Experimental Analysis of the Tournament Size on Genetic Algorithms

Anonymous Author(s)

ABSTRACT

We perform an experimental study about the effect of the tournament size parameter from the Tournament Selection operator. Tournament Selection is a classic operator for Genetic Algorithms and Genetic Programming. It is simple to implement and has only one control parameter, the *tournament size*. Even though it is commonly used, most practitioners still rely on rules of thumb when choosing the tournament size. For example, almost all works in the past 15 years use a value of 2 for the tournament size, with little reasoning behind that choice. To understand the role of the tournament size, we run a real-valued GA on 24 BBOB problems with 10, 20 and 40 dimensions. We also vary the crossover operator and the generational policy of the GA. For each combination of the above factors we observe how the quality of the final solution changes with the tournament size. Our findings do not support the indiscriminate use of tournament size 2, and recommend a more careful set up of this parameter.

CCS CONCEPTS

• **Computing methodologies** → **Genetic algorithms**;

KEYWORDS

Genetic Algorithm, Tournament Selection, Selection Schemes

ACM Reference Format:
Anonymous Author(s). 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, 8 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 [5]. 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 [16].

On the other hand, works in the past 17 years have commonly used very small values for k, such as 2 or 3 [4, 6, 14, 18, 19, 21, 22]. 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 [18] 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 THE TOURNAMENT SELECTION

A selection scheme is a process that aims to improve the average quality of the GA by giving solutions of higher quality a higher probability of composing the next generation [5]. The tournament selection, as any selection mechanism in GA, simply favors the selection of better individuals of a population to influence the next generation.

Tournament Selection is a selection scheme that has a unique set of characteristic that makes it a very popular choice [5]. It is simple to code and its complexity is O(n) [8] which makes it efficient for both nonparallel and parallel architectures. It can increase (decrease) the selection pressure by increasing (decreasing) the tournament size.

The tournament selection is also translation and scaling invariant [15], which means that the behavior of the selection scheme is not affect by the scale of the fitness values [2].

The tournament selection is able to do adjust the selection pressure and can do that easily, because it's only control parameter influences directly the selection pressure. That is an important feature, because any selection scheme should have is the capacity of being able to adjust its selection pressure given different domains. Even so often tournaments are held only between two individuals (binary tournament) but a generalization is possible to an arbitrary group size k called tournament size [5, 8]. We verified that using binary tournaments is still a popular choice.

2.1 The Tournament Selection Procedure

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 [13] 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.

The roulette-wheel and the tournament selection are two of these elitist selection schemes. However, it has been shown that roulette-wheel leads to premature convergence [3].

Nicolau, [18] used the tournament selection in his GA with the goal of solving 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.".

These two studies in addition to the tournament selection popularity and its power to be adjusted for different domains, give us motive to analyze the tournament selection behavior considering the BBOB functions.

2.3 Tournament Pressure

Muhlenbein et al., in [17], 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. [16], 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 [7], 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 would be represented in the population. He recognizes that selection may emphasize highly fit individuals leading to a small number of alternatives considered.

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. They understand that for some functions the selection pressure is not enough.

In 1999, Tsutsui et al. [23] 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 [6], implemented a binary GA and a real-coded GA, with tournament size 2. He aimed to devise a penalty function approach that does not 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". The concluded with declaring that some parametric study should be done on the important parameters to improve the performance.

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.

Sawyerr et al. [21], 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 [9], 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. [12] 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. [11] 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. [14] 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 [18] 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:

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

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 NOISE FREE BENCHMARK BBOB FUNCTIONS

The noise free benchmark BBOB functions [10] are a selected group of functions that we understand suits our objective to evaluate the performance of the tournament selection. That is because these collection of functions are intend to be more difficult than the problems that would be seen in practice, though these difficulties are believed to occur in continuous domain search. Also, all benchmark functions are scalable with the dimension.

