Experimental Analysis of the Tournament Size on Genetic Algorithms

Author1
Institute1
author1@example.com

Author2
Institute2
author2@example.com

ABSTRACT

200 Word Abstract

CCS CONCEPTS

• **Computing methodologies → Genetic algorithms**;

KEYWORDS

Genetic Algorithm, Tournament Selection, Selection Schemes

ACM Reference Format:
Author1 and Author2. 2018. Experimental Analysis of the Tournament Size on Genetic Algorithms. In *Proceedings of the Genetic and Evolutionary Computation Conference 2018 (GECCO '18).* ACM, New York, NY, USA, 7 pages. https://doi.org/10.1145/nnnnnnn.nnnnnnn

1 INTRODUCTION

The Tournament Selection is a very popular selection operator for Genetic Algorithms and Evolutionary Computation in general. Some of the reasons for this popularity are that it is very simple to implement, and that it has a single control parameter, the *Tournament Size*, also known as "Tournament's k".

How should the value of the Tournament Size be chosen? Early works claimed that the value of k could be used to control the balance between exploration and exploitation in the search [6]. As k gets larger, the selection pressure is expected to increase, and better sollutions are expected to exert a larger influence in the following generations [17].

On the other hand, works in the past 15 years have commonly used very small values for k, such as 2 or 3 [4, 5, 7, 15, 19–21]. The reason for this change is hard to identify, as most papers either do not justify their choice of parameters, or lightly mention a concern over computational cost for this rule of thumb.

Precious few works look too closely at their choice of tournament size. One example is Nicolau [19] who tried to empiricaly define a relationship between population size and tournament size for fine tuning a binary GA.

In order to question and further understand these assumptions and rules of thumb, we present an experimental study on the role of the tournament selection size. Our study focuses on a real-valued GA applied to the noise-free Black Box Optimization Benchmark (BBOB) functions.

We vary the mutation operator (uniform or SBX), the generational scheme, the modality of the objective function (unimodal or multimodal), and the dimensionality of the objective function (10,20,40). We observe how the runtime and the quality of the final answer changes as the value of k changes.

Our results show that... and can be used to understand that...

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 [9] 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 [16]. 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.

Often tournaments are held only between two individuals (binary tournament) but a generalization is possible to an arbitrary group size k called tournament size [6, 9]. A tournament selection, as any selection mechanism in GA, simply favors the selection of better individuals of a population to influence the next generation.

2.2 Selection Schemes and BBOB functions

Holtschulte and Moses, in [14] 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.

One of these elitist selection scheme is the fitness-based roulette-wheel. However, it has been shown to lead to premature convergence [3]. Nicolau, [19] 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 [17].

2.3 Tournament Pressure

Muhlenbein et al., in [18], 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. [17], 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 [8], 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. [22] 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 [7], 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. [20], 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 [10], 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. [13] 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. [12] 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. [15] 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 [19] 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:

$$k = P/500, \text{ where P is the population size.} \tag{1}$$

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 [11].

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 [-5, 5], since the target functions are bounded by these values. We considerer two crossover operators when experimenting with the GA: the Uniform crossover and the Simulated Binary crossover (SBX). The other evolutionary operators are elitism and Gaussian mutation. The relevant parameters were set as in the Table 1 and Table 2.

Number of Evaluations	40000
Population Size	800
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 [11].

Generational Scheme. Two generation schemes were used. In te first generation scheme, scheme A, the next population is replaced entirely by the generated population, while in the second generation scheme, scheme B, the best possible solutions from the current *and* the generation population are chosen to compose the next population.

We observed that the BBOB Functions can be separated in Unimodal or multimodal, where a unimodal function is a function with only one local optimum (that is equal to the global optimum) while multimodal means that the function has many local optimum. In our experiments, we consider that the impact the results may be different acknowledged the 2 groups and therefore we analyze the results by grouping all functions all together and by separating them into unimodal or multimodal.

