

Three-Step Parameters Tuning Model for Time-Constrained Genetic Algorithms

Saleem Ramadan¹ & Mahmoud Barghash²

¹ Department of Mechanical and Industrial Engineering, Applied Science Private University, Jordan P. O. Box 166, Postal Code 11931, Amman, Jordan

² Department of Mechanical and Industrial Engineering, the University of Jordan, Amman, Jordan

Correspondence: Saleem Ramadan, Department of Mechanical and Industrial Engineering, Applied Science Private University, Jordan. E-mail: s_ramadan@asu.edu.jo

Received: April 20, 2016

Accepted: May 27, 2016

Online Published: July 7, 2016

doi:10.5539/mas.v10n10p118

URL: <http://dx.doi.org/10.5539/mas.v10n10p118>

Abstract

In this paper a three-step parameters tuning model for time-constrained Genetic Algorithms (GAs) was presented. The first step involved modeling the objective function using multiple regression model where the fitness value was the response variable and the GA parameters were the regressors. The second step involved constraint modeling using the objective function found in the first step and using the upper and lower limits of the GA parameters along with an upper limit on the execution time as constraints. The third step involved optimizing the constraint model found in the second step using a suitable deterministic optimization method to determine the optimal GA parameters taking into consideration four aspects that affect the GA performance. These aspects were: the problem under consideration, the GA parameters used, the execution time, and the power of the computer used.

The validation of this model was demonstrated using two capacitated lot sizing problems. The model was able to predict the fitness values and the optimal parameters of the GA for these problems to a high degree of precision. Moreover, the results showed that tuning the GA parameters using multiple regression along with a suitable deterministic optimization method was an effective and robust method that enhanced the performance of the GA.

The statistical analysis showed that in order to do a proper tuning for a certain GA, the designer of the GA must take into consideration not only the type of problem but also the size of the problem, the allowable execution time, and the hardware used in executing the GA. Furthermore, the results agreed with the "No Free Lunch" theorem.

Keywords: genetic algorithm, tuning, time-constrained genetic algorithm, GA performance

1. Introduction

Genetic algorithms are search algorithms based on the mechanism of the natural selection. The most basic concept is that strong individuals tend to survive while the weak individuals tend to extinguish. That is, optimization is based on evolution and the "Survival of the fittest" concepts. The basic algorithm is as follows:

1.1 Begin

-Initialize population with random candidate solutions

-Evaluate each candidate

-Do while termination condition is not satisfied

(1) Select parents

(2) Recombine parents to generate offspring

(3) Mute the resulting offspring

(4) Evaluate new candidates

(5) Select individuals for the next generation

-End While

GA can have many parameters that need to be tuned to optimize its performance. These parameters include, and not limited to, population size, generation number, crossover rate, execution time, and mutation rate.

It is well known among the GA users that different problems required different strategies in their GAs. Moreover, for the same problem, the set of parameters differs with the complexity of the problem itself. This phenomenon is known in literature as "No Free Lunch" theorem (NFLT). This means that the GA parameters cannot be isolated from the characteristics of the class of functions being optimized, the matter that emphasizes the non-existing of a fixed set of parameters that can optimize the performance of GA for a certain class of functions. The knowledge about the problem integrated into the evolutionary algorithm can enhance considerably the effectiveness of the approach and improve significantly the convergence rate and the quality of the results. The optimization method has shown good efficiency for some applications, but its performance is dependent on the values selected for the parameters of the algorithm. For example, if a Travel Salesman Problem (TSP) is considered, the complexity of the problem increases as the number of nodes increases, the matter that calls for using a different set of parameters in the GA for each different TSP problem according to its complexity. This is true because using different combinations of parameters values with a certain GA gives it different exploration and exploitation power, which makes the underlying GA more or less suitable for the specific problem. Generally speaking, high crossover rate (among other parameters) increases the exploration power of the GAs while high mutation rate (among other factors) increases their exploitation power. Moreover different parameters values may increase or decrease certain inherited problems in the GA such as premature convergence problem and high execution time problem. For example using a small population size may cause a premature convergence problem while using high population size may cause inefficient computational time problem. There is almost an agreement among the GA users about the non-existing of a universal GA. This matter calls the GA designers to tune their GAs to properly suit a specific problem.

It is worth to mention here that parameters tuning problem differs significantly than parameters control problem. In the first problem, the tuning of the parameters happens before the use of the GA and the values of the parameters stay fixed throughout the application of the GA, while in the second problem, the values of the parameters change during the application of the GA. For example Ramadan used a linearly adaptive crossover rate and mutation rate parameters in multiple objective GA to optimize the reliability testing. These adaptive rates are considered as parameters control not parameters tuning as the values of the parameters are changed according to the generation number.

Many attempts were done before to tackle the parameters tuning problem in GA. Meta-EA along with another EA is used to search for the best set of parameters values in. Results showed that the Meta-EA can find a set of parameters values that are better than the recommended default values found in the literatures. Unfortunately, Meta-EA has some inherited problems. These problems can be summarized into three major points as follows: First, not all parameters can be searched by EAs. For example if one of the parameters in the underlying GA is the type of crossover, then this parameter cannot be searched by the Meta-EA as there is no way to define a reasonable distance metric along this dimension. Second, there may be a situation that not all parameters are applicable for a certain GA. For example if there is a parameter for the number of generations at which the GA terminates and the underlying GA does not use this criterion for termination, then the number of generations parameter is not applicable. Finally, this method is time consuming due to its stochastic nature which renders it impractical.

Yuan and Gallagher used Meta-EA along with the Racing statistical method to deal with the tuning problem. Racing searching method is known for its time efficiency comparing to full enumeration method because it quickly identifies weak solutions and discards them from the sample space based on statistical tests. Unlike Meta-EA, the Racing searching method does not need a distance metric; consequently, it can be used to compare different GAs.

In this paper, the GA parameters tuning problem is solved in three steps: the first step involved modeling the objective function using multiple regression model where the fitness value was the response variable and the GA parameters were the regressors. The second step involved constraint modeling using the objective function found in the first step and using the upper and lower limits of the GA parameters along with an upper limit on the execution time as constraints. The third step involved optimizing the constraint model found in the second step using a suitable deterministic optimization method to determine the optimal GA parameters taking into consideration four aspects that affect the GA performance.