These functions can be separated in two groups given their complexity. The first group is composed by the unimodal functions while the second group is composed by the multimodal functions. Here we consider that unimodal functions are functions with only one local optimum (equal to the global optimum) while multimodal functions are functions has many local optimum (with one of them being the global optimum).

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.

While the multimodal functions are: 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.

4 EXPERIMENT DESIGN

4.1 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 [10].

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 possible solution of the GA has size of N real value elements. The values of any solution are defined to the be into the interval [-5, 5], since the target functions are bounded by these interval . 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.

Fitness Function. The fitness function considered are the N dimensions noise free benchmark BBOB functions [10].

Generational Scheme. Two generation schemes were used, the first is named the (λ, λ), also called the comma version, and the second is named the $(\lambda + \lambda)$, also called the plus version. The (λ, λ) scheme has a λ population of parents and for every generation it creates the λ offspring to compose the next population. The $(\lambda + \lambda)$ scheme creates the λ offspring and the best λ individuals from combined population of offspring and parents are select to the next generation [20].

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

4.2 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 with 10, 20 and 40 dimensions, when combined with the Uniform crossover or with the SBX, with the generational scheme A. Then, we analyzed the same relation from before with the same functions and dimensions, but with the generational scheme B and only the SBX crossover. We consider that the impact the results may be different acknowledged the 2 groups of functions and therefore we analyze the results by grouping all functions together, or the unimodal functions or the multimodal functions.

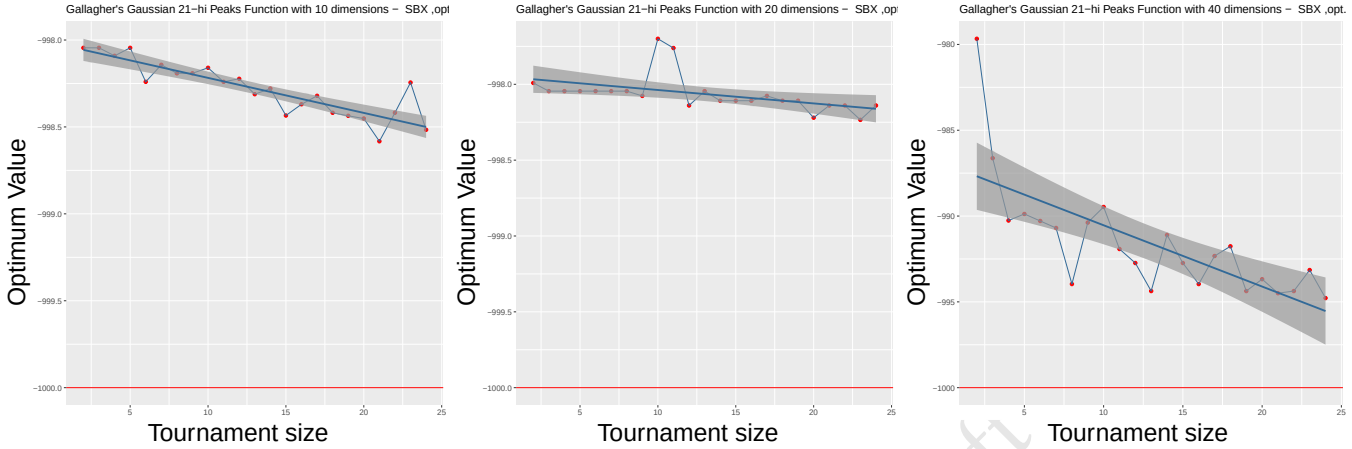
We used the Friedman Test to determine whether any of the for the tournament size show a significant difference in the average performance. In each of these tests, we set $\alpha = 0.05$. For this analysis, the dependent variable being the average performance, the blocking variable is the BBOB function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

