

Influence of the Crossover Operator in the Performance of the Hybrid Taguchi GA

Stjepan Picsek

Faculty of Electrical Engineering
and Computing
Unska 3, Zagreb, Croatia
Email: stjepan@computer.org

Marin Golub

Faculty of Electrical Engineering
and Computing
Unska 3, Zagreb, Croatia
Email: marin.golub@fer.hr

Domagoj Jakobovic

Faculty of Electrical Engineering
and Computing
Unska 3, Zagreb, Croatia
Email: domagoj.jakobovic@fer.hr

Abstract—This paper investigates the influence of different crossover operators on the efficiency of the hybrid Taguchi genetic algorithm and aims to provide guidelines for algorithm’s usage in continuous optimization. We examine the hybrid Taguchi genetic algorithm (HTGA) with 8 different crossover operators and apply it to 15 benchmark numerical optimization problems. The implementation uses binary representation which maps chromosomes to values in real domain with arbitrary precision. Different crossover operators are used with the HTGA and a detailed statistical analysis is performed to evaluate their performance. The results indicate that the HTGA obtains better results with crossover operators different than the ones commonly reported in literature.

I. INTRODUCTION

Evolutionary algorithms present an important tool in the area of optimization tasks. One of the algorithms from the evolutionary algorithms family which has been successfully applied to the variety of optimization problems is genetic algorithm (GA). In Holland’s version of genetic algorithm (GA) crossover operator played a major role as an exploitation force and mutation operator was generally treated as subordinate to crossover operator [1]. From then, different versions of genetic algorithm and operators have emerged [2] [3].

One of those modifications of genetic algorithm is the hybrid Taguchi genetic algorithm (HTGA). In a HTGA, the Taguchi method is performed between the crossover and mutation operators of the genetic algorithm. The Taguchi method is a design approach that uses ideas from the statistical experimental design. [4]. Hybrid Taguchi genetic algorithm represents combination between traditional GA and Taguchi method. Taguchi method can select better genes that undergo crossover process and by it, enhance the genetic algorithm.

Hybrid Taguchi method can be regarded as a variant of the crossover operator to be used within a conventional GA or as a variant of the genetic algorithm itself. The first approach can be justified by the fact that it is directly connected with the creation of the offspring from the parent individuals which is the main characteristic of a crossover operator. The latter approach is justified by the fact that hybrid Taguchi algorithm generates additional individuals, evaluates them and performs selection. Further justification for the second view may be found in the fact that different “simple” crossover operators may be used within the Taguchi method.

We employ different crossover operators with the hybrid Taguchi approach in order to see what combination gives the best overall results.

First it is necessary to answer the question of whether it is possible to find the best search algorithm. The answer is no, since the “No Free Lunch” theorem demonstrates that when averaged over all problems, all search algorithms perform equally. However, if the individual is working with some background knowledge of the problem, it is possible for him to choose more suitable algorithms.

When performing multiple problem analysis it is important to use the proper statistical methods in the analysis of the results. Many of the papers in this niche do not use the proper statistical methods which can lead to wrong interpretation of the results.

Experiments conducted suggest that there are better choices regarding crossover operator used within HTGA. More specifically, single-point operator, as the most common choice of crossover operator used in HTGA never outperformed some less usual crossover operators.

In Section 2 we present the relevant theory, Section 3 defines the experimental environment and presents results, Section 4 gives a discussion about the results, and finally, Section 5 draws a conclusion.

II. PRELIMINARY

A. Selection Methods

Selection is a mechanism for choosing individuals in the population that create offspring for the next generation [3]. In our experiments, we use roulette-wheel selection and tournament selection. For additional information about these two selection mechanisms refer to [5].

B. Crossover Operator

Crossover is a process where new individuals are created from the information contained within the parents. Crossover operators are usually applied probabilistically according to a crossover rate p_c . In this paper crossover refers to a two-parent case where two individuals are selected as the parents to produce one offspring. Table I enumerates crossover operators used in this paper.

