Hierarchical BOA on Random Decomposable Problems

Martin Pelikan
Missouri Estimation of Distribution Algorithms Laboratory, 320 CCB
Dept. of Math and Computer Science
University of Missouri–St. Louis
One University Blvd.
St. Louis, MO 63121
pelikan@cs.umsl.edu

Kumara Sastry
Illinois Genetic Algorithms Laboratory, 107 TB
University of Illinois at Urbana-Champaign
104 S. Mathews Ave.
Urbana, IL 61801
kumara@illigal.ge.uiuc.edu

Martin V. Butz
Department of Cognitive Psychology
University of Würzburg
Röntgenring 11
97070 Würzburg, Germany
mbutz@psychologie.uni-wuerzburg.de

David E. Goldberg
Illinois Genetic Algorithms Laboratory, 107 TB
University of Illinois at Urbana-Champaign
104 S. Mathews Ave.
Urbana, IL 61801
deg@illigal.ge.uiuc.edu

Categories and Subject Descriptors

I.2.8 [**Artificial Intelligence**]: Problem Solving, Control Methods, and Search;
G.1.6 [**Numerical Analysis**]: Optimization

General Terms

Algorithms

Keywords

Hierarchical BOA, evolutionary algorithms, decomposable problems, performance analysis, scalability, optimization.

1. INTRODUCTION

There are three important approaches to testing optimization algorithms: (1) Testing on the boundary of the design envelope using artificial test problems, (2) testing on classes of random problems, and (3) testing on real-world problems or their approximations.

The primary purpose of this work is to introduce a class of random additively decomposable problems, which can be used to test optimization algorithms that address nearly decomposable problems. There are three goals in the design of the proposed class of problems: scalability, known optimum, and easy generation of random instances.

Additionally, we apply several simple and advanced genetic and evolutionary algorithms to random instances of the proposed class of problems. Specifically, we consider standard genetic algorithms (GAs), the univariate marginal distribution algorithm (UMDA), hill climbing (HC), and the hierarchical Bayesian optimization algorithm (hBOA) [1]. The results provide important insights into the performance of genetic and evolutionary algorithms in decomposable problems and the difficulty of decomposable problems.

We first present a brief description of the proposed class of problems, then present selected experimental results, and finally provide pointers to more information about this work.

GECCO'06, July 8–12, 2006, Seattle, Washington, USA.

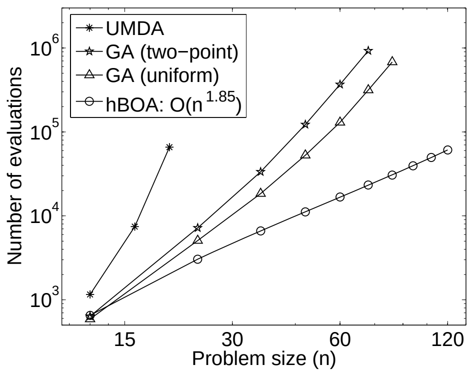
2. RANDOM ADDITIVELY DECOMPOSABLE PROBLEMS

Here we consider additively decomposable problems where the overlap is relatively simple and the optimum can be verified using an efficient procedure based on dynamic programming. The order of all subproblems is fixed to a constant k and the amount of overlap is specified by a parameter $o \in \{0, 1, \ldots, k-1\}$ called *overlap*.

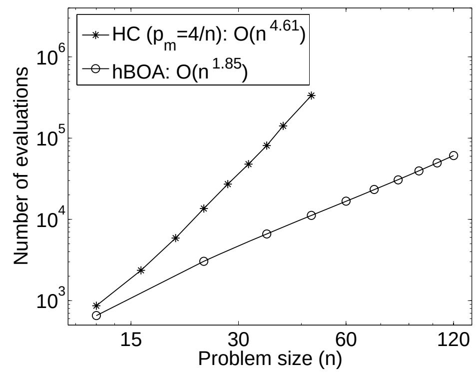
The first subproblem is defined in the first k string positions. The second subproblem is defined in the last o positions of the first subproblem and the next $(k-o)$ positions. All the remaining subproblems are assigned string positions analogically, always defining the next subproblem in the last o positions of the previous subproblem and the next $(k-o)$ positions. To ensure that the subproblems are not always located in consequent string positions, the string can be reordered according to a randomly generated permutation. Each subproblem is defined by a k-bit function and the overall fitness is defined as the sum of all subproblems.