5 RESULTS

5.1 (λ, λ) scheme

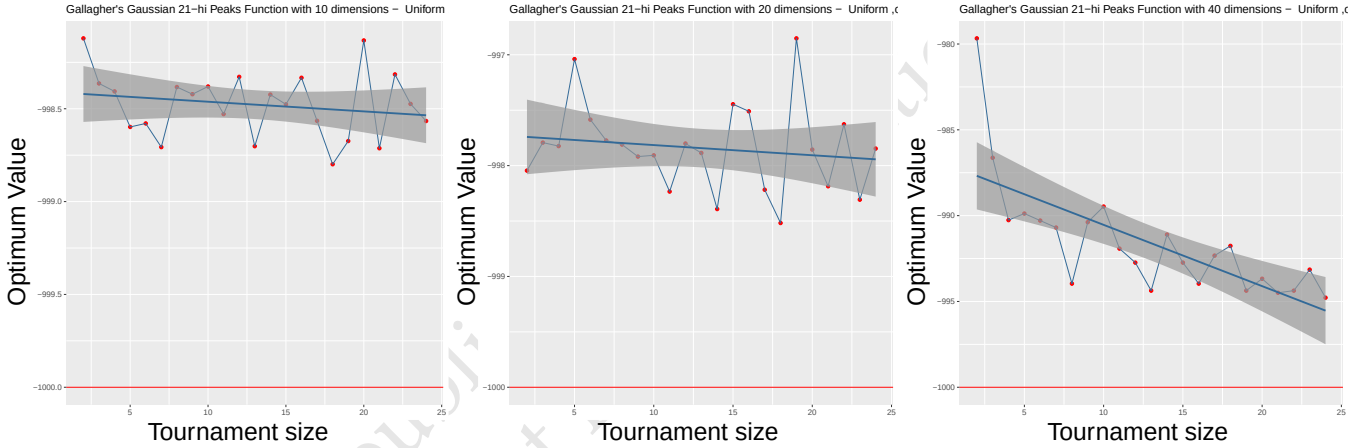
The Friedman Test showed significant difference in values among tournament size values across all functions, for any configuration, with the uniform crossover. This results hold for every scenario with 10, 20, and 40 dimensions, given: all functions (unimodal or multimodal); or only the multimodal functions. There is except when considering the results with the uniform crossover, for the unimodal group of functions. Also, for group of all function with 40 dimensions and the uniform crossover, there was found no significant difference in the results. The results can be better analyzed in the following Tables **??**, 4.

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



(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 - (λ, λ) scheme.



(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 - (λ, λ) scheme.

at each generation, the function target value with a given tournament size. All Figures represent the mean of 40 repetitions given a certain dimension.

5.2 Tournament Size by Functions Analysis

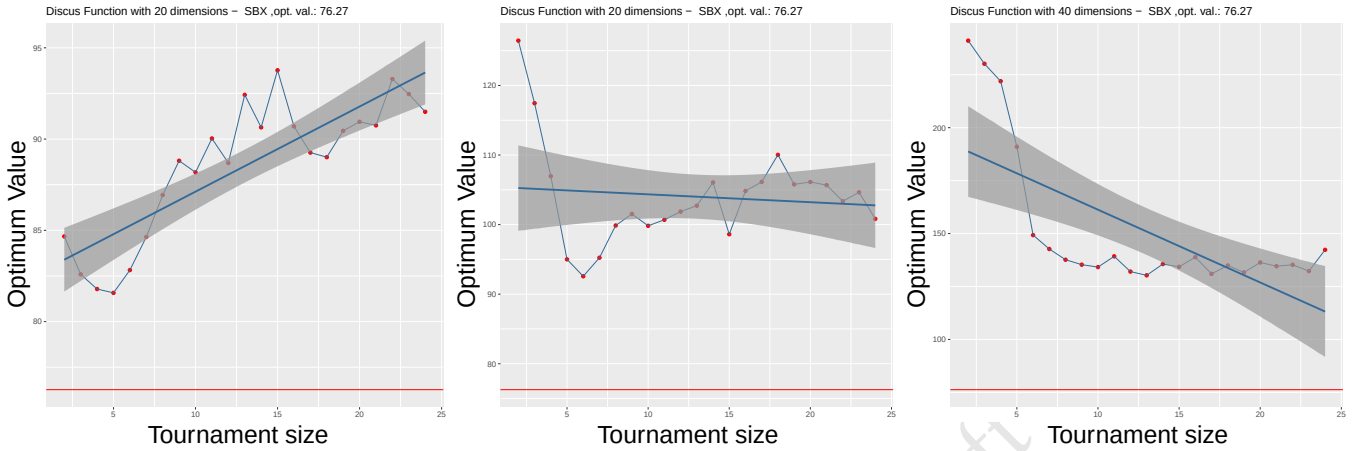
The Figures 1 and 2 exemplify that for the Gallagher's Gaussian 21-hi Peaks Function (with 10, 20 and 40 dimensions) changing the tournament size for higher values tend to increase the quality of the results, with greater variation when the SBX crossovers, Figure 1, in used when comparing with the uniform 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 40 repetitions is shown as bullets and the red line is shows the objective target value for the function.