TABLE I
CROSSOVER OPERATORS USED IN THE EXPERIMENTS

Crossover operator	Reference
Single-point crossover	[3] [5]
Two-point crossover	[5]
Half-uniform crossover	[6]
Uniform crossover	[3] [5]
Shuffle crossover	[2]
Segmented crossover	[2] [7]
Reduced surrogate crossover	[2]
Non-geometric crossover	[8]

Additionally, here we briefly present less known crossover operators.

1) *Reduced Surrogate Crossover*: To reduce the chance of producing clones Booker suggested examining the selected parents to define suitable crossover points [9]. A reduced surrogate crossover operator reduces parent strings to a skeletal form in which only those bits that differ in two parents are represented. Recombination is then limited only to the positions of the bits in reduced surrogates. Single-point crossover is used for the recombination of the skeletal forms of parents. If at least one crossover point occurs between the first and last bits in reduced surrogate, then the offspring will never duplicate the parents [2] [10].

2) *Segmented Crossover*: Segmented crossover represents a variant of multi-point crossover. In this crossover the number of crossover points is not constant. Fixed number of crossover points is replaced by the segment switch rate s , which specifies the probability that the segment will end at any point in the string. Starting from first position in a string, one real-valued number q and one natural number j are generated. The number q represents the probability that j will be crossover point. In experiments we conducted, segment switch rate s is set to the value of 0.2 [2] [7].

3) *Shuffle Crossover*: First, a single crossover position is selected. Before the variables are exchanged, they are randomly shuffled in both parents. After the recombination, the variables in the offspring are unshuffled in reverse. This removes positional bias as the variables are randomly reassigned each time a crossover is performed. In a way, shuffle crossover is similar to the uniform crossover. The difference is that the uniform crossover exchanges bits and not segments like the shuffle crossover [2].

4) *Non-geometric Crossover*: Non-geometric crossover is a crossover operator that generates offspring outside the segment between offspring's parents. First, a primary parent needs to be chosen, either randomly or based on the fitness of the parents. Offspring is created as a copy of primary parent, and then the bits of the offspring are bit-flipped with the prespecified probability when the values of the bits are the same in the both of the parents [8].

C. Hybrid Taguchi Genetic Algorithm

In their work, Leung and Zhang [11] stated that some phases of GAs can be regarded as experiments. They proposed incorporating an orthogonal array design into GAs in order to make it more statistically sound.

After parent solutions are sampled for the reproduction, an orthogonal array is used so the parent solutions can produce a small but representative set of offspring. Parameters of an orthogonal array are Q as the number of factor levels, M as the number of experiments (rows of the orthogonal array), and N as the number of factors (columns of the orthogonal array). An orthogonal array is a fractional factorial matrix, which assures a balanced comparison of levels of any factor or interaction of factors. The array is called orthogonal because all columns can be evaluated independently of one another. Instructions for the creation of the orthogonal array with an arbitrary number of parameters and more detailed explanations about the orthogonal arrays can be found in [12]. For factors of an orthogonal array we use the dimensions of a problem. Offspring are created based on the factors from the two parents and the values from the orthogonal array. From M offspring, n are chosen to become the part of next generation. In our experiments, n is set to the value of 2. For the offspring selection scheme the best offspring and the best parent are chosen since it implicitly provides elitism. The fundamental principle of the HT method is to improve the quality of a product by minimizing the effect of the causes of variation without eliminating the causes. To achieve that goal, two major tools used are the signal-to-noise ratio (SNR) which measures quality and the orthogonal arrays which are used to study many design parameters simultaneously [4]. After generating offspring, their fitness values and SNR are calculated. The equation for calculating SNR is

$$SNR = \frac{1}{y^2} . \quad (1)$$

where y represents fitness function value. Next, the effects of the various factors (variables) are calculated, which can be defined as follows:

$$E_{fl} = \sum (SNR_i) . \quad (2)$$

where factor f is at level l , and index i in SNR_i defines the experiment number. The optimal level for a factor is the level that gives the highest value of E_{fl} in the experimental region. After the optimal levels for each factor are selected, an optimal chromosome is obtained. For a detailed instruction on a HTGA refer to [4].

