Global Journal of Computer Sciences: Theory and Research

Volume 07, Issue 2, (2017) 58-67

www.gjcs.eu

Optimization of potential field parameters using genetic algorithm

Vahid Babaei Ajabshir, Computer Engineering Department, Ankara University, Golbasi 50.yil Yerleskesi Bahçelievler Mh, Ankara 06830, Turkey.

Mehmet Serdar Guzel*, Computer Engineering Department, Ankara University, Golbasi 50.yıl Yerleskesi Bahçelievler Mh, Ankara 06830, Turkey.

Serhat Can, Computer Engineering Department, Ankara University, Golbasi 50.yil Yerleskesi Bahçelievler Mh, Ankara 06830, Turkey.

Erkan Bostanci, Computer Engineering Department, Ankara University, Golbasi 50.yil Yerleskesi Bahçelievler Mh, Ankara 06830, Turkey.

Suggested Citation:

Ajabshir, V. B., Guzel, M. S., Can, S. & Bostanci, E. (2017). Optimization of potential field parameters using genetic algorithm. *Global Journal of Computer Sciences: Theory and Research. 7*(2), 58-67.

Received April 23, 2017; revised June 10, 2017; accepted August 5, 2017.

Selection and peer review under responsibility of Prof. Dr. Dogan Ibrahim, Near East University, North Cyprus.

Abstract

This paper addresses a metaheuristics approach to optimize the parameters of the potential fields (PF) method. This method is an important algorithm and is primarily used for local navigation problems. However, estimating the appropriate parameters is essential for safe and smooth navigation. For instance, complex scenarios that include long and thin corridors or cluttered environments having numerous obstacles require reliable parameter estimation. Accordingly, the genetic algorithm is utilized to estimate the appropriate algorithms to overcome conventional navigation problems based on the PF method. The experimental results verify the reliability and efficiency of the proposed approach.

Keywords: Path planning, potential fields (PF) method, genetic algorithm (GA), DEAP, robot navigation, parameter optimization.

* ADDRESS FOR CORRESPONDENCE: **Mehmet Serdar Guzel,** Computer Engineering Department, Ankara University, Golbasi 50.yil Yerleskesi Bahçelievler Mh, Ankara 06830, Turkey. *E-mail address*: mguzel@ankara.edu.tr

1. Introduction

An autonomous mobile robot needs to detect and avoid obstacles during the navigation process in both static and dynamic environments. Local path planning is considered as a reactive strategy in which sensory data is collected by an agent's sensors to perceive the environment and perform planning in a fast and reactive manner. The potential field (PF) method is one of the most well-known and popular local navigation algorithms used for mobile robot navigation. Despite its simplicity, it provides reliable and safer navigation than most of its counterpart algorithms. One of the main advantages of the PF method is its ability to consider both obstacle avoidance and motion planning simultaneously [1]. However, the PF approach has two main problems to overcome: the conventional local minima problem and the parameter optimization problem [2].

Local minima problem mainly occurs when the robot gets trapped and stops in front of an obstacle rather than its goal. This occurs when the attractive force of the goal point and the repulsive force of the obstacle become equal. Environments, including concave obstacles and mazes, may cause this problem [3]. There are comprehensive methods in the literature that can tackle the local minima problem [3–5].

The next critical problem of the PF method is to estimate the optimum or close-to-optimum parameters used by the PF approach. The PF method, in essence, comprises four parameters that are mainly responsible for smooth, reliable and safe navigation. These parameters, defining the behaviour of the PF, are scale factors ∞, β and spread (*s*) values of goal (s_g) and obstacle (s_o), defining the circle of extent. Finding the functioning set of parameters for a simple scenario may not always be too difficult; however, when taking path smoothness and length into consideration the problem can easily become challenging. Furthermore, once the scenario is complex, estimating appropriate values for these parameters seems almost impossible. Oscillation between two obstacles is another problem that should be taken into account in the PF approach and can only be reduced significantly by selecting the proper parameter values.