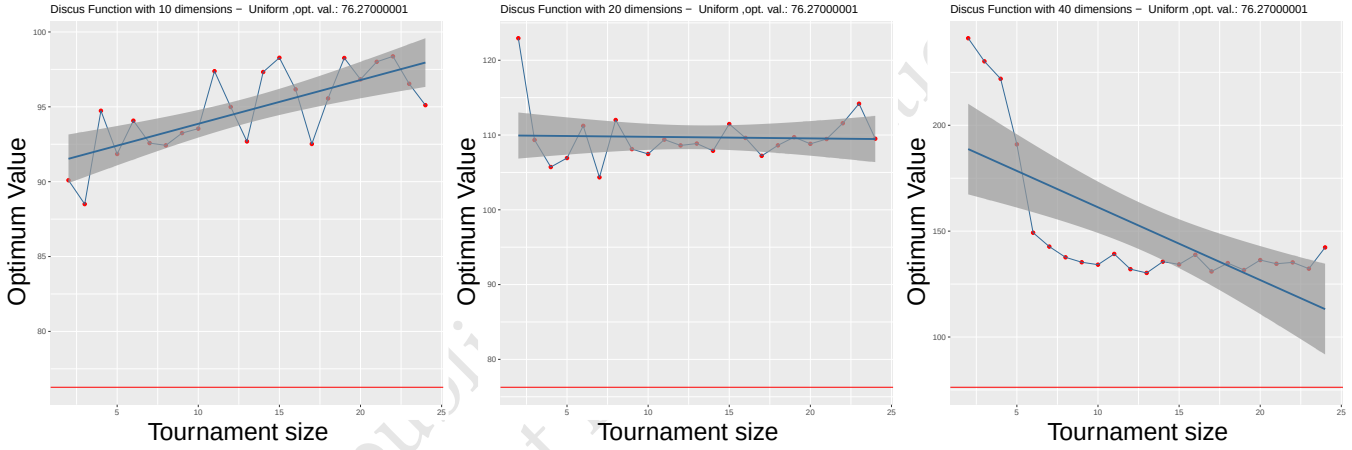
5.3 Convergence Analysis

The Figures 9 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.



(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 -(λ, λ) scheme.



(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 - (λ, λ) scheme.

5.4 $(\lambda + \lambda)$ scheme

The Friedman Test showed 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.

As for the generation (λ, λ), 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 30 repetitions given a certain dimension.

5.5 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 30 repetitions is shown as bullets and the red line is shows the objective target value for the function.

Function Group	Dimension size	Chi-squared	P-value
All	10	75.156	9.975e-08
Unimodal	10	26.637	0.2253
Multimodal	10	66.904	2.012e-06
All	20	43.771	0.003788
Unimodal	20	19.698	0.6019
Multimodal	20	57.525	5.152e-05
All	40	23.967	0.349
Unimodal	40	22.3	0.4421
Multimodal	40	37.091	0.02312

Table 3: Friedman Test results for Uniform Crossover - (λ, λ) scheme.