In its original form, HTGA uses single-point crossover operator for the creation of all M offspring. Since there are many available crossover operators, it is advisable to evaluate the efficiency of alternate operators within the HTGA, which has not been investigated previously. In this work we apply every operator given in Table I to each of the test problems described below to estimate HTGA performance.

D. Test Functions

Test functions given in the Table II have been selected for the investigation of crossover efficiency. These functions represent well known problems for evaluating the performance of evolutionary algorithms. Most of the problems are taken from the CEC 2005 Special Session on Real-Parameter optimization [15]. Main difference is that we usually did not shift global optimum location. The table shows the formula for the function, domain range of the problem, and the reference where additional information on the function can be found.

E. Binary Encoding of Chromosomes

Binary coded chromosome represents a real value in the interval $[lbound, ubound]$. The real value can be decoded using the integer value of the binary string (bin_val) of length n :

$$real_val = lbound + \frac{bin_val}{(2^n - 1)} (ubound - lbound). \quad (3)$$

Alternatively, the analogous integer binary value may also be obtained using the desired real value of the chromosome:

$$bin_val = \frac{real_val - lbound}{ubound - lbound} (2^n - 1). \quad (4)$$

III. ENVIRONMENTAL SETTINGS AND RESULTS

In all the experiments, binary-coded genetic algorithm with 3-tournament or roulette-wheel selection is used. Individuals are binary vectors which represent real values [7]. Parameters of the genetic algorithm that are in common for every round of the experiments are the following: simple bit mutation with mutation probability p_m of 0.01 per bit, population size N of 30, precision is set to 3 digits after the decimal point (which is sufficient to produce large enough number of possible solutions for used test problems), number of independent runs for each experiment is 30, dimensionality D of all the test problems is set to 50, and the number of fitness evaluations is set to 500000 (in accordance with the required number of fitness evaluations as stated in [15]). For all the test functions, finding global minimum is the objective.

All parameters are additionally displayed in Table III for clarity.

TABLE III
PARAMETERS USED IN EXPERIMENTS

Mutation probability p_m	0.01 per bit
Population size N	30
Precision of solution	3 digits after the decimal point
Number of runs	30
Dimensionality of the problem D	50
Number of fitness evaluations	500000

Naturally, by choosing different mutation parameters it is possible to expect different results of analysis. We decided to use the simplest mutation operator and a constant p_m

value since the mutation operator is not of primary interest in this paper. Experiments where roulette-wheel selection is used have an initial phase where the parameter tuning is done. The objective is to find the best value of p_c for every crossover operator.

In this phase, each operator is run 30 times on each test function with different values of p_c , ranging from 0.1 to 1 in steps of 0.1. For every combination the p_c value which gives the smallest mean error on best individuals in 30 runs was chosen. The results from the initial phase are not displayed here, but it can be concluded that for all the test cases crossover probabilities in the range 0.1-0.3 have reached the best results. In all the experiments, k-tournament selection method has a k value of 3. The goal of the rest of the experiments is to find the best overall operator for both selection schemes on the set of all the test functions. As a performance measure we use error rate obtained for every operator.

This structure of input data is in accordance with previous analysis performed over multiple algorithms and test problems [15] [16]. When conducting a statistical analysis, first it is necessary to decide whether to use parametric or nonparametric statistical tests. To be able to use the parametric tests, it is necessary to check if the data satisfies the independency, normality, and heteroscedasticity conditions [16] [17]. Independence of the events is obvious since there are independent runs of the algorithms. To check the normality we use Shapiro-Wilk test and for heteroscedasticity we use Levene test. The results show that the normality and heteroscedasticity conditions are not satisfied and because of that the nonparametric statistical tests should be used. The tests were conducted for a level of significance α of 0.05. References regarding the statistical methods used can be found in [17] and [18]. As a genetic algorithm test suite the Evolutionary Computation Framework (ECF) was used. ECF is a C++ framework intended for the application of any type of the evolutionary computation, developed at the University of Zagreb [19].

