Initial Population for Genetic Algorithms: A Metric Approach

Pedro A. Diaz-Gomez and **Dean F. Hougen**
School of Computer Science
University of Oklahoma
Norman, Oklahoma, USA

Abstract - *Besides the difficulty of the application problem to be solved with Genetic Algorithms (GAs), an additional difficulty arises because the quality of the solution found, or the computational resources required to find it, depends on the selection of the Genetic Algorithm's characteristics. The purpose of this paper is to gain some insight into one of those characteristics: the difficult problem of finding a good initial population. We summarize previous approaches and metrics related to this problem and we suggest the center of mass as an alternative metric to measurement diversity at the population level. This theoretical approach of analysis and measure of the diversity of the initial random population is important and could be quite necessary for the design of GAs because of the relation of the initial population to other GA parameters and operators and because of its relation to the problem of premature convergence.*
Keywords: Population size, diversity, entropy, Hamming distance, center of mass.

1 Introduction

Genetic Algorithms (GAs) are an appealing tool to solve optimization problems [2]. To encode a problem using Genetic Algorithms, one needs to address some questions regarding the initial population, the probability and type of crossover, the probability and type of mutation, the stopping criteria, the type of selection operator, and the fitness function to be used in order to solve the problem. All of these parameters and operators affect the performance of a GA and they are inter-related, i.e., they form a system.

In this paper we study different metrics at the gene and, chromosome levels and propose the population level center of mass metric that gathers the previous ones, can be computed partially while generating the initial population, and goes beyond the previous ones, taking into consideration the actual structure of the population. We have taken this approach because we are considering *diversity* as the variety and difference at the gene, chromosome, and population level. However, other definitions of diversity can be found [13] that could be considered and, at the same time, there are other metrics to use [25], that are beyond the scope of the present study or could be computationally expensive.

The organization of this article is as follows: Section 2 describes previous work, focusing on population sizing for GAs; Section 3 indicates the possible variables that should be taken into account when generating an initial population randomly; Section 4 presents and evaluates different metrics at the gene, chromosome, and population level; Section 5 compares and analyzes those metrics; and Section 6 contains the conclusions and future work.

2 Previous Work

Population sizing has been one of the important topics to consider in evolutionary computation [1], [18]. Researchers usually argue that a "small" population size could guide the algorithm to poor solutions [17], [18], [12] and that a "large" population size could make the algorithm expend more computation time in finding a solution [15], [14], [10], [12]. So, here we are facing a trade-off that needs to be approximated feeding the algorithm with "enough" chromosomes [24], in order to obtain "good" solutions. "Enough," for us, is directly related to instances in the search space and diversity.

Goldberg et al. [9] study the problem taking care of the *building blocks* (BBs) that should be supplied to GAs in order to obtain good results. Others [17], [24], [10] agree that the population size is in direct relation to the difficulty n of the problem, i.e., the bigger n the more individuals are needed. Pelikan et al. [17] study population sizing using the *Bayesian Optimization Algorithm* (BOA), where the initial population is

generated randomly and, after that, in each iteration the best individuals are selected and the worst ones are replaced with new ones generated randomly, building in this way the Bayesian network. Harik and Lobo [10] conclude, then, that the population size needed in order to obtain a good solution is proportional to the number of building blocks in the problem, which means at the same time, that if not enough BBs are supplied in the initial population the algorithm may not find a correct solution. However, it could be the case that, if the population is increased, some algorithms can be worse in finding good solutions, which can occur in problems where the variables are dependent. Yu et al. [24] continue this line of work and, likewise, highlight the linkage between bits in the BBs, but use the entropy concept; they study the connection between selection pressure and population size, that ratifies the concept of interdependence of parameters and operators in GAs. Other authors take advantage of solutions already known for problem of size $n-1$ and apply *seeding* in which, in order to solve the problem of size n, the algorithm is fed with the best individual—or more if there are more—as part of the possible solutions, and the rest of the solutions are generated randomly [22], [19].

Harik and Lobo [10] have addressed the problem of sizing the population using self-adaptation. In this case, there are two principal approaches: (1) the use of self-adaptation previous to running the algorithm [10] [3], in which case the population size remains the same during all the iterations and (2) the use of self-adaptation during the entire run of the GA, having different population sizes depending on parameters like the fitness value—that gives the lifetime of a chromosome—and the reproduction ratio that gives the proportion of increase in the population at each step [11].

