International Journal on Emerging Technologies **10**(1): 206-210(2019)

ISSN No. (Print) : 0975-8364
ISSN No. (Online) : 2249-3255

Quantified Performance Analysis of Selection Algorithms Implemented in Mobile Robot Path Planning

Suresh K.S., Kesudh G., Hemanath S. and K.R. Sekar
School of Computing,
SASTRA Deemed University, Thanjavur (Tamilnadu), India

(Corresponding author: Suresh K.S.)
(Received 10 April 2019, Revised 06 June 2019, Accepted 12 June 2019)
(Published by Research Trend, Website: www.researchtrend.net)

ABSTRACT: **Mobile Robot Path Planning Problem (MRPP) can be solved by applying different methodologies suitable for the specific problem space. One of the approaches to solve the MRPP is using Genetic Algorithm (GA) to find the optimal solution path, out of large solution space consists of candidate paths. The GA operators such as crossover, mutation and selection operators are applied on the population. Mainly the quality of the path is based on the selection algorithm employed to select the better individuals to the next generation. Specifically, the Roulette wheel and Tournament and Rank selection algorithms are popular algorithms for creating next generation. For MRPP, the qualified paths are to be pushed to the next generation to establish a qualitative population to get the better path at the end. These three algorithms are analyzed and compared with different parameters for different problem space and the efficiency is quantified with a Performance Factor.**

Keywords: Mobile Robot Path Planning Problem – Genetic Algorithm – Selection Algorithms – Tournament Selection – Roulette Wheel Selection – analysis of algorithms.

I. INTRODUCTION

Mobile robot is an inevitable research domain which paved the way for industrial automation [1]. The mobile robot path planning problem is a NP-Hard problem [2], so different approaches are being used to solve the problem. The hybridization of approaches [3] is tried for the betterment of efficiency. The Meta Heuristic approaches [4-5] may not give the exact solution compared to the classical approaches [6], but could give the optimal solution with lesser time. One of the Meta Heuristic approaches is Genetic Algorithm (GA) which is a population based algorithm to derive the optimized solution from randomly generated population. The fitness function is determined using the length of the path. The objective function is to evaluate the optimal paths. For each path, the fitness value is calculated by employing the fitness function. The crossover and mutation operators are applied on the initial solution space. These operators are employed to avoid convergence to the local optimal path and to search the new possible paths. After the completion of each iteration, the population has to undergo the selection process to move towards the optimal solution. Therefore the selection operator influences the quality of the solution in each generation. The convergence time is decided by the number of generations to be iterated or saturation of fitness value. The convergence into the optimal solution is influenced [7] by various parameters employed during different phases of the GA. Not only the implementation methods for crossover and mutation, but also the rate of crossover and mutation operation [8] influences the efficiency of the algorithm, since the rate decides the participation of population [9] in producing the next generation. The crossover and mutation operators are supposed to add to the randomization of the population to move the solution phase towards the optimal solutions.

II. GENETIC ALGORITHM PRINCIPLE FOR ROBOT PATH PLANNING

Genetic Algorithm involves methods similar to evolution of biological species where the fittest one will survive for the next generation. In case of mobile robot path planning, GA principle [10] is applied to find the optimal path out of generated candidate paths. The optimal path [11] may consider single objective as path length or multiple conflicting objectives [12],[13] such as length, smoothness or safety. In the path-planning problem, different strategies are exercised for the different phases of algorithm to test the quality of the solution produced. Repetitive application of the genetic algorithm mimics the real life offspring with properties of evolution, giving a population filled with more optimal offspring at the end of every iteration.