As aforementioned, the PF approach is completely parameter dependent, and therefore, it needs a reliable parameter estimation approach. In order to overcome the parameter estimation problem for different mobile robot navigation scenarios, a metaheuristic approach is employed. The technique utilized in this approach exploits the genetic algorithm (GA) that essentially provides an approximation to the best possible values for the PF parameters. The main advantage of this algorithm is its ability to find an optimum parameter set in a reasonable amount of time, even for the challenging scenarios.

The remainder of this paper is organized as follows. Discussion of PF is represented in Section 2. Sections 3 and 4 are dedicated to introduction and discussion of GA and the way it is applied for the given problem. The experimental results are given in Section 5. Finally, the study is concluded in Section 6.

2. Overview of Potential Fields

The PF method is one of the commonly used offline path planning methods for autonomous robots. It was first introduced in [6]. In this method, the robot is regarded as a particle that moves through different repulsive and attractive fields. Another way to think about the PF is to consider the robot as a rolling marble in a terrain with obstacles, like hills, which exert a repulsive force on the robot, while the goal is a valley that exerts an attractive force on the robot. The final robot path direction in each step is calculated according to the sum of the two forces in that state.

The first force that is exerted on the robot is the attractive force of the goal, which is a result of the goal-seeking behaviour and is represented as an action vector. The forces are represented as a gradient vector, which is $\Delta_G = [\Delta_G x, \Delta_G y]$. In order to calculate $\Delta_G x$ and $\Delta_G y$ formulas, Eqs. (1–3), are adapted from the corresponding studies [7–9].

$$if\ d < r \qquad \Delta_G x = \Delta_G y = 0 \tag{1}$$

$$if\ r \le d \le s + r \quad \Delta_G x = \alpha(d - r)\cos(\theta)\ and \quad \Delta_G y = \alpha(d - r)\sin(\theta) \tag{2}$$

$$if\ d > s + r \quad \Delta_G x = \alpha s\cos(\theta)\ and \quad \Delta_G y = \alpha s\sin(\theta) \tag{3}$$

In these formulas, *d* represents the distance between robot and goal; θ is the angle between robot and goal, while *s* is the spread around the goal. On the other hand, the forces that are exerted on the robot by the obstacles are represented as $\Delta_o = [\Delta_o x, \Delta_o y]$ and are calculated as follows:

$$if\ d < r \quad \Delta_o x = -sign(\cos(\theta))\infty\ and \quad \Delta_o y = -sign(\sin(\theta))\infty \tag{4}$$

$$if\ r \le d \le s + r \quad \Delta_o x = -\beta(s + r - d)\cos(\theta)\ and \quad \Delta_o y = -\beta(s + r - d)\sin(\theta) \tag{5}$$

$$if\ d > s + r \quad \Delta_o x = \Delta_o y = 0 \tag{6}$$

The total force exerted on the robot is then calculated by (7, 8)

$$\Delta x = \Delta_o x + \Delta_G x \tag{7}$$

$$\Delta y = \Delta_o y + \Delta_G y \tag{8}$$

3. Genetic Algorithm

Inspired by natural selection and the evolution theory of Darwin, the evolutionary algorithm is introduced and has been widely used in problems of engineering and computer science. Evolutionary algorithms are stochastic optimisation algorithms that provide heuristic solutions to complex problems. They are based upon natural biological evolution principles [10–12]. In these algorithms, randomly initialized groups of candidate solutions are utilized to evolve an optimum solution that operates better in the environment. By nature, one vital principle of the GA is survival of the fittest individuals. Through the evolution process, each new generation comes out to be better than the previous generation. This is due to the high production probability of elite and good chromosomes.

The main objective of this study is to approximate the optimum parameters of the PF approach, namely, obstacle spread (S_o), goal spread (S_g), attractive force (α) and repulsive force (β). This is an extremely important task, as the local navigation behaviour of a robot entirely depends on them. The steps of the proposed GA solution for the given problem are detailed in the following subsections.

