Robot Path Planning using Gray Wolf Optimizer

L.DOĞAN[1], U. YÜZGEÇ[2]

[1] Department of Computer Engineering, Bilecik Seyh Edebali University, Bilecik/Turkey
lokmandogan34@gmail.com

[2] Department of Computer Engineering, Bilecik Seyh Edebali University, Bilecik/Turkey
ugur.yuzgec@bilecik.edu.tr

***Abstract* - Path planning problem plays an important role in mobile robot works. The robotic systems use intelligence algorithms to plan the path of the robot from one point to the other point. The main goal of path planning is to find the allowable movements of a robot in an environment with obstacles. These motions involve a path free of collision from the start position to the target position. In this study, Gray Wolf Optimization (GWO) algorithm was adapted to solve robot path planning problem. GWO algorithm imitates the hunting behavior and social leadership of gray wolves in nature. The leadership hierarchy consists of four grey wolf groups: alpha, beta, delta, and omega wolves. This algorithm comprises hunting mechanism with three stages: searching for prey, encircling prey, and attacking prey. In the test simulations of the robot path planning, we used a map with three circular obstacles. GWO algorithm was adapted to this problem. While finding the candidate solutions in path planning, three coordinate points are used between start and target points. For each iteration, these coordinate points are updated by GWO algorithm. If the solution point is in the obstacle zone, then violation is added to the cost function. The performance of GWO algorithm was evaluated with those of meta-heuristic algorithms for solving the robot path planning problem. The results obtained by GWO algorithm show that the optimal path is found for used test map.**

***Keywords* - Robot Path Planning, Meta-Heuristic Algorithm, Gray Wolf Optimizer.**

I. INTRODUCTION

In last three decades, meta-heuristic algorithms have become very popular for the optimization problems. Meta-heuristic algorithms are inspired by the evolution concepts or the physics rules or the social behavior of swarms, flocks of animals in nature Meta-heuristic algorithms are classified into, physical based algorithms, evolutionary based algorithms swarm intelligence algorithms, bio-inspired algorithms and other nature-inspired algorithms [1][2]. In the physical based algorithms, solving the optimization problem begins with a single solution and it is updated by physical equations at each iteration. Tabu Search algorithm (TS) [3][4], Simulated Annealing algorithm (SA) [5][6] can be given as examples to physical based algorithms. Genetic algorithm (GA) [3][4] and Differential Evolution (DE) algorithm [9]–[11] are the well known examples of the evolutionary based meta-heuristic algorithms. Some of swarm intelligence algorithms include Particle Swarm Optimization (PSO) algorithm by Kennedy & Eberhart [12][13], Artificial Bee Colony (ABC) algorithm by Karaboga [14][15], Ant Colony Optimization (ACO) algorithm by Dorigo et al. [16][17] and Fish Swarm Algorithm (FSA) by Li et al. [18][19].

The bio-inspired algorithms mimics the activities of biological organisms. The most important examples of such algorithms are Artificial Immune algorithm (AI) [20][21] and Bacterial Foraging Optimization algorithm (BFO) [22]–[24]. Some of the other nature inspired meta-heuristic algorithms are Cuckoo Search Algorithm (CSA) [25][26], Firefly algorithm (FA) [27][28], Fruit Fly Optimization Algorithm (FOA) [29], Gravitational Search Algorithm (GSA) [30][31], Imperialist Competitive Algorithm (ICA) [32][33], Antlion Optimizer (ALO) [34][35], Dragonfly Optimization Algorithm (DOA) [36][37], Whale Optimization Algorithm (WOA) [38].

Gray Wolf Optimization (GWO) algorithm which was proposed by Mirjalili in 2014, imitates the hunting strategy and social leadership of gray wolves [39]. In this algorithm, gray wolves are classified into four levels according to the social hierarchy: alpha, beta, delta, and omega wolves. For example, an alpha wolf is a leader of wolf group, omega wolves are the grey wolves at the lowest level. In addition to the social leadership mechanism, gray wolf hunting strategy is another interesting mechanism of GWO algorithm. Although GWO algorithm is a new meta-heuristic algorithm, the studies about improvement and application on GWO can be found in the literature. Some of the studies are : a modified GWO algorithm based on complex-valued encoding [40], chaotic maps based GWO algorithm [41], a Levy flight-based GWO algorithm [42], optimal control of dc motor using GWO algorithm [43], hybrid maximum power point tracking (MPPT) algorithm with GWO algorithm [44], modified discrete grey wolf optimizer algorithm (MDGWO) for multilevel image thresholding [45].

Robot path planning problem for a mobile robot has been popular especially in the last decades and many approaches have been proposed for a robot in an area with a set of fixed obstacles. In this problem, main objective is to find collision-free trajectories for robots. Mobil robot should reach the target location as fast as possible and as short as possible distance between start and target locations [46]–[48]. The problem of path planning consists of the start point of the robot, the desired target point, the geometric description of the zone including the positions of the obstacles and boundaries of the zone.