III. IMPLEMENTATION

A. Representation of an individual and initial population generation

In the path planning algorithms, the working space of the robot will be modelled as a grid and the size of an each cell in the grid is considered as the size of the robot. The grid is implemented as the occupancy matrix with 0s and 1s where 0 represents the free cell and 1 represents the blocked cell. The given starting point and destination point are defined in terms of indices of the matrix. The possible candidate paths are expressed in terms of the indices of the corresponding cells of the occupancy matrix. Since the paths are of variable length, to improve the performance, instead of calculating the length of the individual path every time after applying different operators, path length is updated as an additional value with the indices of the individual path so that it can be used to calculate the fitness value for the paths.

The implementation of the genetic algorithm begins with generation of the possible solution space for the given problem domain using brute force methodology. The solution space has to be processed through different phases of GA. Though each phases of GA has an array of alternatives for implementation, the quality of the population for each generation is ensured by the selection phase by exploiting the better characteristics of the existing solution space. The crossover and mutation operators are to find the new individual to be added into the population and also not to get into the local minima.

While evaluating the initial population, the paths involving loops are not considered, as this would increase the size of the initial population exponentially, and also a path involving a loop is not optimal and reduce the performance of the system.

IV. SELECTION ALGORITHMS

After the randomly generated population, the selection operator is to be applied for each generation which decides the merit of the next generation. Whatever may be the selection algorithm used, the stochastic process is the fundamental approach to select the competent individual for the next generation.

The random members of the population are chosen for the selection process by a time-seeded random function, which uses the current timestamp to generate random numbers which are either used to match with the index of a member of the population in the case of Tournament selection or to generate the force of the ball thrown in the case of Roulette wheel and Rank selection.

For a robot path planning problem, the primary objective for the fitness function is deriving the path with optimized length. The Fitness Value (FV) for individual member(i) of the population is evaluated as the difference between the Maximum Path Length (PLmax) and the Path length of the individual member (PLi). The maximum path length is considered based on the size of the occupancy matrix.

$$FVi = PLmax - PLi \qquad (1)$$

The fitness value is calculated (1) with reference to the assumed maximum path length. As and when the path is formed, the length of the path is affixed with the individual paths which are being used to assess the fitness value. The length of the path is inversely proportional to the fitness value.

A. Roulette wheel

In the case of Roulette wheel, the perimeter of the wheel(WP) where the ball has to be rolled is estimated by summing up the fitness value (FV_i) of all the members of the population evaluated from the equation(1).

$$WP = \sum_i \mathrm{FVi} \qquad (2)$$

The individual paths are selected randomly for the next generation by using the range calculated by (2), assumed as perimeter of the wheel. This ball being rolled is iterated as many times as required for each iteration of population selection. The area occupied by a member of the population is directly proportional to how optimal a solution it is, based on the path length. Thus the probability of the ball landing on an optimal solution is higher than it landing on a non-optimal solution. The ninety percentage of the population is selected for the next generation expecting the probability of selecting the higher fitness value is high.

B. Tournament selection

As the principle of this selection method, multiple tournaments between randomly selected members of the population are carried out and the winners of these tournaments are passed to the population for the next iteration. The fitness function is calculated for evaluating the winner of these tournaments, and the factors used for finding the fitness function are path length.

C. Rank Selection

According to the Rank selection, the population generated in each generation is sorted based on the fitness value. Since the robot path planning problem is a minimization problem, the individual population which has lowest fitness value will be assigned with lowest rank and the rank will be increasing with the fitness value. The probability of selecting a path for the next generation will be based on the rank rather than the fitness function as in the Roulette Wheel selection algorithm. Therefore the probability of selection of individual is distributed.

IV. CROSSOVER

The single point crossover is employed to acquire the diversity over the population. Applying crossover (Fig. 1) operator to a good percentage of population builds a wide range for the solution space. While performing crossover, it poses two complications. a) Identify the crossover point for randomly selected individuals. b) After crossover the length of the paths are to be updated.