The multimodal functions are [11]: Rastrigin Function, Weierstrass Function, Schaffers Function, Schaffers Function ,moderately ill-conditioned, Composite Griewank-Rosenbrock Function F8F2, Schwefel Function, Gallagher's Gaussian 101-me Peaks Function, Gallagher's Gaussian 21-hi Peaks Function, Katsuura Function, and Lunacek bi-Rastrigin Function.

While the unimodal functions are: Ellipsoidal Function, Linear Slope, Atractive Sector Function, Step Ellipsoidal Function, Rosenbrock Function - original, Rosenbrock Function - rotated, Discus Function, Bent Cigar Function, Sharp Ridge Function, and Different Powers Function.

3.2 Experiments

We made a experiment 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, when combined with the Uniform crossover or with the SBX.

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.

4 RESULTS

4.1 Generational Scheme A

The Friedman Test showed no significant difference in values among tournament size values across all functions, for any configuration, with the uniform crossover. This results hold for every scenario with 40 dimensions, given: all functions (unimodal or multimodal); only the unimodal functions; or only the multimodal functions.

When considering 20 dimensions, the results vary more. For the unimodal functions no significant difference was found. But when multimodal functions are considered, there is a significant difference in the results. Also while considering all functions (unimodal or multimodal) a even higher significant difference in the results was found. Detailed data can be visualized on the following Table 3.

Function Group	Dimension size	Chi-squared	P-value
All	10		
Unimodal	10		
Multimodal	10		
All	20	48.405	0.000959
Unimodal	20	15.243	0.8516
Multimodal	20	58.953	3.188e-05
All	40	25.466	0.2753
Unimodal	40	21.222	0.5071
Multimodal	40	30.232	0.1129

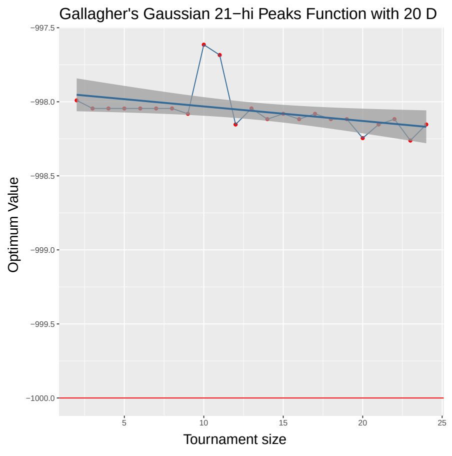
Table 3: Friedman Test results for Uniform Crossover - Generational Scheme A.

For the SBX crossover with 40 or 20 dimensions there is a significant difference in the results, when considering: all functions (unimodal or multimodal); only the unimodal functions; or only the multimodal functions. Detailed data can be visualized on the following Table 4.

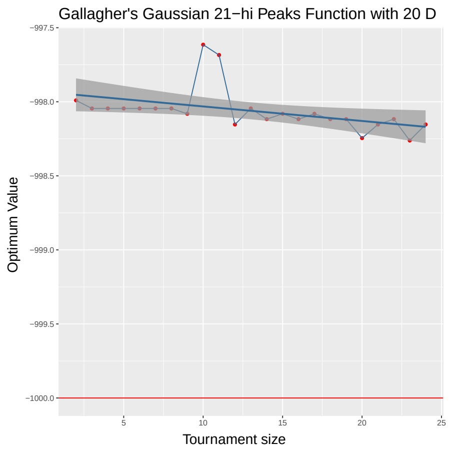
Function Group	Dimension size	Chi-squared	P-value
All	10		
Unimodal	10		
Multimodal	10		
All	20	178.97	< 2.2e-16
Unimodal	20	103.96	1.301e-12
Multimodal	20	88.275	6.727e-10
All	40	250.65	< 2.2e-16
Unimodal	40	143.19	< 2.2e-16
Multimodal	40	116.92	6.296e-15

Table 4: Friedman Test results for SBX Crossover - Gen. Scheme A.