In this study, the GWO algorithm is proposed to find the most suitable path from the starting point to the target point without touching any obstacle. To evaluate the performance of GWO algorithm, we used the zone including three circle obstacles with different radius. GWO algorithm was compared with well-known meta-heuristic algorithms, such as Differential Evolution (DE) algorithm, Particle Swarm Optimization (PSO) algorithm, Artificial Bee Colony (ABC) algorithm and Firefly Optimization Algorithm (FOA).

II. GRAY WOLF OPTIMIZER (GWO)

The Gray Wolf Optimizer (GWO) is based on the behaviors of hunting strategy and social hierarchy of gray wolves. According to the hierarchy of gray wolves, there are four groups, namely alpha, beta, delta, and omega wolves. The leader or dominant wolf is called alpha and alpha wolf follows the other wolves in the group. The alpha is best wolf in terms of managing the group. The second in the social hierarchy of wolf group is beta wolf. Beta helps the leader wolf (alpha) in many activities. Delta wolf has to submit to alpha and beta wolves, but it adjudges the omega wolves. In this group, there are scouts, guards, elders, hunters, and caretakers. Omega wolf is gray wolf at the lowest level [39].

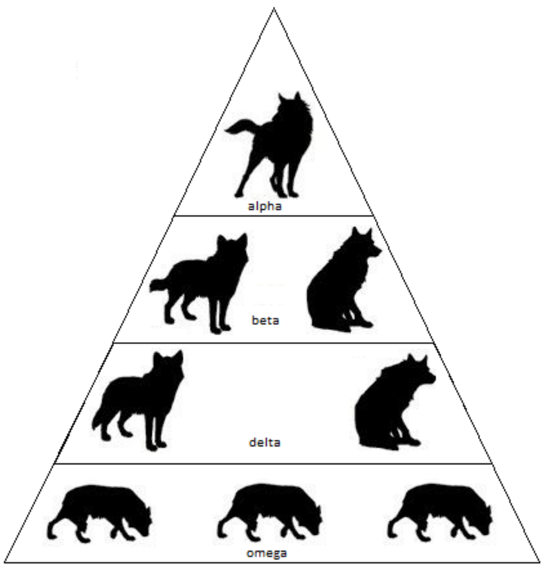

Figure 1: The hierarchy of gray wolves.

The group hunting strategy is another interesting social behavior of gray wolves. In this strategy of the gray wolves, firstly, they recognize the location of prey and encircle it under the leadership of the alpha wolf. In mathematical model of the hunting strategy of gray wolves, it is assumed that the alpha, beta and delta wolves provide better knowledge about the potential location of prey. As a result, the first three best solutions (alpha, beta, delta) are used to update the positions of wolves in GWO algorithm. There is no omega wolves in GWO code [39]. The mathematical model regarding hunting mechanism of gray wolves is given below:

$$\vec{D}_\alpha = \left|\vec{C}_\alpha \cdot \vec{X}_\alpha - \vec{X}_i\right| \quad (1)$$
$$\vec{D}_\beta = \left|\vec{C}_\beta \cdot \vec{X}_\beta - \vec{X}_i\right| \quad (2)$$
$$\vec{D}_\delta = \left|\vec{C}_\delta \cdot \vec{X}_\delta - \vec{X}_i\right| \quad (3)$$
$$\vec{U}_\alpha = \vec{X}_\alpha - \vec{A}_\alpha \vec{D}_\alpha \quad (4)$$
$$\vec{U}_\beta = \vec{X}_\beta - \vec{A}_\beta \vec{D}_\beta \quad (5)$$
$$\vec{U}_\delta = \vec{X}_\delta - \vec{A}_\delta \vec{D}_\delta \quad (6)$$

$$\vec{X}_i = \left(\vec{U}_\alpha + \vec{U}_\beta + \vec{U}_\delta\right)/3 \quad (7)$$

where $\vec{D}_\alpha, \vec{D}_\beta, \vec{D}_\delta$ are distance vector between prey and wolf (alpha, beta, delta), $\vec{X}_\alpha, \vec{X}_\beta, \vec{X}_\delta$ indicate the position vector of the prey for alpha, beta, delta wolves, $\vec{X}_i$ denotes the position vector of gray wolf at i_{th} iteration, $\vec{C}_\alpha, \vec{C}_\beta, \vec{C}_\delta, \vec{A}_\alpha, \vec{A}_\beta, \vec{A}_\delta$ indicate the coefficient vectors of alpha, beta, delta wolves, $\vec{U}_\alpha, \vec{U}_\beta, \vec{U}_\delta$ stand for the trial vector for alpha, beta, delta wolves. The coefficient vectors for alpha, beta and delta wolves are calculated as given below:

