and Technology of China, China, and Ph.D. degree from the University of Bath, United Kingdom. She has sixty scientific publications in journals and conferences on visual information processing, compression, retrieval etc. She is an associate editor of International Journal of Image and Graphics (World Scientific), an editorial board member of Journal of Multimedia (Academy Publisher), and a guest editor of a special issue of Signal Processing (Elsevier). She was a chair of some conference sessions, and a member of program committees of many IEEE/ACM conferences. She is a reviewer for several IEEE transactions, other international journals and conferences. She is a senior member of the IEEE and the IEEE Signal Processing Society.