The rest of the paper will be organized as follows: Section 2 contains a discussion about the aspects affecting GA performance, section 3 presents the proposed model, section 4 describes the GA used for validation, section 5

validates the model and discusses the results, and finally, section 6 concludes the paper.

2. Aspects Affecting GA's Performance

The aspects affecting the GA's performance were divided into four main categories and were discussed individually. These aspects are: the problem under consideration, the GA parameters, the execution time, and finally, the power of the computer used. Tests of hypotheses were conducted to test whether there is a significance difference between the average performances of the GA under the associated aspect.

2.1 The Problem Aspect

The problem type and size are considered the major elements of this aspect. The NFLT guarantees that there is no universal GA; therefore, each type of problems must have its own GA's strategies to perform well.

The other element of the problem aspect is the size of the problem. To show whether there is a significant change in the average performance of the GA under different problem sizes or not, a TSP problem was solved using the same GA parameters, execution time, and computer power. The following hypothesis was tested:

$H_{0,1}$: There is no significance difference between the mean performance of the GA when solving a small TSP problem and the mean performance of the GA when solving the same problem with larger size at significance level of 0.05.

$H_{1,1}$: There is a significance difference between the mean performance of the GA when solving a small TSP problem and the mean performance of the GA when solving the same problem with larger size at significance level of 0.05.

A100 simulations were performed on the standard Eil51 TSP problem and another 100 simulations were performed on standard Eil101 TSP problem. The fitness values were standardized using the following standardization equation:

$$FV_{si} = \frac{FV_i - BFV}{BFV} \quad (1)$$

Where FV_{si} is the standardized fitness value for simulation i , FV_i is the fitness value for simulation i , and BFV is the best solution ever found in the literature for this problem. Table (1) gives a summary of the data

Table 1. Summary of data for hypothesis $H_{0,1}$

Sample	N	Mean	StDev	SE Mean
1	100	0.1025	0.0170	0.0017
2	100	0.04355	0.00524	0.00052

Note. The difference between two means T -Test results were as follows:

Difference = $\mu(1) - \mu(2)$

Estimate for difference: 0.05895

95% CI for difference: (0.05543, 0.06247)

T -Test of difference = 0 (vs not =): T -value = 33.14 P -value = 0.000 DF = 117.

The results show that the P -value is less than 0.05, which means that the null hypothesis is rejected and the alternative one is accepted. Therefore, the mean performance of the GA differs significantly when changing the size of the problem, which means that the size element of the problem aspect does indeed affect the performance of the GA.

2.2 The GA Parameters Aspect

To show whether there is a significant change in the average performance of the GA under different GA parameters or not, the Eil101 TSP problem was solved twice using different set of GA parameters while keeping the other three aspects fixed. The following hypothesis was tested:

$H_{0,2}$: There is no significance difference between the mean performances of the GA when changing the GA's parameters at significance level of 0.05.

$H_{1,2}$: There is a significance difference between the mean performances of the GA when changing the GA's parameters at significance level of 0.05.

Two sets of 100 simulations were performed on the Eil101 problem. Each set had a different parameters for the GA. Table (2) gives a summary of the data

Table 2. Summary of data for hypothesis $H_{0,2}$

Sample	N	Mean	StDev	SE Mean
1	100	695.05	9.66	0.97
2	100	678.87	8.60	0.86

Note. The difference between two means T -Test results were as follows:

Difference = $\mu(1) - \mu(2)$

Estimate for difference: 16.17

95% CI for difference: (13.62, 18.72)

T -Test of difference = 0 (vs not =): T -value = 12.51 P -value = 0.000 DF = 195.

The results show that the P -value is less than 0.05, the matter that allowed us to reject the null hypothesis and to accept the alternative one. This means that the mean performance of the GA differs significantly with respect to the GA parameters used. This result shows clearly that GA parameters aspect has an effect on its performance.

2.3 The Execution Time Aspect

To show whether there is a significant change in the average performance of the GA under different allowed execution times or not, the Eil101 TSP problem was solved twice using 5 seconds and 60 seconds while keeping the other three aspects fixed. The following hypothesis was tested:

$H_{0,3}$: There is no significance difference between the mean performance of the GA when changing the allowed execution times at significance level of 0.05.

$H_{1,3}$: There is a significance difference between the mean performance of the GA when changing the allowed execution times at significance level of 0.05.

Two sets of 100 simulations were performed on the Eil101 problem. The first set had 5 seconds and the other one had 60 seconds. Table (3) gives a summary of the data

Table 3. Summary of data for hypothesis $H_{0,3}$

Sample	N	Mean	StDev	SE Mean
1	100	693.5	10.7	1.1
2	100	666.00	7.66	0.77

Note. The difference between two means T -Test results were as follows:

Difference = $\mu(1) - \mu(2)$

Estimate for difference: 27.47

95% CI for difference: (24.87, 30.07)

T -Test of difference = 0 (vs not =): T -value = 20.87 P -value = 0.000 DF = 179.

The results show that the P -value is less than 0.05, which means that the null hypothesis is rejected and the alternative one is accepted, hence, the mean performance of the GA differs with respect to the execution time used, and therefore, the execution time affects the performance of the GA.

2.3. The Power of the Computer Used Aspect

To show whether there is a significant change in the average performance of the GA under different computers powers or not, the Eil101 TSP problem was solved twice using two computers: One with i7 core computer and the other one with AMD Phenom (tm) IIX6 while keeping the other three aspects fixed. The following hypothesis was tested:

$H_{0,4}$: There is no significance difference between the mean performances of the GA when changing the power of the computer used at significance level of 0.05.

$H_{1,4}$: There is no significance difference between the mean performances of the GA when changing the power of the computer used at significance level of 0.05.

Two sets of 100 simulations were performed on the Eil101 problem. The first set using the AMD Phenom (tm) IIX6 computer and the other one using the i7 core computer. Table (4) gives a summary of the data.

Table 4. Summary of data for hypothesis $H_{0,4}$

Sample	N	Mean	StDev	SE Mean
1	100	693.5	10.7	1.1
2	100	669.37	7.54	0.75

Note. The difference between two means T -Test results were as follows:

Difference = $\mu(1) - \mu(2)$

Estimate for difference: 24.11

95% CI for difference: (21.53, 26.69)

T -Test of difference = 0 (vs not =): T -Value = 18.42 P -Value = 0.000 DF = 177.