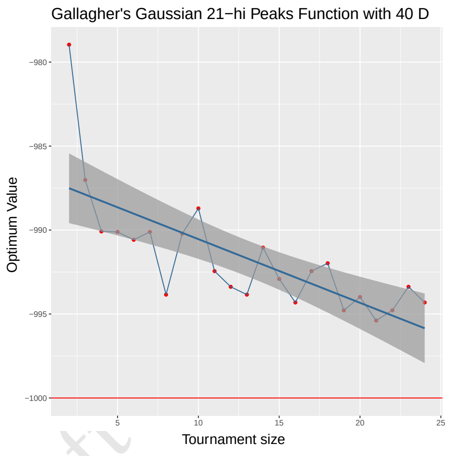
To get a finer intuition about the results, we show some visual examples, separated in two groups of Figures. The first group, shows the mean value achieved by the GA given a function. The second one, shows the convergence plot with the mean of the values found at each generation, the function target value with a given tournament size. All Figures represent the mean of 34 repetitions given a certain dimension.



(a) 10 dimensions.

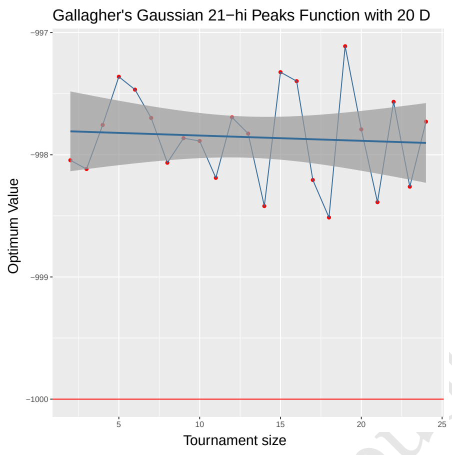


(b) 20 dimensions.

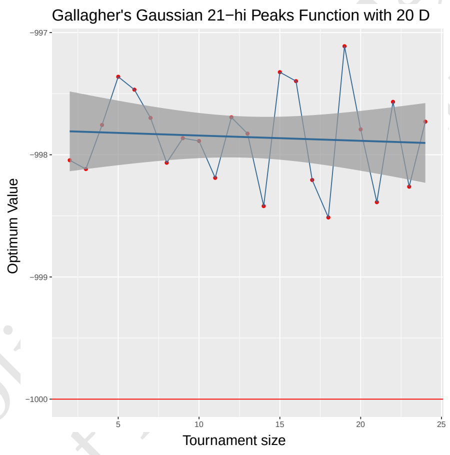


(c) 40 dimensions.

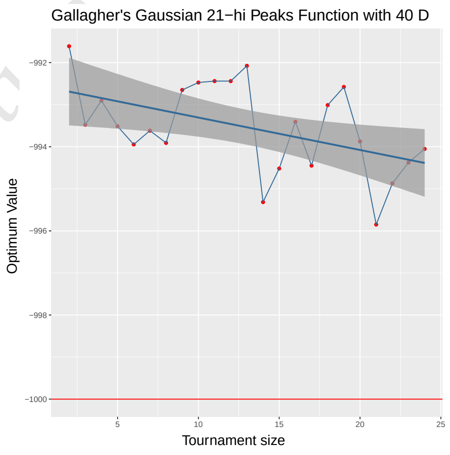
Figure 1: Average performance on different tournament size for the Gallagher's Gaussian 21-hi Peaks Function, when using the SBX crossover - Gen. Scheme A.



(a) 10 dimensions.



(b) 20 dimensions.



(c) 40 dimensions.

Figure 2: Average performance on different tournament size for the Gallagher's Gaussian 21-hi Peaks Function, when using the uniform crossover - Gen. Scheme A.

4.2 Tournament Size by Functions Analysis

The Figures 1 and 2 exemplify that for the Gallagher's Gaussian 21-hi Peaks Function (with 10, 20 and 40 dimensions) changing the tournament size for higher values tend to increase the quality of the results, with greater variation when the SBX crossovers, Figure 1, in used when comparing with the uniform crossover. Figure 2.

For the Discus Function, the Figures 3, 4, (with 10, 20 and 40 dimensions) show that the same trend is observed.

