A Saw-Tooth Genetic Algorithm Combining the Effects of Variable Population Size and Reinitialization to Enhance Performance

Vlasis K. Koumousis and Christos P. Katsaras

***Abstract*—A genetic algorithm (GA) is proposed that uses a variable population size and periodic partial reinitialization of the population in the form of a saw-tooth function. The aim is to enhance the overall performance of the algorithm relying on the dynamics of evolution of the GA and the synergy of the combined effects of population size variation and reinitialization. Preliminary parametric studies to test the validity of these assertions are performed for two categories of problems, a multimodal function and a unimodal function with different features. The proposed scheme is compared with the conventional GA and micro GA (μGA) of equal computing cost and guidelines for the selection of effective values of the involved parameters are given, which facilitate the implementation of the algorithm. The proposed algorithm is tested for a variety of benchmark problems and a problem generator from which it becomes evident that the saw-tooth scheme enhances the overall performance of GAs.**

***Index Terms*—Genetic algorithm (GA), evolutionary computation, optimization methods, population reinstallation.**

I. Introduction

GENETIC algorithms (GAs) are search algorithms based on the concept of natural selection [1], [2]. The efficiency of GAs relies heavily on the successful selection of a number of parameters, such as population size (n), selection scheme, type of crossover, probability of crossover (p_c), probability of mutation (p_m), etc. A number of methods have been developed to improve the robustness and computational efficiency of GAs [3]. Moreover, an effort to classify the developed methodologies that control the parameters of evolutionary algorithms was presented by Eiben *et al.* [4].

A simple GA uses a population of constant size and guides the evolution of a set of randomly selected individuals through a number of generations that are subject to successive selection, crossover, and mutation, based on the statistics of the generation (standard GA). Population size is one of the main parameters that affect the robustness and computational efficiency of the GAs. Small population sizes may result in premature convergence to nonoptimal solutions, whereas large population sizes give a considerable increase of computational effort. Expressions estimating the optimal population size, based on various parameters that account for the complexity of the problem, were suggested by Goldberg *et al.* [5]. Moreover, Harik and Lobo [6] have proposed a parameterless GA which runs multiple populations of increasing size in a cascade-like manner and eliminates the ones that converge.

Variable population size GAs have been proposed in the literature. Smith [7] proposed an algorithm that adjusts the population size based on the probability of selection error. Koumousis and Dimou [8] applied a sinusoidal oscillating population size for structural problems such as the reliability based optimal design of trusses. The variable population size scheme improved the capacity of the GA to refine its solutions and reduced the sensitivity in selecting the GA parameters.

Several methods have been proposed in the literature that attempt to increase the diversity of the population and avoid premature convergence. Cobb and Grefenstette [9] have examined alternative mutation strategies, such as replacing part of the population of each generation with random individuals and triggered hypermutation whenever there is degradation in the performance. Eshelman's CHC algorithm [10] represents a GA with elitist selection and a highly disruptive recombination operator which restarts the search (reinitialization) when the population diversity drops below a threshold.

Another GA that utilizes population reinitialization, suggested by Goldberg [11], is the so-called micro GA or μGA. In general, μGA is a small population GA which evolves for many generations. When after a number of generations the GA population converges, the evolutionary process is reinitialized by preserving the best individual and substituting the rest of the population with randomly generated individuals. The first implementation of μGA was reported by Krishnakumar [12], who used a population size of five individuals, tournament selection, single-point crossover with probability $p_c = 1.0$, elitism, and no mutation. The population was considered converged when less than 5% of the population bits were different from the bits of the best individual. Krishnakumar has shown that μGA can avoid premature convergence and performs better than a simple GA for selected multimodal problems. Carroll [13] has shown that uniform crossover improves the performance of μGA. Several other applications of μGA have appeared in the literature ([14]–[16]) that also resulted in a better performance as compared to the standard GA.

In this paper, a variable population size with periodic reinitialization is proposed that follows a saw-tooth scheme with a specific amplitude and period of variation (saw-tooth GA). In each period, the population size decreases linearly and at the beginning of the next period randomly generated individuals are appended to the population. The reinitialization of the process

Manuscript received December 12, 2003; revised August 3, 2004.
The authors are with the National Technical University of Athens, Athens GR-157 80, Greece (e-mail: vkoum@central.ntua.gr).
Digital Object Identifier 10.1109/TEVC.2005.860765

at specific periods resembles the features of the μGA. Therefore, comparison of the proposed scheme is based on both the standard GA and μGA of equal computing cost.

II. Variable Population Size

Varying the population size between two successive generations affects only the selection operator of the GA. Let n_t and n_{t+1} denote the population size of the current and the subsequent generation, respectively. The selection of the individuals can be considered as a repetitive process of n_{t+1} selection operations, with p_j being the probability of selection of the jth individual. For most of the selection operators, such as fitness proportionate selection and tournament selection with replacement, the selection probability p_j remains constant for the n_{t+1} selection operations. The expected number of copies of the jth individual after selection can be expressed as

$$m(j, t+1) = m(j, t)\, p_j n_{t+1} \tag{1}$$

where $m(j,t)$ is the number of copies of the jth individual at generation t. The expected number of the jth individual is directly proportional to the population size of the subsequent generation. Therefore, the portion of the population related to the jth individual after selection can be expressed as

$$\rho(j, t+1) = \frac{m(j, t+1)}{n_{t+1}} = \frac{m(j, t)\, p_j n_{t+1}}{n_{t+1}} = m(j, t)\, p_j \tag{2}$$

which is independent of the population size of the subsequent generation, provided that the variation of the population size is not significant enough to modify the probability p_j. However, a GA with decreasing population size has bigger initial population size and smaller final population size, as compared to a constant population size GA with the same computing cost (i.e., equal average population size). This is expected to be beneficial, because a bigger population size at the beginning provides a better initial signal for the GA evolution process; whereas, a smaller population size is adequate at the end of the run, where the GA converges to the optimum. The above ascertainment motivated this paper, offering good arguments for a successful statistical outcome.

To study closely the effect of variable population size on the performance of GAs, three different population size schemes are compared. The first has a constant population size of $n = 100$ individuals, the second has a linearly decreasing population size from $n_1 = 150$ to $n_2 = 50$, and the third has a linearly decreasing population size from $n_1 = 195$ to $n_2 = 5$, where n_1 is the initial population size and n_2 is the final population size. All three cases evolve for 200 generations. The three cases are applied using a typical GA for the case of the Goldberg and Richardson [17] multimodal function given in Appendix A (tournament selection of two individuals with replacement, single point crossover with $p_c = 0.85$, bit-to-bit mutation with $p_m = 0.019$, and elitism).