The results show that the P -value is less than 0.05, which means that the null hypothesis is rejected and the alternative one is accepted. This means that the mean performance of the GA differs with respect to the power of the computer used. This result shows clearly that the power of the computer used to run the GA affects the performance of the GA.

The overall result of this section suggests that in order to do a proper tuning for a certain GA, the designer must take into consideration not only the type of problem but also the size of the problem, the allowable execution time, and the hardware used in executing the GA. The next section proposes a tuning model that accounts for the four different aspects affecting the performance of any GA.

3. GA Parameters Tuning Model

The GA parameter tuning problem is the problem of tuning the GA's parameters values to optimize the performance of the GA under consideration. In this model the objective function represents the estimated fitness function of the GA at a certain set P of GA's parameters values (decision variables), while the constraints are the upper and lower limits of these parameters along with an upper cap on the execution time. The objective function will be found using multiple regression analysis where the response variable is the fitness value of the GA and the regressors are the GA's parameters values. "Maximum Execution Time" termination strategy will be formulated and used as a constraint on the maximum allowable execution time for the underlying GA to account for the execution time aspect. The complete model can be expressed mathematically as follows:

$$\text{Max (Min)} \hat{Y} = \varphi(P) \quad (1)$$

$$P_l \leq P \leq P_h \quad (2)$$

$$\vartheta(P) \leq t_{max}, \quad (3)$$

$$P \geq 0 \quad (4)$$

Some parameters can be integers such that \hat{Y} is the estimated fitness, $\varphi(\cdot)$ and $\vartheta(\cdot)$ are two multiple regression models corresponding to \hat{Y} and the execution time respectively. P is the set of GA parameters, P_l and P_h are two sets containing the lower and the upper limits of the GA parameters respectively, t_{max} is the maximum allowable execution time.

The effectiveness of this model was demonstrated using a GA for Capacitated Lot Sizing Problem (CLSP) with five parameters. In the CLSP the quantity and timing of the orders are selected under the resource availability and minimum total cost constraints.

The lot size for a certain period depends on the demand for that period and the demands of the subsequent periods. In general, the objective of the CLSP is to find the production schedule that minimizes the total production cost given the unit production cost, the inventory holding cost, the customer demand, the set up cost, and the discrete and finite production horizon under the constraint of production capacity. The lot sizing decision will be made at the beginning of each period in the time horizon such that the customer needs are satisfied by the end of the time horizon.

GA will be used to solve this problem. In this GA, a binary chromosome representation is used where "1" in a certain period means that this period will cover exactly the need for this period and the needs for all other "0" periods that come to the right of it until the first period that has "1" is again reached. Figure 1 shows a demonstration of the chromosome representation for three products and 4 periods CLSP.

Period	Product 1				Product 2				Product 3			
	1	2	3	4	1	2	3	4	1	2	3	4
Chromosome	1	0	0	1	1	1	0	1	1	0	1	1

Figure 1. Chromosome representation for 3 products and 4 periods

Moreover, Mode crossover strategy is proposed and used where the mode of the corresponding genes for certain number of randomly selected chromosomes (NCHC) is found (from the best 25% chromosomes of that generation) and then copied to the corresponding gene in the offspring. It must be noticed here that the NCHC must be an odd integer number which is less than or equal to the number of chromosomes involved in the crossover to assure that there is only one mode for each gene. Moreover, This crossover strategy guarantees that the offspring is feasible. The probability of performing crossover is governed by the crossover rate (P_c). Figure 2 shows how the Mode Crossover strategy works.

Chromosome 2

Period	Product 1				Product 2				Product 3			
	1	2	3	4	1	2	3	4	1	2	3	4
Chromosome	1	1	0	1	1	0	1	0	1	1	1	0

Chromosome 3

Period	Product 1				Product 2				Product 3			
	1	2	3	4	1	2	3	4	1	2	3	4
Chromosome	1	1	1	0	1	1	0	1	1	0	1	0

Gene Mode 1 1 0 1 1 1 0 1 1 0 1 0

Offspring

Period	Product 1				Product 2				Product 3			
	1	2	3	4	1	2	3	4	1	2	3	4
Chromosome	1	1	0	1	1	1	0	1	1	0	1	0

Figure 2. Example for the Mode Crossover strategy

Random mutation strategy is used in this GA in which a random gene in the offspring is located and its value is changed such that if the original value is "1" it will be changed to "0" and vice versa. Caution should be taken here such that the first gene for each product must always be "1" to guarantee that the demand for the first period is always covered. The probability of applying mutation to a certain offspring is governed by the offspring mutation rate ($OffP_m$) while the probability of mutating a gene in that offspring is governed by the mutation rate (P_m). The population size used in this GA is (P_S).

Often in real life situations, the GA has to run under the pressure of time constraint, so the GA parameters must be tuned to reach the optimal parameters combination that will give the best fitness value within the allowed execution time. Therefore, this GA will terminate when t_{max} seconds of execution time is elapsed.

The fitness function for CLSP is described in details in Suer et al. [14] and is repeated here for convenience.

$$Min z = \sum_{t=1}^T \sum_{i=1}^N (A_i X_{it} + h_i I_{it}) + \sum_{t=1}^T (R_t rr + OT_t or + S_t sr)$$

Note.

N: Number of products

T: Number of periods

A_i: Fixed cost (ordering cost) for product *i*

h_i: Inventory carrying cost for product *i*

rr: Labor rate for regular capacity

or: Labor rate for overtime hours

sr: Labor rate for subcontracting hours

OT_t: Overtime hours needed for period *t*

R_t: Regular labor hours needed for period *t*.

S_t: Subcontract hours needed for period *t*.

I_{it}: Inventory for product *i* in period *t*.

X_{it}: Binary variable with 0 if no production in period *t* of product *i* and 1 otherwise.

4. Results and Discussion

Two CLSPs from Suer et al. [14] are used to demonstrate the effectiveness of the tuning model, one with 5 products and 12 periods and the other one with 15 products and 6 periods. The relevant data for the two CLSPs is summarized in Table (5).