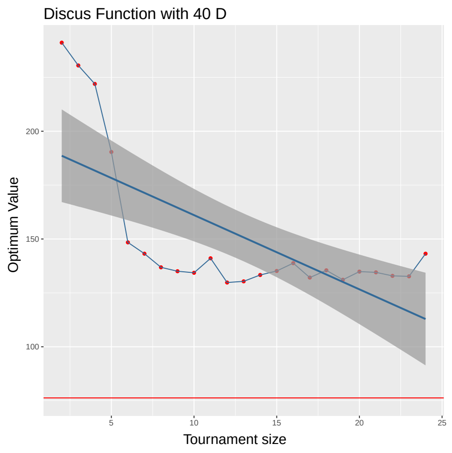
The gray shaded area represents the 95% confidence level interval for predictions from a linear model for each scenario showed and demonstrate that choosing small values for the tournament size, as 2 or 3 can be a very poor choice. For all Figures, the mean of 34 repetitions is shown as bullets and the red line is shows the objective target value for the function.

4.3 Convergence Analysis

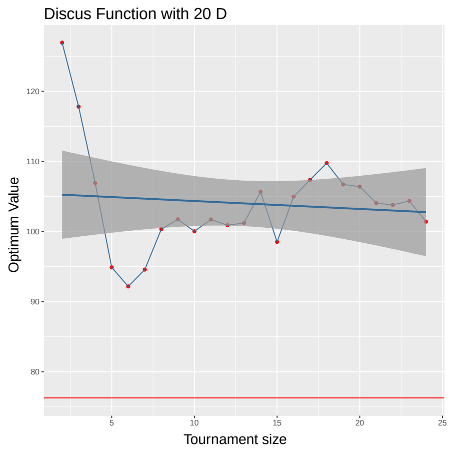
The Figures 5 and 6 exemplify that for the Discus Function with 40 dimensions and with the SBX crossover the GA is converging towards the optimum target value, represented by the bottom, red, horizontal line. The same convergence is observed with the uniform crossover for the Katsuura Function. We found similar behavior in most of the functions, given any crossover operator.

4.4 Generational Scheme B

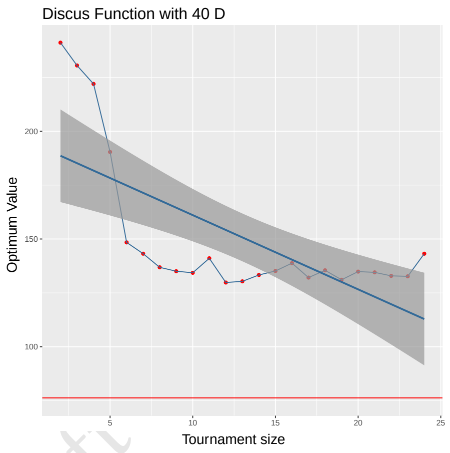
The Friedman Test showed no significant difference in values among tournament size values across all functions, for any configuration, with the SBX crossover. This results hold for every scenario with 40 dimensions, given: all functions (unimodal or multimodal); only the unimodal functions; or only the multimodal functions.



(a) 10 dimensions.

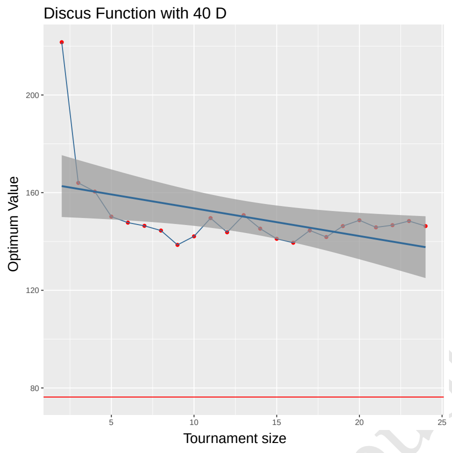


(b) 20 dimensions.