In Fig. 1, the average and the maximum fitness time histories are presented for the three cases. Each curve corresponds to the mean of 100 runs with different random seeds. The fitness histories are plotted versus the actual function evaluations (i.e., individuals that are not modified by mutation or recombination are not re-evaluated). It is observed that the linearly decreasing schemes attain an overall better performance (3.4% better final fitness) for the same computing cost as compared to the constant population size algorithm. Near the end of the run the average fitness of the linearly decaying schemes evolves in a wavy manner. This is because the variation of the population size becomes significant and the behavior deviates from the average trend expressed by (2). The statistical significance of the results is examined by performing a t test between the results of different algorithms (i.e., probability that the difference in the two observed means is smaller than the observed value purely by chance—one tailed test, where 0.00 stands for samples from different populations and 0.50 from the same population). The performance level of the variable population size schemes is distinct from the constant population size scheme as indicated by the t-test values (0.13 between the constant population size case with $n = 100$ and the variable population size with $n_1 = 150$, $n_2 = 50$, and 0.00 for the comparison between $n = 100$ and $n_1 = 195$, $n_2 = 5$). This indicates that for bigger differences between the initial and final population sizes the performance is increased.

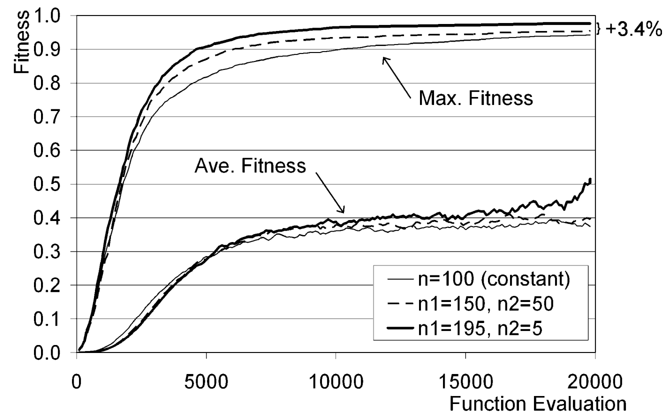

Fig. 1. Effect of variable population size.

III. Population Reinitialization

Reinitialization of the population of a GA at a specific generation occurs when a portion of the population is substituted with new randomly generated individuals. This portion may be selected randomly; however, it is preferable to select the n' least fit individuals, where $1 \leq n' \leq n-1$ and n is the population size. The effect of population reinitialization is in a sense similar to the mutation operator. Both operators introduce random changes in the population to increase diversity and achieve better exploration of the search space. The mutation operator is applied at every bit of all chromosomes with a small probability; whereas, population reinitialization introduces new randomly generated chromosomes and modifies drastically the subsequent population. This effect is favorable when the GA population has prematurely converged to a certain point or local optimum and further improvement is not likely. Therefore, population reinitialization represents a good strategy especially for the case of multimodal problems.

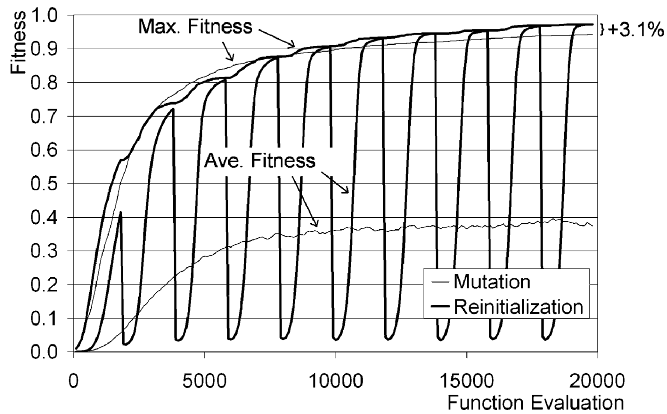

Fig. 2. Effect of population reinitialization.

To study closely the effect of population reinitialization, the constant population size algorithm presented in the previous paragraph is compared with a GA that performs eight reinitializations every 20 generations, instead of mutation, which occur at generations 20, 40,…,180. This is established using the same average number of modified bits for the Goldberg and Richardson [17] multimodal function by utilizing the following relation:

$$t_{\max} n p_m l = 8 n' p'_m l \tag{3}$$

where $t_{\max} = 200$ is the total number of generations, $n = 100$ is the population size, $p_m = 0.019$ is the probability of mutation, and l is the length of the chromosome. Moreover, n' is the number of individuals substituted with new ones at every reinitialization with their bits modified in average with a probability $p'_m = 0.5$. Solving relation (3) with respect to n', the following relation is obtained:

$$n' = \frac{t_{\max} n p_m}{8 p'_m}. \tag{4}$$

This corresponds to $n' = 95$ individuals for this particular study. Both schemes are applied for the case of the Goldberg and Richardson [17] multimodal function (Appendix A).

In Fig. 2, the average and the maximum fitness time histories are presented for the mutation case and the reinitialization case. Each curve corresponds to the mean of 100 runs with different random seeds. For the case of the reinitialization scheme, the average fitness oscillates from a very low minimum value, after reinitialization has occurred, to a very high maximum value, when the population has converged. The maximum fitness advances in steps between the reinitializations, which correspond to better solutions evolving by the recombination of the best individuals with the randomly generated ones. It is observed that the reinitialization scheme attains an overall better performance (3.1% better final fitness) for the same computing cost as compared to the mutation case. Furthermore, a t-test value of 0.00 indicates that the two schemes reach distinct levels of performance. The results of this paragraph are indicative of the influence of reinitialization for multimodal problems; although, in general, equivalence between the two distinct operations of mutation and reinitialization cannot be established. In fact, the proposed saw-tooth GA reaches better performance by utilizing both a low mutation scheme and reinitialization.

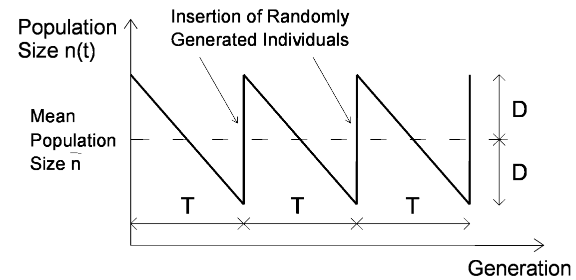

Fig. 3. Population variation scheme of saw-tooth GA.

IV. Proposed Saw-Tooth GA

Based on the previous remarks, a variable population size scheme combined with population reinitialization can be introduced to the GA to improve further its performance. The proposed GA utilizes a variable population size following the periodic scheme presented in Fig. 3 in the form of a saw-tooth function. The mean population size $\overline{n}$ of the periodic scheme corresponds to the constant population size GA having the same computing cost. Moreover, the scheme is characterized by the amplitude D and the period of variation T. Thus, at a specific generation t, the population size $n(t)$ is determined as

$$n(t) = \operatorname{int}\left\{\overline{n} + D - \frac{2D}{T-1}\left[t - T\operatorname{int}\left(\frac{t-1}{T}\right) - 1\right]\right\}. \tag{5}$$

Therefore, $n(1) = \overline{n} + D, n(T) = \overline{n} - D, n(T+1) = \overline{n} + D$, etc. The saw-tooth GA scheme was easily implemented into the Fortran source code developed by Carroll [18].

