A Comparison of GAs using Penalizing Infeasible Solutions and Repairing Infeasible Solutions on Restrictive Capacity Knapsack Problem

[Extended Abstract]

Jun He
School of Computer Science
University of Birmingham
Birmingham, B15 2TT, UK
j.he@cs.bham.ac.uk

Yuren Zhou
School of Computer Science and Engineering
South China University of Technology
Guangzhou 510640, China
yrzhou@scut.edu.cn

Categories and Subject Descriptors

F.2 [**Theory of Computation**]: Analysis of Algorithms and Problem Complexity; G.1.6 [**Mathematics of Computing**]: Optimization

General Terms

Algorithm, Performance

Keywords

Constrained optimization, Knapsack problem, Computation times

1. INTRODUCTION

In the real world almost every optimization problem contains more or less constraints. It is very important to process constraints when GAs are used in solving optimization problems in practice. Different constraint handling techniques have been incorporated with GAs [1]. However, most of current studies are based on computer experiment. The observations from computer experiments need a theoretical analysis and explanation [2]. This paper aims at providing a theoretical analysis to the experimental results reported in [1].

The GAs $A_p[i], i = 1, 2, 3$ using penalty function method and $A_r[i], i = 1, 2$ using repair method are almost the same as those in [1], excepted the crossover is not used.

The following lemma gives the status of the initial population:

LEMMA 1. *If an initial individual is chosen at random, then the probability for all individuals in the initial population being an infeasible solution is not less than* $1-\exp(-n/16)^N$, *where* N *is the population size.*

Let τ_f be the mean number of generations for a GA to find a feasible solution. This number can be rigorously defined by a first hitting time:

$$\tau_f = \{t \geq 0;\ \xi^{(t)} \text{ is feasible.}\}$$

GECCO'07, July 7–11, 2007, London, England, United Kingdom.
ACM 978-1-59593-697-4/07/0007.

Let τ_n be the number of generations for a GA to find a feasible solution or an infeasible solution only 1 bit away from feasible solution, defined by

$$\tau_n = \{t \geq 0;\ H(\xi^{(t)}, S_f) \leq 1\},$$

where $H(\cdot, \cdot)$ is the Hamming distance, S_f the set of feasible solutions.

THEOREM 1. *Assume the initial individual is chosen at random. If an* $(N + N)$ $A_p[1]$ *is used to solve this problem, then it cannot find a nearly feasible solution with a probability not less than* $1 - \exp(-n/16)^N$.

THEOREM 2. *Assume the initial individual is chosen at random. If an* $(N + N)$ $A_p[i](i = 2, 3)$ *is used to solve the problem, then the first hitting time* $\bar{\tau}_n$ *is* $O(n \ln n/N)$ *and the number of fitness evaluations is* $O(n \ln n)$.

THEOREM 3. *Assume the initial individual is chosen at random. If an* $(N + N)$ $A_r[i](i = 1, 2)$ *is used to solve the uncorrected restrictive knapsack, then the expected first hitting time* $\bar{\tau}_n$ *is* 1 *and expected number of fitness evaluations is* $O(n)$.

The results have confirmed the observations from computer experiments [1]: the GAs using repairing infeasible solutions are better than those using penalizing infeasible solutions at the finding a feasible solution.

2. ACKNOWLEDGMENTS

J. He was supported by the UK Engineering and Physical Research Council under Grant No. EP/C520696/1. Y. Zhou was supported in part by the Natural Science Foundation of Guangdong Province of China under Grant No. 04300157, and the Science and Technology Plan Project of Guangdong Province of China under Grant No. 2005B10101048.

3. REFERENCES

[1] Z. Michalewicz. *Genetic Algorithms + Data Structure = Evolution Program.* Springer Verlag, New York, third edition, 1996.

[2] Y. Zhou and J. He. A runtime analysis of evolutionary algorithms for constrained optimization problems. *IEEE Transactions on Evolutionary Computation.* (in press).