Genetic Algorithms for continuous function optimisation

BITNARAE KIM, University of Exeter, UK

Abstract- Genetic Algorithms has three main operators, selection, mutation and crossover that help obtain the optimal solution for optimisation problems. Although each operator has its importance, the selection operator is essential for the optimal performance of the algorithm. In this report, three selection methods are used to see which produce the best results for the GA, and six benchmark functions have been used for comparison. The overall results of the experiment showed that Tournament Selection showed the best performance for five out of the six benchamrk functions.

KEYWORDS : genetic algorithms, continuous functions optimisation, tournament selection, roulette wheel selection, linear rank selection

1 INTRODUCTION

Genetic Algorithm (GA) is a population-based stochastic algorithm, and is the most efficient method when trying to solve problems that have limited information. Genetic Algorithms can effectively handle both unconstrained and constrained optimisation problems [Kumar et al. 2010]. A GA has three main operators: selection, crossover, and mutation. Through this process, the algorithm maintains the best solutions in each generation and uses them to improve other solutions [Haq et al. 2019]. As a result, the entire population becomes better generation by generation. Every solution corresponds to a chromosome with each parameter representing a gene. The GA uses an objective function to evaluate the fitness of each individual in the population [Mirjalili 2019]. To improve poor solutions, the best solutions are chosen probabilistically with a selection scheme according to their fitness values. Then crossover and mutation is performed on the selected solutions to produce offspring. The next generation is formed by an alternative mechanism between parents and their offspring, and this process of evolution is repeated until the end condition is satisfied [Jebari and Madiafi 2013]. The GA has various parameters that need to be set and decision made on about the following:

- the population size
- how the parents will be selected for creating offspring
- mutation and crossover of the individuals
- the stopping criteria

The ultimate goal is to choose the right parameters for the algorithm [Goldberg 1988]. There has been ongoing research to find techniques that improve the quality of the solution. The majority of past research has focused on the population size and the genetic operators exploration [Eiben and Smit 2011]. Out of the various parameters, this report focuses on the selection operator. The right balance between exploitation and exploration is important for the GA, and this can be adjusted by the selection operator [Michalewicz 1992]. The selection operator exploits the best characteristics of candidate solutions to improve these solutions throughout the generations, which leads the GA to converge to a sufficient solution to the optimisation problem. Therefore, the selection operator is the most crucial part of the genetic algorithms that influences the performance of the GA [Bäck and Hoffmeister 1991].

The rest of this report is organised as follows: Section 2 includes the literature review, Section 3 discusses the search algorithm and optimisation problems, Section 4 discusses the experiments and results, and Section 5 includes the conclusion.

Author's address: Bitnarae Kim, University of Exeter, Exeter, UK, bk339@exeter.ac.uk.

2 LITERATURE REVIEW

As previously discussed in Section 1, selection of parent population is one of the main and crucial steps in the GA. There have been several past research that have studied the performance of GA using different selection strategies.

Jadaan et al. [2008] compared the proportional roulette wheel and rank-based roulette wheel selection method using several mathematical fitness functions. They found that rank-based selection outperformed proportional roulette wheel selection in the number of generations to come out with the optimal solution. Furthermore, they observed that rank-based selection was steadier, faster and more robust towards the optimum solutions than the proportional roulette wheel selection.

Mashohor et al. [2005] compared three selection approaches, deterministic selection, Tournament Selection (TS) and Roulette Wheel Selection (RWS) to evaluate the performance of Genetic Algorithm based printed circuit board inspection system. The paper found that deterministic selection out-performed the other two methods, showing maximum fitness.

Zhong et al. [2005] did a comparative study of the selection techniques, where they considered RWS and TS. They chose seven test functions and the results obtained showed that genetic algorithm outperformed with TS while RWS produced average performance.

Julstrom [1999] investigated two rank-based assignments of probabilities, linear normalization, exponential normalization, and two tournament selection schemes (2-tournament selection without replacement and tournament selection with replacement). Computational time was considered as a parameter to examine the efficiency of the selection schemes. From the analysis, tournament selection was preferred over rank-based selection because repeated tournament selection was quicker than sorting the population compared to rank-based selection.