The selection of the $\overline{n}$, T, and D parameters affects the performance of the saw-tooth GA. For amplitude, $D = 0$, regardless of the period T, and the proposed algorithm is reduced to a constant population size GA. For bigger amplitude values D, the population size decreases with a constant decay and reinitialization is enforced every T generations. The effect of population reinitialization is more drastic as the amplitude D increases from zero to $\overline{n} - 1$. Moreover, the selection of the period T is critical as it controls the duration of the decay process before reinitialization occurs. Parametric studies in the subsequent paragraphs determine the optimum range and the sensitivity of the saw-tooth GA parameters.

Combining the effects of population size variation and reinitialization alters the evolution dynamics of the GA in a way that is expected to enhance the overall performance [19]. In Fig. 4, the constant population size GA (standard GA), the variable population size GA, and the population reinitialization GA presented in the previous paragraphs are compared to a saw-tooth GA scheme of period $T = 20$ and amplitude $D = 95$ that represents the combination of variable population size and reinitialization (100 runs, Goldberg and Richardson [17] multimodal function). For comparison reasons, no mutation was used in the saw-tooth GA, although mutation generally improves its performance. The saw-tooth GA outperforms the standard GA from the beginning of the runs. After 20 000 function evaluations, the four different algorithms reach a mature

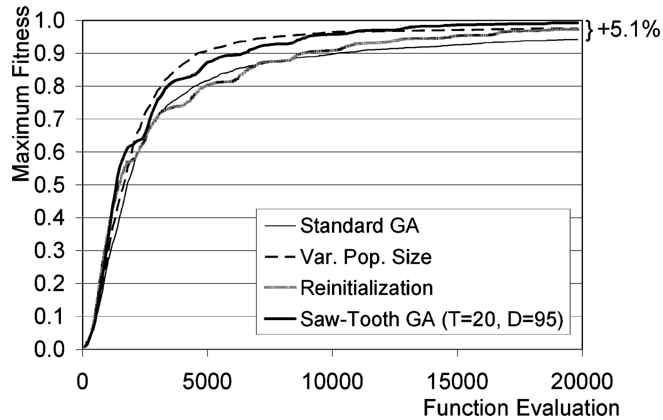

Fig. 4. Contribution of variable population size and reinitialization to performance of saw-tooth GA scheme.

Fig. 5. Performance comparison of algorithms (Goldberg and Richardson multimodal function).

state which permits a fair comparison. The improvement of the final fitness for each case as compared to the standard GA is as follows: variable population size is $+3.4\%$, population reinitialization is $+3.1\%$, combined effects of variable population size and population reinitialization (i.e., saw-tooth GA) result in $+5.1\%$. A mutation level of 0.005 further improved the performance of the saw-tooth GA to $+5.4\%$. Furthermore, the saw-tooth GA scheme reaches a distinct level of performance versus all the other schemes as indicated by the 0.00 values of the corresponding t tests. In addition, the rank of the algorithms does not change if more function evaluations are performed. After 60 000 function evaluations, the saw-tooth GA is the first algorithm to reach the optimal performance (i.e., 100 out of 100 runs find the global optimum); whereas, the standard GA still improves.

The periodic scheme of the saw-tooth GA can be applied to any existing GA. However, it is preferable to use elitism and a selection scheme such as tournament selection. The reason is that after reinitialization the mating pool may contain individuals of very different fitness values, especially if the amplitude D is very big. To avoid premature convergence, the selection scheme of the genetic algorithm should be insensitive to this fact. Fitness proportionate selection such as a roulette wheel is not appropriate, as the individuals with high fitness will easily take over the population in very few generations, preventing a smooth evolutionary process. On the other hand, selection schemes such as tournament selection and ranking schemes are suitable because they reproduce individuals by comparing the fitness rank rather than using absolute fitness values. For example, in tournament selection (with the tournament size of two individuals) the best individual participates in two tournaments, on average, and wins both of them. After selection, the best individual will get two copies on average, no matter how big the fitness value is, as compared to the rest of the population.

Elitism ensures that the best individual is copied to the subsequent population. This operator is useful after the population reinitialization, where the good solutions may constitute a small portion of the population that may be accidentally lost when recombined with the new individuals. In general, mutation improves the saw-tooth GA performance.

V. Parametric Study for Multimodal Problem

To evaluate the performance of the proposed algorithm, a parametric study is conducted for the case of the Goldberg and Richardson multimodal function given in Appendix A. The influence of the three parameters is examined, namely the mean population size $\overline{n}$, the period T, and the amplitude D. The proposed saw-tooth GA is compared to the standard GA and μGA of equal computing cost. To avoid an unfair comparison between the algorithms, the critical parameters of the standard GA and μGA were included in the parametric study with no restriction to the number of modified bits by mutation and reinitialization imposed by (3). In particular, for the standard GA the population size n, the probability of crossover p_c, and the probability of mutation p_m are included. Furthermore, for the μGA, the population size n and the value of the reinitialization criterion p_r (i.e., the percentage of the population bits different than the bits of the best individual) are considered. Due to the reinitializations, elitism is essential for μGA and the saw-tooth GA. Moreover, elitism improved considerably the performance of the standard GA for the examined problem. For each of the aforementioned cases, 25 runs having different random seeds are performed. Every run of the parametric study is terminated at the generation when roughly 10 000 function evaluations have been performed, which sets an equal computing cost basis for the comparison. While all the major parameters of the algorithms are examined in the parametric study, the function evaluation limit is kept equal to 10 000, which corresponds to practical optimization problems, where fitness evaluation is time consuming. The different cases examined are presented in Table I. They correspond to 7662 cases, i.e., 191 550 runs and approximately 1.92×10^9 function evaluations.

In Fig. 5, the final fitness (mean value of the 25 runs) is presented for the best performing case of each algorithm with respect to the population size. For the Goldberg and Richardson multimodal function, μGA performed better than the standard GA, which verifies the suitability of μGA for multimodal problems. Furthermore, μGA performs better for small population sizes (15 to 30) versus the standard GA for moderately big population sizes (150 to 250). The proposed saw-tooth GA performed better than the standard GA and μGA for almost every population size. Moreover, the performance of the saw-tooth GA ap-

