Analysis Effect of Tournament Selection on Genetic Algorithm Performance in Traveling Salesman Problem (TSP)

S Prayudani¹, A Hizriadi², E B Nababan² and S Suwilo³

¹Jurusan Teknik Komputer dan Informatika, Politeknik Negeri Medan, Medan, Indonesia
²Fakultas Ilmu Komputer dan Teknologi Informasi, Universitas Sumatera Utara, Medan, Indonesia
³Fakultas Matematika dan Ilmu Pengetahuan Alam, Universitas Sumatera Utara, Medan, Indonesia

E-mail: santiprayudani@polmed.ac.id

Abstract. This study discusses effect of tournament selection on the way individuals compete on the performance of Genetic Algorithms so which one tournament selection is most suitable for the Traveling Salesman Problem (TSP). One algorithm in solving TSP is Genetic Algorithm, which has 3 (three) main operators, namely selection, crossover, and mutation. Selection is one of the main operators in the Genetic Algorithm, where select the best individuals who can survive (the shortest travel route). Tournament selection compares a number of individuals through a match to choose the best individual based on each fitness value, so that the winning individual (the individual going to the next generation) will be chosen. There is two way to compete in an individual in tournament selection is by tournament selection with replacement (TSWR) and without replacement (TSWOR). The final results of the study conducted TSWR gets the best fitness, even though the generation that gets the best fitness is reaching the maximum generation (takes longer to get the best fitness).

1. Introduction
One of the combinatorial optimization problems that is often discussed is the Traveling Salesman Problem (TSP). TSP is very closely related to the field of transportation, where a salesman must go through several cities with a certain route and return to the city of the initial salesman trip exactly 1 (one) time. The problem that often occurs is how to determine the most optimal travel route to obtain the minimum distance, travel time, and minimum (least) travel cost [1]. One algorithm that can be used to solve TSP is Genetic Algorithm. Genetic Algorithm is based on analogies with genetic structure (genes) and chromosome behavior in individual populations. Genetic Algorithms maintain chromosome populations (solutions) related to their fitness values. Parents are chosen to be mated (reproduction) based on its objective value to produce offspring/child, so that fit individuals are given more opportunities to breed. After random initial population, Genetic Algorithm through several operators, namely: selection, selecting individuals where fit individuals are the survivors/selected (survival of the fitness); crossover, represents crossbreeding between individuals (crossing chromosomes in individuals); and mutations, introducing random modifications to genes [2].
Many operator selection techniques in Genetic Algorithms. One of them is tournament selection [3]. Tournament selection differs from other selection operator techniques, which generally do calculations on the value of fitness to choose the best individual to be chosen to go to the next population. Tournament selection compares (matches) between a number of individuals based on the value of each individual’s fitness. In general, individuals with high fitness scores will win the match and will be forwarded to the next population. The greater number of individuals competing (tour size), the greater the selection pressure given to each individual to be selected. Some research that has been done beforehand is related to tournament selection, conducting research by increasing tour size and selecting individuals who will be selected when the match is held. In 2001 Goldberg and Sastry analyzed the effect of tournament selection with replacement on convergence time and population size. The research shows that the execution time remains the same (it does not affect whether tournament selection is done with or without a replacement). However, to achieve the same level of accuracy, tournament selection with replacement requires a larger population size than tournament selection without replacement. Estimates of the population size model used are based on adding noise to the tournament selection with replacement [4]. Razali and Geraghty in 2011, conducted a study to compare the performance of Genetic Algorithm in solving TSP problems by using tournament selection, roulette wheel selection, and rank-based roulette wheel selection. The results stated that tournament selection, namely binary tournament selection is better in achieving the best solution quality with low computational time. The results of this study also reveal that tournament selection is superior to small problems, but becomes susceptible to early convergence when problem size increase[5]. El Moujahid et al in 2011 introduced a new technique, namely FATS (Fuzzy Adapting Tournament Selection) which is used to dynamically determine tour size using the fuzzy clustering method. The basic idea is to adjust selection pressure in accordance with the diversity of the population (diversity population) with the aim of balancing exploitation and exploration. FATS precisely adjusts selection pressure according to search results dynamically using different tour sizes at different stages of genetic algorithms by considering the nature of the problem described[6]. Gupta, S. and Panwar, P., in 2013 solve the problem of Traveling Salesman Problems by using the genetic algorithm approach, which starts from the Euclidean distance matrix which is calculated between cities and randomly chooses cities as initial population than using tournament selection with the sorting method (heuristic method), 2 points crossover and interchange mutation. The results show that by combining heuristic methods and genetic algorithms can find a good solution for the Traveling Salesman Problem[7].

