Experimental Analysis of the Tournament Size on Genetic Algorithms*

CCS CONCEPTS

• **Computer systems organization** → **Embedded systems**; *Redundancy*; Robotics; • **Networks** → Network reliability;

KEYWORDS

ACM proceedings, LaTeX, text tagging

ACM Reference format:
. 2018. Experimental Analysis of the Tournament Size on Genetic Algorithms. In *Proceedings of the Genetic and Evolutionary Computation Conference 2018, Kyoto, Japan, July 15–19, 2018 (GECCO '18),* 4 pages.
https://doi.org/10.1145/nnnnnnn.nnnnnnn

1 INTRODUCTION

Genetic Algorithm (GA) is a well-known meta-heuristic that is used to seek solutions given a search space. The GA relies on operators such as the crossover operator, the mutation operator and the selection operator. These operators have their own and specific role in the GA. The selection operator selects solutions to compose the next generation given their measured fitness function quality value, with higher solutions having higher probability to compose the next generation. The crossover operator generates new solutions by recombining pairs of good solutions to generate new ones. The mutation operator is applied to the population to sustain diversity, by changing the value of a part of a solution. Both crossover and mutation operators are applied to explore new points in the search space [6].

One of the most used selection operator is the tournament selection operator. It selects solutions based on their fitness value and on one parameter, the tournament size. Controlling the tournament size parameter is argued to be one way of adjusting the balance between the exploration and explotation [6], which governs the search process in GA. Although it is a very important balance, it cannot be set directly [9]. The selection pressure of tournament selection is expected to increase as the tournament size becomes larger. That is, the higher the selection pressure, the more the better solutions influences the next generation [19].

As a rule of thumb, small values, such as 2 or 3, are widely used. Filipovic [9] argued that small vales lead to small selection pressure and may be good choices, but, very often, the search process converges too slowly with smaller tournament size (and too fast with bigger tournament size).

We would like to reconsider the idea that exists a relationship between exploration versus exploitation and the tournament size parameter by verifying, experimentally, the real impact of the tournament size on the Genetic Algorithm.

Consequently, our goal in this paper is to observe the performance of the tournament selection in a real-valued Genetic Algorithm. For that we applied the Genetic Algorithm to 24 noise-free BBOB benchmark functions with 10, 20 and 40 dimensions [13]. We explore the number of solutions to be selected by the Tournament operator, with values from 2 to 25. Then we analyze the results to verify any relationship among the tournament size value, the benchmark functions, and the performance. We also explore briefly the idea that the value of the tournament size value had different effects at different points in the algorithm.

A review of the literature shows that there is a preference to small values for the tournament size, as 2 or 3 [1, 4, 5, 8, 10, 12, 14, 15, 17, 21–24]. This has been a popular choice since 1991. Most of the papers use little scientific justification for their choice for the tournament size value. On the other hand, Nicolau [21] tried to define empirically the value of the tournament size for his binary GA.

2 TOURNAMENT SELECTION

The selection scheme is intended to improve the average quality of the GA by giving solutions of higher quality a higher probability of composing the next generation [6].

Goldberg and Deb [11] stated that there exists several selection schemes commonly used in GA, such as the rank selection, the roulette-wheel selection and the tournament selection. They also declare that the complexity of the tournament selection is O(n) and say that tournaments are often held with tournament size s = 2, although larger tournament sizes may be considered.

2.1 The Tournament Selection Procedure

Tournament selection is translation and scaling invariant [18]. That means that the behavior of the selection method is not affect by the scale of the fitness values [2].

Tournament selection is conducted as: choose a number K of solutions randomly from the population and select the best solution from this group as a parent for the crossover operator. Repeat this operation for each parent necessary. This K is the tournament size parameter that we investigate in this paper.

2.2 Selection Schemes and BBOB functions

Holtschulte and Moses, in [16] used rank selection in their GA implementation and applied it to the BBOB functions. They declared that rank selection should be replaced by an elitism selection scheme to improve the performance of their GA implementation by spreading of high quality solutions through the population.

