Investigating Parent Selection Mechanisms In Evolutionary Algorithms for Continuous Function Optimization

Group 65: John Gatopoulos, Elias Kassapis, Philipp Ollendorff, Konstantin Todorov

1 Introduction

In artificial intelligence, an evolutionary algorithm (EA) is a subset of evolutionary computation, a generic population-based metaheuristic optimization algorithm. Here candidate solutions are represented as individuals, who are as a whole represent a population, and though evolution are trying to maximize the fitness function, which determines the quality of the solutions

Inspired by biological evolution, EA use mechanisms like reproduction, where it selects the best-fit individuals (Parents) to breed new individuals (off springs) through crossover and mutation. This selection is crucial for the performance of the EA, since it distinguish the individuals that they will push towards the quality solutions of the algorithm, exploiting the search space.

In the current work our aim was to examine the effect of different parent selection schemes on algorithm performance in multidimensional optimization problems. We hypothesized that different parent selection mechanisms amount to different behaviour of the evolutionary search, by adjusting the balance of exploration versus exploitation. To test our hypothesis we designed and implemented four instances of the same EA, differing only in the parent selection mechanism, in maximizing three 10-dimensional continuous optimization problems, known as Bent Cigar ($\mathbf{f_{Cig}}$), Katsuura ($\mathbf{f_{Kat}}$), and the Schaffers F7 function ($\mathbf{f_{Sch}}$). Each test problem addresses different aspects of the exploitative and exploratory properties of the search. We then compared the performance of each algorithm on each of these problems to analyze the effect of each parent selection scheme. In particular, we looked at Roulette wheel selection with linear ranking (Roulette L.), Roulette wheel selection with exponential ranking (Roulette E.), Tournament selection and Uniform selection, used as a control.

2 Experimental Setup

Algorithm Description

Representation	Real-valued vectors
Recombination	One-Point Crossover
Recombination Size	40
Mutation	Uncorrelated mutation with n step sizes
Mutation Size	40
Parent Selection	Uniform, Tournament and different Rankings
Population Management	Steady-state Model
Generational Gap Size	No gap
Survival Selection	Elitism with N = 20
Population Size	100
Number of Offspring	80
Initialization	Uniform distribution on interval [-5, 5]
Termination Condition	Given evaluation limit

Table 1: Algorithm Summary.

Our setup is loosely based on Evolutionary Strategies. We used the steady-state model, keeping the 20 best individuals from the previous generation and all offspring are transferred into the next generation. We evaluate all individuals in each cycle and for the Bent Cigar function the termination criterion is 100 cycles of evaluations, for the Schaffers F7 function 1.000 and for the Katsuura function 10.000. We stop the mutation process for the last 5 percent of cycles to boost exploitation of good results. Algorithm specifications are shown in Table 1.

Quantification

To quantify the effectiveness of each algorithm we used the mean best fitness measure (**MBF**) over all runs. This is the average fitness value of the best individual in each run for each algorithm, in each problem over 100 runs.

To quantify the efficiency of each algorithm we used the average number of evaluations to a solution (**AES**) measure; that is, we evaluated the average of the cycle number in which the highest fitness first occurs over 100 runs.

We also produced progress curves for each algorithm on each problem by plotting the MBF against a time axis (units are per evolutionary cycle).

Test problems

(1) **Bent Cigar function:** $f_{\mathbf{Cig}} = x_1^2 + 10^6 \sum_{i=2}^{n} x_i^2$

Properties: A unimodal function with a smooth but narrow ridge. Used to assess the ability of the search to fine-tune good solutions.

(2) **Schaffers F7 function:** $f_{\mathbf{Sch}} = \left[\frac{1}{n-1}\sqrt{s_i} \cdot (sin(50.0 s_i^{\frac{1}{5}}) + 1)\right]^2$

Properties: A higly multimodal function with high variance in frequency and amplitude of the modulation. Used to assess the balance of exploration and exploitation of the search.

(3) **Katsuura Function:** $\mathbf{f_{Kat}} = \frac{10}{D^2} \prod_{i=1}^{D} \left(1 + i \sum^{3} 2_{j=1} \frac{|2^j\_i - [2^j\_i]|}{2^j}\right)^{10/D^{1.2}} - \frac{1}{D^2} + f_{\mathrm{pen}}(\mathbf{x})$

Properties: A highly repetitive and highly rugged function. Used to assess global search behaviour.