In this study, we discusses the effect of choice tournament selection on the performance of Genetic Algorithms on the Traveling Salesman Problem (TSP). The sample data gets from TSPLIB, which is data of 3 (three) cities [8]. City data consisting of coordinate points of a city and the number of cities varies.

2. Method
The general research of this study can be seen in Figure 1.
The first step starts with taking sample data from TSPLIB as test data. Then do the encoding steps on the test data to be processed using Genetic Algorithms. Then determine the parameters in the Genetic Algorithm and process the test data. At the operator selection stage, testing of 2 (two) tournament selection operator selection techniques, namely Tournament Selection With Replacement (TSWR) and Tournament Selection Without Replacement (TWWOR). The final result is comparing the test results of 2 (two) tournament selection in getting the best fitness.

2.1. Genetic Algorithm

Genetic Algorithm is an algorithm that is used to solve the search for value in optimization problems. In Genetic Algorithms, individuals who are called candidate solutions to problems are often coded as bit strings. Individu consist of chromosomes - special elements of candidate solutions. The gene contained a chromosome forming element (in the TSP gene is a city, a chromosome is a collection of several cities, individu are a series of cities that form a travel route). Each gene has an allele (character value) at the locus (string position) consisting of 0 or 1. Crossovers consist of mating genes between two individual parents. Mutations consist of exchanging alleles by choosing a locus randomly.

The Genetic Algorithm process starts from the population which is a set of randomly generated solutions. Each member of the set represents 1 (one) problem solution called an individu that consisting of genes. Individu in a population evolve based on the value of fitness which is an objective function of the problem being optimized. Individu are selected according to their fitness values. Strong individuals have a high chance of survival in the next generation. Then a crossover and mutation process are performed on the selected individual (parent) to form a new individu (offspring). From 2 (two) processes above a new generation is formed. The crossover and mutation will be repeated continuously until it reaches a convergent, that is, as many generations as desired.

For more details can be seen in Figure 2 below:

![Flowchart Genetic Algorithm](image-url)

2.2. Tournament Selection With Replacement (TSWR)

At TSWR, individu who have competed in one match can be re-elected to compete in the next match. In addition, the same individu can compete with each other, making TSWR produce noise. Noise here means that the possibility of an individu being chosen many times for a match is very large, so the chance (possibility) of another individu being selected for a match is very small. The step TSWR can be seen below:
• If $r < ps$, then enter individu to matingpool. If $r \geq ps$, then delete individu.
• Choose $k$ individu sebanyak $s$ at randomly from the tournament, where $s$ is tournament size $[2, N]$. $N$ is population size.
• Select the best individu among the $s$ individu by comparing its fitness (occurrence of the tournament (match)):
  o If fitness of individu $k1 >$ fitness of individu $k2$, then individu $k1$ is winner.
  o If fitness of individu $k1 ==$ fitness of individu $k2$, then individu winner is chosen randomly between those compared.
• The selected individu (winner) from step 3 is entered or copied into the matting pool.
• Winning and losing individuals are copied back into the tournament (winning and losing individuals can be re-elected).
• Number of individuals in the matting pool = number of individuals currently in matting pool + 1.
• Repeat step 2 to 6 to the number of individuals in the matting pool = $N$.
• Individuals in the matting pool will go to crossover process.

2.3. Tournament Selection Without Replacement (TSWOR)
At TSWOR, individu who have been selected for a match cannot be selected again for the next match and no same individu competes in a match, so that each individu has the same opportunity to be selected for a match [5]. The step TSWOR can be seen below:
• If $r < ps$, then enter individu to matingpool. If $r \geq ps$, then delete individu.
• Choose $k$ individu sebanyak $s$ at randomly from the tournament, where $s$ is tournament size $[2, N]$. $N$ is population size.
• Select the best individu among the $s$ individu by comparing its fitness (occurrence of the tournament (match)):
  o If fitness of individu $k1 >$ fitness of individu $k2$, then individu $k1$ is winner.
  o If fitness of individu $k1 ==$ fitness of individu $k2$, then individu winner is chosen randomly between those compared.
• The selected individu (winner) from step 3 is entered or copied into the matting pool.
• Winning and losing individuals are copied back into the tournament (winning and losing individuals can’t be re-elected).
• Number of individuals in the matting pool = number of individuals currently in matting pool + 1.
• Repeat step 2 to 6 to the number of individuals in the matting pool = $N$.
• Individuals in the matting pool will go to crossover process.