Table 5. Relevant data for the two problems used in the experimentation

Type of information	Problem 1: 5 products and 12 periods	Problem 2: 15 products and 6 periods
Order cost for different products	[500 300 600 400 900]	[500 300 600 400 900 800 400 600 400 600 400 500 400 500 800]
Carrying cost for different products	[1.5 2 1.75 1.5 2.2]	[1.5 2 1.8 1.5 2.2 1.6 1.5 1.2 2.2 1.8 2.3 2.2 2.2 1.6 1.6]
Processing time for different products	[20/60 19/60 26/60 22/60 18/60]	[20/60 19/60 26/60 22/60 18/60 15/60 20/60 30/60 16/60 26/60 23/60 16/60 16/60 26/60 21/60]
Setup time for different products	[5 3 4 4 6]	[5 3 4 4 4 4 1 4 4 1 2 4 5 4];
Labor Rate	[10 15 20]	[10 15 20]
Net Requirement for different products for different periods	[25 21 21 28 26 24 25 21 21 28 26 24;24 26 28 24 25 20 24 26 28 24 25 20;28 26 24 27 27 24 28 26 24 27 27 24;24 25 26 27 24 22 24 25 26 27 24 22;20 34 21 15 17 18 20 34 21 15 17 18]	[25 21 21 28 26 24;24 26 28 24 25 20;28 26 24 27 27 24;24 25 26 27 24 22;20 34 21 15 17 18;25 23 21 25 27 29;20 25 26 20 31 24;25 26 27 28 29 21;20 21 25 23 25 28;28 29 26 26 24 22;24 25 26 27 26 24;20 23 25 26 24 21;21 20 23 25 24 20;22 26 25 27 20 22;24 26 28 24 25 20]

As this paper discusses the tuning problem of the GA parameters, it is irrelevant to discuss in details the different GA strategies used as we are only interested on the effect of the parameters values on the performance of the GA. Therefore, the GA will be considered as a black box where the inputs to this box are the GA parameters values and the output is the fitness value. Thus, only the values of the parameters will be presented. Table (6) represents the upper and lower limits for the GA parameters that have been used in the experimentation.

Table 6. Parameters limits

Parameter Description	Parameter Abbreviation	Lower Limit	Upper Limit
Number of chromosomes used in the Mode Crossover operation	<i>NCHC</i>	3	9
Probability of performing crossover	<i>P_c</i>	0.1	1
Probability of mutating an offspring	<i>OffP_m</i>	0.1	1
Probability of mutating a gene	<i>P_m</i>	0.1	1
Size of the population	<i>P_s</i>	20	100

4.1 Experimentation on Problem 1

One hundred random and distinct combinations (within the parameters limits) were generated for the parameters values. For each combination, the GA was simulated 100 times using i7 core computer and the average fitness value at that combination was recorded.

Multiple linear regression model was used to model the fitness function $\varphi(P)$ of this problem. as the fitness function of the model and all its constraints are linear, the mixed linear programming (MLP) optimization method was used to optimize the model. Verification of using the multiple linear regression model to model the $\varphi(P)$ for this problem will follow.

Minitab software was used to carry out the statistical analysis and Microsoft Excel solver was used to optimize the model. Table (7) shows the relevant statistical analysis for the regression model.

Table 7. The statistical summary for the multiple regression models

Predictor	Coef	SE Coef	T	P	VIF
Constant	19085.6	18.1	1054.	33	0.000
<i>P_s</i>	-0.4798	0.2811	-1.71	0.091	1.362
<i>P_c</i>	-17.06	38.37	-0.44	0.658	1.364
<i>P_m</i>	305.54	14.79	20.66	0.000	1.046
<i>OffP_m</i>	-81.467	6.583	-12.38	0.000	1.015
<i>NCHC</i>	4.790	1.138	4.21	0.000	1.073
Analysis of Variance					
Source	DF	SS	MS	F	P
Regression	5	290420	58084	109.96	0.000
Residual Error	94	9655	528		
Total	99	340076			

Note: S = 22.9836 R-Sq = 85.4% R-Sq(adj) = 84.6%.

The significance of the model was 0.000. This clearly shows that our model is significant at 5% significance level; hence, the GA parameters can indeed predict the fitness value with high precision. The adjusted R-square value for the regression model was 84.6%, which indicates that our model has accounted for 84.6% of the variation in the fitness values. Moreover, the table also shows that *P_m*, *OffP_m*, and *NCHC* have the biggest impact on the model as their absolute *t*-statistics values are large and their *p*-values are small. The results showed that the other parameters (*P_c* and *P_s*) have little impact on the fitness value as their *p*-values are relatively high.

The Collinearity diagnostics shows that model 1, which included only *p_m* parameter accounted for 59.2% of the variation in the fitness values (Adjusted R-square =0.592). The inclusion of the *OffP_m* into model 2 resulted in an additional 22.8% of the variation in the fitness values being explained with a total Adjusted R-square of 82.%. The third model 3 also included *NCHC*, and this model resulted of an additional 2.4% of the variation in the fitness values being explained with a total of 84.40% of the variation in the fitness values being accounted for. The forth model, model 4, also included *P_s* and *P_c* together and this model resulted of an additional 0.2% of the variation in the fitness values being explained with a total of 84.6% of the variation in the fitness values being accounted for. Moreover, this Collinearity diagnostics shows that the parameters can be considered independent as their VIF values are close to 1.

To test the linearity and homoscedasticity assumptions made in using the multiple regression model, the standardized residual errors versus the fitted values are given in Figure 3. The figure shows that there is a linear relationship between the GA parameters and the fitness value. Moreover, the figure shows that the assumption of homoscedasticity is also applied, therefore, the variance errors are the same across all levels of GA parameters. The Kolmogorov-Smirnov tests of normality shows a value of 0.081 which indicates that the normality assumption is valid. These results along with the high value of R^2 justifies the use of multiple linear regression model to model the relationship between the GA parameters and the fitness value.