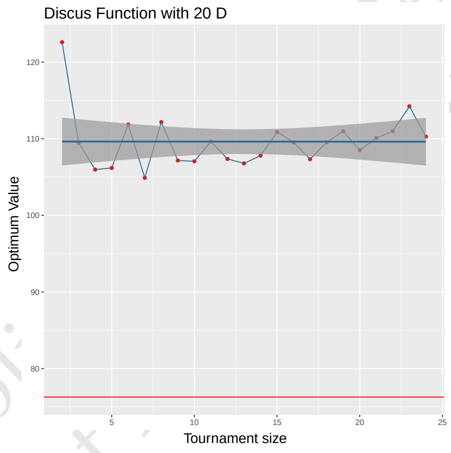


(c) 40 dimensions.

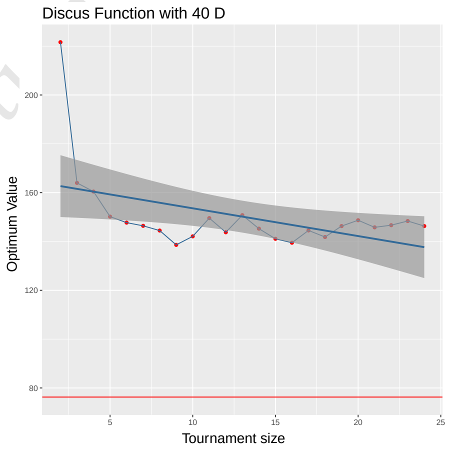
Figure 3: Average performance on different tournament size for the Discus Function, when using the SBX crossover - Gen. Scheme A.



(a) 10 dimensions.



(b) 20 dimensions.



(c) 40 dimensions.

Figure 4: Average performance on different tournament size for the Discus Function, when using the uniform crossover - Gen. Scheme A.

As for the generation Scheme A, we show visual examples 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. All Figures represent the mean of 34 repetitions given a certain dimension.

4.5 Tournament Size by Functions Analysis

The Figure 7 that exemplifies that for the Gallagher's Gaussian 21-hi Peaks Function (with 10, 20 and 40 dimensions) changing the tournament size does not lead to significant better final values found by the GA. The Figure 8 shows the same trend, but for the Discus Function.

The gray shaded area represents the 95% confidence level interval for predictions from a linear model for each scenario showed and demonstrate that choosing small values for the tournament size, as 2 or 3 can be a very poor choice. For all Figures, the mean of 34 repetitions is shown as bullets and the red line is shows the objective target value for the function.

4.6 Convergence Analysis

The Figures 5 and 6 exemplify that for the Discus Function with 40 dimensions and with the SBX crossover the GA is converging towards the optimum target value, represented by the bottom, red, horizontal line. The same convergence is observed with the uniform

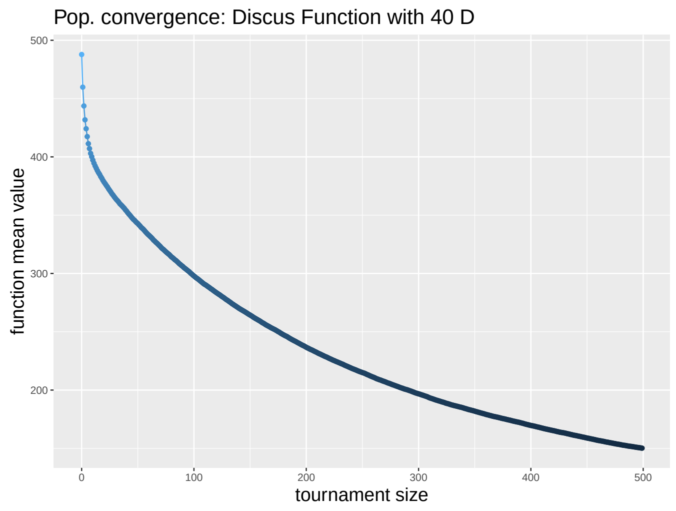


Figure 5: Population convergence for Discus Function.