3. Result and discussion
Test data is taken from TSPLIB. The selected TSPLIB data can be seen in table 1 below:

| Data       |
|------------|
| rat783.tsp |
| rl5934.tsp |
| rl11849.tsp |

Source: http://comopt.ifi.uni-heidelberg.de/software/TSPLIB95/XML-TSPLIB/instances/ [8]

As for the Genetic Algorithm parameter [12], the same value for each tournament selection is used, can be seen in table 2 below:
Table 2. Data TSPLIB.

| Population Size | 100 |
|-----------------|-----|
| Number of Generation/Iteration | 500 |
| Tour Size | 10 |
| Selection Probability | 0.7 |
| Crossover Probability | 0.9 |
| Mutation Probability | 0.1 |

3.1. Testing Tournament Selection With Replacement (TSWR)

The test is carried out using tournament selection with replacement and using predefined parameters. After being executed up to 3 (three) times, the average value of the generation in each city is taken. The test results can be seen in table 3.

Table 3. Tournament Selection With Replacement (TSWR).

| City        | Gene-ration | Best Fitness | Distance | Genera- tion | Best Fitness | Distance | Genera- tion | Best Fitness | Distance | Genera- tion | Best Fitness | Distance | Genera- tion | Best Fitness | Distance |
|-------------|-------------|--------------|----------|--------------|--------------|----------|--------------|--------------|----------|--------------|--------------|----------|--------------|--------------|----------|
| rat783.tsp  | 499         | 8.99E-03     | 111.267  | 8.76E-03     | 114.216.6   | 95       | 9.09E-03     | 109.986.161 | 500      | 9.09E-03     | 109.986.161 | 500      | 9.09E-03     | 109.986.161 | 500      |
| rl5934.tsp  | 499         | 2.73E-05     | 36.728.92| 8.873        | 36.728.92   | 8.873    | 2.74E-05     | 36.451.742   | 498      | 3.50E-05     | 36.451.742   | 498      | 3.50E-05     | 36.451.742   | 498      |
| rl11849.tsp | 500         | 1.26E-05     | 79.519.47| 3.974        | 79.519.47   | 3.974    | 1.27E-05     | 78.878.734   | 499      | 1.27E-05     | 78.878.734   | 499      | 1.27E-05     | 78.878.734   | 499      |

Table 3 above shows that by using tournament selection with replacement, the average generation in getting the best fitness can be seen from:

- The highest best fitness value is obtained by rat783.tsp, which is 9.09E-03
- The lowest best fitness value is obtained by rl11849.tsp, which is 1.27E-05
- The fastest generation in getting the best fitness is 500 generation.
- The longest generation to get the best fitness is the 498 generation.

3.2. Testing Tournament Selection Without Replacement (TSWOR)

The test is carried out using tournament selection without replacement and using predefined parameters. After being executed up to 3 (three) times, the average value of the generation in each city is taken. The test results can be seen in table 4.

Table 4. Tournament Selection Without Replacement (TSWOR).

| City        | Gene-ration | Best Fitness | Distance | Genera- tion | Best Fitness | Distance | Genera- tion | Best Fitness | Distance | Genera- tion | Best Fitness | Distance | Genera- tion | Best Fitness | Distance |
|-------------|-------------|--------------|----------|--------------|--------------|----------|--------------|--------------|----------|--------------|--------------|----------|--------------|--------------|----------|
| rat783.tsp  | 489         | 7.19E-03     | 139.134  | 7.15E-03     | 139.883.940 | 496      | 7.29E-03     | 137.094.802 | 492      | 7.29E-03     | 137.094.802 | 492      | 7.29E-03     | 137.094.802 | 492      |
| rl5934.tsp  | 497         | 2.52E-05     | 39.700.9 | 2.53E-05     | 39.572.660.938 | 498      | 2.51E-05     | 39.927.488.530 | 498      | 2.51E-05     | 39.927.488.530 | 498      | 2.51E-05     | 39.927.488.530 | 498      |
| rl11849.tsp | 500         | 1.19E-05     | 84.209.3 | 1.20E-05     | 83.463.499.891 | 494      | 1.20E-05     | 83.541.29.998 | 498      | 1.20E-05     | 83.541.29.998 | 498      | 1.20E-05     | 83.541.29.998 | 498      |