*Produces the permission block, and copyright information

GECCO '18, July 15–19, 2018, Kyoto, Japan

ACM ISBN 978-x-xxxx-xxxx-x/YY/MM…$15.00
https://doi.org/10.1145/nnnnnnn.nnnnnnn

One of these elitist selection scheme is the fitness-based roulette-wheel. However, it has been shown to lead to premature convergence [3]. Nicolau, [21] uses the tournament selection in his GA aiming to solve the BBOB functions. He declares "for harder problem domains, such as the ones on the BBOB-Benchmark suite, the tournament selection scheme is more appropriate.".

Then, in this work we chose to analyze the tournament selection behavior in the BBOB functions.

Also, tournament selection is said to be simple to code, efficient in both parallel and non parallel architectures, and its pressure is able to be adjusted for different domains [19].

2.3 Tournament Pressure

Muhlenbein et al., in [20], introduced the concept of selection intensity, which is to measure the pressure of selection schemes. The selection intensity is defined as the expected average fitness of the population after selection.

Miller et al. [19], argued that with higher tournament sizes, the selection pressure rises resulting in higher convergence rates, because the convergence rate of a GA is largely determined by the selection pressure. Accordingly, if the selection pressure is too low, the convergence rate will be slower if it is too high, there is an increased chance of the GA prematurely converging.

We understand that it is important to explore the values for tournament size, since this value seems to determine the balance of exploration and exploitation, by changing the selection intensity. It is possible that this balance could change given a domain, once it is expected that every domain holds a specific characteristic that would need a different proportion of this balance.

2.4 Tournament Size in the Literature

In 1991, Goldberg [10], analyzed the tournament selection with size 2 in a real-coded GA. He recognized, empirically, that after some time only individuals with relatively high function values will be represented in the population. Also he compares the importance of mutation operators and the tournament selection operator and states that "the addition of some small amount of creeping mutation or other genetic operators should not materially affect these results".

Agrawal [1], in 1995, implemented real-coded GA with tournament selection with size 2. Their goal was to compare the results of the simulated binary crossover (SBX) with the binary single-point crossover and the blend crossover. They concluded that the SBX has search power similar to the single-point crossover and the SBX performs better in difficult test functions than the blend crossover.

In 1999, Tsutsui et al. [24] proposed a simplex crossover (SPX), a multi-parent recombination operator for real-coded genetic algorithms. For their experiments, they used a real-coded GA with tournament selection with size 2 and showed that "SPX works well on functions having multi modality and/or epistasis with a medium number of parents: 3-parent on a low dimensional function or 4 parents on high dimensional functions.".

Later, in 2000, Deb [8], implemented a binary GA and a real-coded GA, with tournament size 2. He aimed to devise a penalty function approach that does require any penalty parameter. Because all of their search spaces are defined in the real space, the real-coded GA was more suited in finding feasible solutions, but in all cases, this approach " [has] a niche over classical methods to handle constraints".

In 2001, Beyer et al. [4], analyzed the results of using the simulated binary crossover (SBX), the fuzzy recombination operator and the blend crossover (BLX), all with no mutation and tournament size 2 in a real-coded GA. The results indicated that those crossovers exhibit similar performance and linear convergence order.

Bhunia et al. [5], in 2009, created a tournament GA application to solve an economic production lot-size (EPL) model.

Sawyerr et al. [22], in 2015, proposed the RCGAu, hybrid real-coded genetic algorithm with "uniform random direction" search mechanism. They used tournament selection with size 3 to explore the suite of noiseless black-box optimization testbed and the results showed great performance in some functions as f_7 and f_{21}.

Goldberg et AL [12], in 1993, implemented a binary GA using the tournament selection as a convenient way to control selection pressure. They wanted to study the theory aspects of the GA focusing in the mixing building block. The actual effect of the tournament value was not investigated empirically in the work.