This succinct presentation is finished mentioning maybe the most common method to initialize and size the initial population: that is the empirical method [7], in which the algorithm is tested with various population sizes and the one that gives best results is the one that is reported. However as Lobo and Goldberg [14] mention, the empirical method is used because of the difficulties of sizing the population with some of the methods just enumerated and because the problem itself can be difficult to estimate.

3 Initial Population

The first step in the functioning of a GA is, then, the generation of an initial population. Each member of this population encodes a possible solution to a problem. After creating the initial population, each individual is evaluated and assigned a fitness value according to the fitness function. However, we approach this article having in mind that the problem to find a good initial population and the optimal population size is a hard problem [7] and a general rule can not be applied to every type of problem or function to be evaluated [16].

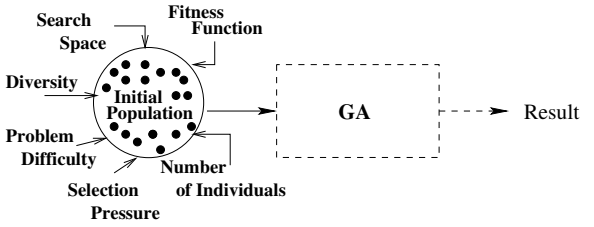

Fig. 1. Some factors to take into account when the initial population is generated randomly.

Figure 1 shows some factors that could influence the initial population or that should be taken into account when an initial population is generated randomly: the search space, the fitness function, the diversity, the problem difficulty, the selection pressure, and the number of individuals.

It has been recognized that if the initial population to the GA is good, then the algorithm has a better possibility of finding a good solution [4], [26] and that, if the initial supply of building blocks is not large enough or good enough, then it would be difficult for the algorithm to find a good solution [15]. Sometimes, if the problem is quite difficult and some information regarding the possible solution is available, then it is good to seed the GA with that information [5], [20], i.e., the initial population is seeded with some of those possible solutions or partial solutions of the problem. A measure of diversity plays a role here in the sense that, when we have no information regarding a possible solution, then we could expect, that the more diverse the initial population is, the greater the possibility to find a solution is, and of course, the number of individuals in the population to get a good degree of diversity becomes important.

Some authors have suggested that diversity could be good in terms of performance of the algorithm [4], [26], and diversity has been used not only to generate the initial population but also as a way to guide the algorithm to avoid premature convergence [8], [13].

The selection pressure should be taken into account in the initial population size [24]. We can say that, if a

selection pressure sp_1 is greater than a selection pressure sp_2, then, when using selection pressure sp_1 the population size should be larger than when using selection pressure sp_2, because a higher selection pressure can cause a decrease in diversity [10] of the population at a greater rate, perhaps causing the algorithm to converge prematurely.

The fitness function can be taken into account, in the sense that, depending on that, it could be better to generate the initial population in a pseudo-random way [21] than letting it go only with randomness or that there could be a degree of correlation between diversity and the fitness function [4]. Besides that, the fitness evaluation of the initial population can be used as a metric of diversity, looking, for example, at the initial standard deviation of fitness values and evaluating the dispersion of such values. For the present, however, we are not going to analyze the first fitness evaluation of the initial population as a metric because in this article we are not considering specific functions for evaluation.

4 Metrics to Evaluate Diversity

We consider basically three types of measures for a fixed length population of chromosomes: one at the gene-level, one at the chromosome-level, and one at the population level. At the gene-level, diversity is measured at each locus of the entire population; at the chromosome-level, diversity is measured in each chromosome of the entire population; and, at the population level, the position of each bit of each chromosome of the entire population is pondered.

4.1 Grefenstette Bias

In order to find *diversity* at the gene-level in a population, a formula that can be used is the *bias* measure suggested by Grefenstette [2] that is defined as

$$b(P(t)) = \frac{1}{l \cdot N} \sum_{j=1}^{l} max \left\{ \sum_{i=1}^{N} (1 - a_{i,j}), \sum_{i=1}^{N} a_{i,j} \right\} \tag{1}$$

where $b(P(t))$ is the $bias$ of the population $P(t)$ at time step t; l is the length of the chromosome, N is the number of individuals in the population, and $a_{i,j}$ is the j-gene of the i-individual.