Function Group	Dimension size	Chi-squared	P-value
All	10	128.77	< 2.2e-16
Unimodal	10	49.997	0.000587
Multimodal	10	81.653	8.645e-09
All	20	182.09	< 2.2e-16
Unimodal	20	105.91	5.893e-13
Multimodal	20	88.885	5.302e-10
All	40	269.01	< 2.2e-16
Unimodal	40	144.89	< 2.2e-16
Multimodal	40	118.46	3.316e-15

Table 4: Friedman Test results for SBX Crossover - (λ, λ) scheme.

Function Group	Dimension size	Chi-squared	P-value
All	10	118.67	3.035e-15
Unimodal	10	54.011	0.0001639
Multimodal	10	75.724	8.073e-08
All	20	157.29	< 2.2e-16
Unimodal	20	73.078	2.15e-07
Multimodal	20	95.524	3.867e-11
All	40	190.59	< 2.2e-16
Unimodal	40	101.52	3.502e-12
Multimodal	40	87.318	9.762e-10

Table 5: Friedman Test results for SBX Crossover - $(\lambda + \lambda)$ scheme.

5.6 Convergence Analysis

The Figures 9 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.

6 CONCLUSION

We proposed an experimental analysis of the impact of the tournament size on the BBOB benchmark functions. We verified that

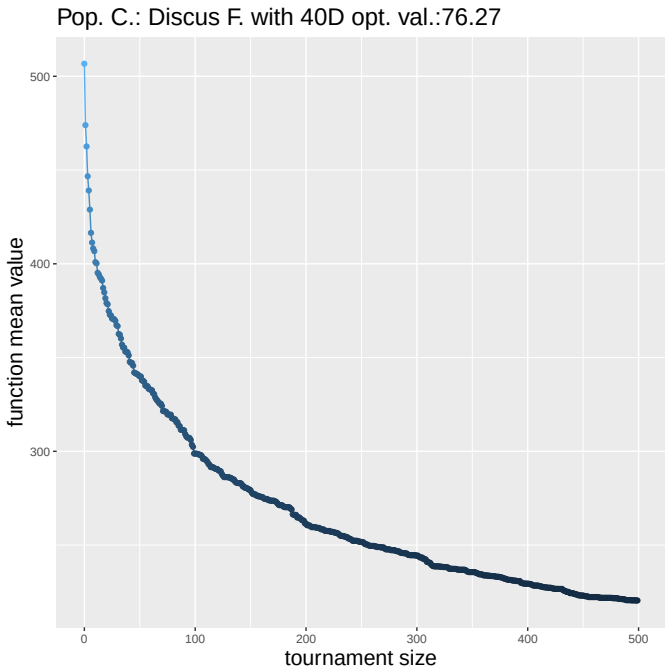


Figure 5: Population convergence for Discus Function.