In 1999, Harik et al. [15] studied a simple GA and experimented with the tournament size. Their work shows that a given solution quality can be obtained faster by the tournament size equal to 4 or 8, instead of tournament size equal to 2 or 40.

Also, Harik et al. [14] implement a binary coded GA using tournament selection. They analyze the impact of the tournament size on their GA with a 100-bit one-max function. They inspected this implementation with tournament sizes of 2, 4, and 8. They observed that the proportion of the building blocks changed with the tournament size, being higher for smaller tournament sizes.

In 2007, Kaelo et al. [17] compared the usage of the tournament size in a real-coded GA. They discuss the impact of using tournament size of 2 and tournament size of 3. The results of comparing these two tournament sizes indicated that the tournament with size 3 is more greedy, reduces the number of functions evaluations greatly and also improves CPU usage, therefore, it impacts negatively in the success rate of their GA.

Also in 2009, Nicolau [21] implemented a binary GA to solve the noise-free BBOB 2009 testbed. Trough a quick experimentation with a subset of benchmark functions which yielded the following equation: $k = P/500$, where P is the population size. (1)

From the works showed here, it is possible to understand that there is a preference to small values for the tournament size. We traced back to 1991, and we found out that it has been a popular choice ever since. We must highlight that these values are chosen usually chosen without much justification.

3 EXPERIMENT DESIGN

3.1 The Genetic Algorithm

To test the influence of the tournament size we implemented a simple real-valued Genetic Algorithm to explore the search space of the 24 noise free N dimension BBOB benchmark functions [13].

Genome Representation and Evolutionary Tools. Each solution is represented as real valued array, where each element is one input to a N dimension noise free benchmark BBOB function. Therefore, each solution of the GA has size of N real value elements. Each

solution's values is set to the be into the interval [-4, 4], since the target functions are bounded by these values. It uses the Uniform crossover, elitism and Gaussian mutation as evolutionary operators. The relevant parameters were set as in the Table 1 and Table 2.

Number of Evaluations	40000
Population Size	500
Crossover chance	0.9
Mutation Chance	0.1
Elitism size	1

Table 1: GA Relevant Parameters

Mean	0
Standard Deviation	1
Ind. Prob. Attribute to be Mutated	0.1

Table 2: Gaussian Mutation parameters

Fitness Function. The fitness function considered are the N dimensions noise free benchmark BBOB function [13].

Implementation Details. Chuang et al. [7], proposed a Genetic Algorithm specific to run on the noise free BBOB benchmark functions. Some good practices presented in this work were incorporated to the GA. These practice are the concept of "restart strategy" and the concept of "alleviate stagnation".

The restart strategy concept is defined by Chuang et al. as: "For each restart, the initial population is uniformly and randomly sampled within the search space. Whenever the restart condition is met, the algorithm will be reinitialized without using any information about the last test run. This process is iterated until the stopping criteria are met, i.e., maximum number of function evaluations has been reached, or the function value is less than the target precision."

The restart condition used in the GA is value of the standard deviation of the current population. If it is smaller that 10^{-12}, then the restart condition is triggered. In this case the algorithm population is reinitialized randomly, as it is in the first initialization.

In the GA, we consider information about the last run, by utilizing the Elitism strategy on the best solution prior to the restart. The maximum number of function evaluations chosen is $(4 * 10^5)$ and the target precision chosen is (10^{-8}), values compatible with the ones in the previous work.

3.2 Experiments

We made two experiments aiming to verify the impact of different values for the tournament size. In the first experiment, we analyzed the relation between the tournament size and the performance on the BBOB benchmark functions. In the second experiment, we analyze the impact of changing the values of the tournament size *during* the execution.

We used the Friedman Test to determine whether any of the for the tournament size show a significant difference in the average performance. In each of these tests, we set $\alpha = 0.05$. For this analysis, the dependent variable being the average performance, the blocking variable is the BBOB function number and the treatment variable is the tournament size value.