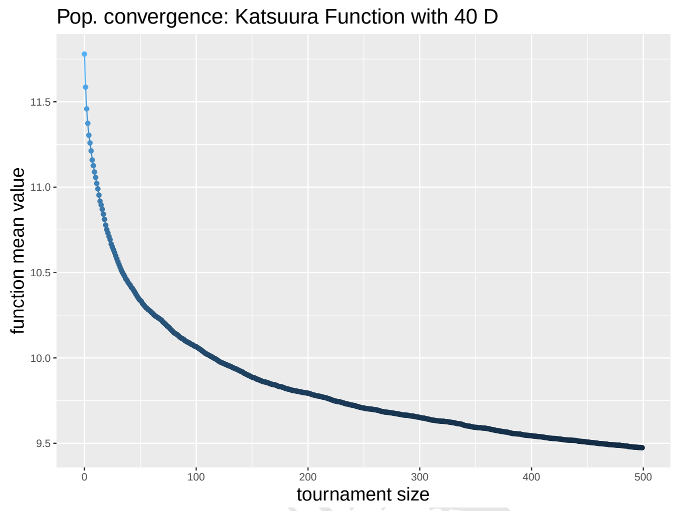


Figure 6: Population convergence for Katsuura Function.

crossover for the Katsuura Function. We found similar behavior in most of the functions, given any crossover operator.

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 on different selection operators and we found that using the tournament size with values like 2 or 3 may not be good choices, depending on the characteristic of the problem you are exploring. For that, we propose that some sort of self-adaptative procedure should be tried, aiming to minimize the impact of bad choice of values for the tournament size.

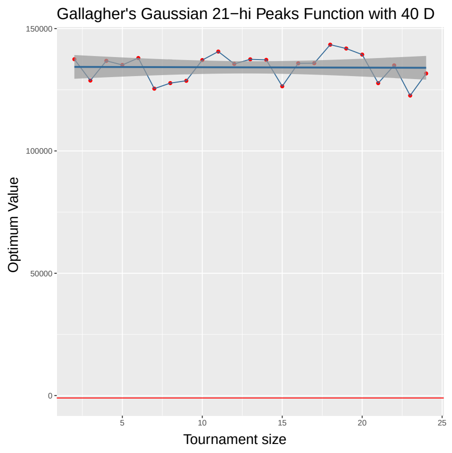
We are aware that our work is limited to few tournament size values. Also, we understand that the other parameters may have some influence in the results. We know that there is much more that could be done in this field of study, therefore we propose that more investigations need to be done, in order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

Function Group	Dimension size	Chi-squared	P-value
All	10		
Unimodal	10		
Multimodal	10		
All	20		
Unimodal	20		
Multimodal	20		
All	40	27.335	0.1988
Unimodal	40	31.111	0.09387
Multimodal	40	25.928	0.2548

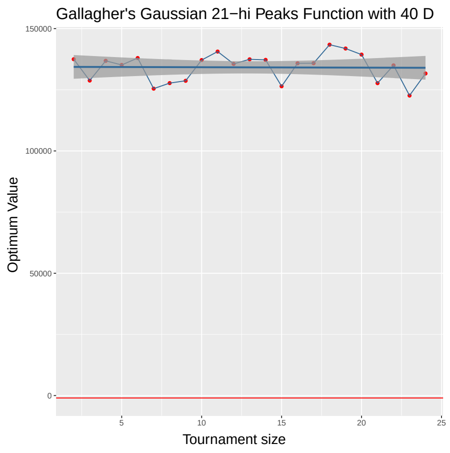
Table 5: Friedman Test results for SBX Crossover - Gen. Scheme B.

