# Multi-Objective Bayesian Optimization Algorithm

**Nazan Khan, David E. Goldberg & Martin Pelikan**

IlliGAL Report No. 2002009
March 2002

Illinois Genetic Algorithms Laboratory
University of Illinois at Urbana-Champaign
117 Transportation Building
104 S. Mathews Avenue Urbana, IL 61801
Office: (217) 333-2346
Fax: (217) 244-5705

# Multi-Objective Bayesian Optimization Algorithm

**Nazan Khan**, **David E. Goldberg**, **Martin Pelikan**
Illinois Genetic Algorithms Laboratory
Department of General Engineering
University of Illinois at Urbana-Champaign
Urbana, IL - 61801
{nkhan1, deg,mpelikan}@uiuc.edu Phone: 1-217-333-2346

March, 2002

**Abstract**

This paper proposes a competent multi-objective genetic algorithm called the multi-objective Bayesian optimization algorithm (mBOA). mBOA incorporates the selection method of the non-dominated sorting genetic algorithm-II (NSGA-II) into the Bayesian optimization algorithm (BOA). The proposed algorithm has been tested on an array of test functions which incorporate deception and loose-linkage and the results are compared to those of NSGA-II. Results indicate that mBOA outperforms NSGA-II on large loosely linked deceptive problems.

## 1 Introduction

Recently, significant development in the theory and design of compentent genetic algorithms has been achieved (Goldberg, 1999). By competent genetic algorithm we mean genetic algorithms that can solve hard problems quickly, accurately, and reliably. Several competent genetic algorithms have been proposed that can solve problems of bounded difficulty in subquadratic time(Pelikan, Goldberg, & Cantú-Paz, 1999; Harik, 1999).

However, most of the aforementioned competent genetic algorithms focus only on single-objective optimization although many real-world problems contain more than one objective. Independently of the development of competent genetic algorithms, a number of approaches to solve such multiobjective problems have been proposed (Zitzler, Deb, & Thiele, 2000). However, there has been little or no effort to develop competent multiobjective operators that efficiently identify, propagate, and combine important partial solutions of the problem at hand.

This study makes an effort toward multiobjective competency by combining the best of both the worlds. Specifically, the study combines competent genetic algorigthms with advanced techniques for finding and maintaining a diverse set of nondominated solutions defining the Pareto front. In particular, one of the most advanced competent genetic algorithms called the Bayesian optimization algorithm (BOA)(Pelikan, Goldberg, & Cantú-Paz, 1999) is combined with one of the leading multiobjective solvers called the nondominated sorting genetic algorithm (NSGA-II) (Deb, Pratap, & Meyarivan, 2000). The proposed method is shown to perform well on all tested problems and to discovere a superior Pareto front compared to the NSGA-II. The next section overviews background of multiobjective optimization and the probabilistic model-building genetic algorithms. Section 3 describes multi-objective optimization NSGA-II. Section 4 describes Bayesian optimization algorithm. Section 5 describes the multi-objective Bayesian optimization algorithm. Section 6 describes the test functions used to test the algorithm. Section

7 gives the results obtained and comparison with results obtained by using NSGA-II. Summary and conclusion are provided in Section 8.

