A Two-level Hierarchical EDA using Conjugate Priori

Bo Wang, Hua Xu, Yuan Yuan
State Key Laboratory of Intelligent Technology and Systems
Tsinghua National Laboratory for Information Science and Technology
Department of Computer Science and Technology, Tsinghua University, Beijing 100084, China
wbthu@outlook.com, xuhua@tsinghua.edu.cn, yyxhdy@gmail.com

ABSTRACT

Estimation of distribution algorithms (EDAs) are stochastic optimization methods that guide the search by building and sampling probabilistic models. Inspired by Bayesian inference, we proposed a two-level hierarchical model based on beta distribution. Beta distribution is the conjugate priori for binomial distribution. Besides, we introduced a learning rate function into the framework to control the model update. To demonstrate the effectiveness and applicability of our proposed algorithm, experiments are carried out on the 01-knapsack problems. Experimental results show that the proposed algorithm outperforms cGA, PBIL and QEA.

Categories and Subject Descriptors

I.2.8 [**ARTIFICIAL INTELLIGENCE**]: Problem Solving, Control Methods, and Search;
G.1.6 [**NUMERICAL ANALYSIS**]: Optimization

Keywords

Artificial intelligence, Combinatorial optimization, Empirical study

1. INTRODUCTION

In evolutionary computation community, researchers have proposed a kind of algorithms named estimation the distribution algorithms (EDAs), also called probabilistic model-building genetic algorithms and iterated density estimation evolutionary algorithms [4]. EDAs evolve an explicit probabilistic model to guide the search. In EDAs, the exploration and exploitation are fundamental problems just like other evolutionary algorithms (EAs)[1]. Exploration is the process of searching the new regions, while exploitation is the process of visiting the neighborhood of previously visited points. Excessive exploration results in unsatisfactory solution and excessive exploitation leads population into local optima. Hence an evolutionary algorithm needs to balance these two antagonistic cornerstones to get superior performance.

Inspired by Bayesian inference, we proposed a two-level hierarchical EDA (THEDA) adopting beta distribution. We also proposed a learning rate function alongside the update of the two-level model to balance the exploration and exploitation explicitly. At the beginning of the searching, the learning rate is small, so to promote exploration. With the increase of the number of generations, learning rate become larger and algorithm tend to exploitation.

We provide in section 2 a description of the proposed algorithm. Experimental results on 0-1 knapsack problems are in section 3. Finally, section 4 concludes with a summary and a discussion of future work.

2. A TWO-LEVEL HIERARCHICAL EDA

The framework of THEDA is basically the same with other EDAs except that THEDA uses a two-level model. Algorithm 1 gives the pseudocode of the THEDA.

Algorithm 1 Framework of THEDA

$t \leftarrow 0$
initialize the probability vector PV
while termination criteria not met **do**
  $P \leftarrow$ generate individuals by sampling PV
  $S \leftarrow$ select from P according to their fitness
  $betaV \leftarrow updateModel(S, t)$
  $PV \leftarrow$ generate by sampling $betaV$
  $t \leftarrow t + 1$
end while
procedure UPDATEMODEL(S, t)
  for $i = 1, 2, \ldots, l$ **do**
    $c_1 \leftarrow$ number of 1s in the ith dimension of S
    $c_0 \leftarrow$ number of 0s in the ith dimension of S
    $(betaV_i).\alpha \leftarrow 1 + c_1 \times f_L(t)$
    $(betaV_i).\beta \leftarrow 1 + c_0 \times f_L(t)$
  end for
  return $betaV$
end procedure

The first level is from beta vector (βV) to probability vector (PV) and the second level is from PV to population consisted of solutions. In our approach, the selected individuals are used to estimate a βV. The βV can generate the PV from which the individuals are sampled. The βV is an array of pairs. Each pair (e.g. α_i, β_i) includes two numbers defined on $[1, +\infty)$ and it determines a beta

GECCO'14, July 12–16, 2014, Vancouver, BC, Canada.
ACM 978-1-4503-2881-4/14/07.
http://dx.doi.org/10.1145/2598394.2598470.

distribution. Therefore, a βV represents an array of beta distribution. We limited the domain of numbers in beta vectors to be on the interval $[1, +\infty)$ in spite of that, parameters of beta distribution are numbers greater or equal to 1. This restriction ensures that the mode does not occurs at one or both ends. Beta distribution has only two parameters which have intuitive significance. In Bayesian inference, α and β of beta distribution represent the number of $1s$ and $0s$ in observed data respectively. Therefore, the update of beta vector just needs a count of the number of $0s$ and $1s$ for each position. This is computationally efficient. We designed a learning rate function for controlling the search. A learning rate function is defined as a simple linear function as $f_L(t) = c \times t$, where c is the coefficient determining the learning rate and t is the number of generations.