We suggest a slight change to Equation 1 to

$$d(P(t)) = 2 * (1 - b(P(t))) \tag{2}$$

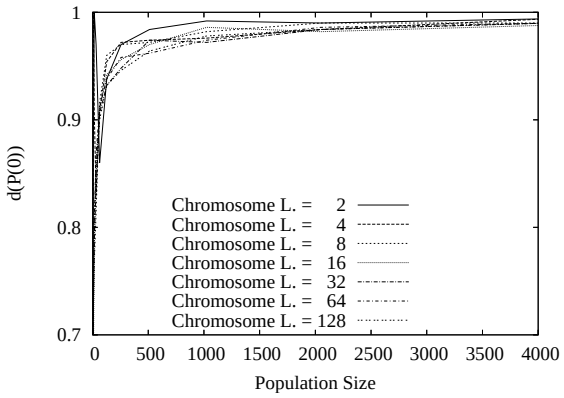

Fig. 2. Diversity value vs. chromosome length. Chromosomes Length from 2 to 128. As population size grows, Tendency of diversity as in Equation 2 becomes independent of chromosome length.

so we can get values from $[0.0, 1.0]$ where values near to 1 are those that are more diverse. (Diversity as in Equation 1 is in the range $[0.5, 1.0]$; the nearer to 0.5 the more diverse the population is [2]).

Equation 1 is such that if $N \gg l$, then the term l in the denominator vanishes because the term $max\{\sum_{i=1}^{N}(1 - a_{i,j}), \sum_{i=1}^{N} a_{i,j}\}$ dominates the equation—see Figure 2 for population size less or equal to $4,000$. Likewise, if the length $l \gg N$, then the term N vanishes. Formally, if $N \to \infty$ and the initial random generation of genes is uniformly distributed, then

$$b(P(0)) = \frac{1}{l \cdot N} \sum_{j=1}^{l} max \left\{ \frac{N}{2}, \frac{N}{2} \right\} = \frac{1}{l \cdot N} \frac{N \cdot l}{2} = \frac{1}{2} \tag{3}$$

That corresponds to $d(P(0)) = 1$ in Equation 2, a fact that shows that in order to obtain "good" diversity—according with Grefenstette formula—the number of individuals in the population should be "big enough."

It should be reinforced that Equation 2 can not be seen as if all individuals in the population are different. If, for example, in an initial population of 8 individuals, 4 individuals are 11111111 and 4 individuals are 00000000, then $d(P(0)) = 1.0$ (the best), but it turns out that there are only two types of individuals. However, all positions of the chromosome have been represented with the possible values 0 and 1. If, on the contrary, all individuals in the initial population different, as is the case of a base in an n-dimensional space, then $d(P(0)) = 2*\{1-[(n-1)\cdot l/l\cdot n]\} = 2*\{1-[(n-1)/n]\}$, clearly equal to 1.0 only when $n = 2$. These examples show some considerations that should be taken into account in Equation 2, if applied alone to measure

diversity in the initial population.

4.2 Gene-Level Entropy

One common measure of uncertainty is *entropy*, defined as [23]

$$H_j = -\sum_{i=1}^{N} p_{ij} * log_2 p_{ij} \tag{4}$$

where p_{ij} is the probability of occurrence of independent event $a_{i,j}$, and N is the number of trials (that is the population size in this case).

A measure of entropy at the gene-level for a population could, then, be then the equation

$$H(P(0)) = \frac{1}{l}\sum_{j=1}^{l} H_j \tag{5}$$

where H_j corresponds to the entropy as in Equation 4 for locus j of the entire population.

Looking at entropy values for different chromosome lengths, we find the same patterns as just discussed in Section 4-1—as l and N increase, entropy values are quite similar independently of l and N. However, it seems that Equation 2 is stronger in the measure of diversity—at gene-level—than Equation 5. Equation 4 (which is part of Equation 5)—takes into account the probability of occurrence of $1's$ and $0's$ for each locus in the entire population. On the contrary, Equation 2 takes into account the maximum number of $1's$ or $0's$, whichever it is, for each locus of the entire population. For example, for 3 bits with values 1, 0, 0, according to equation 4, $H = -(\frac{1}{3}log_2\frac{1}{3} + \frac{2}{3}log_2\frac{2}{3}) = -(-0.5283-0.39) = 0.9183$ and, according to Equation 2, $d = 2 * (1 - \frac{2}{3}) = \frac{2}{3} = 0.6667$.