## 2 Related Work

There has been a growing interest in the multi-objective optimization algorithm. The multi-objective genetic algorithm (MOGA) proposed by Fonseca and Fleming Fonseca and Fleming (1993), the non-dominated sorting algorithm (NSGA) proposed by Srinivas and Deb (Srinivas & Deb, 1995), the niched-Pareto genetic algorithm (NPGA) proposed by Horn, Nafploitis, and Goldberg (Horn, Nafpliotis, & Goldberg, 1994), and the non-dominated sorting algorithm II (NSGA-II) proposed by Deb, Agarwala, Pratap and Meyarivan (Deb, Pratap, & Meyarivan, 2000) are multi-objective genetic algorithms which make efforts to preserve diversity in the solutions on the Pareto-optimal front. They use non-dominated sorting to assign fitness to the solutions. A more complete review of this literature is available elsewhere. Here, we concentrate on NSGA and its successor NSGA-II. The NSGA was proposed to solve multi-objective problems but some difficulties became evident following its publication. The major disadvantage encountered was that the user needed to specify the sharing parameter which was not an easy task. The success of the algorithm depends on the correctness of the sharing parameter value specified. This led to the development of a new multi-objective algorithm called NSGA-II. It eliminated the need to specify any parameters. The introduction of crowding distance led to the development of a selection operator that was successful in maintaining diversity on the Pareto-optimal front, and hence it is used extensively. There is also a growing interest in sampling new solution using probabilistic models (Baluja, 1994). The simple genetic algorithm relies on crossover and mutation operator to generate new solution. The new solutions generated are different from parents and hopefully better than parents. This theory of genetic algorithms deals with partial solutions or building blocks (BB)(Goldberg, 1989). It is important for the success of genetic algorithms that proper growth and mixing of building blocks is achieved. Simple crossover and mutation operators have a high tendency to disrupt building blocks and lose them. This sometimes leads to errors involving convergence to a local optima. Hence, the development of competent genetic algorithms took place. The fast messy genetc algorithms (mGA) (Goldberg, Deb, Kargupta, & Harik, 1993), linkage learning genetic algorithms (LLGA)(Harik & Goldberg, 1996), Bayesian optimization algorithms are few of the popular competent genetic algorithms. Hierarchical Bayesian optimization algorithm (Pelikan & Goldberg, 2000) has been introduced to solve hierarchical decomposable functions. One of the works which combines the power of competent genetic algorithms and multi-objective genetic algorithms is done by Bosman and Thierens (1999). He proposed an algorithm based on mixture-based iterated density estimation (MIDEA) to solve multi-objective problems.

## 3 NSGA-II

The non-dominated sorting algorithm-II (NSGA-II) is a technique to solve multi-objective problems. It uses crowding distance to incorporate niching. In particular, NSGA-II has a special selection operator that creates a mating pool by combining its parent and offspring populations and selecting the best with respect to fitness and spreading them out on the Pareto-optimal front. It uses a special selection operator that uses crowding distance to obtain a good spread of solutions on the Pareto-optimal front. The selection operator used is quiet similar to binary tournament selection. Randomly pick two individuals from the population, say I1 and I2. R1 and R2 are their respective ranks and C1 and C2 are their respective crowding distance values. Then based on one of the following conditions, we select one of the two individuals as the winner.

```
1.  If (R1 < R2)    I1 is the winner
2.  If (R1 > R2)    I2 is the winner
3.  If (R1 == R2)
            If (C1 >C2)    I1 is winner
            If (C1 <C2)    I2 is winner
            If (C1==C2)  randomly pick one of the two.
```

# 4 Bayesian Optimization Algorithm

The Bayesian optimization algorithm (BOA) (Pelikan, Goldberg, & Cantú-Paz, 1999) is a probabilistic-model building genetic algorithm(PMBGA).Probabilistic model building genetic algorithms replace genetic operators like crossover and mutation with a probabilistic model building and sampling process. It builds probabilistic models and uses these model to sample new solutions from the old solutions. It identifies, reproduces, and mixes building blocks up to a specified order. Acyclic Bayesian networks are used as the probabilistic model and can be mathematically represented by the following equation: , where$X = (X_0, ..., X_{l-1})$ is a vector of Boolean variables, $x_i$ is the set of parents of $X_i$ in the network, and $p(X_i|x_i)$ is the conditional probability of $X_i$ conditioned on the variables $x_i$. BOA uses Bayesian Dirichlet (BD) metric (Heckerman, Geiger, & Chickering, 1994) to test the quality of the Bayesian network it has produced. BOA searches for a good network. The quality of solution depends on the accuracy of the model generated. After constructing the network the conditional probabilities of each variable are determined. These conditional probabilities are then used to create new solutions. BOA generates a new model at each generation. It performs better than a simple GA in case of loosely linked deceptive problems of large problem size. It does not need prior knowledge of the problem. More detail is available elsewhere (Pelikan, Goldberg, & Cantú-Paz, 1999)

