ABSTRACT

PROBLEMS WITH CROSSOVER BIAS FOR BINARY STRING

REPRESENTATIONS IN GENETIC ALGORITHMS

By

Brian Cleary

August 2011

Genetic algorithms, a popular technique for optimization, traditionally uses binary

pressure will be applied to make up for the bias against some values.

PROBLEMS WITH CROSSOVER BIAS FOR BINARY STRING REPRESENTATIONS IN GENETIC ALGORITHMS

A THESIS

Presented to the Department of Computer Science and Computer Engineering

California State University, Long Beach

In Partial Fulfillment

of the Requirement for the Degree

Master of Science

By Brian Cleary

B.S., 2003, California State University, Long Beach

August 2011

WE, THE UNDERSIGNED MEMBERS OF THE COMMITTEE,

HAVE APPROVED THIS THESIS

PROBLEMS WITH CROSSOVER BIAS FOR BINARY STRING

REPRESENTATIONS IN GENETIC ALGORITHMS

By

Brian Cleary

COMMITTEE MEMBERS

Shui-Fung Lam, Ph.D. (Chair)	Computer Science
Kenneth James, Ph.D.	Computer Science
Burkhard Englert, Ph.D.	Computer Science

ACCEPTED AND APPROVED ON BEHALF OF THE UNIVERSITY

Kenneth James, Ph.D.
Department Chair, Computer Science

California State University, Long Beach

August 2011

TABLE OF CONTENTS

LIST OF FIGURES

FIGURE Page

LIST OF TABLES

TABLE Page

CHAPTER 1

INTRODUCTION

Optimization

Improvement means finding better solutions to problems. These problems may be to win a game [1][2]. Any system for which the quality of a function.[1] Some situations require method is the Genetic Algorithm, the subject of this work.

The simplest optimization method is to examine every solution in the search space, to compute, time is often given in iterations rather than seconds.

Mastermind

Consider a simple code-breaking game to be referred to henceforth as Mastermind [3]. The objective of the player is to guess a multi-digit sort of difficult problem for which approximate optimization is often used [4][5].

The objective function is defined in equation 1. The motivation behind it will be explained in section 1.

$$f(code) = correct + log(partiallycorrect) \tag{1}$$

Random Search

Another obvious optimization method would be to repeatedly guess while keeping probability of having found the optimum in k trials given a search space of size n given by the following equation and shown for various values of n in Figure 1.

[1] Some objective functions make the goal minimization rather than maximization, but the process is effectively the same.

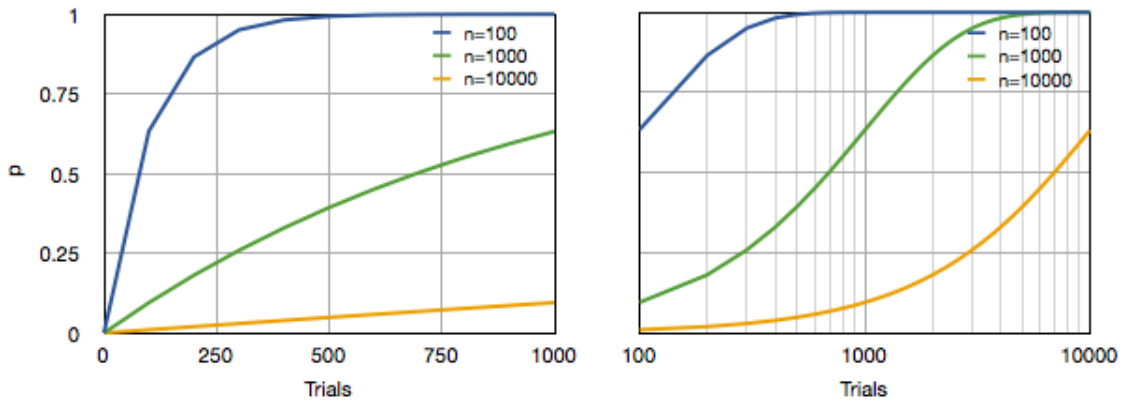

FIGURE 1. Probability of having guessed optimal solution in k trials given search size n, for n=100, 1000, and 10000.

$$p(k|n) = 1 - \left(1 - \frac{1}{n}\right)^k \tag{2}$$

Given the probability of guessing the solution in n trials is approximately 63%the objective function for random search is shown in Figure 2, normalized to 1 for the maximum possible value. This shows that, while

Most evolutionary algorithms include notions of a population, a fitness function, a selection method, and a breeding method [9]. The population contains potential solutions to an optimization problem, the quality of which is judged by the fitness function. A generic evolutionary algorithm is shown in Table 1.