Statistics

In order to obtain statistical significance in our comparisons we used Kruskal-Wallis tests on the results, as all the data sets were non-parametric. The normality of the data distributions was checked using the Kolmogorov-Smirnov test.

All bar charts display mean, and the error bars represent standard deviation. Statistical significance is indicated on graphs using standard conventions: n.s.d. (non-significant): $p > 0.05, * : p \leq 0.05, ** : p \leq 0.01, *** : p \leq 0.001$. All experiments were repeated 100 times.

The analyses were performed using Prism 5 (GraphPad Software). The significance level was set at $p = 0.05$.

3 Experimental Results

We began by evaluating the MBF of each algorithm throughout each independent run on each test function to identify any difference in effectiveness between them. For all test problems; that is, $\mathbf{f_{Cig}}$, $\mathbf{f_{Sch}}$ and $\mathbf{f_{Kat}}$ we found a statistically significant difference between the different algorithms, as determined by a Kruskal-Wallis test ((H = 306.2, p < 0.0001), (H = 337.0, p < 0.0001) and (H = 8.281, p = 0.0405) respectively). To compare the difference between each algorithm in each test problem we used a Dunn's multiple comparison post-hoc test. For $\mathbf{f_{Cig}}$ we found that that there was a significant difference in all algorithm comparisons except in the Roulette L. vs Tournament comparison. For $\mathbf{f_{Sch}}$ we found that there was a significant difference between all algorithms. For $\mathbf{f_{Kat}}$ we found no significant difference between any algorithms.

With respect to the MBF, the Roulette E. selection algorithm outperformed all other algorithms in all test problems, although it was only significant in $\mathbf{f_{Cig}}$ and $\mathbf{f_{Sch}}$. The Roulette L. and Tournament selection algorithms were second best, outperforming the Uniform selection algorithm in all test problems, but it was also only statistically significant in $\mathbf{f_{Cig}}$ and $\mathbf{f_{Sch}}$. The Roulette L. selection method performed significantly better than the Tournament selection algorithm in $\mathbf{f_{Sch}}$ but not in the other test problems. Results are displayed in Table 2, and graphically in Figure 1.

Alg.	$\mathbf{f_{Cig}}$	$\mathbf{f_{Sch}}$	$\mathbf{f_{Kat}}$
Uniform Selection	0.005743 ± 0.03411	0.007566 ± 0.01424	0.04792 ± 0.1207
Roulette Wheel L.	7.915 ± 0.7509	6.886 ± 1.437	0.1686 ± 0.2919
Roulette Wheel E.	9.629 ± 0.7064	8.907 ± 2.513	0.2935 ± 0.3694
Tournament Selection	6.974 ± 2.201	2.400 ± 1.504	0.1803 ± 0.3285

Table 2: Computational results. MBF $\pm$ Standard deviation (to 4 s.f.)

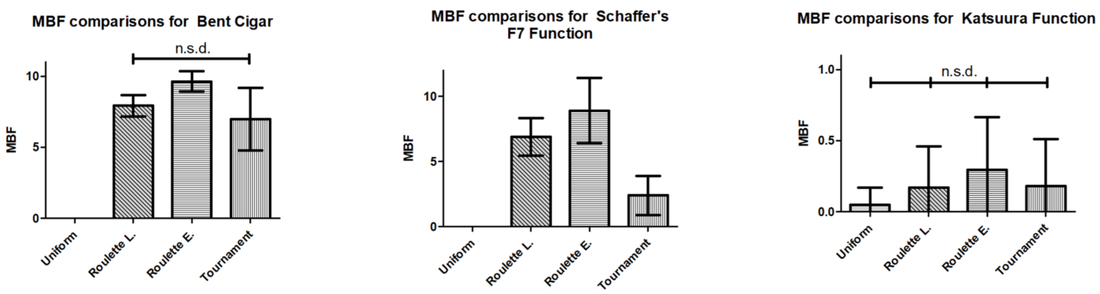

Figure 1: Algorithm effectiveness vs. Test problem

Only the differences that are not significant (n.s.d.) are displayed as all other differences are significant at $p < 0.0001$.