# 5 Multi-objective Bayesian Optimization Algorithm

The working of multi-objective Bayesian optimization algorithm can be explained by following steps. At the first step, initial population is generated randomly. The population is evaluated and assigned fitness. Probabilistic model building technique is used to build model. The Bayesian network is then used to model the solutions. New solutions are generated using this model. The exact procedure of building models and generating new solutions from them is explained in detail elsewhere (Pelikan, Goldberg, & Cantú-Paz, 1999). The new individuals generated using models are called children. This child population is also evaluated. The parent population and child populations are merged together to form a double size population. This population can be called the combined population. The size of the parent population and child population are equal. The individuals of the combined population are ranked and crowding distance is calculated for each individual in the combined population. At this stage, every individual in the combined population has a rank and a crowding distance. Rank and crowding distance calculation is explained in more detail elsewhere (NSGA-II) (Deb, Pratap, & Meyarivan, 2000). If the size of the parent population is N, the best N members are taken from the combined population to form the new parent population. This is done by taking solutions from the 1st rank, 2nd rank , 3rd rank and so on till we get total of N individuals in the new parent population. In the last rank taken it is possible that we can not take all individuals as the total number of solutions in the new parent population may then exceed N. So, in the last rank we use the selection operator of NSGA-II. We take the individuals with higher crowding distance.

# 6 Test Functions

We test our algorithm on 5 test functions:

- T1 - Multiple interleaved minimal deceptive problem
- T2 - Complement of T1
- T3 - Multiple interleaved 5-bit trap function
- T4 - Complement of T3
- T5 - Multiple interleaved 6-bit bipolar deceptive function

In all caeses the function was parameterized on the number of building blocks. In the remainder of this section, each of the function is described in more detail. The test functions are difficult in four aspects namely deception (Deb & Goldberg, 1993), loose-linkage, multi-modality (Deb, Horn, & Goldberg, 1993; Goldberg, Deb, & Horn, 1992) and large problem size.

## 6.1 Interleaved Minimal Deceptive Problem (T1 & T2)

The results were tested on a loosely linked minimal deceptive problem (Goldberg, 1987). The loose linkage was incorporated by dividing the string into two halves and one bit from each half was coupled with one corresponding bit from the other half. In figure 1, this loose linkage is explained further. The rectangular blocks represent the bit positions. The blocks having the same pattern are linked. It can be a 0 or a 1 depending on the string. Let $\ell$ be the problem size. $0^{th}$ bit was coupled by $(\frac{\ell}{2}+1)^{th}$ bit. $(\frac{\ell}{2})^{th}$ bit was coupled by last bit and bits falling in between were coupled with bit from other half correspondingly. Now, the first fitness function was evaluated as, if the two coupled bit were both 1 we add 1 to fitness function and if they are both 0 we add 0.9 and rest of the time we add 0 to the fitness function. Hence the optima of this function lies at all bits equal to 1. T2 is complement of T1. It's optima is at all bits equal to 0.