Chudasama et al. [2011] and Razali et al. [2011] compare different selection techniques for solving the Travelling Salesman Problem. Razali et al. [2011] used TS, RWS, and Rank-based Roulette Wheel Selection. They concluded from their research that tournament selection acheived the best solution quality compared to both RWS and rank-based roulette wheel selection, and with low computing times. On the other hand, Chudasama et al. [2011] used TS, RWS, and Elitism as their selection methods. Their research showed that while in the early stages, all three selection methods worked similarly, in the end, elitism generated better results compared to TS and RWS.

3 PROBLEM AND IMPLEMENTATION

There are several selection methods that are used in GAs, and this report focuses on Tournament Selection (TS), Roulette Wheel Selection/Fitness Proportionate Selection (RWS), and Linear Rank Selection (LRS).

TS is the most popular selection method in GA because of its efficiency and simple implementation. In tournament selection, 'n' individuals are randomly selected from the population, and the selected individuals compete against each other. The individual with the highest fitness wins and is included as one of the next generation population. TS also allows all individuals to be selected which preserves diversity [Blickle 2000]. The pseudocode shown in Algorithm 1 shows how TS works.

In RWS, individuals are selected with a probability that is directly proportional to their fitness values. RWS calculates the sum of the fitness values of each individual in the population. Then it calculates the fitness value of each individual and their probability of selection. The roulette wheel is then split according to the calculated probabilities. Then the wheel is spun 'n' times and when the wheel stops, the section where the pointer is pointing at corresponds to the selected individual [Lipowski and Lipowska 2012]. Algorithm 2 shows the pseudocode of RWS.

In LRS, the individuals are sorted according to their fitness value and then assigned ranks. The best individual is ranked as 'N' and the worst is ranked as '1'. The selection probability is then assigned linearly to the individuals according to their given ranks. Each individual is given a different rank even if they have the same probabilities [Shukla et al. 2015]. The working of the algorithm is shown in Algorithm 3.

Algorithm 1: Tournament Selection

```
Select k individuals from population and perform a
  tournament amongst them
Select the best individual from the k individuals
Repeat process 1 and 2 until the desired amount of
  population
```

Algorithm 2: Roulette Wheel Selection

```
P = size of the population
while population size < P do
    Generate P random number r
    Calculate cumulative fitness, total fitness, and sum of
      proportional fitness
    Spin wheel P times
    if sum of proportional fitness < r then
        Select first chromosome
        Otherwise select jth chromosome
    else
    end
end
return chromosomes with fitness value proportional to size
  of wheel section
```

The three main operators of the genetic algorithm are selection, crossover, and mutation. These operators help obtain the optimal solution for constrained optimisation problems. The process of the GA can be described by the pseudocode shown in Algorithm 4. The three selection methods mentioned above was used on the GA with single-point crossover, bit-flip mutation, and a population size of 100. To evaluate the performance of these three selection methods, several benchmarks have been used. The chosen benchmarks are Ackley, Bohachevsky, Schaffer, Egg-holder, Drop-wave, and Schwefel. The benchmark functions and their optimum value and properties can be seen in Table 1.

Algorithm 3: Linear Rank Selection

```
Input  : population P and reproduction rate of the worst
         individual
Output : The population after selection
Sort population according to fitness from worst individual to
  best
for each individual 1 <= i <= n do
    Linearly assign selection probability to individuals
      according to their rank
    Generate a random number r
    for 1 <= j <= n do
        if P(j) <= r then
            Select the jth individual
        else
    end
end
```

Algorithm 4: Genetic Algorithm

```
Initial population of individuals: P(0)
Evaluate the fitness of all individuals of P(0)
Maximum number of generations: t_max
while end condition not satisfied and t<t_max do
    t ← t + 1
    Select parents for offspring production
    Apply reproduction and mutation operators
    Create a new population of survivors: P(t)
    Evaluate P(t)
end
return the best individual of P(t)
```