A more specific version of evolution is genetic evolution, which treats the gene as the main element of evolution rather than the organism [10]. of genes, and the breeding methods should be akin to genetic crossover and mutation.

Heuristics

A major way in which algorithms, including optimization methods, can be improved is by designing it to take advantage of knowledge and assumptions made by its

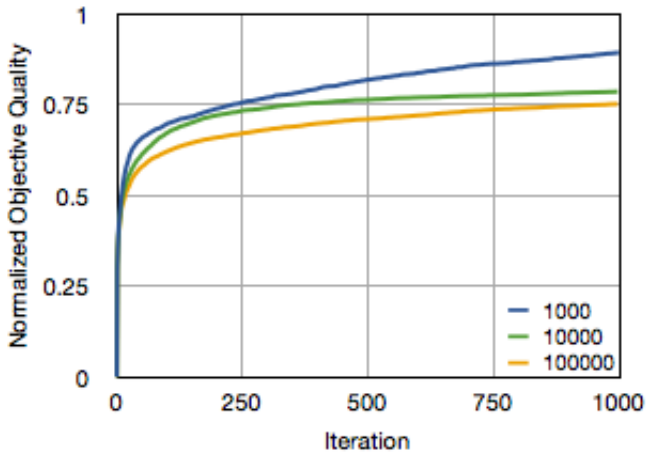

FIGURE 2. Normalized objective function quality for random search of Mastermind for k iterations given search size n=100, 1000, 10000. Averaged over 1000 trials.

TABLE 1. Generic Evolutionary Algorithm

```
population ← random initial values
WHILE fitness of best population member ≤ required fitness
    new_population ← empty list
    REPEAT
        Select parents from the population proportionally to fitness
        Breed parents to create children.
        Add children to new_population.
    UNTIL size(new_population) == size(population)
    population ← new_population
```

CHAPTER 2

OVERVIEW OF GENETIC ALGORITHMS

One of the most direct imitations of genetic evolution used for optimization is the

Genetic Algorithms Operations and Their Purposes

Parts of the Genetic Algorithm

The main part of the genetic algorithm is the gene, which is a problem parameter

CHAPTER 3

COMPLICATIONS OF GENETIC ALGORITHMS

The many variations in genetic algorithms have been created to address numerous genetic algorithm effectiveness.

Diversity

One of the most fundamental problems that can occur in a genetic algorithm is the probability between 0.01 and 0.001 per bit, the delay would have a high expected duration [11][12]. For this reason, some on a diversity measure of the population, raising it when the diversity becomes too low.

Another issue of low genetic diversity in the population is that it represents poor Hamming distance between the chromosomes selected for breeding [13]. Conversely, when it is desirable to explore multiple local optima, the same measure could be used to promote crossover within these niches and even partition the population for separate parallel exploration [15].

similar integer values often have very dissimilar binary representations, as shown in Equation 3 by the linear distance Hamming distance for the integers seven and eight. If the schemata being explored were $x111_2$ and the optimum value for the problem happened to be 1000_2, the space worse for a larger pair of values such as 1000000000000000_2 and 0111111111111111_2.

$$|7 - 8| = 1 \qquad \neq \qquad ||0111_2 - 1000_2||_1 = 4 \tag{3}$$

representations. 1101_2 and 1100_2 are close together by both linear and would be useless since no schema would have any approximate meaning.

CHAPTER 4

PROBLEMS WITH COMBINATORIAL CROSSOVER BIAS FOR BINARY STRING GENETIC REPRESENTATIONS

Distribution of Features and Values

When using a limited range of integer values, crossover as the recombination of

Similarity Problems with Categorical Variables

As noted previously, another problem with the binary string representation is that

TABLE 2. Crossover Outcomes for the Range 0 to 2

$00_2 \times 00_2$	$\rightarrow$	$\mathbf{00}_2$, 00_2
$00_2 \times 01_2$	$\rightarrow$	$\mathbf{01}_2$, 01_2
$00_2 \times 10_2$	$\rightarrow$	$\mathbf{10}_2$, 00_2
$01_2 \times 00_2$	$\rightarrow$	$\mathbf{00}_2$, 01_2
$01_2 \times 01_2$	$\rightarrow$	$\mathbf{01}_2$, 01_2
$01_2 \times 10_2$	$\rightarrow$	$\mathbf{00}_2$, 11_2
$10_2 \times 00_2$	$\rightarrow$	$\mathbf{10}_2$, 00_2
$10_2 \times 01_2$	$\rightarrow$	$\mathbf{11}_2$, 00_2
$10_2 \times 10_2$	$\rightarrow$	$\mathbf{10}_2$, 10_2