A. Experiments and Results

Table IV gives the results for a Friedman two-way analysis of variances by ranks, which represents the most well known procedure for testing the differences between more than two related samples [17]. The objective of this test is to show that there are statistical differences between groups (crossover operators) for both selection methods. If this is true, then additional *post-hoc* statistical analysis can be performed to discover where those differences are.

With the level of significance α of 0.05 both the Friedman and Iman-Davenport statistic show significant differences in operators with test values of 39.22 and 8.35, respectively, and $p < 0.001$ for roulette-wheel selection. Friedman and Iman-Davenport statistic for tournament selection case amounts to the values of 40.88 and 8.92, respectively, and $p < 0.001$ which indicates significant statistical differences.

In the post-hoc analysis we applied the Bonferroni-Dunn, Hochberg, Finner and Li tests [18] over the results of

TABLE II
BENCHMARK FUNCTIONS USED IN EXPERIMENTS

Test function	Domain Range	Reference
$f(x) = \sum_{i=1}^D x_i^2$	[-5.12, 5.12]	[13]
$f(x) = \sum_{i=1}^D i \cdot x_i^2$	[-5.12, 5.12]	[13]
$f(x) = \sum_{i=1}^D 5 \cdot i \cdot x_i^2$	[-5.12, 5.12]	[13]
$f(x) = \sum_{i=1}^D \left(\sum_{j=1}^i x_j^2 \right)$	[-65.536, 65.536]	[13]
$f(x) = \sum_{i=1}^{D-1} 100 \cdot (x_{i+1} - x_i^2)^2 + (1 - x_i)^2$	[-2.048, 2.048]	[14]
$f(x) = 10 \cdot D + \sum_{i=1}^D (x_i^2 - 10 \cdot \cos(2 \cdot \Pi \cdot x_i))$	[-5.12, 5.12]	[13]
$f(x) = \sum_{i=1}^D -x_i \cdot \sin\left(\sqrt{ x_i }\right)$	[-500, 500]	[12]
$f(x) = \sum_{i=1}^D x_i^2/4000 - \prod_{i=1}^D \cos(x_i/\sqrt{i}) + 1$	[-600, 600]	[14]
$f(x) = -20 \cdot e^{-0.2 \sqrt{\sum_{i=1}^D x_i^2/D}} - e^{\sum_{i=1}^D \cos(2\Pi x_i)/D} + 20 + e$	[-32.768, 32.768]	[13]
$f(x) = -\sum_{i=1}^D \sin(x_i) \cdot \left(\sin(i \cdot x_i^2/\Pi)\right)^{20}$	[0, 3.14]	[13]
$f(x) = \sum_{i=1}^D (10^6)^{(i-1/D-1)} \cdot x_i^2 - 450$	[-100, 100]	[15]
$f(x) = \sum_{i=1}^D x_i + \prod_{i=1}^D x_i $	[-10, 10]	[12]
$f(x) = \sum_{i=1}^D 2 \cdot D + \sum_{i=1}^{D-1} [\sin(x_i + x_{i+1}) + \sin(2 \cdot x_i \cdot x_{i+1}/3)]$	[3, 13]	[14]
$f(x) = 1/D \cdot \sum_{i=1}^D (x_i^4 - 16 \cdot x_i^2 + 5 \cdot x_i)$	[-5, 5]	[12]
$f(x) = \sum_{i=1}^D \left(\sum_{k=0}^{20} [0.5^k \cdot \cos(2\Pi \cdot 3^k (x_i + 0.5))] \right) - D \sum_{k=0}^{20} [0.5^k \cdot \cos(\Pi \cdot 3^k)]$	[-0.5, 0.5]	[15]

TABLE IV
AVERAGE RANKINGS OF THE ALGORITHMS (FRIEDMAN)

Algorithm	Ranking Tournament	Ranking Roulette-wheel
Single-point	3.87	4.07
Two-point	3.6	4.2
Half-uniform	4.2	4.2
Uniform	3.73	3.07
Segmented	3.23	3
Shuffle	7.07	6.67
Reduced surrogate	3.47	3.87
Non-geometric	6.8	6.93

Friedman procedure. The analysis indicates the level of significance with which the control operator is better than each of the remaining operators (i.e. for which the null hypothesis is rejected, where null hypothesis represents the case when there are no significant differences) and the adjusted p values are shown in Table V. The results highlight segmented crossover as the best operator for both

selection methods, so the post- hoc analysis is performed with segmented crossover as the control method.