4.3 Chromosome-Level Hamming Distance

The *Hamming distance* measures the number of bits at which two individuals differ; it is defined as [2] [8]

$$\rho_H(C_1, C_2) = \sum_{k=1}^{l}(|C_{1,k} - C_{2,k}|) \tag{6}$$

where C_1, C_2 are chromosomes and, $C_1, C_2 \in \{0,1\}^l$.

In order to calculate an average Hamming distance, we need to calculate the Hamming distance between each pair of chromosomes in the population. The Hamming distance of a population of size N, averaged by the total number of computations ($(N-1) * N/2$) is

$$\Gamma_H(P(0)) = \frac{2 * \sum_{i=1}^{N-1}\sum_{j=i+1}^{N}\rho_H(C_i, C_j)}{N * (N-1)} \tag{7}$$

For a uniformly randomly generated population, the average Hamming distance tends to $l/2$, because it is assumed that half of the bits are $1's$, half are $0's$, and that they are through equally distributed. Two chromosomes are totally different, with this metric, if their Hamming distance is l. So the nearer to l the better. Cases where $\Gamma_H(P(0)) < l/2$ may be an indication of poor diversity at the chromosome level.

Taking again the example of 8 individuals, the first 4 being all genes $1's$ and the last 4 being $0's$, it is obtained $\Gamma_H(P(0)) = 2 * (8*4 + 8*4 + 8*4 + 8*4)/(8*7) = 4.5714$. This value is greater than $l/2$, and that could be considered as a good grade of diversity. However, for the GA's purposes, there are only two types of individuals.

4.4 Chromosome-Level Neighborhood Metric

We can use the Hamming distance with the concept of *neighborhood* defined by Bäck [2], as a metric of diversity at the chromosome-level

$$B_k(C_m) = \{C_j \in \{0,1\}^l | \rho_H(C_m, C_j) = k\} \tag{8}$$

where $k \in \{0, 1, ..., l\}$ and $\rho_H(C_m, C_j)$ is as in Equation 6.

Equation 8 gives the neighbors of a chromosome C_m at a distance equal to k. In other words, if, for example, we want to find the individuals equal to C_m, then $k = 0$.

In order to use Equation 8, a pivot C_m is needed and, after that, the set of neighbors at each distance k from the pivot is found. The cardinality $|B_k(C_m)|$ of each set is multiplied by the corresponding distance k, added, and the result is finally divided by the sum of all cardinalities to obtain:

$$\mathcal{N}(C_m) = \frac{\sum_{k=0}^{l}|B_k(C_m)| * k}{\sum_{k=0}^{l}|B_k(C_m)|} \tag{9}$$

with $|B_k(C_m)|$ the cardinality of $B_k(C_m)$, i.e., the number of neighbors of C_m at a distance k in the whole population.

It is appreciated—using Equation 9—that as the population size grows, the tendency of the chromosome-level neighborhood is toward a distance of $l/2$ and the standard deviation is smaller, there is a concentration of points around the median point C_m at a distance $k = l/2$.

Again take the example of a population of 8 individuals, 4 of which are $1's$ and the last ones are $0's$. In order to calculate the neighborhood $B_k(C_m) =$

$\{C_j \in \{0,1\}^l | \rho_H(C_1, C_j) = k\}$, a pivot is needed; that could be any chromosome in the initial population. If we choose a midpoint as the chromosome of all $0's$, then C_m is going to have 3 neighbors at a distance $k = 0$, and 4 neighbors at a distance $k = 8$, then the average neighborhood distance as in Equation 9 is $(3*0+4*8)/(3+4) = 4.5714$; that is exactly the same as the Hamming distance metric calculated in Section 4-3.

4.5 Population-Level Center of Mass

We are proposing in this article a population-level diversity metric that looks at all genes in a population as a matrix of two types of genes (particles) and calculates the center of mass with respect to an origin $(0,0)$—that is located at the left-top of the matrix. For the case of the x coordinate of the center of mass for a gene with value 1, it is suggested

$$\overline{x}_1 = \frac{\sum_{i=1}^{N}\sum_{j=1}^{l} C(a_{i,j})}{\sum_{i=1}^{N}\sum_{j=1}^{l} a_{i,j}} \tag{10}$$

where $C(a_{i,j})$ is the *column position* j of gene $a_{i,j}$ where the gene has value 1, and $\sum_{i=1}^{N}\sum_{j=1}^{l} a_{i,j}$ is the number of those genes (i.e., where $a_{i,j} = 1$).