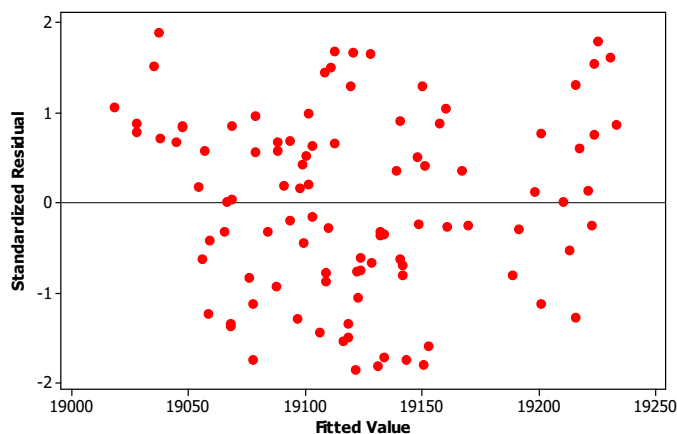


Figure 3. Standardized residuals versus fitted values

The low and high limits of the parameters used in the examples were $P_l = [3 \ 0.1 \ 0.1 \ 0.1 \ 20]$ and $P_h = [9 \ 1 \ 1 \ 1 \ 100]$. It should be noticed here that the $NCHC$ and P_s parameters should be integers with $NCHC$ odd.

The objective function shows that as the values of $OffP_m$, P_c , and P_s increase, the average fitness value decreases (enhances). Furthermore, as the values of $NCHC$ and P_m increase the average fitness value increases (deteriorates). This result can be understood on the light of the termination criterion used, which is, the GA terminates when 5 seconds of execution time is elapsed. As the values of $NCHC$ and P_m increase, the computation time for a single generation increases, consequently, the overall number of generations that can be done within the 5 seconds time frame decreases. This matter deteriorates the fitness value.

Table (8) shows the effectiveness of the multiple regression models in predicting the actual fitness value. The table shows the estimated fitness values and the actual fitness values for 10 different randomly selected combinations of parameters values along with their absolute residuals. It is clear from the table that the multiple linear regression equation predicts the fitness values precisely with a maximum absolute residual of 0.166% and minimum absolute residual of 0.001%.

Table 8. Effectiveness of the multiple regression model for problem 1

Ave(NGR)	Parameters					Actual Fitness Value	Predicted Fitness Value	Abs(residual)%
	P_s	P_c	P_m	$OffP_m$	$NCHC$			
33.85	60	0.1	0.1	0.2	3	19076.66	19083.76	0.037%
34.65	60	0.1	0.2	0.1	5	19124.83	19132.04	0.038%
32.62	40	0.25	0.1	0.5	3	19066.69	19066.35	0.002%
72.68	40	0.1	0.1	0.2	3	19088.9	19093.35	0.023%
32.15	60	0.1	0.5	1	9	19164.13	19169.54	0.028%
71.1	40	0.1	0.5	0.5	7	19210.55	19210.29	0.001%
33.82	60	0.1	0.1	0.5	3	19049.98	19059.32	0.049%
33.33	60	0.1	0.1	0.2	7	19099.38	19102.91	0.019%
30.77	40	0.25	0.1	1	9	19058.35	19054.36	0.021%
18.6	80	0.1	0.2	1	9	19036.61	19068.28	0.166%

Using the same procedure above the estimated execution time $\vartheta(P)$ can be given as:

$$\vartheta(P) = 4.925 + 0.000118NCHC + 0.301P_c + 0.00307OffP_m - 0.00584P_m + 0.00199P_s.$$

MLP optimization can find the optimal combination for the parameters values that optimizes the performance of the underlying GA. The complete MLP model is given as:

$$\min \hat{Y} = 19085.62 + 4.79NCHC - 17.06P_c - 81.47OffP_m + 305.53P_m - 0.4798P_s \quad (5)$$

$$3 \leq NCHC \leq 9 \quad (6)$$

$$0.1 \leq P_c \leq 1 \quad (7)$$

$$0.1 \leq OffP_m \leq 1 \quad (8)$$

$$0.1 \leq P_m \leq 1 \quad (9)$$

$$20 \leq P_s \leq 100 \quad (10)$$

$$4.925 + 0.000118NCHC + 0.301P_c + 0.00307OffP_m - 0.00584P_m + 0.00199P_s \leq 5 \quad (11)$$

$$P_s, P_m, OffP_m, P_c, NCHC \geq 0 \quad (12)$$

$$NCHC, P_s \text{ integers} \quad (13)$$

$$NCHC \text{ odd} \quad (14)$$

By solving this model using Excel solver, the estimated optimal fitness value using the model was 19015.7 with optimal parameters values $[NCHC \ P_c \ OffP_m \ P_m \ P_s]$ of $[3 \ 0.1 \ 1 \ 0.1 \ 66]$ respectively. These parameters were used in the underlying GA where the GA was simulated for 100 times. The average fitness value was 19027.16 and the average number of generations was 33.1. This value is better than any of the other fitness values that were found in the 100 combinations evaluated earlier. This indicates that the MLP method is able to tune the underlying GA parameters to give the best performance within the 5 seconds time frame.

The performance of the tuned GA was compared with the performance of the GA when one parameter's value was changed at a time.

Figure (4) shows a comparison between the average fitness value for the tuned GA and the average fitness value for the GA when $NCHC$ value changes from its optimal value of 3 to 7.

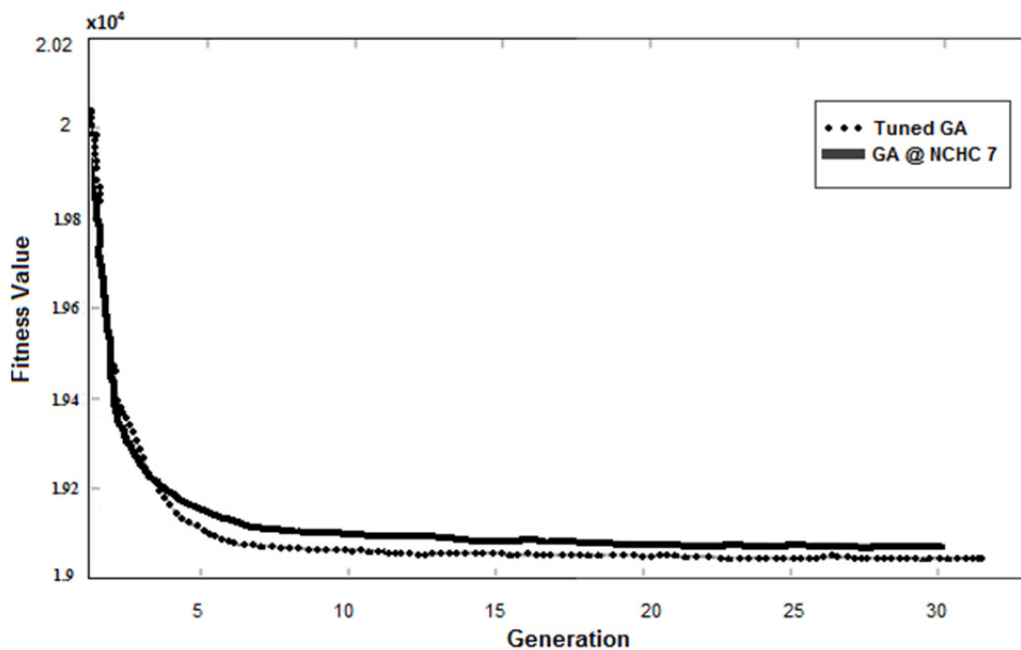


Figure 4. The performance of the GA when changing $NCHC$ from 3 to 7

Figure (5) shows a comparison between the average fitness value for the tuned GA and the average fitness value for the GA when the $OffP_m$ value was changed from its optimal value of 1 to 0.5.