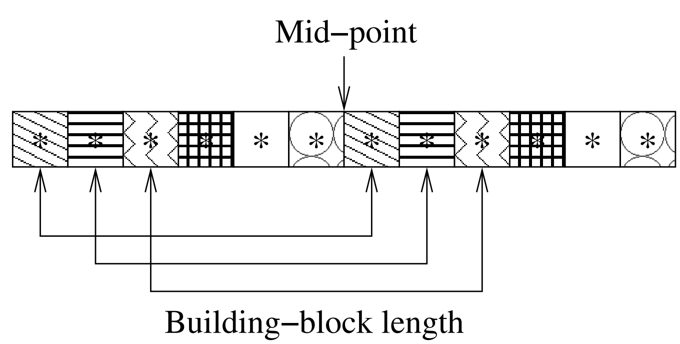



Figure 1: Figure describing the linkage in the loosely linked minimal deceptive problem in T1 and T2

## 6.2 INTERLEAVED 5-BIT TRAP FUNCTION (T3 & T4)

The algorithms were tried on deceptive problem (Deb & Goldberg, 1993) which composed of trap-5 functions. Figure 3(a) describes the trap function for T3. To introduce loose linkage , the 5 bits used to evaluate the fitness function were not consecutive bits but were at a distance of $\frac{\ell}{5}$ apart. Here, $\ell$ is the problem size. The linkage of bits is shown in figure 2.The blocks represent bit position. Bit with same pattern are linked. The five coupled bits have the same pattern. $0^{th}$ bit, $(\frac{\ell}{5})^{th}$ bit, $2(\frac{\ell}{5})^{th}$ bit, $3(\frac{\ell}{5})^{th}$ bit and $4(\frac{\ell}{5})^{th}$ bit are coupled. In this case the defining length of building block is very large. The function is extensively multi-modal. The optima of T3 is at all bits equal to 0. T4 is complement of T3. It is shown in figure 3(b). It has optima at all bits equal to 1.

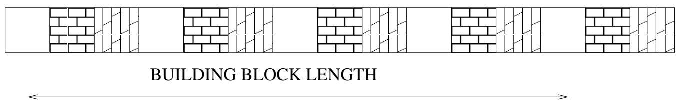
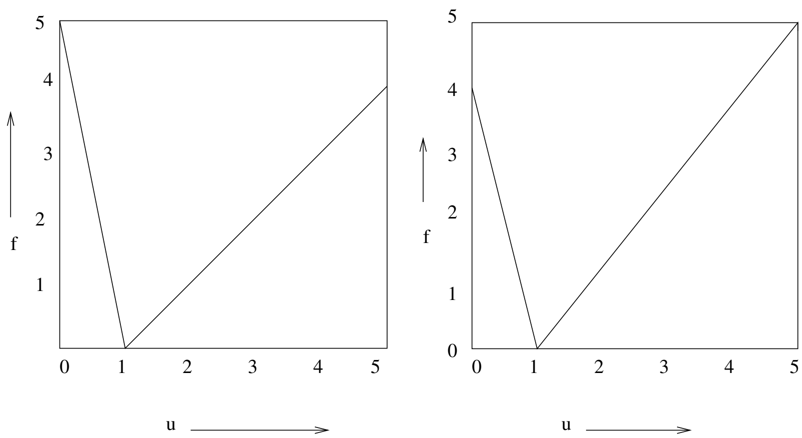



Figure 2: The figure describing the coupling of the bits in the loosely linked trap-5 function of T3 and T4




(a) Trap-5 function of T3 (b) Trap-5 function of T4

Figure 3: The figure describing the Trap-5 functions

### 6.3 Interleaved 6-Bit Bipolar Function (T5)

The algorithm was tried on the loosely linked 6-Bit deceptive bipolar deceptive function (Deb, Horn, & Goldberg, 1993). The first three bits ( $1^{st}$, $2^{nd}$, $3^{rd}$ position) and last three bits ($\ell-3$, $\ell-2$, $\ell-1$ positions) are linked to calculate six-bit bipolar deceptive function. Then next three bits ($4^{th}$, $5^{th}$, $6^{th}$) and second-last three bits ($\ell-6$, $\ell-5$, $\ell-4$ positions) are linked and so on. Figure 4 shows 6-Bit deceptive bipolar function. The linkage is shown in the figure 5 where linked bits have same pattern. The building block length is not constant. For the first combination it is equal to the string length. And for the last combination that is the middle six bit it equal to six.