Fixed tournament size. To analyze the impact of the tournament size into the GA in terms of quality of result, the following experiments were performed. First we analyze different values for the tournament size applied to the BBOB benchmark functions with 10, 20 and 40 dimensions. The tournament sizes were select, arbitrarily, from 2 to 25. For each combination of BBOB benchmark function, dimension, and tournament size we performed 40 repetitions and used the mean final result as the performance for that combination.

4 RESULTS

The Friedman Test showed no significant difference in values among tournament size values across all functions. The test was repeated for dimensions: 10, 20 and 40 dimensions and similar results were found. Detailed data can be visualized on the following Table 3.

Dimension size	Chi-squared	P-value
10	31.497	0.1111
20	27.903	0.2195
40	23.288	0.444

Table 3: Friedman Test results.

To get a finer intuition about the results, we show some visual examples, separated in two groups of Figures. The first group, shows the mean value achieved by the GA given a function. The second one, shows the convergence plot with the mean of the values found at each generation, the function target value with a given tournament size.

4.1 Tournament Size by Functions Analysis

The Figures **??**, **??**, **??** exemplify that for the F5, F12, and F21 function (with 10, 20 and 40 dimensions) that changing the tournament size does not lead to significant better final values found by the GA. The gray shaded area represents the 95% confidence level interval for predictions from a linear model for each scenario showed and demonstrate that no value of the tournament size has consistently better results. For all Figures, the mean of 40 repetitions is shown as bullets and the bars represent the standard deviation, with 40 dimensions.

4.2 Convergence Analysis

The Figure **??** exemplifies that for the F1 function (with 10 dimensions), the GA is converging towards the optimum target value, represented by the bottom, horizontal line. We found similar behavior in most of the functions.

5 CONCLUSION

We proposed an experimental analysis of the impact of the tournament size on the BBOB benchmark functions. We verified that there are little mathematical or experimental studies that supports a choice of value for the tournament size. Usually, the tournament size is chosen to be 2 or 3, based on "previous results".

We analyzed a group of tournament size values and we found that no significant relationship was discovered.

We are aware that our work is limited to few tournament size values. Also, we understand that the other parameters may have some influence in the results. We know that there is much more that could be done in this field of study, therefore we propose that more investigations need to be done, in order to truly understand the role of the tournament size value related to the quality of final results. For that, we suggest that some kind of adaptative algorithm should be explored. Also, we would like to explore whether our results were mainly related with a particular set of GA operators.

REFERENCES

[1] Ram Bhushan Agrawal, K Deb, and RB Agrawal. 1995. Simulated binary crossover for continuous search space. *Complex systems* 9, 2 (1995), 115–148.

[2] Thomas Bäck, David B Fogel, and Zbigniew Michalewicz. 2000. *Evolutionary computation 1: Basic algorithms and operators*. Vol. 1. CRC press.

[3] James E Baker. 1987. Reducing bias and inefficiency in the selection algorithm. In *Proceedings of the second international conference on genetic algorithms*. 14–21.

[4] H-G Beyer and Kalyanmoy Deb. 2001. On self-adaptive features in real-parameter evolutionary algorithms. *IEEE Transactions on evolutionary computation* 5, 3 (2001), 250–270.

[5] AK Bhunia, S Kundu, T Sannigrahi, and SK Goyal. 2009. An application of tournament genetic algorithm in a marketing oriented economic production lot-size model for deteriorating items. *International Journal of Production Economics* 119, 1 (2009), 112–121.

[6] Tobias Blickle and Lothar Thiele. 1995. A Mathematical Analysis of Tournament Selection.. In *ICGA*. 9–16.

[7] Yao-Chen Chuang and Chyi-Tsong Chen. 2012. Black-box optimization benchmarking for noiseless function testbed using a direction-based RCGA. In *Proceedings of the 14th annual conference companion on Genetic and evolutionary computation*. ACM, 167–174.