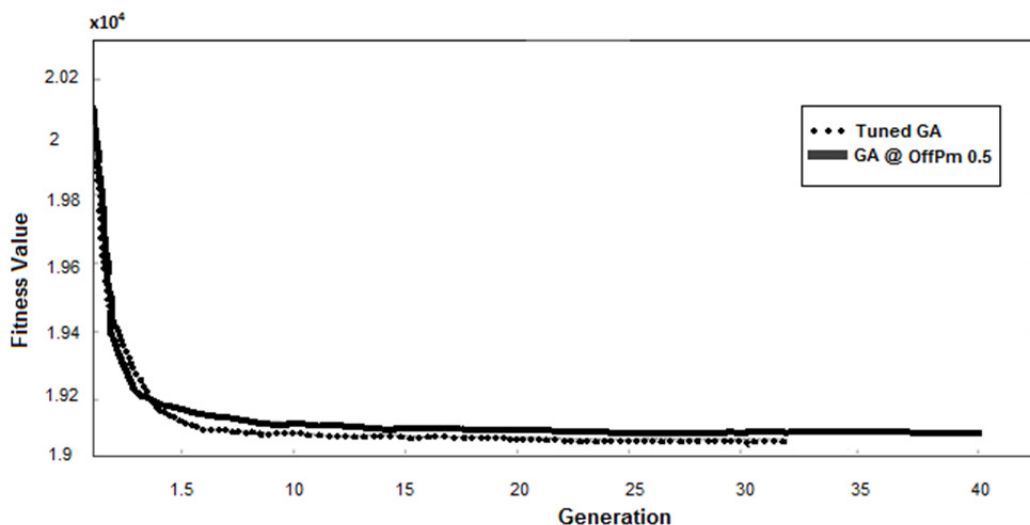


Figure 5. The performance of the GA when changing OffP_m from 1 to 0.5

Figure (6) shows a comparison between the average fitness value for the tuned GA and the average fitness value for the GA when the P_m value was changed from its optimal value of 0.1 to 0.5.

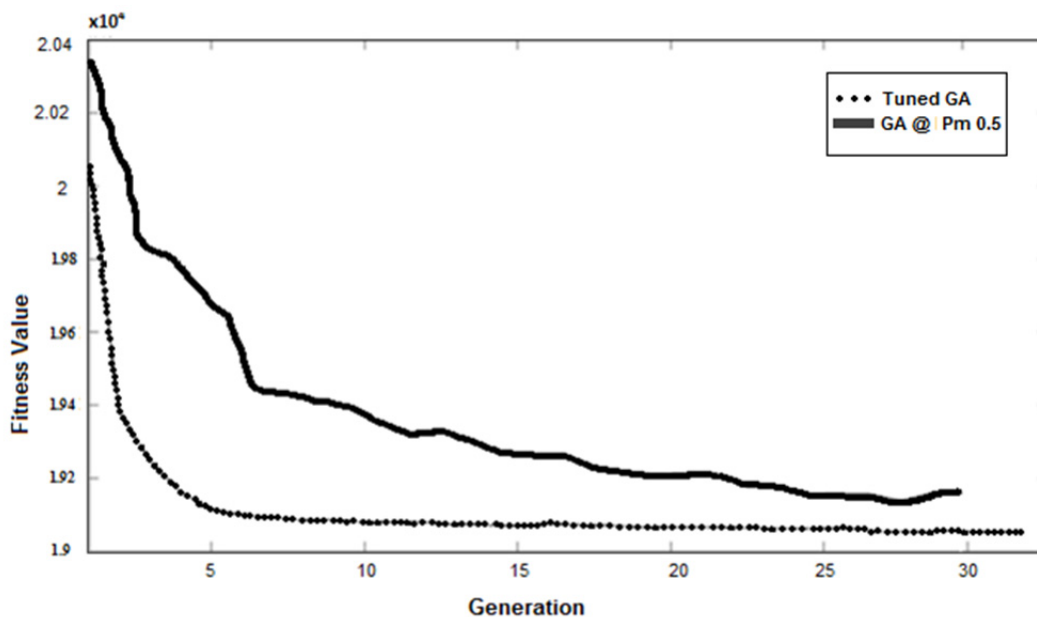


Figure 6. The performance of the GA when P_m value was changed from 0.1 to 0.5

Figure (7) shows a comparison between the average fitness value for the tuned GA and the average fitness value for the GA when the parameters values were changed from their optimal values to the following values [7 0.1 0.5 0.2 60].