For the Bonferroni-Dunn test, a *critical difference* (CD) [16] has a value of 2.41. The interpretation of this measure is that the performance of two algorithms is significantly different only if the corresponding mean ranks differ by at least a critical difference, which is depicted in Fig. 1 and Fig. 2. A cut line is drawn at height equal to the sum of critical difference and ranking of the control algorithm. The bars that exceed this line are associated with the algorithms that have worse performance than the control algorithm. Bonferroni Dunn's test distinguish that segmented crossover is better than shuffle or non-geometric crossover. Post-hoc analysis confirms that for the both selection methods, segmented crossover is better than shuffle crossover and non-geometric crossover with $\alpha = 0.05$. These results coincide with hypotheses rejected by Bonferroni Dunn test.

A *contrast estimation* procedure based on medians [18] can be used to estimate the differences between each two crossover operators. In this test the performance of the algorithms is reflected by the magnitudes of the differences in error rates,

TABLE V
POST-HOC COMPARISON (CONTROL OPERATOR: SEGMENTED CROSSOVER)

Selection	Algorithm	unadjusted p	p_{Bonf}	$p_{Hochberg}$	p_{Finner}	p_{Li}
Roulette-wheel	Non-geometric	0.000011	0.000077	0.000077	0.000077	0.000184
	Shuffle	0.000041	0.00029	0.000248	0.000145	0.000696
	Two-point	0.179712	1.257987	0.665128	0.370125	0.751532
	Half-uniform	0.179712	1.257987	0.665128	0.370125	0.75153
	Single-point	0.233038	1.631266	0.665128	0.370125	0.796837
	Reduced surrogate	0.332564	2.327946	0.665128	0.376057	0.848421
	Uniform	0.940584	6.58409	0.940584	0.940584	0.940584
Tournament	Shuffle	0.000022	0.000151	0.000151	0.000151	0.000122
	Non-geometric	0.000078	0.000546	0.000468	0.000273	0.000441
	Half-uniform	0.296718	2.077023	0.823063	0.560152	0.626443
	Single-point	0.502335	3.516345	0.823063	0.705124	0.73952
	Uniform	0.601845	4.212915	0.823063	0.724531	0.772803
	Two-point	0.709388	4.965717	0.823063	0.763481	0.80037
	Reduced surrogate	0.823063	5.761443	0.823063	0.823063	0.823063

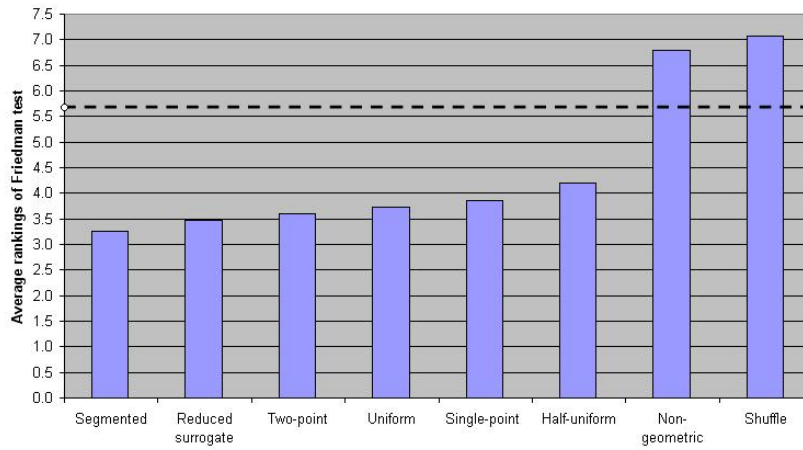


Fig. 1. Bonferroni-Dunn's test for tournament selection, critical difference = 2.41, control operator: segmented crossover