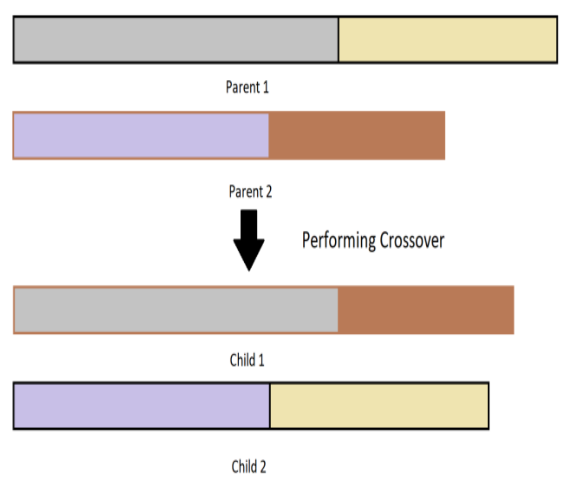

Fig. 1. Crossover operation on two parent paths.

Specifically in the robot path planning solution space, the crossover operator cannot be exercised at a random point as in the other domain applications, because, if the crossover operator is applied randomly, the path becomes incoherent. Therefore before implementing, the compatible point must be identified for the randomly selected paths, and then the crossover is employed. The length of the paths will vary for the newly created paths, hence to be updated for both. If the length of the path of the two selected individuals are x and y respectively and j and k are corresponding crossover points to apply crossover, then the new length is calculated (3&4) after crossover.

Length of Parent 1 – x units
Length of Parent 2 – y units
Point of crossover for Parent 1 - j
Point of crossover for Parent 2 - k
Length of Child 1 =Point of crossover of Parent 1 + (Length of Parent 2 – Point of crossover of Parent 2)

$$LCH1 = j + (y - k) \tag{3}$$

Length of Child 2 =Point of crossover of Parent 2 + (Length of Parent 1 – Point of crossover of Parent 1)

$$LCH2 = k + (x - j) \tag{4}$$

The two parents chosen for crossover are taken from the population which undergoes the selection process, hence population size doesn't change between selection and crossover.

V. MUTATION

Only one percentage of the stochastically selected paths is involved in the mutation process to explore possibility of the new paths. Since the individual must comply with the solution set after mutation, the point to be mutated must satisfy the following two constraints.
The new point must be adjacent to the point it is replacing.
The new point should be adjacent to the previous and next point in the solution path sequence.
The phases of GA are exercised for each generation which derives the optimal path for the given environment.

VI. RESULTS AND INFERENCES

The Table 1 compares the Tournament (TT), Roulette Wheel (RW) and Rank(RK) selection algorithms for different environments (Maze 1, Maze 2, Maze 3, Maze 4, Maze 5 and Maze 6) varied with percentage of obstacles (35%, 50% and 65%) and size of the environment (10 x 10, 13 x 13 and 15 x 15).

Table 1: Comparison of Selection Algorithms for arbitrary environments with 3 different size and percentage of obstacles.

Block percentage		**35%**						**50%**					
		Maze 1			**Maze 2**			**Maze 3**			**Maze 4**		
Selection algorithm		TT	RW	RK	TT	RW	RK	TT	RW	RK	TT	RW	RK
10x10	**Initial**	35.36			28.61			24.697			23.32		
	Final	10.1	27.25	26.3	10.6	25	27.2	10.4	24.45	23	11	22.44	25
	Min	10			10			10			11		
	PF	99	36.69	38.1	94.33	40	36.76	94.33	40	36.76	100	49.01	44
13x13	**Initial**	29.7			33.79			23.55			27.26		
	Final	15	29	30.2	13.6	31	32.7	15	21.63	24.2	19	13	293
	Min	14			13			15			13		
	PF	93.33	48.27	46.2	95.58	41.93	39.75	100	69.34	61.98	68.42	100	44.36
15x15	**Initial**	34.17			35.81			26.22			31.47		
	Final	17	25.82	35.2	15.8	26.56	33.52	15.3	24.3	25.31	17	31	32.57
	Min	16			15			15			16		
	PF	94.11	61.96	45.5	94.93	56.47	44.47	94.93	56.47	44.74	94.11	51.61	49.12