In order to obtain the y coordinate for a gene of type 1 a similar equation is suggested

$$\overline{y}_1 = \frac{\sum_{j=1}^{l}\sum_{i=1}^{N} R(a_{i,j})}{\sum_{j=1}^{l}\sum_{i=1}^{N} a_{i,j}} \tag{11}$$

where $R(a_{i,j})$ is the *row position* i of gene $a_{i,j}$ where the gene has value 1, and $\sum_{j=1}^{l}\sum_{i=1}^{N} a_{i,j}$ is the number of such genes as in Equation 10.

Equations 10 and 11 can be applied taking into account $0's$ instead of $1's$, in order to find the center of mass with respect to gene value 0. If the number of $1's$ and the number of $0's$ are uniformly distributed then it is expected that $(\overline{x_1}, \overline{y_1}) \approx (\overline{x_0}, \overline{y_0}) \approx (\frac{l}{2}+\frac{1}{2}, \frac{N}{2}+\frac{1}{2})$. The $\frac{1}{2}$ comes from the fact that the first gene is considered at a position $(1,1)$; however if we consider the first gene to be at position $(0,0)$ the origin, then the fraction $\frac{1}{2}$ should be dropped.

Let us analyze the special case presented in section 4-1, where we have an initial population of 8 individuals, the first 4 of which are 11111111 and the last 4 individuals are 00000000. According to Equation 10—for the case of gene 1, $\overline{x}_1$ coordinate—$\sum_{j=1}^{8} C(a_{i,j}) = 1+2+3+4+5+6+7+8 = 36$ and, as there are 4 chromosomes with $1's$, then, the numerator is $36*4$. As $\sum_{i=1}^{N}\sum_{j=1}^{l} a_{i,j} = 8*4$ then, it is obtained $\overline{x}_1 = 36*4/8*4 = 4.5$. The same procedure applied in order to find the y_1 coordinate gives $\overline{y}_1 = 2.5$. This example gives as a result $(\overline{x}_1, \overline{y}_1) = (4.5, 2.5)$, i.e., the center of mass, for gene 1, is exactly in the middle coordinate of the first 4 chromosomes, with respect to origin $(0,0)$. If the same calculation is performed for the center of mass for gene value 0, it is obtained $(\overline{x}_0, \overline{y}_0) = (4.5, 6.5)$. Clearly the $y's$ coordinates of the two centers of mass for 1 and 0, are showing an "unbalance" ($\overline{y}_1 = 2.5 \neq \overline{y}_0 = 6.5$) in the distribution of $1's$ and $0's$ in the matrix. At the x coordinate, the population is "balanced", i.e., $\overline{x}_1 = 4.5 = \overline{x}_0$.

We are suggesting to use this metric in *uniformly randomly generated genes* in a population because there could be cases—non randomly generated—where we get a perfect center of mass $(\frac{l}{2}+\frac{1}{2}, \frac{N}{2}+\frac{1}{2})$ for x and y coordinates but the population couldn't be diverse at all. As a matter of example, if we have $N = 11$ chromosomes with length $l = 11$, and genes with value 1 are in positions $(1,1)$, $(1,12)$, $(12,1)$, and $(12,12)$, and the rest of the genes are equal to 0, then the center of mass with respect to origin $(0,0)$ is $(\overline{x}_1, \overline{y}_1) = (\overline{x}_0, \overline{y}_0) = (6,6)$. However, extreme cases like a population equal to a base in the $\mathcal{B}^n$ space, is well captured by the center of mass, but not for other metrics just studied—see Section 4-1.

5 Analysis of Metrics

Studying metrics at gene level, similarities are observed among them when diversity is computed with two bits. Gene-level Grefenstette and gene-level entropy evaluate two bits with the same value. Bits 00 and 11 are evaluated with diversity/entropy value of 0, and bits 01 and 10 are evaluated as 1 (see Equations 2 and 5). With more than two bits, as is shown in Section 4-2, the evaluation differs for all combinations where there are different proportion of $1's$ with respect to $0's$, and it turns out that gene-level Grefenstette is stronger[1] than gene-level entropy. The computation time needed in order to evaluate these metrics is $\mathcal{O}(N*l)$, where N is the size of the population and l is the chromosome length.

Chromosome-Level neighborhood metric—as in Equation 9—takes less computation time ($\mathcal{O}(N*l)$) than Chromosome-level Hamming distance diversity

[1] *Stronger* in this context means that diversity values for the same set of genes are graded lower for the stronger diversity metric than the other one under consideration.