## 7 Experiments

The multi-objective BOA was tried on various combinations of test problems and was compared to the original NSGA-II. Since, multi-objective BOA and NSGA-II work on different underlying principles it would not be fair to run both the algorithms for same number of generation with

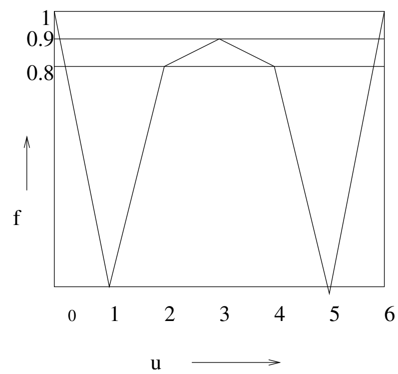



Figure 4: Figure describing the 6-Bit deceptive bipolar function of T5

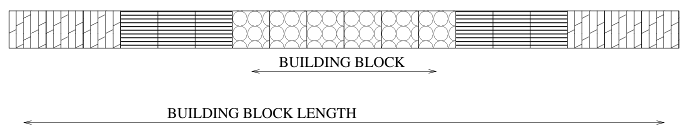
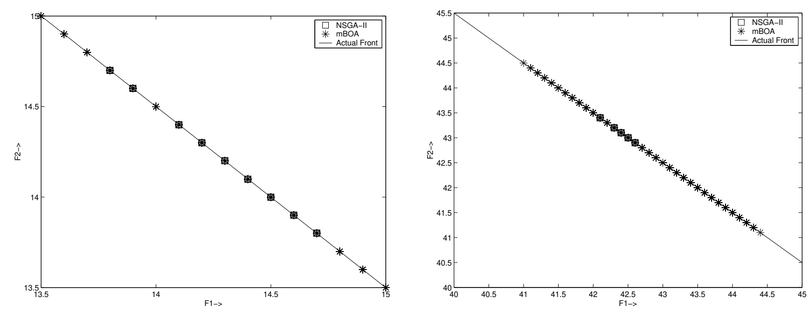



Figure 5: Figure describing the linkage in 6-Bit bit bipolar function of T5

same population size. So, instead the total number of fitness evaluations was kept the same for both the algorithms. The population size and the number of generations were set to values that suited reasonably well to each algorithm. Multi-objective BOA generates Bayesian networks and learns the linkage between different bit positions. It needs a larger population size as compared to NSGA-II. But, it converges very fast. The population size used for multi-objective BOA was determined in the same way as for single objective Bayesian optimization algorithm. NSGA-II was run for $2\ell$ number of generations, where $\ell$ is the problem size. Population size for NSGA-II was calculated using the following equation $n_{fe} = n_1 t_{c_1} = n_2 t_{c_2}$. Where $n_{fe}$ is total number of fitness function evaluations, $n_1$ is the population size for mBOA and $t_{c_1}$ is the number of generations taken by mBOA, $n_2$ is the population size for NSGA-II and $t_{c_2}$ is the number of generations taken by NSGA-II. In NSGA-II, binary tournament selection was used. The crossover probability was set to 0.9. Mutation rate of 0.01 used. Single point crossover was found to perform better than uniform crossover. So, single point crossover was used. The following problems were tried on multi-objective BOA and NSGA II and the results are plotted on the same graph for the ease of comparison. The actual Pareto-optimal front is also plotted. Both the fitness functions are maximized.