		65%					
		Maze 5			**Maze 6**		
Selection algorithm		TT	RW	RK	TT	RW	RK
10x10	**Initial**	20.03			20.64		
	Final	10	16.51	15.3	10.5	14.26	15.45
	Min	10			10		
	PF	100	60.56	65.6	95.23	70.12	64.72
13x13	**Initial**	25.27			24.14		
	Final	13.1	27	26.2	13.3	25	22.12
	Min	13			13		
	PF	99.23	48.14	49.6	97.74	52	58.77
15x15	**Initial**	31.38			27.37		
	Final	29	31	33.2	15	31	30.25
	Min	29			15		
	PF	100	93.54	87.2	100	48.38	49.58

TT – Tournament Selection, RW - Roulette Wheel Selection, RK – Rank Selection

The algorithms are implemented with Genetic Algorithm principle and analyzed to get the statistical substantiation.

From the tabulated obtained results, it is very difficult to come to the conclusion by comparing the numerical values derived by implementing different algorithms for varied configurations. Hence the performance factor is used to compare the competences of the algorithm.

VII. QUANTIFICATION OF PERFORMANCE

A performance factor is defined to compare the different selection algorithms as follows.

Performance factor = (Minimum path length possible for given maze/ Final average path length) ×100

The performance factor (PF) is used to quantify and compare the efficiency of convergence of the GA to the minimum path length possible for the selection algorithms. The PF is defined in this way because the direct comparison of the average path values on GA would be ambiguous due to different environments considered.

To remove these inconsistencies in comparison, each value is normalized to evaluate the PF by comparing it to the minimum possible value for that specific environment.