$\mathcal{O}(N^2 * l)$, a fact that leads us to suggest using it at the Chromosome-Level. Both metrics are computing diversity at the chromosome level and both are using the Hamming distance; this makes the corresponding chromosome-level diversity values quite similar—at least for the current study.

Population-level diversity can capture the distribution of $1's$ and $0's$ in the whole population taking into account the structure of it. In order to see this, let us take the example of three individuals 101, 000, 101 [6]. In this case the Grefenstette metric as in Equation 2 evaluates to 0.4444, the Entropy metric as in Equation 5 evaluates to 0.6122, the Hamming metric as in Equation 7 evaluates to 1.3333, the Neighborhood metric evaluates to 2.000, and the center of mass with respect to $1's$ and $0's$ evaluates to $(2, 2)$. If we change the structure as 110, 110, 000, then all metrics evaluate the same except the center of mass, that evaluates $(\overline{x}_1, \overline{y}_1) = (1.5, 1.5)$ and $(\overline{x}_0, \overline{y}_0) = (2.4, 2.4)$. The center of mass with respect to 1 and the center of mass with respect to 0 should tend to the middle of the matrix, and if that is not the case, that could be an indication that there is some bias toward a specific region of the search space, as this is the case. However, what is a benefit for center of mass, could be a bias in small populations, as the case where we have a small change like 101, 010, 101 where the center of mass with respect to $1's$ and $0's$ is the same as in 101, 000, 101. The computational complexity of population diversity is $\mathcal{O}(N * l)$, where N is the size of the population and l is the chromosome length.

6 Conclusions and Future Work

The evaluation of diversity in a random population, using some of the metrics analyzed in this paper, depends on the population size and chromosome length. However, as the population size and chromosome length increase, diversity becomes independent of them, and this is part of the difficulty of sizing an initial random population using only this approach. So, for small population sizes, one of the previous methods could be applied and, perhaps for big population sizes, we could measure diversity in a small portion of it or try to generate chromosomes uniformly distributed in the fitness landscape.

We have suggested evaluating the initial population and, depending of the problem we are solving, we can choose gene-level diversity, chromosome-level diversity, population-level diversity, or a combination of those, having in mind that some metrics are more computationally expensive than others. We have proposed a measure of population-level diversity that measures gene-level and chromosome level at the same time, with the $\overline{y}$ and $\overline{x}$ coordinate of the center of mass respectively. It is expected that the computation time invested in calculating diversity and analyzing the initial population is going to be rewarded in the quality of the solution and number of iterations to get it.

We have not taken in this study the evaluation of the landscape the GA has to travel. This does not mean it is not important for us but certainly an initial evaluation of the initial population with the fitness function is another metric that could be considered as another statistical measure. Empirical research must be done in order to test some of the conjectures and suggestions of the present. Also any effort done in the analysis of the initial population is going to impact the possible finding of a solution, in the sense that the initial population is the input and, as such, it is expected that from there, any point in the search space should be reachable [21].

7 Acknowledgments

Thanks to Abraham Bagher, from the University of Houston, for help with calculating entropy of a population, and Dr. Dee Wu from the OU Children's Hospital, for help with calculating the center of mass.

8 References

[1] J. T. Alander, "On optimal population size of genetic algorithms," in *Proceedings of the IEEE Computer Systems and Software Engineering*, 1992, pp. 65–69.

[2] T. Bäck, *Evolutionary Algorithms in Theory and Practice*. Oxford University Press, 1996.

[3] T. Bäck, A. Eiben, and V. der Vaart, "An empirical study on GAs "without parameters"," in *Proceedings of the 6th International Conference on Parallel Problem Solving from Nature*, 2000, pp. 315–324.

[4] E. K. Burke, S. Gustafson, and G. Kendall, "Diversity in genetic programming: An analysis of measures and correlation with fitness," *IEEE Transactions on Evolutionary Computation*, vol. 8, no. 1, pp. 47–62, 2004.

[5] D. A. Casella and W. D. Potter, "New lower bounds for the snake–in–the–box problem: Using evolutionary techniques to hunt for snakes," in *Proceedings of the Florida Artificial Intelligence Research Society Conference*, 2004, pp. 264–268.

[6] K. Deb and S. Jain, "Running performance metrics for evolutionary multi-objective optimization," Indian Institute of Technology Kanpur, Tech. Rep., 2004.