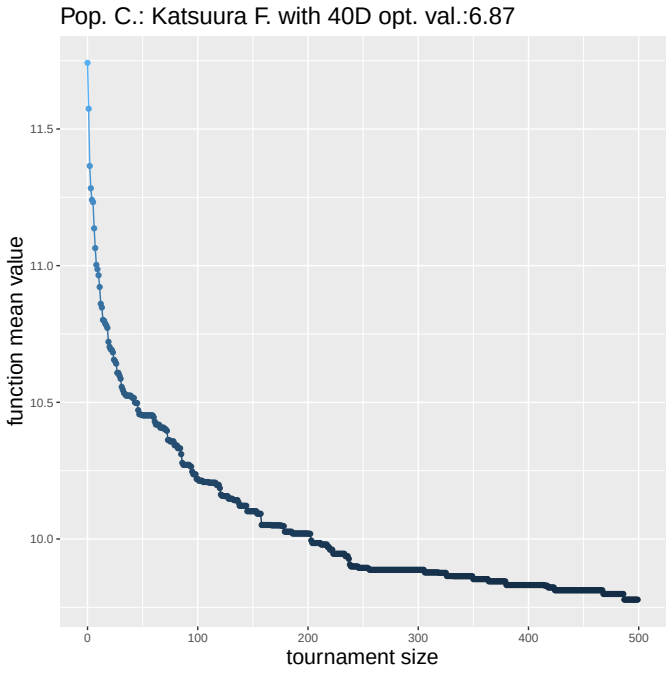
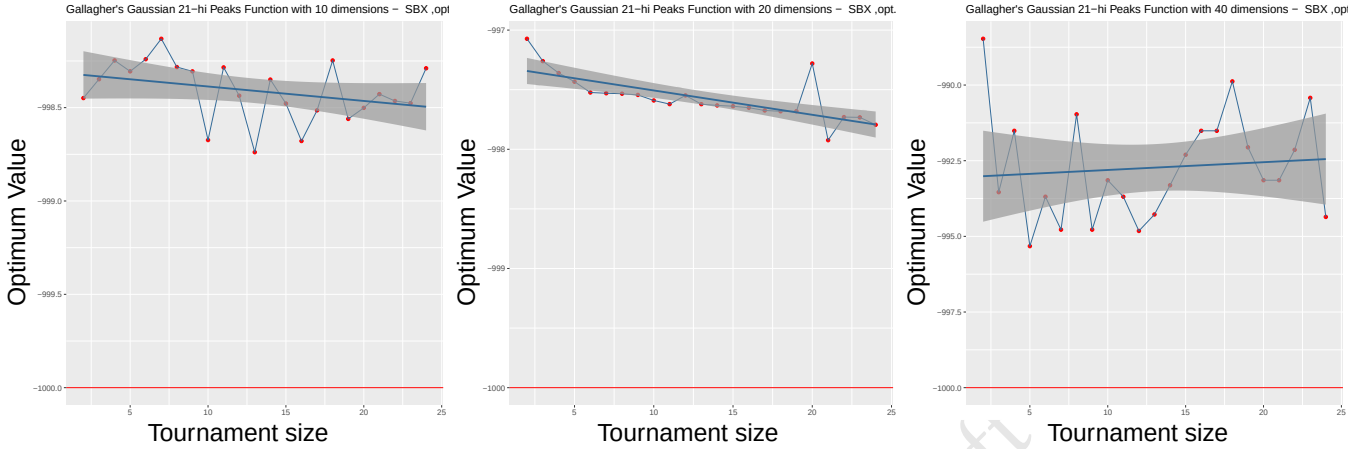


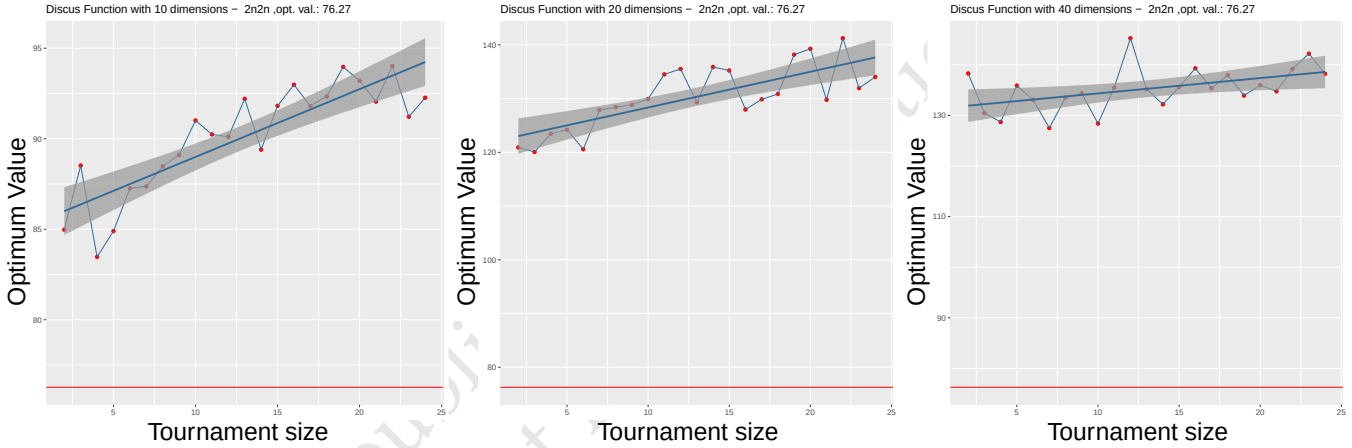
Figure 6: Population convergence for Katsuura Function.