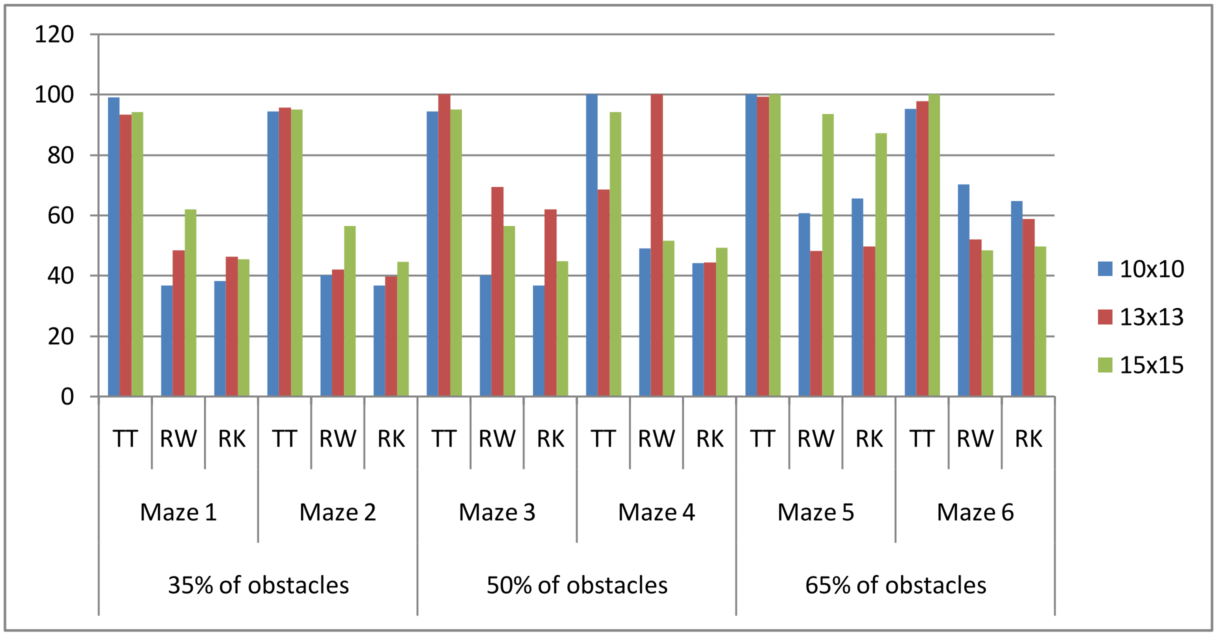

Chart 1. Comparison of Performance Factor for Tournament, Roulette Wheel and Rank Selection Algorithms for arbitrary environments with 3 different size(10×10,13×13 and 15×15 and percentage of obstacles(35%, 50% and 65%).

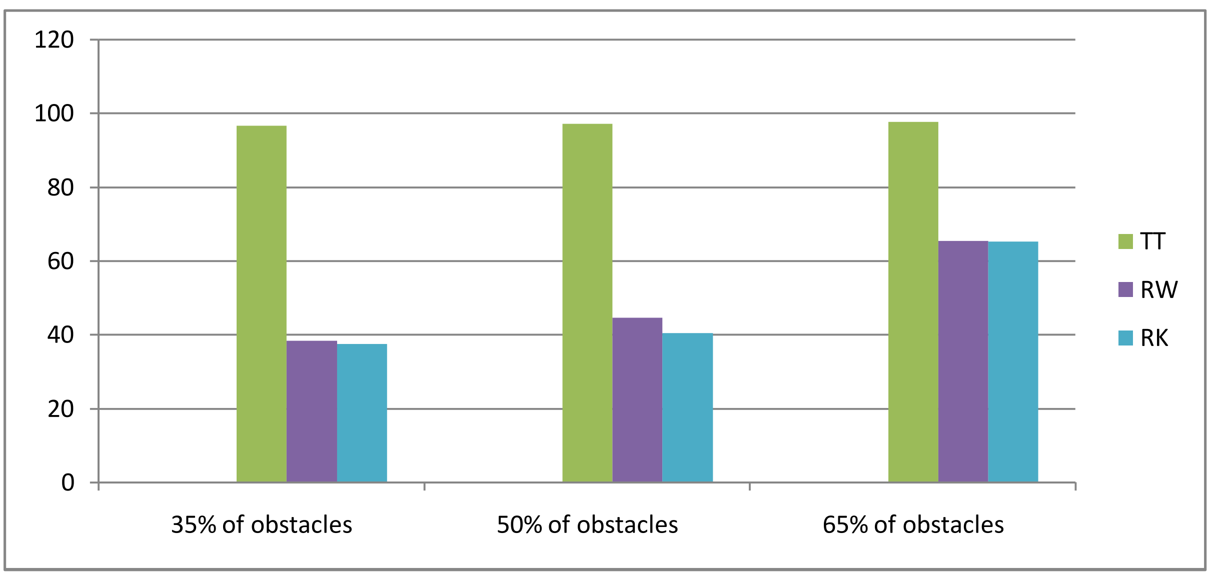

Chart 2. Comparison of average of Performance Factor for Tournament, Roulette Wheel and Rank Selection Algorithms for arbitrary environments with 3 different size (10×10,13×13 and 15×15 and percentage of obstacles (35%, 50% and 65%).

For normalization, the ratio of overall Minimum Path Length (MPL) of the maze and Final Average Path Length (FAPL) of each maze with different size and percentage of obstacles is calculated.

$$PF = \frac{MPL}{FAPL} X\ 100$$

The following table shows the PF calculated for each of the individual mazes for three different selection algorithms and the average of performance factors for the algorithms is tabulated.

By comparing the values of average PF it is inferred that Tournament selection algorithm is very efficient for robot path planning compared to Roulette Wheel and Rank selection algorithms.