3.1. Chromosomes

Defining an effective representation for a chromosome is one of the significant GA design decisions, as it determines the evaluation method for fitness value among the types of genetic operators. Considering that the chromosomes represent a solution to the problem, each gene (allele) of the chromosome represents a value for the given parameters. In accordance with the desired values of the PF method's parameters, it is chosen to encode real values in chromosome representation. In the proposed design, chromosome genes represent α, β, goal spread (S_g) and obstacle spread (S_o) parameters, respectively. A sample chromosome is represented below, which is also the best chromosome values for scenario 1 (see Section 5)

3.12	2.20	2.96	2.76

The upper and lower bounds of each variable are selected by trial-and-error method. Values of the variables will not cross these boundaries during the runtime of genetic operators. The min and max values are shown as follows:

[1.0, 6.0]	[1.0, 8.0]	[0.2, 4.2]	[0.4, 3.6]

3.2. Fitness Calculation

The fitness value of a chromosome enables us to rank the chromosomes. Individuals with higher fitness values are more likely to be selected for reproduction to contribute their genetic makeup to the upcoming generations. For this problem, the fitness value is calculated according to the distance travelled and collision avoidance. In other words, in the evaluation process, the number of collisions (obstacle avoidance) and the overall navigation time to reach the goal are considered. The solutions leading to collisions are discarded and other solutions are ranked according to the path length that the robot traverses from the start point to the goal point.

4. Genetic Operators and Genetic Application

Genetic operators are used for creating offspring from chromosomes of the current population. They must be designed in such a way that avoid premature convergence of population and maintain genetic diversity. Crossover (Recombination) and mutation are the two primary genetic operators.

3.3.Mutation

With respect to chromosome representation, among the various methods of mutation, Gaussian mutation is a good match for our chromosome structure. Gaussian mutation is utilized in real-value representations. In this approach, a randomly generated value from a Gaussian distribution is added to an allele that is chosen with a probability of 0.2. The probability of choosing a chromosome for mutation is 0.3 meaning that from a population with 50 chromosomes 1.5 ones are chosen for mutation.

3.4.Crossover

Crossover which is the other commonly used genetic operator, in which two offsprings are created. Some of the well-known crossover methods are uniform, one-point, and multi-point. It is selected to apply two-point crossover in which a randomly chosen segment of two parents is swapped between two parents for offspring creation. In the case of our algorithm, crossover is the only one phase and there is no need for later offspring allele corrections. The probability of selecting a chromosome among the current population for crossover is 0.8.

3.5. Selection

Selection is the genetic operator that determines the individuals selected for next generation. In this paper, we used roulette wheel selection which is one of the popular fitness proportionate selection methods. The selection is similar to the roulette wheel in casinos. The individual with higher fitness has more chance to survive to next generation. This method might choose one individual for several times.

3.6. Genetic Algorithm for Finding Potential Fields Parameters

Obstacle avoidance is one of the most common tasks that robots should be able to manage. In order to accomplish the task, there is a need for finding right parameter settings for PF approach. PF method has four parameters, each taking a real value number, as previously mentioned. The range which each parameter can take is limited and specified by some trial and error on each parameter. Although ranges are limited, still we are facing an almost infinite search space and it is not possible to test all values. This is what makes, GA to be a good choice for this problem. Even though it doesn't guarantee the best answer, it approximates an optimal solution.

In order to test and evaluate the solutions generated as chromosomes in GA we developed a simulating environment tool consisting of swarm of robots. For designing the environments, we used Robotic Operating System (ROS) and stage among with python's DEAP package.

The ROS is a flexible platform for writing robot software and simulators. It provides a range of tools and libraries that simplify the process of generating complex and powerful robot behaviours [13]. Stage which is used as a part of this simulating environment provides virtual world for mobile robots with wide range of sensors on them along with simulated objects to sense and manipulate [14].

Genetic part of this research is implemented in DEAP which is an evolutionary computation framework that allows rapid prototyping in python programming language [15].