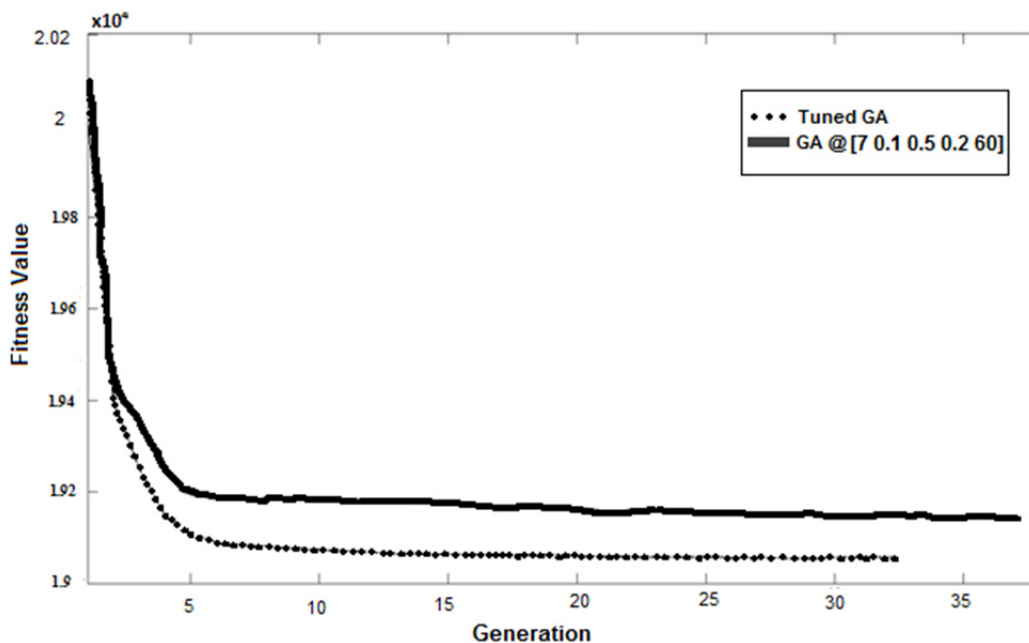


Figure 7. The performance of the GA with the following parameters [7 0.1 0.5 0.2 60]

Table (9) shows a summary of Figures 4, 5, 6 and 7.

Table 9. Summary of Figures 4, 5, 6 and 7

Parameter vales	Average Fitness value	Average number of generations
[7 0.1 1 0.1 66] @ 5 seconds	19058	30.1
[3 0.1 0.5 0.1 66] @ 5 seconds	19066	42.4
[3 0.1 1 0.5 66] @ 5 seconds	19168	29.8
[7 0.1 0.5 0.2 60] @ 5 seconds	19145	37.9

When the number of chromosomes used in the crossover changed from the optimal value 3 to 7 the average number of generations dropped from 33.1 to 30.1 the matter that deteriorated the fitness value as less exploration, for the sample space, happened. In the case where the rate of mutation of the offspring reduced from 1 to 0.5, the exploitation power of the GA reduced and in the same time the exploration power increased due to the increase in the number of generations from 33.1 to 42.4. The overall result was deterioration in the fitness value.

In the case where the mutation rate increased from 0.1 to 0.5, the fitness function deteriorated. In one hand, this increase in the crossover rate allowed for more exploitation (the matter that should enhance the fitness function), in the other hand, this increase reduced the number of generations from 33.1 to 29.8 the matter that reduced the exploration power of the GA and thus the net effect was a deterioration in the fitness value. Finally, changing all the parameters values deteriorated the fitness value due to different effects that affected the exploration and the exploitation powers of the GA.

To assess the sensitivity of the optimal parameters values to the changes in the GA strategies, the mating strategy was changed from mating among the Best 25% of the population to mate the Best 10% of the population with the Worst 10% of the population in the generation. The same problem was solved 100 times under the optimal conditions ([3 0.1 1 0.1 66]) found earlier and the average fitness value was 19038 while the average number of generations was 30.8. This clearly shows that the optimal values of the parameters found under the old mating strategy are not optimal under the new mating strategy.

To appreciate the power of the computer used aspect on the performance of the GA, a 100 simulations were done using AMD Phenom (tm) IIX6 computer using the same optimal conditions [3 0.1 1 0.1 66] found earlier and the average fitness value was 19059.1 while the average number of generations was 9. This clearly shows that the optimal values of the parameters found using the i7 core computer are not optimal under the AMD Phenom (tm)

IIX6 computer as the average fitness value deteriorated due to the sever decrease in the average number of generation.

4.2 Experimentation on Problem 2

The same procedure used to optimize problem 1 was used to optimize problem 2 that contains 6 periods and 15 products. The only difference is that the best regression model for the fitness function was nonlinear. It is worth to mention here that the only parameter involved in the fitness function regression model for this problem was parameter $OffP_m$. This shows clearly that as the size of the problem changes, the behavior of the GA differs significantly, consequently, the whole schema of the regression model changes. This agrees with what was concluded in section 2 about the problem aspect and the NFLT. The resulting model is:

$$\min \hat{Y} = 50882.6 \cos(OffP_m + 106.806) - 23168.6 \cos(2 \times OffP_m + 106.806) + 4862.46 \cos(3 \times OffP_m + 106.806) \quad (15)$$

$$3 \leq NCHC \leq 9 \quad (16)$$

$$0.1 \leq P_c \leq 1 \quad (17)$$

$$0.1 \leq OffP_m \leq 1 \quad (18)$$

$$0.1 \leq P_m \leq 1 \quad (19)$$

$$20 \leq P_s \leq 100 \quad (20)$$

$$4.81 + 0.000205NCHC + 0.346P_c + 0.008887OffP_m + 0.000605P_m + 0.002342P_s < 5 \quad (21)$$

$$P_s, P_m, OffP_m, P_c \geq 0 \quad (21)$$

Note. $NCHC$: P_s integer

$NCHC$ odd.

Solving this model using Excel solver, the optimal values for the parameters [$NCHC$ P_c $OffP_m$ P_m P_s] were [3 0.1 1 0.1 63] respectively. Using these parameters in the underlying GA and simulating the GA for 100 times gave an average fitness value of 32453.12 and variance of 2190.7. The average number of generation was 26.1 generations.

To see whether the optimal values obtained for problem 1 were valid for this problem also, problem 2 was evaluated using the same optimal set of parameters values for problem 1 [3 0.1 1 0.1 66]. The average fitness value was 32491.1 with variance of 1908.5. The average number of generation was 22.9 generation. The following test of hypothesis was conducted:

H_0 : The mean performance of the GA for problem 2 under parameters values [3 0.1 1 0.1 63] is the same as the mean performance of the GA for problem 2 under parameters values [3 0.1 1 0.1 66].

H_1 : The mean performance of the GA for problem 2 under parameters values [3 0.1 1 0.1 63] is less than the mean performance of the GA for problem 2 under parameters values [3 0.1 1 0.1 66].

Table (10) gives a summary of the data.