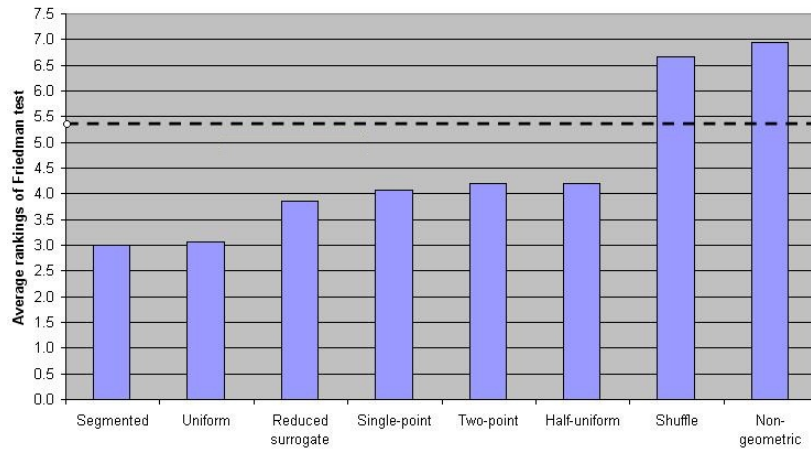


Fig. 2. Bonferroni-Dunn's test for roulette-wheel selection, critical difference = 2.41, control operator: segmented crossover

TABLE VIII
AVERAGE RANKINGS OF THE ALGORITHMS (FRIEDMAN)

Algorithm	Ranking
Segmented tournament	2.67
Reduced surrogate tournament	2.6
Segmented roulette-wheel	2.27
Uniform roulette-wheel	2.47

and test values are shown in Table VI and Table VII for the roulette-wheel and tournament selection, respectively. A negative value for the operator in a given row indicates that the operator performs better than the operator in a given column. For the tournament selection shuffle and non-geometric crossover operators obtained significantly worse results than all the other operators, so they were excluded from the Table VII for clarity. These results highlight the segmented and reduced surrogate operators as the best performing ones when tournament selection is used, and segmented and uniform crossover as the best operators when roulette-wheel selection is used.

Finally, additional comparison was performed to find the best overall crossover operator regardless of the selection method. In this analysis only the two best algorithms for each selection method were used, and results obtained are displayed in Table VIII.

Since Friedman and Iman Davenport statistics can not show significant differences in operators when $\alpha = 0.05$ no additional post-hoc analysis has been made, rather we conducted only contrast estimation analysis. From Table IX it can be concluded that the best overall algorithm is HTGA with segmented crossover and roulette-wheel selection.

IV. DISCUSSION

The results show that there exist significant statistical differences in performance when alternative crossover operators are used within the Taguchi part of the genetic algorithm. There are also differences depending on the selection method used, where two most common selection methods are employed. It should be stressed that the roulette-wheel selection, unlike the tournament variant, includes an additional crossover probability parameter. This parameter influences the ratio at which the hybrid Taguchi procedure is applied within the genetic algorithm operator flow. Overall, the best results (in most cases) were obtained with the probability of crossover of 0.1 – 0.3, which could be attributed to the exploratory nature of the hybrid Taguchi approach.

For both the roulette-wheel selection and the tournament selection segmented crossover operator performed the best. However, there are differences when comparing the best algorithms with both selection methods. Hybrid Taguchi genetic algorithm with roulette-wheel selection gives better results than HTGA with k-tournament selection. Furthermore, it can be observed that the algorithms are much more stable when roulette-wheel selection is used. That means that the best algorithms for some