4. Experimental Results

In order to evaluate and demonstrate the effectiveness of the proposed method, three scenarios are defined. Experiments were performed with an Intel Core 2 Quad machine running at 2.2 GHz, with three Gbytes of RAM memory in a Linux Mint Xfce edition operation system. ROS is employed to design the experimental setup. All scenarios are created in a stage that a mobile robot is allowed to navigate between predefined start and goal points while trying defeat different obstacles. A solution to our problem is defined by generating parameters of PF algorithm using GA algorithm in which chromosomes are generated and evaluated using fitness function. The evaluation process includes collision free path, smoothness and the length of the path that is travelled within the generated set of parameters. In this process, if a simulated robot of a solution hits an object (collision), or reaches a goal after a predefined time interval, the corresponding parameter set is marked as rejected by assigning a 1 value to its fitness property. The other solutions, achieving to lead the robot towards the goal by the time limit, are assigned a value between (1,0). The best solutions are assigned 0 in which a simulated robot completes the task in the shortest time interval. This is only achievable when a robot directly goes from a start point to the end point while not encountering any obstacles located during its path. Accordingly, four different scenarios are considered and analysed as follows.

4.1. Scenario 1

The first scenario illustrates an environment that has one obstacle located between the start and goal points, as shown in Figure 1. While Figure 1(a) shows the worst case scenario in which the robot collides with the obstacle, Figure 1(b) illustrates the navigation results of the robot using randomly generated parameters with trial-and-error method. On the other hand, Figure 1(c) simulates the navigation path of the robot whose parameters are calculated using the GA approach. As it is shown, this figure provides a smooth and short path. Also, Figure 2 shows the fitness value of the best individual in each generation for this scenario that the best solution is found in the 19th generation.

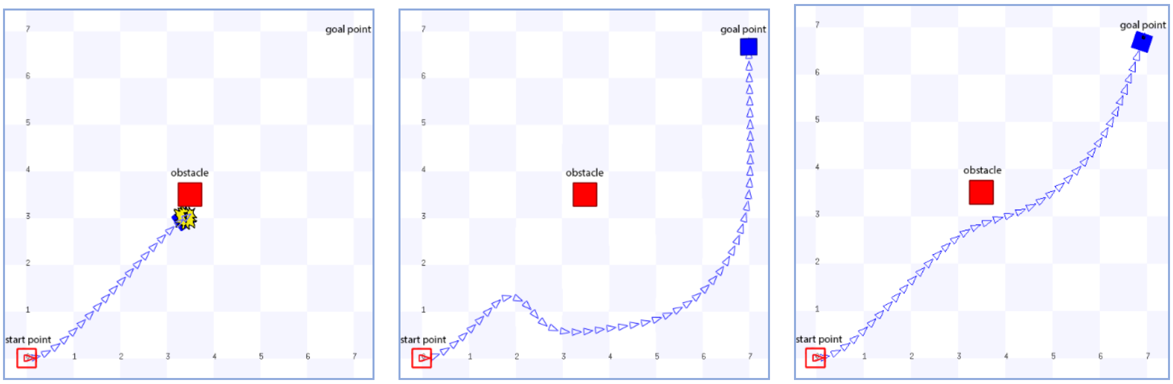

Figure 1. Solutions for scenario 1. (a) Worst solution, (b) random solution and (c) the best found solution

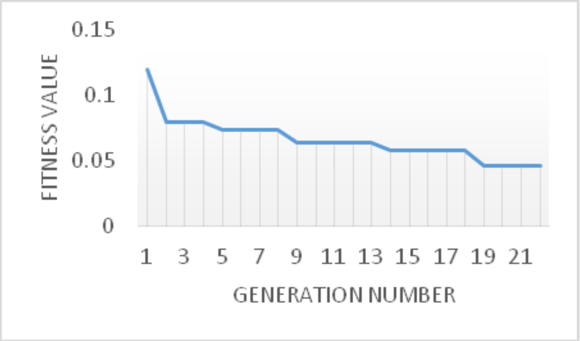

Figure 2. Fitness value generation graph for the first scenario

4.2. Scenario 2

This scenario includes three obstacles obscuring the path between the initial point and the goal points. The best result for this scenario is obtained at the 13th generation and shown in Figure 3. As it can be seen, this is a complex scenario for the PF algorithm to handle. Figure 3 illustrates that the algorithm finds the first acceptable solution in the fourth generation meaning that none of the individuals in these generations are even capable of guiding the robot to the goal point. The final resultant path has the shortest length and the smoothest shape (with less oscillation) created in all these generations. The values shown in the graph are the fitness value of the best individual in the corresponding generation (see Figure 4).