TABLE I
PARAMETRIC STUDY CASES

Standard GA: tournament selection, single point crossover, bit-to-bit mutation, elitism	
n=	5, 10, 20, 30, 40, 50, 60, 70, 80, 90, 100, 125, 150, 175, 200, 225, 250
p_c=	0.50, 0.55, 0.60, 0.65, 0.70, 0.75, 0.80, 0.85, 0.90, 0.95, 1.00
p_m=	0.000, 0.005, 0.010, 0.015, 0.020, 0.025, 0.030, 0.035, 0.040, 0.045, 0.050, 0.100, 0.150, 0.200
μGA: tournament selection, single point crossover p_c=1.0, elitism, reinitialization	
n=	5, 10, 15, 20, 25, 30, 35, 40, 45, 50, 60, 70, 80, 90, 100, 125, 150, 175, 200, 225, 250
p_r=	0.01, 0.02, 0.03, 0.04, 0.05, 0.06, 0.07, 0.08, 0.09, 0.10, 0.11, 0.12, 0.13, 0.14, 0.15, 0.16, 0.17, 0.18, 0.19, 0.20
Saw-Tooth GA: tournament selection, single point crossover p_c=0.90 bit-to-bit mutation p_m=0.010, elitism	
$\bar{n}$=	5, 10, 20, 30, 40, 50, 60, 70, 80, 90, 100, 125, 150, 175, 200, 225, 250
$T/\bar{n}$=	0.2, 0.3, 0.4, 0.5, 0.6, 0.7, 0.8, 0.9, 1.0, 1.1, 1.2, 1.3, 1.4, 1.5, 1.6, 1.7
$D/\bar{n}$=	0.00, 0.10, 0.20, 0.30, 0.40, 0.50, 0.60, 0.70, 0.75, 0.80, 0.85, 0.90, 0.92, 0.94, 0.96, 0.98, 0.99

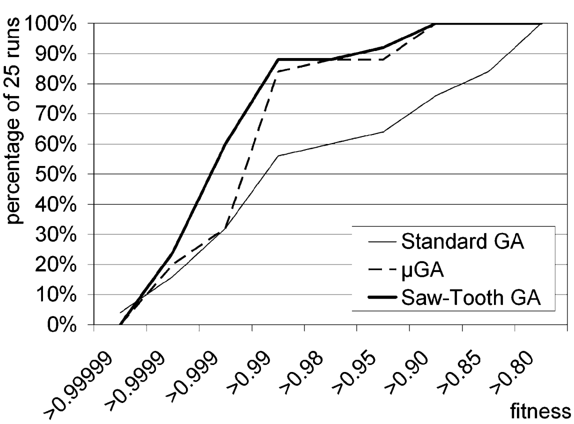

Fig. 6. Fitness distribution of 25 runs for best performing case of each algorithm (Goldberg and Richardson multimodal function).

Fig. 7. Performance comparison of algorithms (Rosenbrock unimodal function).

pears insensitive to the selection of population size, where, on the contrary, this remains a critical parameter for the success of the other GAs.

The parameters of the best performing case of each algorithm are the following: standard GA is $n = 225$, $p_c = 0.90$, and $p_m = 0.005$, μGA is $n = 25$, $p_r = 0.09$; saw-tooth GA is $\overline{n} = 60$, $T/\overline{n} = 0.70$, and $D/\overline{n} = 0.99$. In Fig. 6, the cumulative fitness distribution of the 25 runs for the best performing case of the compared algorithms is presented. Generally, the order of the algorithms in terms of performance remains the same as before. All the runs of the saw-tooth GA and the μGA found a solution in the global optimum peak (i.e., fitness > 0.85), whereas 16% of the standard GA runs missed the global optimum. The saw-tooth GA refined better the solutions, and 60% of the runs found the global optimum with less than 0.001 error. On the other hand, only 32% of the standard GA and μGA runs found the global optimum with less than 0.001 error. For the best performing cases of each algorithm, the results of the statistical significance t test are 0.00 and 0.21 between the saw-tooth GA runs and the standard GA and μGA runs, respectively. This indicates that the saw-tooth GA reaches a distinct performance level from the standard GA; whereas, for μGA there is little relevance.

VI. Parametric Study for Unimodal Problem

The three algorithms are also compared for the case of the Rosenbrock unimodal function given in Appendix A. The parametric study of the previous paragraph is repeated for this problem. In Fig. 7, the final fitness (mean value of the 25 runs) is presented for the best performing case of each algorithm with respect to the population size. For the Rosenbrock unimodal function, the standard GA performs better than μGA, and the performance was insensitive with respect to the selection of the population size. This suggests that μGA may not be preferable for problems with not many local peaks. The saw-tooth GA obtains the best performance of the three compared algorithms for mean population size $\overline{n}$ in the range of 30 to 120 individuals. The parameters of the best performing case of each algorithm are the following: standard GA is $n = 90$, $p_c = 0.85$, and $p_m = 0.045$; μGA is $n = 45$, $p_r = 0.18$; and saw-tooth GA is $n = 40$, $T/\overline{n} = 0.40$, and $D/\overline{n} = 0.85$. The above values indicate that a strong mutation or reinitialization improves the

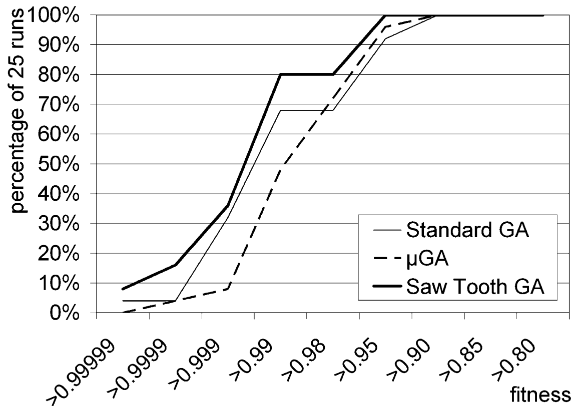

Fig. 8. Fitness distribution of 25 runs for best performing case of each algorithm (Rosenbrock unimodal function).

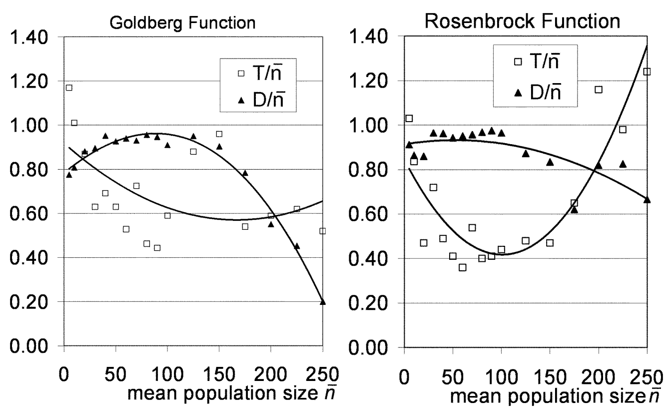

Fig. 9. Optimum selection of T and D parameters of saw-tooth GA with respect to mean population size $\overline{n}$.

performance of GAs for Rosenbrock unimodal function. In Fig. 8, the cumulative fitness distribution of the 25 runs for the best performing case of the compared algorithms is presented. The saw-tooth GA achieves better fitness distribution for all the fitness values. Furthermore, 80% of the saw-tooth GA runs found a solution with error smaller than 0.01; whereas, 68% of standard GA runs and 48% of μGA runs found a solution in this fitness range. For the best performing case of each algorithm, the t-test results of statistical significance are 0.04 and 0.01 between the saw-tooth GA runs and the standard GA and μGA runs, respectively. This indicates that the algorithms reach distinct levels of performance.