[8] Kalyanmoy Deb. 2000. An efficient constraint handling method for genetic algorithms. *Computer methods in applied mechanics and engineering* 186, 2 (2000), 311–338.

[9] Vladimir Filipović. 2012. Fine-grained tournament selection operator in genetic algorithms. *Computing and Informatics* 22, 2 (2012), 143–161.

[10] David E Goldberg. 1991. Real-coded genetic algorithms, virtual alphabets, and blocking. *Complex systems* 5, 2 (1991), 139–167.

[11] David E Goldberg and Kalyanmoy Deb. 1991. A comparative analysis of selection schemes used in genetic algorithms. *Foundations of genetic algorithms* 1 (1991), 69–93.

[12] David E Goldberg, Kalyanmoy Deb, and Dirk Thierens. 1993. Toward a better understanding of mixing in genetic algorithms. *Journal of the Society of Instrument and Control Engineers* 32, 1 (1993), 10–16.

[13] Nikolaus Hansen, Anne Auger, Steffen Finck, and Raymond Ros. 2010. *Real-parameter black-box optimization benchmarking 2010: Experimental setup*. Ph.D. Dissertation. INRIA.

[14] George Harik, Erick Cantú-Paz, David E Goldberg, and Brad L Miller. 1999. The gambler's ruin problem, genetic algorithms, and the sizing of populations. *Evolutionary Computation* 7, 3 (1999), 231–253.

[15] Georges R Harik, Fernando G Lobo, and David E Goldberg. 1999. The compact genetic algorithm. *IEEE transactions on evolutionary computation* 3, 4 (1999), 287–297.

[16] Neal J Holtschulte and Melanie Moses. 2013. Benchmarking cellular genetic algorithms on the BBOB noiseless testbed. In *Proceedings of the 15th annual conference companion on Genetic and evolutionary computation*. ACM, 1201–1208.

[17] P Kaelo and MM Ali. 2007. Integrated crossover rules in real coded genetic algorithms. *European Journal of Operational Research* 176, 1 (2007), 60–76.

[18] Michael de la Maza and Bruce Tidor. 1993. An analysis of selection procedures with particular attention paid to proportional and Boltzmann selection. In *Proceedings of the 5th International Conference on Genetic Algorithms*. Morgan Kaufmann Publishers Inc., 124–131.

[19] Brad L Miller, David E Goldberg, et al. 1995. Genetic algorithms, tournament selection, and the effects of noise. *Complex systems* 9, 3 (1995), 193–212.

[20] Heinz Mühlenbein and Dirk Schlierkamp-Voosen. 1993. Predictive models for the breeder genetic algorithm i. continuous parameter optimization. *Evolutionary computation* 1, 1 (1993), 25–49.

[21] Miguel Nicolau. 2009. Application of a simple binary genetic algorithm to a noiseless testbed benchmark. In *Proceedings of the 11th Annual Conference Companion on Genetic and Evolutionary Computation Conference: Late Breaking Papers*. ACM, 2473–2478.

[22] Babatunde A Sawyerr, Aderemi O Adewumi, and M Montaz Ali. 2015. Benchmarking RCGAu on the Noiseless BBOB Testbed. *The Scientific World Journal* 2015 (2015).

[23] Babatunde A Sawyerr, Montaz M Ali, and Aderemi Oluyinka Adewumi. 2011. A comparative study of some real-coded genetic algorithms for unconstrained global optimization. *Optimization Methods and Software* 26, 6 (2011), 945–970.

[24] Shigeyoshi Tsutsui, Masayuki Yamamura, and Takahide Higuchi. 1999. Multi-parent recombination with simplex crossover in real coded genetic algorithms. In *Proceedings of the 1st Annual Conference on Genetic and Evolutionary Computation-Volume 1*. Morgan Kaufmann Publishers Inc., 657–664.