problems never perform the worst for other problems from employed benchmark set. When tournament selection is used, it can be observed that best algorithms for some of the test problems, performed among the worst for the other problems. A likely reason for worse performance of tournament selection may be the fact that tournament selection by itself contributes a great deal of selection pressure, and Taguchi method also contributes to quicker convergence. When applying tournament selection and Taguchi method together, the overall selection pressure is very high and it can result in trapping in suboptimal solution. Segmented crossover operator with roulette-wheel selection performed the best and it can be outlined as a rough recommendation for hybrid Taguchi genetic algorithm when multiple-problem analysis is conducted. Segmented crossover is a variant of a multi-point crossover, in which the fixed number of crossover points is replaced with a segment switch rate. In our experiments, the switch rate is set to the value of 0.2. Multi-point crossover operators can search through larger search spaces more thoroughly, but they also disrupt the schema more easily. The reason that segmented crossover with Taguchi method performed the best possibly lies in the fact that segmented crossover contributed with a search of larger solution space and then the Taguchi method chooses better offspring among those solutions. Other operators that search through larger search space, like uniform crossover, performed well, although somewhat poorer and the reason might be in easier schema disruption. These results show that there is room for improvement of the performance of the hybrid Taguchi genetic algorithm. In further work it will be necessary to experiment with additional test functions. Those new test functions should also include problems that are more natural to solve using binary interpretation.

Test functions have a great influence on the results of the experiments. If the chosen problems are too easy, it can result in too fast convergence to the global optimum for all crossover operators. On the other hand, choosing problems that are too difficult can result in the trapping of the algorithm in the local optima. The test problems used in experiments here were chosen because they represent a standard test suite for examining performance of an algorithm. Additionally, some of these test problems were also used in examining the performance of original HTGA so it was natural to compare performance with the same test problems. Further experiments should also be done to compare hybrid Taguchi genetic algorithm with roulette-wheel selection and segmented crossover, and other top performance methods.

V. CONCLUSION

This paper presents an implementation and a detailed statistical analysis of different crossover operators used within a hybrid Taguchi GA. The presented results can provide guidelines to operator selection in hybrid Taguchi application to numerical optimization problems. The performance is evaluated on a well established set of optimization problems, which should indicate the approximate performance in real-world applications. Since in most cases there is not enough

TABLE VI
CONTRAST ESTIMATION TEST FOR ROULETTE-WHEEL SELECTION

	Single-point	Two-point	Half-uniform	Uniform	Segmented	Shuffle	Red. surr.	Non-geom.
Single-point	0	0.006	0.012	0.057	0.098	-0.24	-0.002	-0.247
Two-point	-0.006	0	0.007	0.052	0.093	-0.246	-0.008	-0.252
Half-uniform	-0.012	-0.007	0	0.045	0.086	-0.252	-0.015	-0.259
Uniform	-0.057	-0.052	-0.045	0	0.041	-0.297	-0.06	-0.304
Segmented	-0.098	-0.093	-0.086	-0.041	0	-0.338	-0.101	-0.345
Shuffle	0.24	0.246	0.252	0.297	0.338	0	0.237	-0.007
Red. surr.	0.002	0.008	0.015	0.06	0.101	-0.237	0	-0.244
Non-geom.	0.247	0.252	0.259	0.304	0.345	0.007	0.244	0

TABLE VII
CONTRAST ESTIMATION TEST FOR TOURNAMENT SELECTION

	Single-point	Two-point	Half-uniform	Uniform	Segmented	Reduced surrogate
Single-point	0	0.004	-0	-0	0.004	0.004
Two-point	-0.004	0	-0.004	-0.004	0.001	0
Half-uniform	0	0.004	0	-0	0.005	0.004
Uniform	0	0.004	0	0	0.005	0.004
Segmented	-0.004	-0.001	-0.005	-0.005	0	-0
Reduced surrogate	-0.004	-0	-0.004	-0.004	0	0

TABLE IX
CONTRAST ESTIMATION TEST FOR THE BEST OVERALL ALGORITHM

	Segmented tour.	Red. surrogate tour.	Segmented roulette-wheel	Uniform roulette-wheel
Segmented tournament	0	0.003	0.221	0.11
Reduced surrogate tournament	-0.003	0	0.218	0.107
Segmented roulette-wheel	-0.221	-0.218	0	-0.111
Uniform roulette-wheel	-0.11	-0.107	0.111	0

time for an exhaustive operator and parameter search, the provided findings may prove useful to researchers in similar optimization environments. The experiments conducted show that hybrid Taguchi genetic algorithm can be modified to a more powerful version with a choice of a different selection and crossover operator. The results singled out HTGA with roulette-wheel selection and segmented crossover as the most successful combination.