VII. Optimum Selection of T and D Parameters

In Fig. 9, the optimum values of the T and D parameters of the saw-tooth GA are presented for the Goldberg and Richardson multimodal function and Rosenbrock unimodal function with respect to the mean population size $\overline{n}$. Each data point corresponds to the mean value of the corresponding parameter for the ten best performing cases of the parametric studies presented previously. For both test problems, the optimum performance of the saw-tooth GA was obtained for typical population sizes in the range of 30 to 120 individuals. In this range, the optimum value of the normalized amplitude $D/\overline{n}$ is from 0.90 to 0.98, which indicates that strong reinitialization is beneficial for both problems. Moreover, moderate periods led to optimum saw-tooth GA performance. Particularly, the optimum ranges are $T/\overline{n} = 0.50$ to 0.70 for the Goldberg and Richardson multimodal function and $T/\overline{n} = 0.40$ to 0.50 for the Rosenbrock unimodal function, with the small difference attributed to the different features of the test problems. For population sizes smaller than 30, the variation of the population size becomes significant and the GA performance drops as it was observed; therefore, bigger $T/\overline{n}$ and smaller $D/\overline{n}$ are more suitable. For population sizes bigger than 120, the convergence of the GA is slower; therefore, bigger $T/\overline{n}$ and smaller $D/\overline{n}$ are more suitable to avoid premature reinitialization, which prevents the GA from potential improvement. The graphs presented in Fig. 9 are useful for an effective selection of the period T and the amplitude D of the saw-tooth GA for problems with similar features to the Goldberg and Richardson multimodal function and Rosenbrock unimodal function.

In Fig. 10, the performance of the saw-tooth GA is presented as a function of the mean population size $\overline{n}$ and the normalized parameters $T/\overline{n}$ or $D/\overline{n}$ for the case of the Rosenbrock unimodal function. It is shown that for typical population sizes of 30 to 120 individuals, the best performance of the saw-tooth GA is obtained for big amplitudes D in the range of $0.85\overline{n}$ to $0.99\overline{n}$ combined with medium periods T in the range of $0.30\overline{n}$ to $0.70\overline{n}$.

The sensitivity of the saw-tooth GA performance with respect to the values of the parameters T and D is presented for the case of the Goldberg and Richardson multimodal function (Fig. 11). The smoothed three-dimensional (3-D) surface covers a wide range of values for the period T and the amplitude D, for a mean population size $\overline{n} = 80$. The best performance is observed for the maximum possible amplitude D of $\overline{n} - 1 = 79$ (i.e., $D/\overline{n} = 0.99$) and moderate periods T of 25–40 (i.e., $T/\overline{n} = 0.30$ to 0.50). Generally, the performance increases for bigger amplitudes D, except for the case of small periods T combined with large amplitudes D, where the saw-tooth GA becomes unstable. From Fig. 11, it becomes evident that there exists a wide medium period range in which the saw-tooth GA performs better. Moreover, if the period T is selected adequately large (i.e., bigger than 20), then the performance of the saw-tooth GA is not sensitive with respect to the period T.

In Figs. 12 and 13, the performance of the saw-tooth GA scheme for optimal values of the parameters $T = 25$ and $D = 79$ is compared with nonoptimal values for a mean population size of $\overline{n} = 80$ individuals (20 runs for the Goldberg and Richardson multimodal function). In Fig. 12, the effect of the amplitude D is examined. For smaller values of the amplitude D, the average fitness converges faster to a smaller value and the population remains converged for many generations until the next reinitialization occurs, resulting in a worse overall performance. In Fig. 13, the effect of period T is examined. The bigger period $T = 50$ achieves a very good performance, which is slightly lower than the optimum $T = 25$. This is because the average fitness reaches a plateau at the end of the period, which indicates convergence of the population. The converged population does not improve the maximum fitness, and therefore a plateau is also observed at the maximum fitness

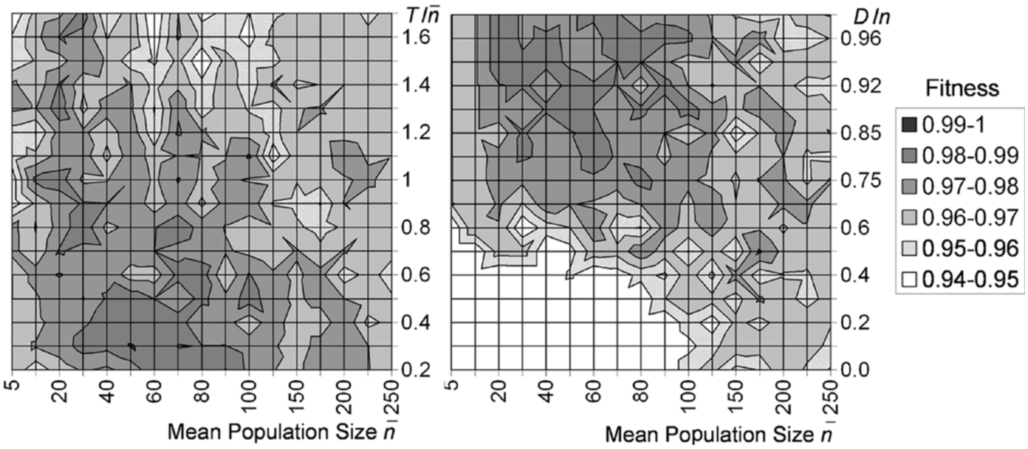

Fig. 10. Performance of saw-tooth GA with respect to mean population size $\overline{n}$ and normalized parameters $T/\overline{n}$ and $D/\overline{n}$ (Rosenbrock unimodal function).

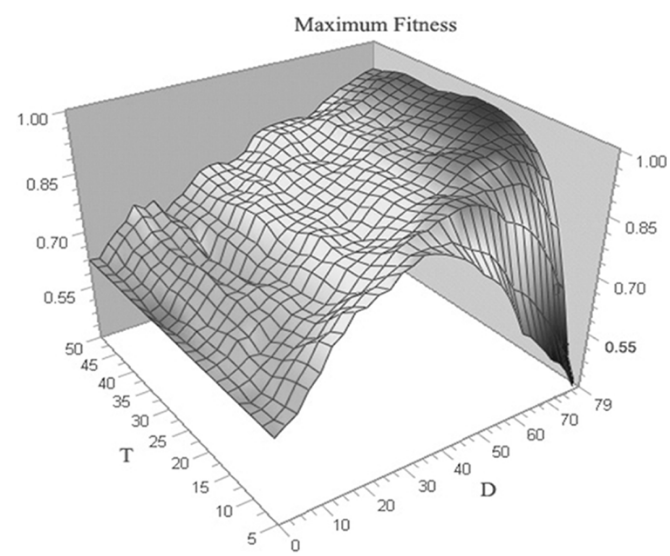

Fig. 11. Final fitness with respect to period T and amplitude D (Goldberg and Richardson multimodal function, mean value of 20 runs, $\overline{n} = 80$).