### 7.1 2-D Multiple Interleaved Minimal Deceptive Problems (T1 vs T2)

The first fitness function was T1 and the second fitness function was T2. Algorithms were tried for problem size 30. The multi-objective BOA was run for 40 generations with a population size of 300. NSGA-II was run for 60 generations with a population size of 200. Both the algorithms did well (Figure 6(a)). Analyzing more closely multi-objective BOA found the extreme optima of the Pareto-optimal front. They correspond to the solution with all-bit-one and all-bit-zero. NSGA-II was not able to find these extreme optimal solutions. Also the total number of different strings lying on the Pareto-optimal front of NSGA-II is 19. While multi-objective BOA was able to find a total number of 93 different strings lying on Pareto-optimal front. This shows that the search power of multi-objective BOA is much higher than the NSGA-II or any other simple GA . Next, the algorithms were tried for problem size 90. The BOA was run for 80 generations with a population size of 700. NSGA II was run for 180 number of generation with a population size of population size of 312. None of the algorithm can find the extreme optimal solutions (Figure6(b)). Multi-objective BOA was able to find 700 different strings in the Pareto-optimal front with none of them being the extreme optima. While, NSGA-II was able to find only 6 different strings in the Paret-optimal front. The whole population of NSGA-II converged to merely 6 points on the Pareto-optimal front. To get 700 different strings in the Pareto-optimal front from NSGA-II or any other simple GA we will have to use extremely high population size. So, at larger problem size we can see the advantage of multi-objective BOA over a simple GA.




(a) Problem size 30

(b) Problem size 90

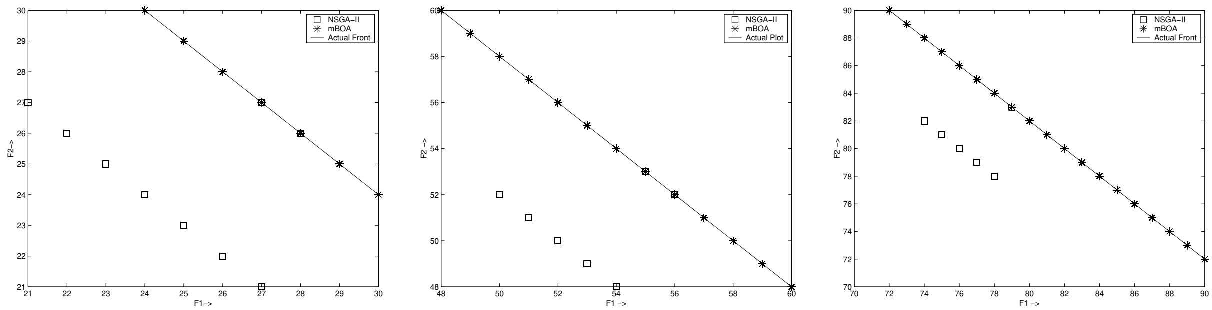
Figure 6: Test run on 2-D multiple interleaved minimal deceptive problem

### 7.2 2-D Multiple Interleaved Trap-5 Functions(T5 vs T4)

The first fitness function was taken to be T3 and the second fitness function was taken to be T4. The algorithms were tried on problem size 30. Multi-objective BOA was run for 40 generations with a population size of 1300. NSGA-II was run for 60 generations with a population size of 868. Multi-objective BOA was able to find 89 different strings in the Pareto optimal front (Figure 7(a)). While NSGA-II could only find 79 string. The algorithms were tried on same function with problem size equal to 60. NSGA-II was run for 120 number of generations with a population size of 1650. Multi-objective BOA was run for 60 generations with a population size of 3300. Multi-objective BOA was able to find 2122 different strings in the Pareto-optimal front while NSGA-II was not able to converge to the real Pareto-optimal front (Figure 7(b)). Multi-objective BOA found both the extreme solutions all-ones and all-zeros. The algorithms were tried on same function with problem size equal to 90. NSGA II was run for 180 numbers of generations with a population size of 2312. Multi-objective BOA was run for 80 generations with a population size of 5200. Multi-objective BOA was able to find 4936 different strings in the Pareto-optimal front while NSGA II was not able to converge to the real Pareto-optimal front (figure 7(c)). Multi-objective BOA found both the extreme solutions all-ones and all-zeros. From above results it clear BOA is more suitable as problem size is increased.