REFERENCES

- [1] J. H. Holland, *Adaptation in Natural and Artificial Systems: An Introductory Analysis with Applications to Biology, Control, and Artificial Intelligence*. Cambridge, USA: The MIT Press, 1992.
- [2] D. Dumitrescu, B. Lazzarini, L. C. Jain, and A. Dumitrescu, *Evolutionary Computation*. Florida, USA: CRC Press, 2000.
- [3] M. Mitchell, *An Introduction to Genetic Algorithms*. Cambridge, USA: The MIT Press, 1999.
- [4] J.-T. Tsai, T.-K. Liu, and J.-H. Chou, "Hybrid taguchi-genetic algorithm for global numerical optimization," *IEEE Trans. Evol. Comput.*, vol. 8, no. 4, pp. 365–377, Aug. 2004.
- [5] T. Weise, *Global Optimization Algorithms Theory and Application*, 2009. [Online]. Available: <http://www.it-weise.de/>
- [6] L. J. Eshelman, "The CHC adaptive search algorithm: How to have safe search when engaging in nontraditional genetic recombination," in *Foundations of Genetic Algorithms*. San Francisco, USA: Morgan Kaufmann, 1991, pp. 265–283.
- [7] Z. Michalewicz, *Genetic Algorithms + Data Structures = Evolution Programs*, 3rd ed. New York, USA: Springer-Verlag, 1996.
- [8] H. Ishibuchi, N. Tsukamoto, and Y. Nojima, "Maintaining the diversity of solutions by non-geometric binary crossover: A worst one-max solver competition case study."
- [9] L. Booker, *Improving Search in Genetic Algorithms*. Pitman, London, and Morgan Kaufman, Los Altos, 1987.
- [10] S. N. Sivanandam and S. N. Deepa, *Introduction to Genetic Algorithms*. New York, USA: Springer-Verlag, 2008.
- [11] Q. Zhang and Y.-W. Leung, "An orthogonal genetic algorithm for multimedia multicast routing," *IEEE Trans. Evol. Comput.*, vol. 3, no. 1, pp. 53–62, Apr. 1999.
- [12] Y.-W. Leung and Y. Wang, "An orthogonal genetic algorithm with quantization for global numerical optimization," *IEEE Trans. Evol. Comput.*, vol. 5, no. 1, pp. 41–53, Feb. 2001.
- [13] H. Pohlheim, "Geatbx examples of objective functions," Available as http://www.geatbx.com/download/GEATbx_ObjFunExpl_v38.pdf, 2006.
- [14] K. Y. Chan, C. K. Kwong, H. Jiang, M. E. Aydin, and T. C. Fogarty, "A new orthogonal array based crossover, with analysis of gene interactions, for evolutionary algorithms and its application to car door design," *Expert Systems with Applications*, vol. 37, pp. 3853–3862, 2010.
- [15] P. Suganthan, N. Hansen, J. Liang, K. Deb, Y. Chen, A. Auger, and S. Tiwari, "Problem definitions and evaluation criteria for the CEC 2005 special session on real parameter optimization," Nanyang Technological University, Tech. Rep., 2005.
- [16] S. Garcia, D. Molina, M. Lozano, and F. Herrera, "A study on the use of non-parametric tests for analyzing the evolutionary algorithms? behaviour: a case study on the CEC 2005 special session on real parameter optimization," *Journal of Heuristics*, no. 15, pp. 617–644,

2009.

- [17] D. Sheskin, *Handbook of Parametric and Nonparametric Statistical Procedures*, 4th ed. Chapman and Hall/CRC, 2007.
- [18] J. Derrac, S. Garcia, D. Molina, and F. Herrera, "A practical tutorial on the use of nonparametric statistical tests as a methodology for comparing evolutionary and swarm intelligence algorithms," *Swarm and Evolutionary Computation*, vol. 1, pp. 3–18, 2011.
- [19] D. Jakobovic and et al., "Evolutionary computation framework," Mar. 2011. [Online]. Available: <http://gp.zemris.fer.hr/ecf/>