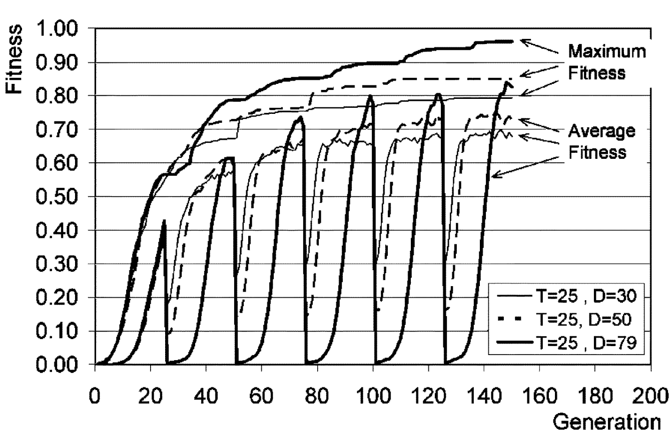

Fig. 12. Saw-tooth GA fitness histories. Comparison with nonoptimal values of amplitude D.

that delays the whole process. The small period of $T = 15$ degrades the performance of the algorithm considerably as premature reinitialization prevents the algorithm from potential improvement.

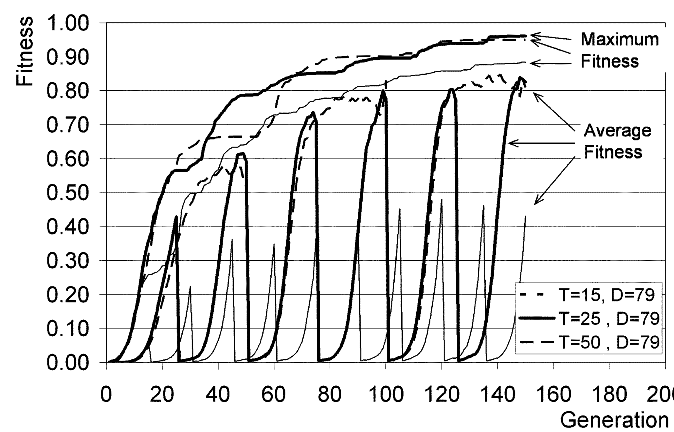

Fig. 13. Saw-tooth GA fitness histories. Comparison with nonoptimal values of period T.

VIII. Application to Test Functions

The proposed saw-tooth GA is applied to a set of test functions and a randomized function generator to test its overall performance. The test functions are presented in Appendix A. The first four correspond to multimodal functions with decreasing peaks and different features (Schwefel, Rastrigin, Ackley, and Griewangk). Eighteen more test functions are generated from the Kennedy multimodal function generator modified to consider unequal peaks. They correspond to combinations of $N = 5, 10$, and 20 variables and $M = 1, 10$, and 100 peaks. For the case of $N = 10$, $M = 100$, six different functions are generated which correspond to different locations of the random peaks. More functions with similar features can be found in the work of Lee and Yao [20] and Ho *et al.* [21]. The same parameters of each algorithm are used for all the different functions. The standard GA and μGA are applied with typical parameters: standard GA is $n = 80$, $p_c = 0.85$, and $p_m = 0.005$; μGA is $n = 5$, $p_r = 0.05$, $p_c = 1.0$, and $p_m = 0.0$. The saw-tooth GA is based on the parameters of the standard GA: $\overline{n} = 80$, $p_c = 0.85$, and $p_m = 0.005$. For the period and amplitude, according to the guidelines of the previous paragraph, $T = 40$ and $D = 75$ are chosen. All the algorithms use single-point crossover, tournament selection, and elitism. For each algorithm, 50 runs with

TABLE II
APPLICATION TO TEST FUNCTIONS AND FUNCTION GENERATOR

Test Function (minimization problems)	Standard GA average	Standard GA % runs in global peak	μGA average	μGA % runs in global peak	Saw-Tooth GA average	Saw-Tooth GA % runs in global peak	t-test* (Saw-Tooth GA & Standard GA)	t-test* (Saw-Tooth GA & μGA)
1. Schwefel	-4111.1	-	-4174.2	-	**-4178.4**	-	0.00	0.23
2. Rastrigin	5.50	-	**2.85**	-	3.02	-	0.00	0.25
3. Ackley	14.26	-	16.18	-	**12.97**	-	0.21	0.01
4. Griewangk	0.092	-	0.229	-	**0.087**	-	0.33	0.00
5. Kennedy N=5, M=1	2.4E-04	-	**2.0E-05**	-	4.3E-05	-	0.00	0.21
6. Kennedy N=5, M=10	6.6E-03	92%	6.0E-05	100%	**1.3E-05**	**100%**	0.01	0.06
7. Kennedy N=5, M=100	1.6E-02	80%	**2.6E-05**	**100%**	9.5E-05	100%	0.00	0.03
8. Kennedy N=10, M=1	0.0033	-	5.1E-04	-	**4.8E-04**	-	0.00	0.39
9. Kennedy N=10, M=10	0.0560	40%	0.0720	14%	**0.0412**	**52%**	0.04	0.00
10. Kennedy N=10, M=100 #1	0.0992	8%	0.0938	14%	**0.0795**	**22%**	0.01	0.04
11. Kennedy N=10, M=100 #2	0.0865	18%	0.0982	10%	**0.0757**	**22%**	0.11	0.00
12. Kennedy N=10, M=100 #3	0.1070	2%	0.0918	16%	**0.0847**	**18%**	0.00	0.20
13. Kennedy N=10, M=100 #4	0.0798	24%	0.0878	18%	**0.0475**	**50%**	0.00	0.00
14. Kennedy N=10, M=100 #5	0.0754	32%	0.0707	34%	**0.0602**	**40%**	0.07	0.15
15. Kennedy N=10, M=100 #6	0.0829	24%	0.0953	12%	**0.0630**	**34%**	0.02	0.00
16. Kennedy N=20, M=1	0.9764	-	0.9738	-	**0.9736**	-	0.00	0.07
17. Kennedy N=20, M=10	0.5553	0%	0.5896	0%	**0.5340**	0%	0.00	0.00
18. Kennedy N=20, M=100	0.3952	0%	0.4046	0%	**0.3861**	0%	0.05	0.01

** probability that the difference in the two observed means is smaller than the observed value purely by chance – one tailed test*
*** bold characters indicate the best performing algorithm*

different random seeds are performed. Each run is terminated at 20 000 function evaluations. Individuals that are not modified by mutation or recombination are not re-evaluated.