4 EXPERIMENTS

The objective of the experiment was to make a comparison between the three selection schemes, Tournament selection, Roulette selection and Linear Rank selection, with four other algorithms by using benchmark functions. The four algorithms chosen for comparison were Evolutionary Strategy (ES), Stochastic Hill Climber (SHC), Random Search (RS), and Differential Evolution algorithm (DE). Differential evolution is a heuristic approach for global optimisation of

Benchmark	Optimum value	Properties
Ackley	0	Multimodal, non-convex
Bohachevsky	0	Multimodal, nonseparable
Schaffer	0	Unimodal, nonseparable
Egg-holder	-959.6407	Nonconvex, multimodal
Drop-wave	-1	Multimodal, nonseparable
Schwefel	0	Multimodal, nonseparable

Table 1. Benchmark functions

nonlinear and non-differentiable continuous functions. Evolutionary Strategy is an optimisation technique that belongs to the class of Evolutionary Algorithms, and works similarly to the GA. Fixed parameters were used to see the overall efficiency of the selection schemes used for the optimisation problems:

- population size: 100
- single-point crossover
- bitflip mutation

To make a comparison, each algorithm was run 30 times and statistical results of the mean value and standard deviation were taken as final results. Six benchmark functions were used as previously mentioned in Table 1.

4.1 Results and Analysis

The algorithms were executed thirty times to get final results of the mean value and standard deviation. All experiments were terminated when the number of generations achieved the maximum numbers of generation. The statistical results of the algorithms can be shown in Table 2 and Table 3.
The results show that on the Ackley benchmark, tournament selection shows the lowest mean and standard deviation compared to other two selection methods, and it obtained the theoretical optimum value on the Schaffer benchmark. Results on the Drop-wave benchamrk was similar for all three selection methods. For the Bohachevsky function, roulette wheel selection performed the worst while tournament selection was best-performing selection technique. Stochastic hill climbing, Evolution Strategy, the Differential Evolution algorithm produced the same result in every run, but for the sake of comparison, they were all run 30 times and the statistical results were calculated. The Differential Evolution algorithm obtained optimum value for the Ackley, Bohachevsky, Schaffer, and Drop-wave function. Random search performed poorly on the majority of the benchmark functions. From the overall results, apart from the Drop-wave function which had similar results for all selection methods, Tournament Selection showed to be the best performing selection technique. Linear rank selection performed better on the Ackely, Bohachevsky, and Schaffer function compared to the Roulette Wheel selection.

5 CONCLUSION

This report focused on the relative performance of three commonly used selection methods (Tournament Selection, Roulette Wheel selection, and Linear Rank selection) to obtain the optimal solution for six benchmark problems. Additional four algorithms of Evolutionary Strategy, Stochastic Hill Climber, Random Search, and Differential Evolution algorithm were used for comparison. The six benchmark functions that were used cover various characteristics that include unimodal, multimodal, convex, non-convex separable, and non-separable. The statistical results of the experiment show that Tournament selection produced the best results for five out of six benchmark functions.

	TS	RWS	LR	ES	SHC	RS	DE
Ackely	0.000324	0.5762873	0.079060167	0.001147	5.381939	7.7807457	0
Bohachevsky	4.3333333E-7	0.21335113	0.0178168	0.000002	2.287498	16.841844	0
Schaffer	0	7.6333333E-6	1.2E-6	0	0.004358	0.337853	0
Egg-holder	-49.032479	-48.9686	-46.783297	-49.032523	-49.025972	-40.654407	-49.03306
Schwefel	830.09834	830.12475	830.42864	830.098347	830.098593	831.95921	830.09831
Drop-wave	-0.95688063	-0.968753	-0.98476307	-0.996160	-0.477784	-0.05774	0

Table 2. Statistical results: Mean

	TS	RWS	LR	ES	SHC	RS	DE
Ackely	0.00023879698	0.41504691	0.11786137	4.3368087E-19	8.8817842E-16	2.873016	0
Bohachevsky	4.9553562E-7	0.16787393	0.045331937	8.4703295E-22	0	10.980774	0
Schaffer	0	6.9353362E-6	2.6255158E-6	0	0	0.23967103	0
Egg-holder	2.1316282E-14	0.035393506	1.3192533	1.4210855E-14	1.4213521E-14	14.720331	1.4210855E-14
Schwefele	1.1368684E-13	0.020888155	0.27467276	5.6843419E-13	2.2737368E-13	5.6326335	2.2737368E-13
Drop-wave	0.024828971	0.022286222	0.023528375	2.220446E-16	1.6653345E-16	0.065908719	0

Table 3. Statistical results: Standard deviation

REFERENCES

Omar al jadaan, L. Rajamani, and C. Rao. 2008. Improved selection operator for GA. *Journal of Theoretical and Applied Information Technology* 4 (01 2008), 269–277.