Assuming that the problem is decomposable according to the above definition and that we know the subsets string positions in each subproblem and the corresponding subfunctions, it is possible to solve any problem instance using dynamic programming in $O(2^k n)$ fitness evaluations.

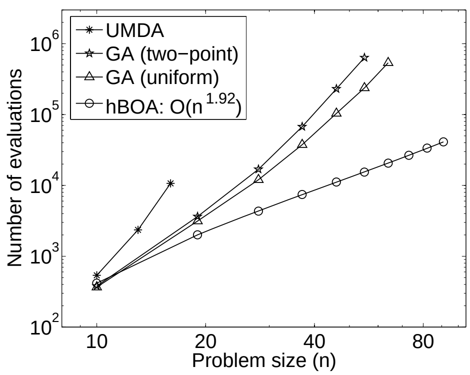
To generate random instances of the proposed class of problems, we must first choose the number m of subproblems, the order k of decomposition, and the overlap o. Then, we generate the 2^k values that specify each subfunction according to the uniform distribution over $[0, 1)$, and the permutation of string positions to eliminate the assumption of tight linkage. Clearly, other distributions can be used to generate the subfunctions and the permutation.



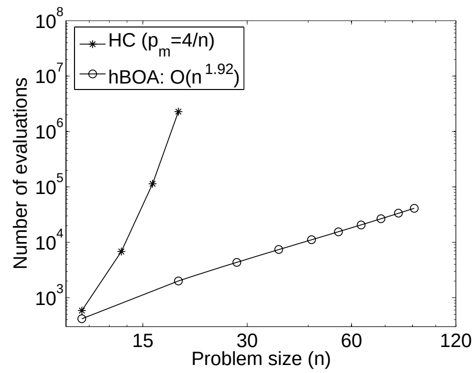
(a) hBOA vs. UMDA and GA for $o = 0$



(b) hBOA vs. HC for $o = 0$



(c) hBOA vs. UMDA and GA for $o = 1$



(d) hBOA vs. HC for $o = 1$

Figure 1: Comparison on random decomposable problems.

3. EXPERIMENTS

All algorithms were compared on problems of varying size for $k = 5$ and overlap $o = 0$ or 1. For each combination of values of n, k, and o, 1000 random problem instances were generated and tested. All algorithms were required to successfully converge to the optimum in 10 out of 10 independent runs. Population size for GA, UMDA, and hBOA was set to its optimal value using bisection method to ensure reliable convergence. For more information, see [2].

Figure 1 compares the performance of hBOA, GA, UMDA, and HC on random problems with $o = 0$ and $o = 1$ (more results can be found in [2]). The results show that the best performance is achieved with hBOA, which can solve all variants of random decomposable problems with only $O(n^{2.02})$ (for $o = 2$) function evaluations or faster. GA, UMDA and HC perform much worse than hBOA, usually requiring a number of evaluations that appears to grow exponentially fast. Recombination-based methods appear to be much less sensitive to overlap than the methods based on local search operators. Although deception is not enforced for any subproblem, linkage learning remains important. Finally, the difficulty of random decomposable problems does not seem to vary much for constant n, k, and o.

For more detailed results and their discussion, please see [2], which can be downloaded at `http://medal.cs.umsl.edu/`, where you can also find the source code of the problem generator and the fitness function in ANSI C. Since it is widely believed that many real-world problems are nearly decomposable and the proposed class of problems covers many potential problems of this form, the proposed class of random problems can be used to provide valuable information about the performance of various optimization algorithms in many real-world problems and to design automated methods for setting algorithm-specific parameters in a robust manner.

Acknowledgments

This work was sponsored by the Air Force Office of Scientific Research, Air Force Materiel Command, USAF, under grant FA9550-06-1-0096, the National Science Foundation under NSF CAREER grant ECS-0547013 and ITR grant DMR-03-25939 at Materials Computation Center, UIUC, and the European Commission contract No. FP6-511931. The U.S. Government is authorized to reproduce and distribute reprints for government purposes notwithstanding any copyright notation thereon.

4. REFERENCES

[1] M. Pelikan and D. E. Goldberg. Escaping hierarchical traps with competent genetic algorithms. *Proceedings of the Genetic and Evolutionary Computation Conference (GECCO-2001)*, pages 511–518, 2001.

[2] M. Pelikan, K. Sastry, M. V. Butz, and D. E. Goldberg. Hierarchical BOA on random decomposable problems. MEDAL Report No. 2006001, Missouri Estimation of Distribution Algorithms Laboratory, University of Missouri–St. Louis, St. Louis, MO, 2006.