The results are listed in Table II, where the average values of the 50 runs of each algorithm are presented. For the case of the Kennedy function generator, the percentage of the runs that find a solution in the global peak is also presented. Moreover, the statistical significance t test is performed between the saw-tooth GA runs and the results of the other two algorithms. The best performing algorithm is indicated by bold characters. The saw-tooth GA has the best performance in 15 out of 18 functions. In the Rastrigin function, the μGA performs better. Moreover, in two out of three Kennedy functions with $N = 5$ variables, the μGA demonstrated better ability to refine the optimum solution; as with 20 000 function evaluations, both μGA and saw-tooth GA reach a saturated state and all the runs find the global optimum. In all other cases the saw-tooth GA reaches the best performance, while the standard GA and μGA alternate in rank, depending on the function. Especially for the Kennedy functions with $N = 10$ variables, where 20 000 function evaluations are considered adequate, the saw-tooth GA performed better in all six different cases. For $N = 20$ variables, the saw-tooth GA was better than the other two algorithms, demonstrating better performance when the function evaluations are less than adequate. This indicates that the saw-tooth GA is more robust as compared to both standard GA and μGA for the great majority of test problems. The t-test results indicate that the algorithms reach distinct levels of performance in most cases and some relevance in others with a maximum value of 0.39 as in the case of Kennedy function with $N = 10$ variables and $M = 1$ peak.

IX. CONCLUDING REMARKS

The proposed saw-tooth GA improves the performance of GAs and can be easily introduced to any existing GA code. It introduces a variable population size and periodic population reinitialization in the form of a saw-tooth function. For a wide, though not exhaustive, range of the examined problems, the performance of the saw-tooth GA was statistically superior as compared to the standard GA and micro GA (μGA) of equal computing cost. Furthermore, the saw-tooth GA was more robust than the standard GA and μGA with respect to the features of all the examined test problems. In addition, the optimum performance of saw-tooth GA schemes is achieved for big amplitudes D combined with medium periods T. Almost optimal performance is achieved for a wide range of both amplitude D and period T parameters, making the algorithm practically insensitive with respect to their selection.

APPENDIX
TEST FUNCTIONS

The test functions that are used in this paper are presented in Table III. Graphic illustrations of the test functions in two dimensions are presented in Fig. 14. The first function (Goldberg and Richardson) [17] is a multimodal function with decreasing peaks where the exponent a control the narrowness of the peaks. For the parametric studies, a set of $N = 4$ parameters discretized with 15 bits each and an exponent $a = 30$ are used to create a reasonably tough problem for the GA. The second function (Rosenbrock) is a unimodal function that has only one minimum in a region of a wide plateau. For the parametric studies, this function is transformed into a maximization problem with

TABLE III
TEST FUNCTIONS

		Test Function $f(x_1,x_2,\ldots,x_N)$	Bounds	Optimum	Variables
1	Goldberg & Richardson (α=30)	$f=\prod_{i=1}^{N} g_i h_i$, where $g_i=[\sin(5.1\pi x_i+0.5)]^{\alpha}$, $h_i=\exp\left(-4.0\log(2.0)\frac{(x_i-0.0667)^2}{0.64}\right)$	$0.0\le x_i\le 1.0$	max f = 1.0, for x_i=0.06684, i=1..N	4 variables, 15 bits each
2	Rosenbrock	$f=\sum_{i=1}^{N-1}100\left(x_{i+1}-x_i^2\right)^2+(1-x_i)^2$	$0.0\le x_i\le 2.0$	min f=0.0 for x_i=1.0, i=1..N	3 variables, 20 bits each
3	Schwefel	$f=-\sum_{i=1}^{N}x_i\sin\left(\sqrt{\lvert x_i\rvert}\right)$	$-500\le x_i\le 500$	min f=-416.99·N for x_i=416.99, i=1..N	10 variables, 10 bits each
4	Rastrigin	$f=\sum_{i=1}^{N}\left[x_i^2-10\cos(2\pi x_i)+10\right]$	$-5.0\le x_i\le 5.0$	min f=0.0 for x_i=0.0, i=1..N	10 variables, 10 bits each
5	Ackley	$f=-20\exp\left(-0.2\sqrt{\frac{1}{N}\sum_{i=1}^{N}x_i^2}\right)-\exp\left(\frac{1}{N}\sum_{i=1}^{N}\cos(2\pi x_i)\right)+20+e$	$-100\le x_i\le 100$	min f=0.0 for x_i=0.0, i=1..N	10 variables, 10 bits each
6	Griewangk (b=4000)	$f=\frac{1}{b}\sum_{i=1}^{N}x_i^2-\prod_{i=1}^{N}\cos\left(\frac{x_i}{\sqrt{i}}\right)+1$	$-50\le x_i\le 50$	min f=0.0 for x_i=0.0, i=1..N	10 variables, 10 bits each
7	Kennedy multimodal function generator	$f=\min_{j=1}^{M}\left(\sum_{i=1}^{N}\left[\frac{1}{1+\exp(-x_i)}-A_{ij}\right]^2+\frac{(j-1)^{0.15}}{15}\right)$ where A_{ij} = random matrix , $i=1..N$, $j=1..M$	$-4\le x_i\le 4$	min f=0.0 for random x_i, i= 1..N	N=5, 10, 20 variables, 10 bits each, M=1,10,100 peaks

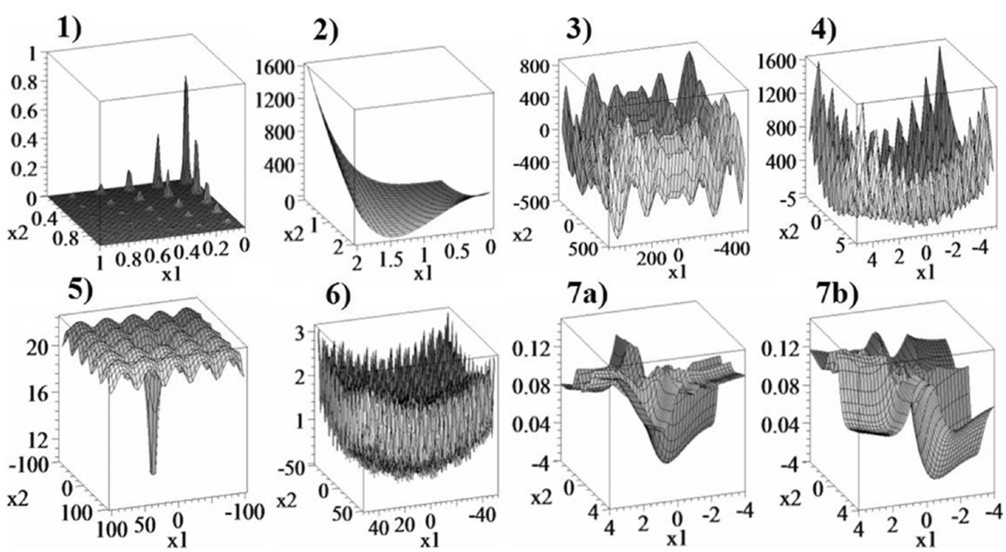

Fig. 14. Test functions in two dimensions: 1) Goldberg and Richardson multimodal function ($\alpha = 30$); 2) Rosenbrock unimodal function; 3) Schwefel function; 4) Rastrigin function; 5) Ackley function; 6) Griewangk function ($b = 4000$); 7a) Kennedy multimodal function generator ($M = 100$ peaks); and 7b) Kennedy multimodal function generator ($M = 100$ peaks, different random seed).