Selection Algorithm→	**TT**	**RW**	**RK**
Average Performance Factor→	95.566	57.257	51.593

VIII. CONCLUSION

The Genetic Algorithm Principle is exercised to derive the optimised path for robot path planning problem. The Tournament, Roulette Wheel and Rank Selection Algorithms are implemented for the environment with different size and percentage of obstacles to analyse and compare the efficacy of the algorithm in robot path planning with obstacle avoidance. The analysis of the problem gives the inference that for robot path planning problem in most of the cases the Tournament Selection algorithm outperforms the Roulette Wheel and Rank Selection Algorithms. This inference is evident by evaluating the Performance Factor for each algorithm and compared.

REFERENCES

[1]. Sun, B., Chen, W.D. and Xi, Y.G. (2005). Particle swarm optimization based global path planning for mobile robots. *Control and Decision,* **20**(9): 1052.

[2]. Xue, Y. and Sun, J.Q. (2018). Solving the path planning problem in mobile robotics with the multi-objective evolutionary algorithm. *Applied Sciences,* **8**(9): 1425.

[3]. Das, P.K., Behera, H.S. and Panigrahi, B.K. (2016). A hybridization of an improved particle swarm optimization and gravitational search algorithm for multi-robot path planning. *Swarm and Evolutionary Computation,* **28**: 14-28.

[4]. Mac, T.T., Copot, C., Tran, D.T. and De Keyser, R. (2016). Heuristic approaches in robot path planning: A survey. *Robotics and Autonomous Systems,* **86**: 13-28.

[5]. Suresh, K.S., Vaithiyanathan, V. and Venugopal, S. (2013). Emergence of meta heuristic algorithm principles in the field of robot motion planning-A survey of paradigm shift. *International Journal of Applied Engineering Research*, **8**(20): 2717-2.

[6]. Masehian, E. and Sedighizadeh, D. (2007). Classic and heuristic approaches in robot motion planning-a chronological review. *World Academy of Science, Engineering and Technology*, **23**(5): 101-6.

[7]. Suresh, K.S., Rajarajan, S., Prabhu, M. and Vaithiyanathan V. (2013). Study of the Influence of Genetic Algorithm Parameters on Travelling Salesman Problem. *International Journal of Applied Engineering Research,* **8**(20): 2013.

[8]. Rexhepi, A., Maxhuni, A. And Dika, A. (2013). Analysis of the impact of parameters values on the Genetic Algorithm for TSP. *International Journal of Computer Science Issues (IJCSI),* **10**(1): 158.

[9]. Suresh, K.S., Ravichandran, K.S., Ananthakrishnan, S. and Venugopal, S. (2017). Adaptive strategy to improve the efficiency of robot path planning using population based algorithm. *International Journal of Mechanical Engineering and Technology,* Volume **8**, Issue 8, pp. 1441–1448.

[10]. Ismail, AT., Sheta, A. and Al-Weshah, M. (2008). A mobile robot path planning using genetic algorithm in static environment. *Journal of Computer Science*. **4**(4): 341-4.

[11]. Raja, P. and Pugazhenthi, S. (2012). Optimal path planning of mobile robots: A review. International Journal of Physical Sciences, **7**(9): 1314–20.

[12]. Suresh, K.S., Ravichandran, K.S., Aananthakrishnan, S. and Ponmani, E. (2017). Emergence of multi objective genetic algorithm in robot path planning –a survey. *International Journal of Mechanical Engineering and Technology*, Volume **8**, Issue 8, pp. 1449–1457.

[13]. Nazarahari, M., Khanmirza, E. and Doostie, S. (2019). Multi-objective multi-robot path planning in continuous environment using an enhanced genetic algorithm. *Expert Systems with Applications*, **1**(115): 106-20.

How to cite this article: Suresh, K.S., G., Kesudh, S. Hemanath and Sekar, K.R. (2019). Quantified Performance Analysis of Selection Algorithms Implemented in Mobile Robot Path Planning. *International Journal on Emerging Technologies*, **10**(1): 206-210.