[7] A. E. Eiben, R. Hinterding, and Z. Michalewicz, "Parameter control in evolutionary algorithms," *IEEE Transactions on Evolutionary Computation*, vol. 3, no. 2, pp. 124–141, 1999.

[8] W. G. Frederick, R. L. Sedlmeyer, and C. M. White, "The hamming metric in genetic algorithms and its application to two network problems," in *Proceedings of the ACM/SIGAPP Symposium on Applied Computing*, 1993, pp. 126–130.

[9] D. E. Goldberg, K. Deb, and J. H. Clark, "Genetic algorithms, noise, and the sizing of populations," *Complex Systems*, vol. 6, pp. 333–362, 1992.

[10] G. R. Harik and F. G. Lobo, "A parameter-less genetic algorithm," in *Proceedings of the Genetic and Evolutionary Computation Conference*, 1999, pp. 258–265.

[11] Z. M. Jaroslaw Arabas and J. Mulawka, "GAVaPS—a genetic algorithm with varying population size," in *Proceedings of the IEEE International Conference on Evolutionary Computation*, 1995, pp. 73–78.

[12] V. K. Koumousis and C. P. Katsaras, "A saw-tooth genetic algorithm combining the effects of variable population size and reinitialization to enhance performance," *IEEE Transactions on Evolutionary Computation*, vol. 10, no. 1, pp. 19–28, 2006.

[13] Y. Leung, Y. Gao, and Z. Xu, "Degree of population diversity—a perspective on premature convergence in genetic algorithms and its Markov chain analysis," *IEEE Transactions on Neural Networks*, vol. 8, no. 5, pp. 1165–1176, 1997.

[14] F. G. Lobo and D. E. Goldberg, "The parameter-less genetic algorithm in practice," *Information Sciences—Informatics and Computer Science*, vol. 167, no. 1-4, pp. 217–232, 2004.

[15] F. G. Lobo and C. F. Lima, "A review of adaptive population sizing schemes in genetic algorithms," in *Proceedings of the Genetic and Evolutionary Computation Conference*, 2005, pp. 228–234.

[16] M. Lunacek and D. Whitley, "The dispersion metric and the CMA evolution strategy," in *Proceedings of the Genetic and Evolutionary Computation Conference*, 2006, pp. 447–484.

[17] M. Pelikan, D. E. Goldberg, and E. Cantú-Paz, "Bayesian optimization algorithm, population sizing, and time to convergence," Illinois Genetic Algorithms Laboratory, University of Illinois, Tech. Rep., 2000.

[18] A. Piszcz and T. Soule, "Genetic programming: Optimal population sizes for varying complexity problems," in *Proceedings of the Genetic and Evolutionary Computation Conference*, 2006, pp. 953–954.

[19] W. D. Potter, R. W. Robinson, J. A. Miller, K. Kochut, and D. Z. Redys, "Using the genetic algorithm to find snake-in-the-box codes," in *Proceedings of the 7th International Conference on Industrial & Engineering Applications of Artificial Intelligence and Expert Systems*, 1994, pp. 421–426.

[20] D. S. Rajan and A. M. Shende, "Maximal and reversible snakes in hypercubes," in *Proceedings of the 24th Annual Australasian Conference on Combinatorial Mathematics and Combinatorial Computing*, 1999.

[21] C. R. Reeves, "Using genetic algorithms with small populations," in *Proceedings of the 5th International Conference on Genetic Algorithms*, 1993, pp. 92–99.

[22] K. Sastry, "Efficient cluster optimization using extended compact genetic algorithm with seeded population," in *Proceedings of the Genetic and Evolutionary Computation Conference*, 2001, pp. 222–225.

[23] C. E. Shannon, "A mathematical theory of communication," *The Bell System Technical Journal*, vol. 27, pp. 379–423, 623–656, 1948.

[24] T.-L. Yu, K. Sastry, D. E. Goldberg, and M. Pelikan, "Population sizing for entropy-based model building in genetic algorithms," Illinois Genetic Algorithms Laboratory, University of Illinois, Tech. Rep., 2006.

[25] P. Zezula, G. Amato, V. Dohnal, and M. Batko, *Similarity Search. The Metric Space Approach*. USA: Springer, 2006.

[26] E. Zitzler, K. Deb, and L. Thiele, "Comparison of multiobjective evolutionary algorithms: Empirical results," *Evolutionary Computation*, vol. 8, no. 2, pp. 173–195, 2000.