a maximum of 1.0. A set of $N = 3$ parameters discretized with 20 bits each are used. Functions 3–6 (Schwefel, Rastrigin, Ackley, and Griewangk) are multimodal functions with different features. A set of $N = 10$ parameters discretized with ten bits each are used. The seventh function (Kennedy) is a multimodal function generator [22]. It generates functions with M random peaks in N dimensions. The second term of the equation represents a modification introduced in this analysis to generate unequal peaks. There is only one global minimum with value 0.00 and the rest of the peaks represent local minima with values greater than 0.067. For the comparison of the algorithms, 14 different functions are generated with $N = 5, 10$, and 20 variables and $M = 1, 10$, and 100 random peaks. Each variable is discretized with ten bits.

REFERENCES

[1] J. H. Holland, *Adaptation in Natural and Artificial Systems*. Ann Arbor, MI: Univ. Michigan Press, 1975.

[2] D. E. Goldberg, *Genetic Algorithms in Search, Optimization and Machine Learning*. New York: Addison-Wesley, 1989.

[3] J. E. Smith and T. C. Fogarty, "Operator and parameter adaptation in genetic algorithms," *Soft Comput.*, vol. 1, pp. 81–87, 1997.

[4] E. A. Eiben, R. Hinterding, and Z. Michalewicz, "Parameter control in evolutionary algorithms," *IEEE Trans. Evol. Comput.*, vol. 3, no. 1, pp. 124–143, Feb. 1999.

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

[6] G. R. Harik and F. G. Lobo, "A parameter-less genetic algorithm," Univ. Illinois, Urbana–Champaign, IlliGAL Rep. 99 009, 1999.

[7] R. E. Smith, "Adaptively resizing populations: An algorithm and analysis," in *Proc. 5th Int. Conf. Genetic Algorithms (ICGA 93)*, 1993, pp. 653–653.

[8] V. K. Koumousis and C. K. Dimou, "The effect of oscillating population size on the performance of genetic algorithms," in *Proc. 4th GRACM Congr. Computational Mechanics*, 2002, Paper 099.

[9] H. G. Cobb and J. J. Grefenstette, "Genetic algorithms for tracking changing environments," in *Proc. 5th Int. Conf. Genetic Algorithms (ICGA 93)*, 1993, pp. 523–530.

[10] L. J. Eshelman, "The CHC adaptive search algorithm: How to have safe search when engaging in nontraditional genetic recombination," in *Proc. Foundations Genetic Algorithms-1 (FOGA-1)*, 1990, pp. 265–283.

[11] D. E. Goldberg, "Sizing populations for serial and parallel genetic algorithms," in *Proc. 3rd Int. Conf. Genetic Algorithms (ICGA 89)*, 1989, pp. 70–79.

[12] K. Krishnakumar, "Micro-genetic algorithms for stationary and nonstationary function optimization," in *Proc. SPIE Intelligent Control Adaptive Systems*, 1989, pp. 289–296.

[13] D. L. Carroll, "Genetic algorithms and optimizing chemical oxygen-iodine lasers," *Devel. Theoretical Appl. Mechanics*, vol. 18, pp. 411–424, 1996.

[14] C. A. C. Coello and G. T. Pulido, "Multiobjective optimization using a micro-genetic algorithm," in *Proc. Genetic Evol. Computation Conf. (GECCO-2001)*, 2001, pp. 274–282.

[15] J. Ryoo and P. Hajela, "Handling variable string lengths in GA based structural topology optimization," in *Proc. 42nd AIAA/ASME/ASCE/AHS/ASC SDM Conf.*, 2001, pp. 3297–3307.

[16] Y. G. Xu and G. R. Liu, "Detection of flaws in composites from scattered elastic-wave field using an improved μGA and a local optimizer," *Comput. Methods Appl. Mechanics Eng.*, vol. 191, no. 36, pp. 3929–3946, 2002.

[17] D. E. Goldberg and J. Richardson, "Genetic algorithms with sharing for multimodal function optimization," in *Proc. 2nd Int. Conf. Genetic Algorithms*, 1987, pp. 41–49.

[18] D. L. Carroll. (1999) FORTRAN genetic algorithm (GA) driver v1.7.0. [Online]. Available: http://www.cuaerospace.com/carroll/ga.html

[19] E. D. De Jong, R. A. Watson, and J. B. Pollack, "Reducing bloat and promoting diversity using multi-objective methods," in *Proc. Genetic Evol. Computation Conf. (GECCO-2001)*, 2001, pp. 11–18.

[20] C. Y. Lee and X. Yao, "Evolutionary programming using the mutations based on the Lévy probability distribution," *IEEE Trans. Evol. Comput.*, vol. 8, no. 1, pp. 1–13, Feb. 2004.

[21] S. Y. Ho, L. S. Shu, and J. H. Chen, "Intelligent evolutionary algorithms for large parameter optimization problems," *IEEE Trans. Evol. Comput.*, vol. 8, no. 6, pp. 522–541, Dec. 2004.

[22] J. Kennedy. (1997) Continuous valued multimodality generator for evolutionary algorithms. [Online]. Available: http://www.cs.uwyo.edu/~wspears/multi.kennedy.html

Vlasis K. Koumousis received the Diploma in civil engineering from the National Technical University of Athens, Athens, Greece, in 1975, and the M.Sc. and Ph.D. degrees in applied mechanics from the Polytechnic University of New York, New York, in 1976 and 1980.

He is currently a Professor at the School of Civil Engineering, National Technical University of Athens. His research interests include structural optimization, evolutionary computation, logic programming, shell structures, and seismic isolation of buildings, bridges, and monuments. He is the author more than 70 journal and conference papers in the above areas. Moreover, he has served as a Consultant in a number of design projects for major buildings, bridges, and restoration of monuments.

Prof. Koumousis is a member of the Structural Optimization Committee of ASCE, founding member of European Group for Structural Engineering Applications of Artificial Intelligence, and member of the New York Academy of Sciences.

Christos P. Katsaras received the Diploma in civil engineering from the National Technical University of Athens, Athens, Greece, in 2002, and the M.Sc. degree from the University of California, Berkeley, in 2004.

From 2002 to 2003, he was a Research Assistant at the National Technical University of Athens for the Seismic Isolation of the Statue of Hermes of Praxiteles in the Museum of Olympia, Greece. His research interests include genetic algorithms and seismic isolation.

Mr. Katsaras is a member of the Technical Chamber of Greece. He was awarded the Fulbright Scholarship for studies in the U.S., in 2003.