Tobias Blickle. 2000. Tournament selection. *Evolutionary computation* 1 (2000), 181–186.

Thomas Bäck and Frank Hoffmeister. 1991. Extended Selection Mechanisms in Genetic Algorithms. Morgan Kaufmann, 92–99.

Chetan Chudasama, SM Shah, and Mahesh Panchal. 2011. Comparison of parents selection methods of genetic algorithm for TSP. In *International Conference on Computer Communication and Networks CSI-COMNET-2011, Proceedings*. Citeseer, 85–87.

A. Eiben and S.K. Smit. 2011. Parameter tuning for configuring and analyzing evolutionary algorithms. *Swarm and Evolutionary Computation* 1 (03 2011), 19–31. https://doi.org/10.1016/j.swevo.2011.02.001

David E. Goldberg. 1988. Genetic Algorithms in Search Optimization and Machine Learning.

Ehtasham-ul Haq, Ishfaq Ahmad, Abid Hussain, and Ibrahim Almanjahie. 2019. A Novel Selection Approach for Genetic Algorithms for Global Optimization of Multimodal Continuous Functions. *Computational Intelligence and Neuroscience* 2019 (12 2019), 1–14. https://doi.org/10.1155/2019/8640218

Khalid Jebari and Mohammed Madiafi. 2013. Selection methods for genetic algorithms. *International Journal of Emerging Sciences* 3, 4 (2013), 333–344.

B.A. Julstrom. 1999. It's all the same to me: revisiting rank-based probabilities and tournaments. In *Proceedings of the 1999 Congress on Evolutionary Computation-CEC99 (Cat. No. 99TH8406)*, Vol. 2. 1501–1505 Vol. 2. https://doi.org/10.1109/CEC.1999.782661

Manoj Kumar, Mohammad Husain, Naveen Upreti, and Deepti Gupta. 2010. Genetic algorithm: Review and application. *Available at SSRN 3529843* (2010).

Adam Lipowski and Dorota Lipowska. 2012. Roulette-wheel selection via stochastic acceptance. *Physica A: Statistical Mechanics and its Applications* 391, 6 (2012), 2193–2196. https://doi.org/10.1016/j.physa.2011.12.004

Syamsiah Mashohor, J.R. Evans, and Tughrul Arslan. 2005. Elitist selection schemes for genetic algorithm based printed circuit board inspection system, Vol. 2. 974 – 978 Vol. 2. https://doi.org/10.1109/CEC.2005.1554796

Zbigniew Michalewicz. 1992. Genetic Algorithms + Data Structures = Evolution Programs. In *Artificial Intelligence*.

Seyedali Mirjalili. 2019. *Genetic Algorithm*. Springer International Publishing, Cham, 43–55. https://doi.org/10.1007/978-3-319-93025-1_4

Noraini Mohd Razali, John Geraghty, et al. 2011. Genetic algorithm performance with different selection strategies in solving TSP. In *Proceedings of the world congress on engineering*, Vol. 2. International Association of Engineers Hong Kong, China, 1–6.

Anupriya Shukla, Hari Mohan Pandey, and Deepti Mehrotra. 2015. Comparative review of selection techniques in genetic algorithm. In *2015 International Conference on Futuristic Trends on Computational Analysis and Knowledge Management (ABLAZE)*. 515–519. https://doi.org/10.1109/ABLAZE.2015.7154916

Jinghui Zhong, Xiaomin Hu, Jun Zhang, and Min Gu. 2005. Comparison of Performance between Different Selection Strategies on Simple Genetic Algorithms., Vol. 2. 1115–1121. https://doi.org/10.1109/CIMCA.2005.1631619