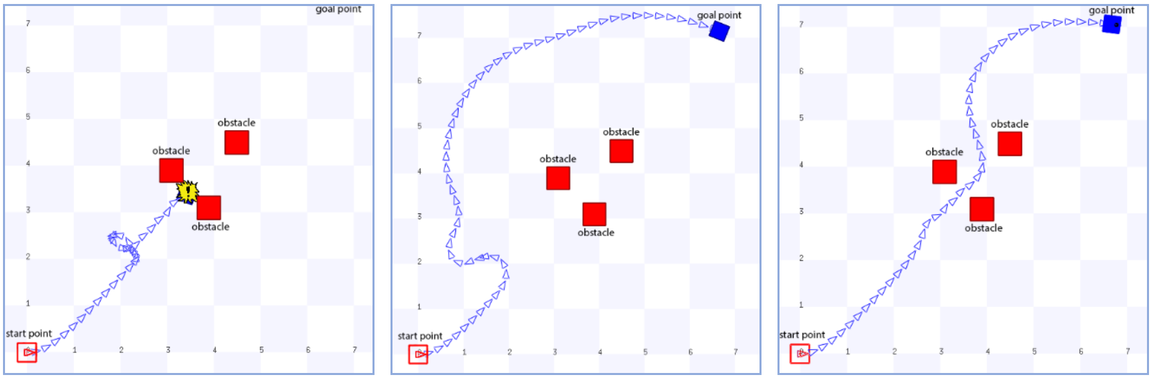

Figure 3. Solutions for the second scenario. (a) Worst solution, (b) random solution and (c) the best found solution

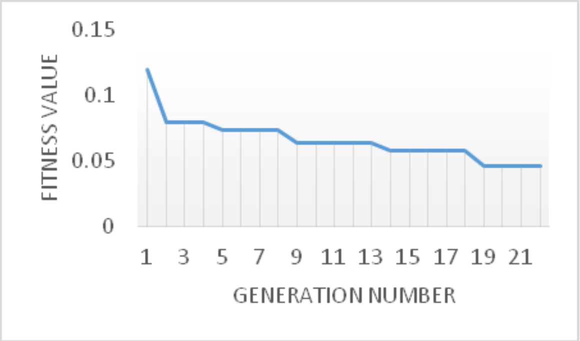

Figure 4. Fitness value generation graph for the second scenario

4.3. Scenario 3

This scenario is designed to simulate one of the toughest problems that are faced as PF algorithm is employed for robot navigation during a corridor. Mostly, when a robot tries to pass a corridor, it is oscillated due to the influence of opposite forces that slows down the robot significantly and also prevents smooth navigation behaviour, as shown in Figures 5(a) and 5(b). Figure 5(c) shows the best answer for the scenario in which the parameters are generated by employing GA algorithm. Accordingly, for this conventional problem of the PF algorithm, the GA is able to find an acceptable parameter set to overcome the corridor navigation problem. The best individual is found in the 18th generation (see Figure 6).