### 7.3 Multiple Interleaved 6-Bit Deceptive Bipolar Function vs Multiple Interleaved Minimal Deceptive Function (T5 vs T2)

The first fitness function was taken to be T5 and the second fitness function was taken to be T2. The problem size was 60. The NSGAII was run for 120 number of generation with a population size of 1500. Multi-objective BOA was run for 90 generation with a population size of 2000. Multi-objective BOA found the unique optima (all zero) while NSGA II was not able to find it (Figure 8).




(a) Problem size 30 (b) Problem size 60 (c) Problem size 90

Figure 7: Test run on multiple interleaved trap-5 deceptive functions

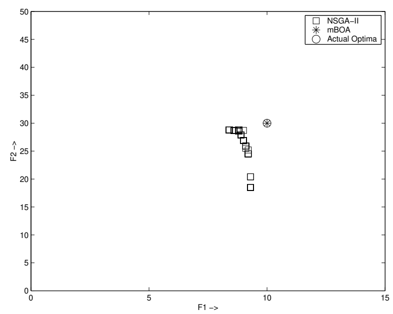



Figure 8: Test run on multiple interleaved 6-Bit deceptive bipolar Function vs multiple interleaved minimal deceptive function with problem size of 60

### 7.4 Multiple Interleaved 6-Bit Deceptive Bipolar Function vs Multiple Interleaved 5-Bit Deceptive Trap Function(T5 vs T3)

The first fitness function was taken to be T5 and the second fitness function was taken to be T3. This problem also has just one optimal solution. Pareto front is just one point. The problem size was equal to 30. Both multi-objective BOA and NSGA II were run for 60 generation with a population size of 1500. Multi-objective BOA was able to converge to the optima while NSGAII was not able to find it (Figure 9). It is important to notice that the problem size 30 is considerably small here. If we increase the problem size the performance of NSGAII will decrease further unless we use a very large population size.

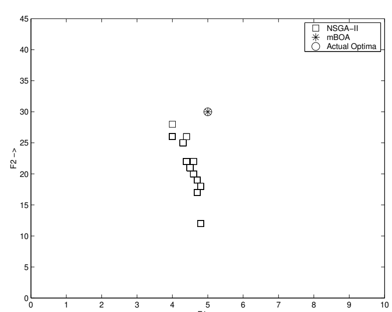



Figure 9: Test run on multiple interleaved 6-Bit deceptive bipolar function vs multiple interleaved Trap-5 deceptive function with problemsize of 30

# 8 Summary and Conclusion

The work sheds light on the use of competent GAs like BOA in the field of multi-objective optimization. Competent genetic algorithms and multi-objective genetic algorithms are important topics in the field of genetic algorithms. The competent genetic algorithms are ahead of simple genetic algorithms in solving complex problems. Multi-objective genetic algorithms are means to handle problems involving more than one objective function. We combined competent genetic algorithms and multi-objective genetic algorithms to have a tool that can solve loosely- linked deceptive multi-objective problems of large problem size. The proposed algorithm outperforms NSGA-II, a representative of simple multi-objective GA schemes. The solution of deceptive multi-objective problems suggests that mBOA can handle a very large class of real world problems quickly, reliably and accurately. More work is needed but we believe that the technique is ready for intial trials on practical problems.

### Acknowledgments

We thank Michael Welge and Barbara Minsker for encouraging continued work in multi-objective genetic algorithms.

The work was sponsored by the Air Force Office of Scientific Research, Air Force Materiel Command, USAF, under grant F49620-00-0163. Research funding for this work was also provided by a grant from the National Science Foundation under grant DMI-9908252. The US Government is authorized to reproduce and distribute reprints for Government purposes notwithstanding any copyright notation thereon.

The views and conclusions contained herein are those of the authors and should not be interpreted as necessarily representing the official policies or endorsements, either expressed or implied, of the Air Force Office of Scientific Research, the National Science Foundation, or the U.S. Government.