3. EXPERIMENTAL RESULTS

This section reports the experimental results of the proposed algorithm on 01-knapsack problem comparing with cGA, PBIL and QEA. The 01-knapsack problems is a well-known combinatorial optimization problem [3]. The data are generated as follows: w_i = uniformly random$[1, 10]$, $p_i = w_i + 5$, where w_i and p_i are the weight and the value of the ith item respectively. Two types of capacity were used, one is the half of the total weight of all items ($C_h = \frac{1}{2}\sum_{i=1}^{N} w_i$), and the other is a fixed value, that is $C_f = 20$. To satisfy the constraint, a random repair operator were used, see the details in [2]. Algorithms were tested on 01-knapsack problems of size 250, 500, 1000, 2000. For each size, two capacities mentioned above were used. These instances were denoted by $P_{knp\_D\_Ch}$, $P_{knp\_D\_Cf}$ where D should be replaced by the size of the problem. Algorithms were tested on each instances within 40,000 evaluations. The best solutions ever found from 50 independent runs by these four algorithms were stored and averaged.

For cGA, the virtual population number is set to be $n = \frac{\sqrt{\pi}}{2}\sqrt{D}\log D$ as reported in [5]. For PBIL, population size n, learning rate L, probability of mutation P_m and the shift amount for mutation S are set to be 20, 0.1, 0.02 and 0.05 respectively. For QEA, all parameters were same as the original literature [2]. In the proposed THEDA, the population size n, proportion of selection P_s and the learning coefficient c are set to be 200, 0.1 and 0.5 respectively.

On $P_{knp\_250\_Ch}$ and $P_{knp\_500\_Ch}$, the performances of cGA, PBIL and the proposed THEDA are similar to each other. For $P_{knp\_1000\_Ch}$ and $P_{knp\_2000\_Ch}$, the THEDA is superior to others. Premature convergence of PBIL resulted in inferior solution. The cGA could find solution having the same quality. But cGA is slower than THEDA.

On $P_{knp\_250\_Cf}$, PBIL was faster than THEDA before $4,000$ evaluations and then decelerate quickly. THEDA had a better mean of the best solutions and a relatively smaller standard deviation than that of PBIL. We found that premature convergence is gradually becoming a serious defect to PBIL as the population size increases. Especially, on $P_{knp\_2000\_Cf}$, PBIL startup quickly till $10,000$ evaluations, but it prematurely converged to solutions whose fitness are about 75. While the fitness of the best solution found by the proposed THEDA maintained steady growth from beginning to $30,000$ evaluations. The mean of the final solutions offered by THEDA was 109.091 with a standard deviation 1.8311. The results on $P_{knp\_2000\_Ch}$ and $P_{knp\_2000\_Cf}$ are shown in 1.

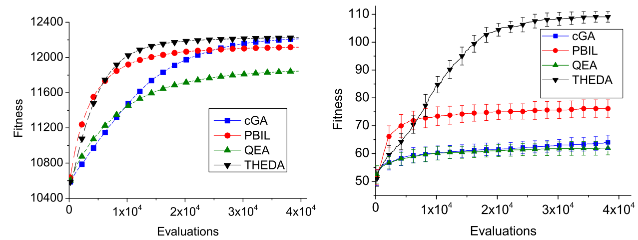

Figure 1: The Result on $P_{knp\_2000\_Ch}$ and $P_{knp\_2000\_Cf}$

4. CONCLUSION AND FUTURE WORK

A proper way to balance exploration and exploitation is an essential part of an effective evolutionary algorithm. The core of EDA is the model and the update method. We regard the model building from fittest individuals as an inference. Inspired by Bayesian inference, we used a two-level hierarchical model involving beta vector and probability vector. We also design a function of generation number to control the exploration and exploitation along with the model updating. THEDA was tested on 01-knapsack problems. The results showed that THEDA outperformed cGA, PBIL and QEA. Its advantage was amplified as the increase of the problem size. There are several issues required to be studied in the future work. For example, the design of this learning rate function will need more detailed research. Besides, THEDA uses prior probability which can contain prior knowledge of the problem. One can try to enhance THEDA by introducing prior knowledge into it. Last but not least, there should be a detailed study on parameter settings.

Acknowledgment

This work is supported by National Natural Science Foundation of China (Grant No: 61175110), National Basic Research Program of China (973 Program) (Grant No: 2012CB 316305), National S&T Major Projects of China (Grant No: 2011ZX02101-004) and National Banking Information Technology Risk Management Projects of China.

5. REFERENCES

[1] M. Črepinšek, S.-H. Liu, and M. Mernik. Exploration and exploitation in evolutionary algorithms: a survey. *ACM Computing Surveys (CSUR)*, 45(3):35, 2013.

[2] K.-H. Han and J.-H. Kim. Quantum-inspired evolutionary algorithm for a class of combinatorial optimization. *Evolutionary Computation, IEEE Transactions on*, 6(6):580–593, 2002.

[3] S. Martello and P. Toth. *Knapsack Problems: Algorithms and Computer Implementations*. John Wiley & Sons, Inc., New York, NY, USA, 1990.

[4] M. Pelikan, D. E. Goldberg, and F. G. Lobo. A survey of optimization by building and using probabilistic models. *Computational optimization and applications*, 21(1):5–20, 2002.

[5] M. D. Platel, S. Schliebs, and N. Kasabov. Quantum-inspired evolutionary algorithm: a multimodel eda. *Evolutionary Computation, IEEE Transactions on*, 13(6):1218–1232, 2009.