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".



(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 - ($\lambda + \lambda$) scheme.



(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 - ($\lambda + \lambda$) scheme.

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 as future works that some sort of self-adaptative procedure should be tried, one that aims to minimize the impact of bad choice of values for the tournament size by adjusting this parameter on-line during the search process.

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 propose that more investigations need to be done, in order to better understand the role of the tournament size value related to the quality of final results. Also, we would like to explore how much the results were influenced by the particular sets of GA operators studied here.

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] Tobias Blickle and Lothar Thiele. 1995. A Mathematical Analysis of Tournament Selection.. In *ICGA*. 9–16.
[6] Kalyanmoy Deb. 2000. An efficient constraint handling method for genetic algorithms. *Computer methods in applied mechanics and engineering* 186, 2 (2000), 311–338.
[7] David E Goldberg. 1991. Real-coded genetic algorithms, virtual alphabets, and blocking. *Complex systems* 5, 2 (1991), 139–167.
[8] 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.

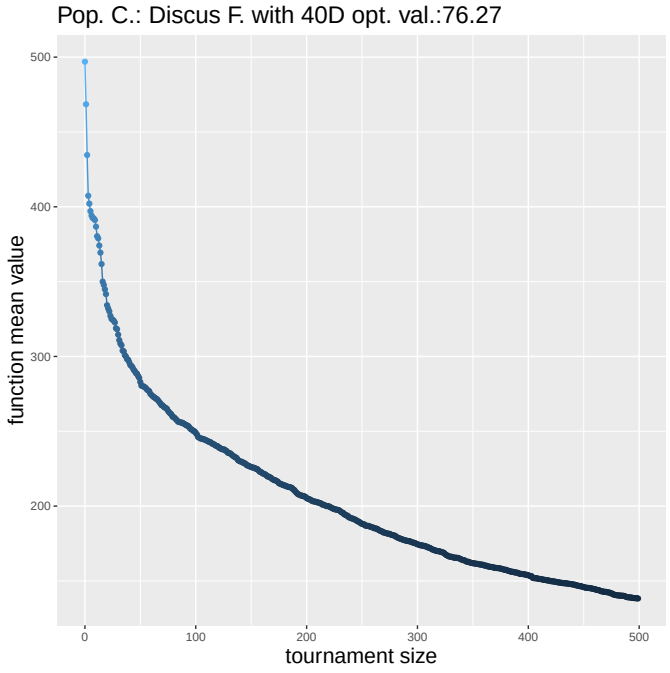


Figure 9: Population convergence for Discus Function.

[9] 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.

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

[11] 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.

[12] 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.

[13] 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.

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

[15] 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.

[16] 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.

[17] 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.

[18] 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.

[19] Asil Oztekin, Lina Al-Ebbini, Zulal Sevkli, and Dursun Delen. 2018. A decision analytic approach to predicting quality of life for lung transplant recipients: A hybrid genetic algorithms-based methodology. *European Journal of Operational Research* 266, 2 (2018), 639–651.

[20] Martin Pilát and Roman Neruda. 2017. Parallel evolutionary algorithm with interleaving generations. In *Proceedings of the Genetic and Evolutionary Computation Conference.* ACM, 865–872.

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

[22] 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.

[23] 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.