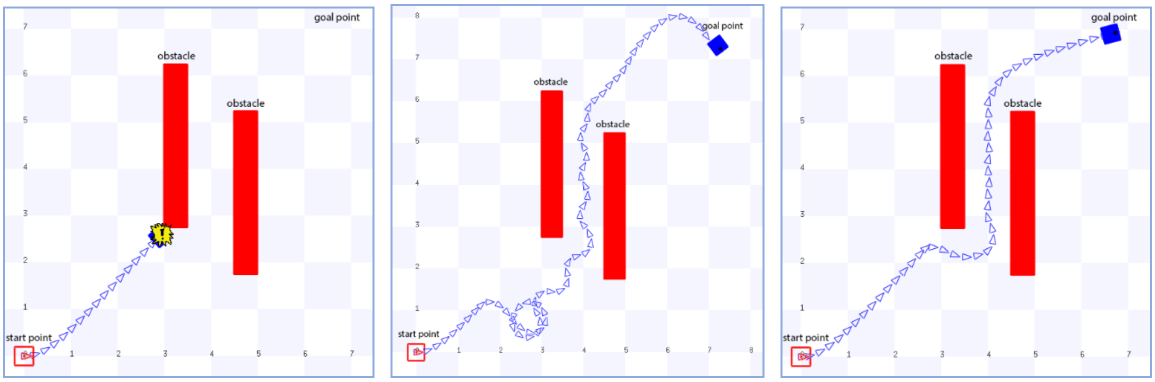

Figure 5. Solutions for the third scenario. (a) Worst solution, (b) random solution, and (c) the best found solution

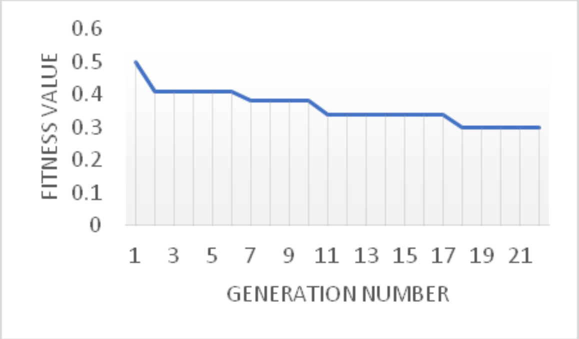

Figure 6. Fitness value generation graph for the third scenario

4.4. Scenario 4

This scenario illustrates another challenging corridor navigation problem that is also difficult for PF method to solve. According to that, Figure 7(a) shows a failed scenario, while Figures 7(b) and 7(c) illustrate a successful scenario, respectively. On the other hand, Figure 6(c) shows a shorter path whose parameters are obtained GA algorithm. The training phase of the GA algorithm to estimate appropriate parameters for this scenario is illustrated in Figure 8.

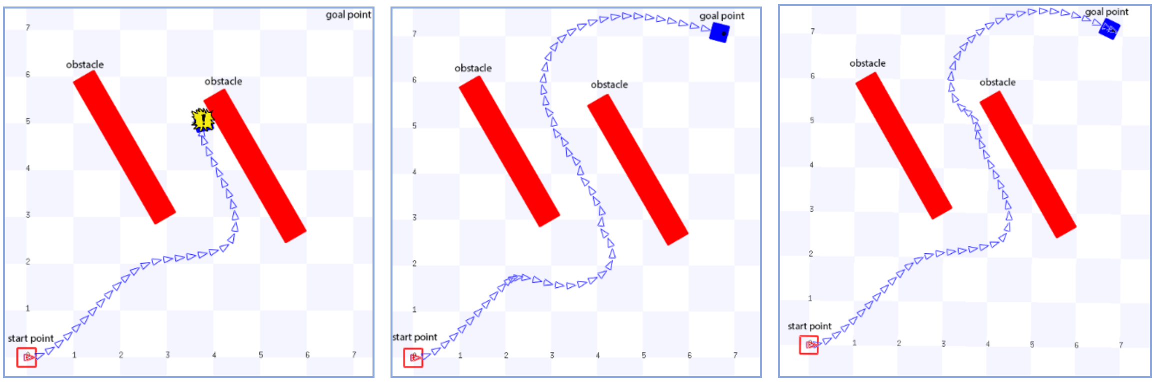

Figure 7. Solutions for the fourth scenario. (a) Worst solution, (b) random solution and (c) the best found solution

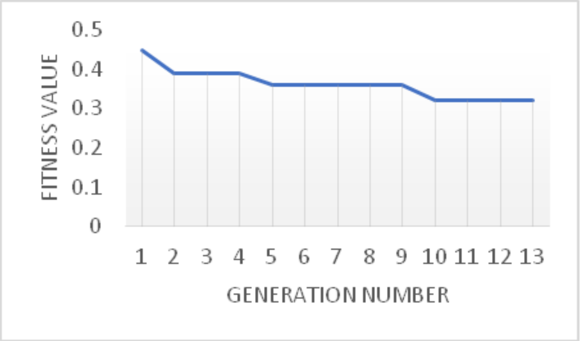

Figure 8. Fitness value generation graph for the fourth scenario

Overall, Table 1 illustrates all generated parameter sets obtained from GA algorithm for corresponding scenarios.