REFERENCES

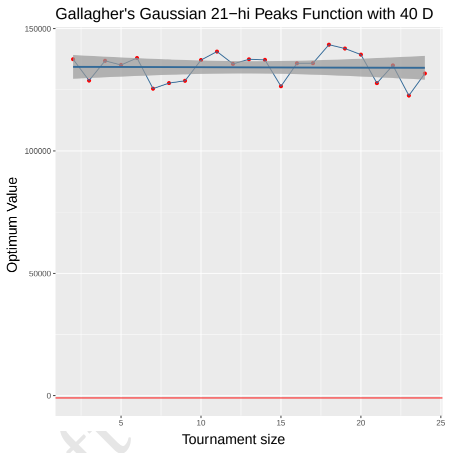
[1] Ram Bhushan Agrawal, K Deb, and RB Agrawal. 1995. Simulated binary crossover for continuous search space. *Complex systems* 9, 2 (1995), 115–148.
[2] Thomas Bäck, David B Fogel, and Zbigniew Michalewicz. 2000. *Evolutionary computation 1: Basic algorithms and operators.* Vol. 1. CRC press.
[3] James 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] Kalyanmoy Deb. 2000. An efficient constraint handling method for genetic algorithms. *Computer methods in applied mechanics and engineering* 186, 2 (2000), 311–338.
[8] David E Goldberg. 1991. Real-coded genetic algorithms, virtual alphabets, and blocking. *Complex systems* 5, 2 (1991), 139–167.
[9] 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.
[10] 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.
[11] Nikolaus Hansen, Anne Auger, Steffen Finck, and Raymond Ros. 2010. *Real-parameter black-box optimization benchmarking 2010: Experimental setup.* Ph.D. Dissertation. INRIA.
[12] 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.
[13] 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.
[14] 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.



(a) 10 dimensions.

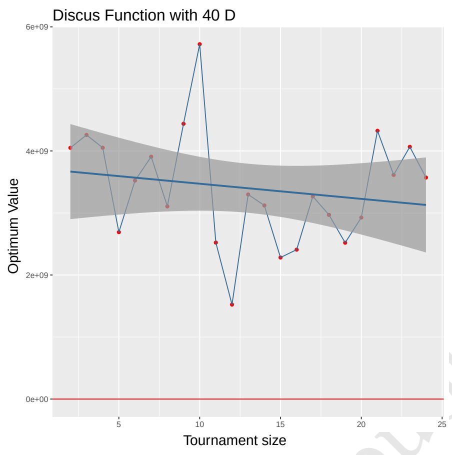


(b) 20 dimensions.

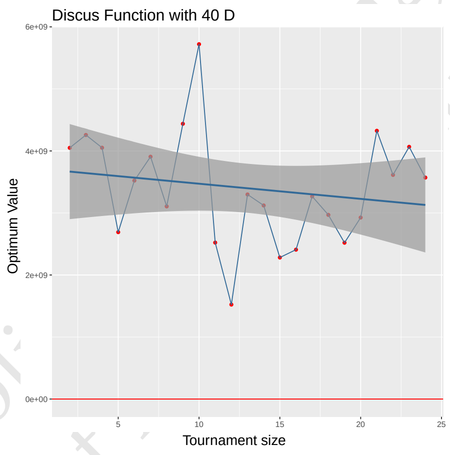


(c) 40 dimensions.

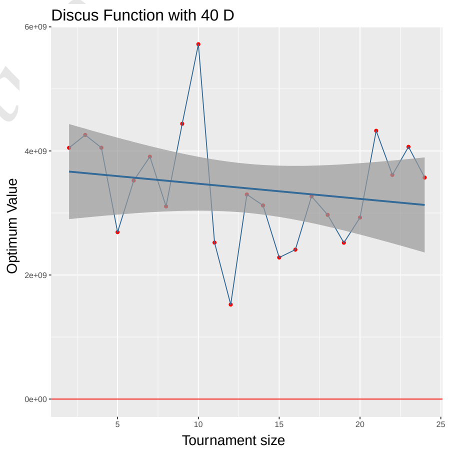
Figure 7: Average performance on different tournament size for the Gallagher's Gaussian 21-hi Peaks Function, when using the SBX crossover - Gen. Scheme B.



(a) 10 dimensions.



(b) 20 dimensions.



(c) 40 dimensions.

Figure 8: Average performance on different tournament size for the Discus Function, when using the SBX crossover - Gen. Scheme B.

[15] P Kaelo and MM Ali. 2007. Integrated crossover rules in real coded genetic algorithms. *European Journal of Operational Research* 176, 1 (2007), 60–76.
[16] 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.
[17] 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.
[18] 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.
[19] 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.
[20] Babatunde A Sawyerr, Aderemi O Adewumi, and M Montaz Ali. 2015. Benchmarking RCGAu on the Noiseless BBOB Testbed. *The Scientific World Journal* 2015 (2015).
[21] 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.
[22] 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.