Table 4 above shows that by using tournament selection without replacement, the average generation in getting the best fitness can be seen from:

- The highest best fitness value is obtained by rat783.tsp, which is 7.19E-03
• The lowest best fitness value is obtained by rl11849.tsp, which is 1,E-05.
• The fastest generation in getting the best fitness is 492 generation.
• The longest generation to get the best fitness is the 498 generation.

4. Result and discussion
From the graph shown in Figure 3 below it appears that the average of the longest generation (maximum generation) in getting best fitness with the average maximum generation set in this study is tournament selection with replacement (TSWR). Meanwhile, the fastest average generation (faster convergent) in getting the best fitness is tournament selection without replacement (TSWOR).

![Comparison of the Average Generation in Getting the Best Fitness](image)

**Figure 3. Chart of Average Generations for Getting the Best Fitness.**

From the graph shown in Figure 4 below it appears that the highest best fitness obtained based on the average generation is tournament selection with replacement (TSWR). Meanwhile, the lowest best fitness based on the average generation in this study is tournament selection without replacement (TSWOR).

![Comparison of Best Fitness With Average Generation](image)

**Figure 4. Chart of Best Fitness with Average Generations.**
5. Conclusion
From the test results with TSWR and TSWOR on 3 (three) sample data, obtained:

- Tournament Selection With Replacement (TSWR) longer to get best fitness, while Tournament Selection Without Replacement (TSWOR) is faster to get best fitness.
- Tournament Selection With Replacement (TSWR) gets the highest best fitness, while Tournament Selection Without Replacement (TSWOR) gets the lowest best fitness.

References
[1] Ahmed, Z. H., 2010. Genetic Algorithm for the Traveling Salesman Problem using Sequential Constructive Crossover Operator. International Journal of Biometrics & Bioinformatics (IJBB) 3(6): 96-105.
[2] Taiwo, O. S., Mayowa, O. O. & Ruka, K. B. 2013. Application of Genetic Algorithm to Solve Traveling Salesman Problem. International Journal of Advance Research,(IJOAR) 1(4): 27-48
[3] Goldberg, D. E. 1989. Genetic Algorithms In Search Optimization And Machine Learning. Addison-Wesley Professional: Massachusetts.
[4] Goldberg, D. E. & Sastry, K. 2001. A Practical Schema Theorem For Genetic Algorithm Design And Tuning. Proceedings of the 3rd Annual Conference on Genetic and Evolutionary Computation 2001, pp. 328-335
[5] Razali, M. N. & Geraghty, J. 2011. Genetic Algorithm Performance with Different Selection Strategies in Solving TSP. Proceedings of the World Congress on Engineering 2011, pp. 1134-1139
[6] El, M. A., Jebari, K., Bouroumi, A. & Ettouhami, A. 2011. A Fuzzy Clustering Technique for Adapting Tournament Selection. International Journal of Soft Computing 6(3): 62-67
[7] Gupta, S. & Panwar, P. 2013. Solving Travelling Salesman Problem Using Genetic Algorithm, International Journal of Advanced Research in Computer Science and Software Engineering (IARCSSE) 3(6): 376-380
[8] http://comopt.uni-heidelberg.de/software/TSPLIB95/XML-TSPLIB/instances/
[9] Negnevitsky, M. 2005. Artificial Intelligence: A Guide to Intelligent Systems. 2nd Edition. Addison-Wesley Peason Education Limited: London.
[10] Purnamawati S, Nababan E B, Tsani B, Taqyuddin R, Rahmat R F 2018. Advertisement scheduling on commercial radio station using genetics algorithm Journal of Physics: Conference Series 978(1),012113
[11] Mitchell, M. 1996. An Introduction to Genetic Algorithms. MIT Press. Cambridge, Massachusetts: London.
[12] Santi. P., 2013. Pengaruh Tournament Selection Terhadap Pengukuran Kinerja Genetic Algorithm Dalam Menyelesaikan Traveling Salesman Problem. Tesis. Tidak Diterbitkan. Fasilkom-TI. Universitas Sumatera Utara: Medan