Table 1. List of estimated parameters with respect to given scenarios

Scenario	α	β	Goal spread	Obstacle spread
1	3.1261	2.2055	2.9602	2.7628
2	5.4693	6.5175	2.3422	2.3274
3	2.0231	7.4883	0.7151	0.5581
4	3.2753	6.0929	2.2476	1.7900

5. Conclusion

Having an appropriate parameter set is vital for the conventional PF approach, especially for the cluttered environments. This paper proposes a new GA-based solution to approximate the best parameter set used by PF method for overcoming different complex navigation scenarios. Estimation of the best possible parameters requires a detailed analysis of a huge search space that essentially proposes the GA as a suitable solution. Results verify the effectiveness of the GA approach for this problem. Besides, it is figured out that the GA-based parameter estimation approach allows the PF algorithm to overcome problems that are not easy to handle even using the alternative parameter estimation methods. This study proves that regardless of the complexity of the scenario, the GA-based approach can approximate best parameter set for the PF algorithm. This motivates authors to make further contribution to the field; therefore, future work will focus to develop generic solution, which can be used for all possible scenarios in different environments.

Acknowledgements

This study was supported by the Scientific and Technological Research Council of Turkey (TUBITAK Project No: 114E648).

References

[1] T. Weerakoon *et al.*, “Dead-lock free mobile robot navigation using modified artificial potential field,” in: *15th International Symposium 2014 Joint 7th International Conference on and Advanced Intelligent Systems*, 2014, pp. 259–264.

[2] P. Nattharith and M. S. Guzel, “Machine vision and fuzzy logic-based navigation control of a goal-oriented mobile robot,” *Adaptive Behavior*, vol. 24, issue 3, pp. 168–180, 2016.

[3] K. M. Krishna and P. K. Kalra, “Solving the local minima problem for a mobile robot by classification of spatio-temporal sensory sequences,” *J. Robot. Syst.*, vol. 17, issue 10, pp. 549–564, 2000.

[4] L. Tang *et al.*, “A novel potential field method for obstacle avoidance and path planning of mobile robot,” in: *IEEE International Conference on Computer Science and Information Technology*, 2010, pp. 633–637.

[5] S. Ge and Y. Cui, “New potential functions for mobile robot path planning,” *IEEE Trans. Robot. Autom.*, vol. 16, issue 5, pp. 615–620, 2000.

[6] O. Khatib, “Real-time obstacle avoidance for manipulators and mobile robots,” *Int. J. Rob. Res.*, vol. 5, issue 1, pp. 90–98, 1986.

[7] R. Arkin, *Behavior-based robotics*. 1998.

[8] R. R. Murphy, “Introduction to AI robotics,” *BJU Int.*, vol. 108, pp. 257–291, 2000.

[9] R. A. Brooks, *Cambrain intelligence: the early history of the new AI*. Cambridge: MIT Press, 1999.

[10] D. Whitley, “A genetic algorithm tutorial,” *Stat. Comput.*, vol. 4, issue 2, pp. 65–85, 1994.

[11] G. R. Harik *et al.*, “The compact genetic algorithm,” *IEEE Trans. Evol. Comput.*, vol. 3, issue 4, pp. 287–297, 1999.

[12] M. Mitchell, *An introduction to genetic algorithms*. Cambridge: MIT Press, 1999.

[13] Y. Chen *et al.*, “Motion planning implemented in ROS for mobile robot,” *Control Decis.*, 2017.

[14] R. Vaughan, “Massively multi-robot simulation in stage,” *Swarm Intell.*, vol. 2, issue 2–4, pp. 189–208, 2008.

[15] C. Gagn, “DEAP: evolutionary algorithms made easy,” *J. Mach. Learn. Res.*, vol. 13, pp. 2171–2175, 2012.