TABLE 3. Crossover Outcomes Frequency for the Range 0 to 2

Outcome	00_2	01_2	10_2	11_2
Frequency	7	5	4	2

CHAPTER 5

METHODOLOGY

This work has two main components. First, demonstrating the existence of the

Measuring Bias

Measuring the Bias of a Base Set

The first bias to be measured was that of the base set from which samples will be

CHAPTER 6

RESULTS

Integer Range Bias

Versus offset

As previously mentioned, it was found that most ranges can be shifted to a lower

CHAPTER 7

CONCLUSIONS

Crossover Bias is Significant

Combinatorial crossover bias is a significant issue with binary strings. Crossover that of the form $0..2^n - 1$, which includes all normal integer type variables that are some whole number of bytes.

Offset Improvement Significant but Insufficient

Simple linear offsets make significant but insufficient improvement to the Pearson's χ^2 test for uniformity, almost all would not.

REFERENCES

REFERENCES

[1] J. Rouet et al., "Genetic algorithms for a robust 3-D MR-CT registration," in *IEEE Trans. Inf. Technol. Biomed.*, 2000, vol. 4, pp. 126-136.

[2] G. Fredericks, *Light-Bot Genetic Algorithms,* 2011, June 16; http://gfredericks.com/sandbox/light_bot

[3] E. W. Weisstein, *Mastermind,* 2011, June 16; http://mathworld.wolfram.com/Mastermind.html

[4] D. Knuth, "The Computer as a Master Mind", *J. Recr. Math.*, 1976, vol. 9, pp. 16.

[5] J. Stuckman and G. Zhang, "Mastermind in NP-Complete," in *INFOCOMP J. Comput. Sci.*, 2006, vol. 5, pp. 25-28.

[6] S. Kirkpatrick et al., "Optimization by Simulated Annealing," in *Science*, 1983, vol. 220, Issue 4598, pp. 671-680.

[7] J. Kennedy et al., *Swarm Intelligence*, Morgan Kaufmann Publishers, 2001, pp. 105-109.

[8] D. B. Fogel, *Evolutionary Computation: Toward a New Philosophy of Machine Intelligence*, 2nd ed., IEEE Press, 2000, pp. 31-51.

[9] J. Kennedy et al., *Swarm Intelligence*, Morgan Kaufmann Publishers, 2001, pp. 133-186.

[10] R. Dawkins, *The Selfish Gene*, 30th Anniversary ed., Oxford University Press Inc., 1976, pp. 46-65.

[11] K. A. DeJong and W. M. Spears, "An Analysis of the Interacting Roles of Population Size and Crossover in Genetic Algorithms," in *Proc. 1st Workshop Parallel Problem Solving from Nature*, Springer-Verlag, 1990, pp. 38-47.

[12] J. J. Grefenstette, "Optimization of Control Parameters for Genetic Algorithms," in *IIEEE Trans. Syst. Man Cybern.*, vol. SMC-16, no. 1, Jan./Feb. 1986, pp. 122-128.

[13] L. Eshelman, "The CHC Adaptive Search Algorithm," in *Foundations of Genetic Algorithms*, G. J. E. Rawlins, ed., Morgan Kaufmann Publishers Inc., 1991, pp. 265-283.

[14] J. Holland, *Adaptation in Natural and Artificial Systems*, University of Michigan Press, 1975.

[15] S. Mahfoud, "Niching Methods for Genetic Algorithms," Ph.D. dissertation, Dept. General Eng., Univ. Illinois, Urbana-Champaign, 1995.

[16] D. H. Wolpert and W. G. Macready, "No Free Lunch Theorems for Optimization," in *IEEE Trans. Evol. Comput.*, vol. 1, no. 67, April 1997.

[17] R. L. Plackett, "Karl Pearson and the Chi-Squared Test," in *International Statistical Review / Revue Internationale de Statistique*, vol. 51, no. 1, Apr. 1983, pp. 59-72.