We also evaluated the AES of each algorithm throughout each independent run on each test function to identify any difference in efficiency between them. Again, for $\mathbf{f_{Cig}}$, $\mathbf{f_{Sch}}$ and $\mathbf{f_{Kat}}$ we found a statistically significant difference between the different algorithm, as determined by a Kruskal-Wallis test ((H = 280.7, p < 0.0001), (H = 257.9, p < 0.0001) and (H = 15.43, p = 0.0015) respectively). To compare the difference between each algorithms in each test problem we also used a Dunn's multiple comparison post-hoc test. For $\mathbf{f_{Cig}}$ we found that that there was a significant difference in all algorithm comparisons except in the Roulette E. vs Tournament comparison. For $\mathbf{f_{Sch}}$ we found that there was a significant difference in all algorithm comparisons except Roulette L. and Roulette E. For $\mathbf{f_{Kat}}$ we only found a significant difference between Uniform and Roulette L., and Uniform and Roulette E.

With respect to the AES, the Roulette E. selection algorithm was better in $\mathbf{f_{Cig}}$ and $\mathbf{f_{Sch}}$, although it was not statistically significant. Again the Uniform selection algorithm had the lowest performance compared in all test problems compared to the other algorithms but in $\mathbf{f_{Kat}}$ the difference with the Tournament selection algorithm was not significant. Results are displayed in Table 3, and graphically in Figure 2.

Alg.	$\mathbf{f_{Cig}}$	$\mathbf{f_{Sch}}$	$\mathbf{f_{Kat}}$
Uniform Selection	20 ± 27	154 ± 209	913 ± 711
Roulette Wheel L.	74 ± 18	815 ± 273	6240 ± 4305
Roulette Wheel E.	96 ± 9	953 ± 53	5258 ± 4117
Tournament Selection	92 ± 16	539 ± 310	4928 ± 4315

Table 3: Computational results. AES $\pm$ Standard deviation (to the nearest cycle)

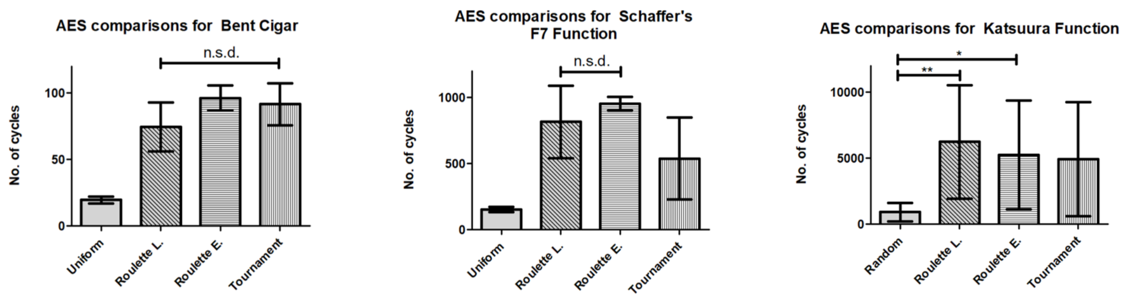

Figure 2: Algorithm efficiency vs. Test problem

For the Bent Cigar and Schaffer's comparison only the differences that are not significant (n.s.d.) are displayed as all other differences are significant at $p < 0.0001$. For the Katsura comparison only the significant differences are displayed as all other differences are not significant

Finally we looked further into the effectiveness and efficiency of each algorithm by constructing progress curves for each test problem. The results are displayed graphically in Figure 3.

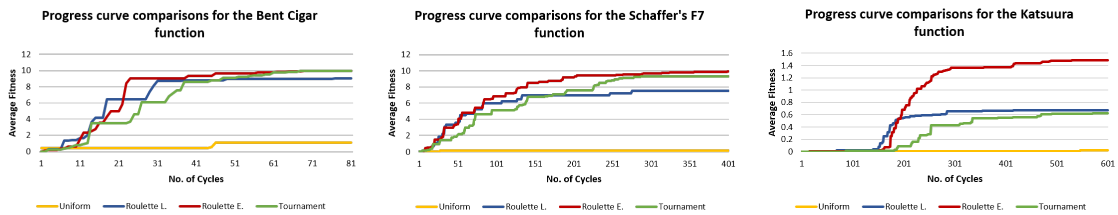

Figure 3: Algorithm Progress Curve vs. Test problem

4 Analysis and Discussion

In the present study we investigated the effect of different parent selection mechanisms in EAs for continuous function optimization tasks by quantifying the performance of four instances of the same basic algorithm, that only differed in parent selection scheme. To examine the impact of each parent selection mechanism on the behaviour of the search process in such tasks, we used three different test problems that are each designed to address and assess different properties of the search process. We found that all algorithms highly fluctuated in performance, as expected due to the stochastic nature of the search process. However, Roulette E. selection was clearly better than the other parent selection schemes in terms of finding the best solution, and was faster in doing so in both the unimodal and multimodal test problems. Roulette L. selection was also better than Tournament selection in finding the best solution to both unimodal and multimodal problems, and it was faster than Tournament selection to reach its best solution to multimodal, but not unimodal problems. Uniform selection had as expected the worst performance in all test problems. Furthermore, all algorithms performed poorly in the Katsuura function