Table 10. Summary of data for hypothesis H_0

Sample	N	Mean	StDev	SE Mean
1	100	32453.1	46.8	4.7
2	100	32491.1	43.7	4.4

Note. The difference between two means T-Test results were as follows:

Difference = $\mu(1) - \mu(2)$

Estimate for difference: -37.98

95% upper bound for difference: -27.40

T-Test of difference = 0 (vs <): T-value = -5.93 P-value = 0.000 DF = 197.

The results show that the P -value is less than 0.05, the matter that allowed us to conclude that the mean performance of the GA under parameters values [3 0.1 1 0.1 63] is less than the mean performance of the GA under

parameters values [3 0.1 1 0.1 66] for problem 2, consequently, the model was able to tune the GA for this specific problem. This result agrees with what was shown in section 2 about the problem size.

5. Conclusions

When designing a GA for a certain problem under execution time constraint and specific computer power, it is important to tune the GA parameters values to optimize the utilization of the available execution time to reach a compromise between the values of the different GA parameters. Unfortunately, this optimal combination of the GA parameters values is valid only for a specific problem under certain environment, hence, parameters tuning process should be conducted each time the type of the problem, the size of the problem, the GA strategies, the execution time, or the computer used are changed. This result agrees with "No Free Lunch" theorem. The results showed that tuning the GA parameters under execution time constraint using multiple regression along with deterministic optimization is an effective and robust method.

Acknowledgment

The authors are grateful to the Applied Science Private University, Amman, Jordan, for the financial support granted to this research.

References

- Bäck, T. (1996). *Evolutionary algorithms in theory and practice*. Oxford University Press, New York.
- Bartz-Beielstein, T. (2003). Experimental Analysis of Evolutionary Strategies-Overview and Comprehensive Introduction. *Technical Report, 3*, 157. University of Dortmund.
- Birattari, M., Stutzle, T., Paquete, L., & Varrentrapp, K. (2002). *A Racing Algorithm for Configuring Metaheuristics*. In: Proceedings of Genetic and Evolutionary Computation Conference (GECCO 2002), 11-18.
- Culberson, J. C. (1996). *On the Futility of Blind Search, Technical Report TR*, 96-18. Department of Computing Science, University of Alberta, Edmonton, Alberta, Canada, July.
- De Jong, K. (1975). *The analysis of the behavior of a class of genetic adaptive systems*. PhD Thesis, University of Michigan.
- de Pina, A. A., Albrecht, C. H., de Lima, B. S. L. P., & Jacob, B. P. (2011). Tailoring the particle swarm optimization algorithm for the design of offshore oil production risers. *Optimization and Engineering, 12*, 1-2, 215-235. <http://dx.doi.org/10.1007/s11081-009-9103-5>
- Eiben, A. E., Hinterding, R., & Michalewicz, Z. (1999). Parameter Control in Evolutionary Algorithms. *IEEE Transactions on Evolutionary Computation, 3*, 124-141.
- Fogel, D., & Ghozeil, A. (1996). *Using Fitness Distributions to Design More Efficient Evolutionary Computations*. In: Proceedings of the 1996 IEEE Conference on Evolutionary Computation, Nagoya, Japan, 11-19. IEEE Press, Piscataway. <http://dx.doi.org/10.1109/icec.1996.542328>
- Grefenstette, J. J. (1986). Optimization of Control Parameters for Genetic Algorithms. *IEEE Transactions on System, Man, and Cybernetics, 16*, 122-128. <http://dx.doi.org/10.1109/TSMC.1986.289288>
- Hart, W. E., & Belew, R. K. (1991). *Optimizing an Arbitrary Function is Hard for the Genetic Algorithm*. In: Proceedings of the Fourth International Conference on Genetic Algorithms, 190-195.
- Ho, Y. C. (1999). The No Free Lunch Theorem and the Human Machine Interface, *IEEE Control Systems Magazine*, June.
- Ho, Y., & Pepyne, D. L. (2001). *Simple Explanation of the No Free Lunch Theorem of Optimization*. Proceedings of the 4th IEEE, Conference on Decision and Control, December.
- Joaquin, P., Rodolfo, P., Juan, F., Guillermo, R., Laura, C., Graciela, M., & Hector, F. (2004). *Self-Tuning Mechanism for Genetic Algorithms Parameters*. An Application to Data-Objective Allocation in the Web. International Conference, Assisi, Italy, Proceedings part IV, 77-86, May.
- Maria, A., & Tania, P. (2011). Tuning Genetic Algorithm Parameters to Improve Convergence Time. *International Journal of Chemical Engineering, 7*. Article ID 646917.
- Pyrz, M. (2004). Evolutionary Algorithm Integrating Stress Heuristics for Truss Optimization. *Optimization and Engineering, 5*(1), 45-57. <http://dx.doi.org/10.1023/B:OPTE.0000013634.36489.32>
- Radeliff, N. J., & Surry, P. D. (1995). Fundamental Limitations on Search Algorithms: Evolutionary Computing

- in Perspective, *Computer Science Today*, 1000, 275-291. <http://dx.doi.org/10.1007/BFb0015249>
- Ramadan, Z. S. (2011). *Bayesian multi-objective design of reliability testing*. PhD Thesis, University of Ohio. <http://dx.doi.org/10.1109/4235.771166>
- Süer, G. A., Badurdeen, F., & Dissanayake, N. (2008). Capacitated lot sizing by using multi-chromosome crossover strategy. *Journal of Intelligent Manufacturing*, 19, 273-282. <http://dx.doi.org/10.1007/s10845-008-0080-x>
- Wolpert, D. H., & Macready, W. G. (1997). No Free Lunch Theorems for Optimization. *IEEE Transactions on Evolutionary Computation*, 11, 67-82. <http://dx.doi.org/10.1109/4235.585893>
- Yuan, B., & Gallagher, M. (2004). *Statistical Racing Techniques for Improved Empirical Evaluation of Evolutionary Algorithms*. In: Proceedings of the 8th International Conference on Parallel Problem Solving from Nature, 172-181. http://dx.doi.org/10.1007/978-3-540-30217-9_18
- Yuan, B., & Gallagher, M. (2005). A hybrid approach to parameter tuning in genetic algorithms. *Evolutionary Computation*, 2, 1096-1103. The IEEE Congress, 2005.

Copyrights

Copyright for this article is retained by the author(s), with first publication rights granted to the journal.

This is an open-access article distributed under the terms and conditions of the Creative Commons Attribution license (<http://creativecommons.org/licenses/by/3.0/>).