# References

Baluja, S. (1994). *Population-based incremental learning: A method for integrating genetic search based function optimization and competitive learning* (Tech. Rep. No. CMU-CS-94-163). Pittsburgh, PA: Carnegie Mellon University.

Bosman, P. A. N., & Thierens, D. (1999). Linkage information processing in distribution estimation algorithms. *Proceedings of the Genetic and Evolutionary Computation Conference 1999: Volume 1*, 60–67.

Deb, K., & Goldberg, D. E. (1993). Analyzing deception in trap functions. *Foundations of Genetic Algorithms 2*, 93–108.

Deb, K., Horn, J., & Goldberg, D. E. (1993). Multimodal deceptive functions. *Complex Systems*, *7*(2), 131–153.

Deb, K., Pratap, A., & Meyarivan, T. (2000). A fast elitist non-dominated sorting genetic algorithm for multi-objective optimization: Nsga-ii. *Parallel Problem Solving from Nature*, *4*(1), 849–858.

Fonseca, C. M., & Fleming, P. J. (1993). Genetic algorithms for multiobjective optimization: Formulation, discussion and generalization. *Proceedings of the Fifth International Conference on Genetic Algorithms*, 416–423.

Goldberg, D. E. (1987). Simple genetic algorithms and the minimal, deceptive problem. In *Genetic Algorithms and Simulated Annealing* (Chapter 6, pp. 74–88).

Goldberg, D. E. (1989). *Genetic algorithms in search, optimization, and machine learning.* Reading, MA: Addison-Wesley.

Goldberg, D. E. (1999). The race, the hurdle, and the sweet spot: Lessons from genetic algorithms for the automation of design innovation and creativity. *Evolutionary Design by Computers*, 105–118.

Goldberg, D. E., Deb, K., & Horn, J. (1992). Massive multimodality, deception, and genetic algorithms. *Parallel Problem Solving from Nature, 2*, 37–46.

Goldberg, D. E., Deb, K., Kargupta, H., & Harik, G. (1993). Rapid, accurate optimization of difficult problems using fast messy genetic algorithms. *Proceedings of the Fifth International Conference on Genetic Algorithms*, 56–64.

Harik, G. (1999). *Linkage learning via probabilistic modeling in the ECGA* (IlliGAL Report No. 99010). Urbana, IL: University of Illinois at Urbana-Champaign, Illinois Genetic Algorithms Laboratory.

Harik, G. R., & Goldberg, D. E. (1996). Learning linkage. *Foundations of Genetic Algorithms 4*, 247–262.

Heckerman, D., Geiger, D., & Chickering, D. (1994). Learning bayesian networks: The combination of knowledge and statistical data. *Tenth Conference on Uncertainty in Artificial Intelligence*, 293–301.

Horn, J., Nafpliotis, N., & Goldberg, D. E. (1994). A niched pareto genetic algorithm for multiobjective optimization. *Proceedings of the First IEEE Conference on Evolutionary Computation*, 82–87.

Pelikan, M., & Goldberg, D. E. (2000). Hierarchical problem solving and the Bayesian optimization algorithm. *Proceedings of the Genetic and Evolutionary Computation Conference 2000*, 267–274.

Pelikan, M., Goldberg, D. E., & Cantú-Paz, E. (1999). BOA: The bayesian optimization algorithm. *Proceedings of the Genetic and Evolutionary Computation Conference 1999: Volume 1*, 525–532.

Srinivas, N., & Deb, K. (1995). Multiobjective optimization using nondominated sorting in genetic algorithms. *Evolutionary Computation*, *2*(3), 221–248.

Zitzler, E., Deb, K., & Thiele, L. (2000). Comparison of multiobjective evolutionary algorithms: Empirical results. *Evolutionary Computation*, *8*(2), 173–195.