Our results show that Roulette E. selection is the best in facilitating exploitation; that is, focusing the evolutionary search in high fitness regions and fine-tuning solutions, as evident from having the best performance in both the Bent Cigar function and the Schaffer's F7 function. Furthermore, having the best performance in the Schaffer's F7 function is an indication that it is also the best of the considered selection schemes in exploring the search space for identification of high fitness regions, due to the multimodal nature of the function. The low performance of all the algorithms in the Katsuura function hints that all parent selection mechanisms described undergo genetic drifting before exploring the complex fitness landscape of this function enough, restricting the search within comparatively low fitness regions; that is, local optima [1].

Although the algorithms had different AES values in all test problems, the differences were not as pronounced as the differences in MBF measures. In addition, the MBF was similar for all algorithms in the Katsuura function, suggesting that interchanging selection methods did not drastically affect rate of genetic drifting. This may be the case because we have only looked at panmictic populations; that is we used a single population of individuals in each generation and applied operators on them as a whole, which is one of the limitations of our study. A future study could include cellularization or island models to encourage variation, reducing the rate of genetic drifting, and therefore improve the sampling of the sampling space [5]. Another reason for this may be that we evaluated the fitness of each individual at every generation, therefore the number of fitness evaluations did not accurately reflect the number of generated points in the search space. Thus AES may not accurately represent the rate of genetic drifting; although judging from the Katsuura function performances this may be insignificant.

Given our results, we conclude that our hypothesis that different parent selection mechanisms amount to different behaviour of the evolutionary search, by adjusting the balance of exploration versus exploitation is correct. Roulette wheel selection methods are better than Tournament selection at fine-tuning good solutions (exploitation). We suggest that this is due to the higher proportion of higher quality individuals in the mating pool, as we see that this is accentuated when Roulette wheel selection is used with exponential ranking rather than linear ranking. We based our conclusion on the premise that the mating pool under Tournament selection consists of lower quality individuals on average relative to the Roulette wheel selection methods, because we used Tournament selection with replacement, and sampling occurs stochastically in contrast to Roulette wheel selection, where sampling is proportional to fitness. In terms of exploitation, our results were less clear, although Roulette wheel selection does seem to outperform Tournament in this domain as well.

A limitation of the study is that we compared the selection mechanisms with a specific recombination and mutation pair, and specific parameter values. They may favour one of the selection schemes, introducing a bias in our results. A further issue of the study is that our performance measures may not be as transparent as other measures used in the literature in dissecting the different properties of the algorithmic search in each instance of the algorithm used (the different selection schemes). For example the MBF only addresses one aspect of the selection method. Also, as pointed out earlier, we didn't cache fitness values for individuals that we have computed earlier in the evolutionary search. Thereby multiple evaluations of the same genotype are possible.The impact of each scheme on the search process dynamics could be described in terms of its interaction with the fitness distribution. This framework allows the derivation of several insights in the properties of each selection method, including the selection intensity, the selection variance and the loss of diversity [2].

To conclude, there are many different parent selection schemes that exert distinct selection pressures to the population, which is a consequence of the fact that each selection scheme draws a mating pool with different formulation from the population, as each individual is selected according to different criteria; thus pushing the search process in different directions. This results adjusts the balance of exploration and exploitation, and therefore lead to different paths of quality improvement during the evolutionary search.

5 Bibliography

[1] Eiben, A.E., Smith, J.E., Introduction to Evolutionary Computing. Springer, 2015, 2nd edition

[2] Tobias Blickle, Lothar Thiele, A comparison of selection schemes used in evolutionary algorithms

[3] Th. Bck: Evolutionary Algorithms in Theory and Practice, Oxford, University Press, NY, 1996.

[4] Nikolaus Hansen, Steffen Finck, Raymond Ros and Anne Auger,"Real-Parameter Black-Box Optimization Benchmarking 2010: Noiseless Functions Definitions" INRIA Research Report RR-6829, March 24, 2012.

[5] E. Alba, B. Dorronsoro, "Cellular Genetic Algorithms", Springer, 2008