$$\vec{A}_\alpha = 2\vec{a}\vec{r}_{\alpha 1} - \vec{a} \quad (8)$$
$$\vec{C}_\alpha = 2\vec{r}_{\alpha 2} \quad (9)$$
$$\vec{A}_\beta = 2\vec{a}\vec{r}_{\beta 1} - \vec{a} \quad (10)$$
$$\vec{C}_\beta = 2\vec{r}_{\beta 2} \quad (11)$$
$$\vec{A}_\delta = 2\vec{a}\vec{r}_{\delta 1} - \vec{a} \quad (12)$$
$$\vec{C}_\delta = 2\vec{r}_{\delta 2} \quad (13)$$

where $\vec{a}$ indicates the vector linearly decreased from 2 to 0 during the optimization, $\vec{r}_{\alpha 1}, \vec{r}_{\beta 1}, \vec{r}_{\delta 1}$ denote the first random vector in [0,1] and $\vec{r}_{\alpha 2}, \vec{r}_{\beta 2}, \vec{r}_{\delta 2}$ denote the second random vector in [0,1].

The hunting mechanism of gray wolf group is illustrated in Fig. 2. The members of gray wolf group update their positions according to the alpha, beta, delta wolves and prey. The gray wolves catch their prey and finish the hunt by attacking the prey. In mathematical model, this situation is defined as decreasing $\vec{a}$ vector given below:

$$\vec{a} = 2 - \frac{2 \cdot Iter}{MaxIt} \quad (14)$$

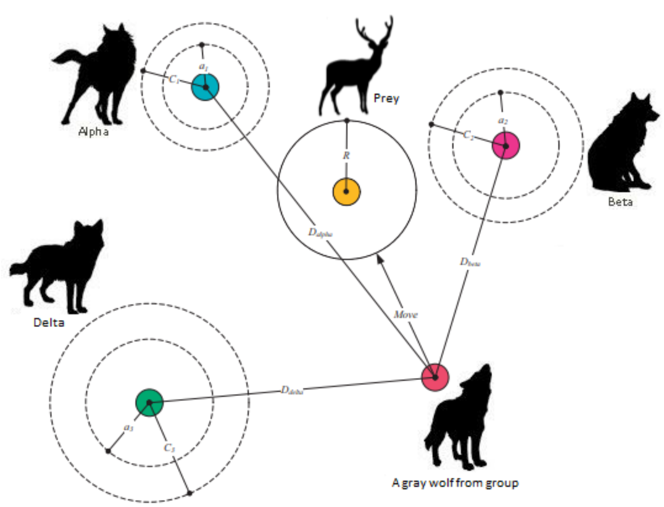

Figure 2: The hunting strategy of gray wolves.

The pseudo code of GWO algorithm is given in Algorithm 1.

Algorithm 1: Pseudo code of GWO algorithm.

Initialize the positions of gray wolves
Calculate the cost values of gray wolves
Save the best gray wolf as alpha wolf
Save the second best gray wolf as beta wolf
Save the third best gray wolf as delta wolf
while (*iteration* < *maximum iteration*)
Decrease $\vec{a}$ using Eq. (14)
for each gray wolf
Generate the coefficient vectors for alpha, beta, delta
Calculate the distance vectors using Eqs. (1-3)
Calculate the trial vectors using Eqs. (4-6)
Update the position of gray wolf using Eq. (7)
end for
Calculate the cost values of updated gray wolves
for each gray wolf
if (gray wolf < alpha wolf)
update alpha wolf
else if (gray wolf < beta wolf)
update beta wolf
else if (gray wolf < delta wolf)
update delta wolf
end if
end for
Update the elite antlion
increase iteration one
end while
return alpha wolf

III. ROBOT PATH PLANNING

The robot path planning problem is a NP-hard optimization problem and this problem is often solved by meta-heuristic algorithms in the literature. The main aim in solving this problem is that the mobile robot should reach from the start point to the target position in the shortest path without touching any obstacles. It consists of the start and target positions, the size of obstacles, the shape of obstacles, the number of obstacles, the zone's boundaries. The objective function of path planning problem is given below:

$$J = \min_{x,y} Q(1 + \beta V) \quad (15)$$

where β is violation coefficient (100), V indicates the violation cost, Q denotes the total distance between start and target points. In calculating the violation for the candidate solution, the following pseudo code was used.

Algorithm 2: Pseudo code of violation's calculation.

Violation ← 0
for each obstacle
Calculate distance vector between the obstacle's center and path
a ← max $(1 - \frac{distance}{radius_{obs}}, 0)$
Violation ← Violation + mean (a)
end for

IV. EXPERIMENTAL RESULTS

To show the performance of GWO algorithm for path planning problem, we have taken an instance scenario from www.yarpiz.com web site [49]. In Fig. 3, this instance scenario is shown. There are three circle-shaped obstacles with different radius in a 6x6 zone. The yellow square indicates the start point of the mobile robot and the green square indicates the target point. We have solved this problem using GWO algorithm and its performance has been compared with the several well-known meta-heuristic algorithms, such as Differential Evolution (DE) algorithm, Particle Swarm Optimization (PSO) algorithm, Artificial Bee Colony (ABC) algorithm and Firefly Optimization Algorithm (FOA).

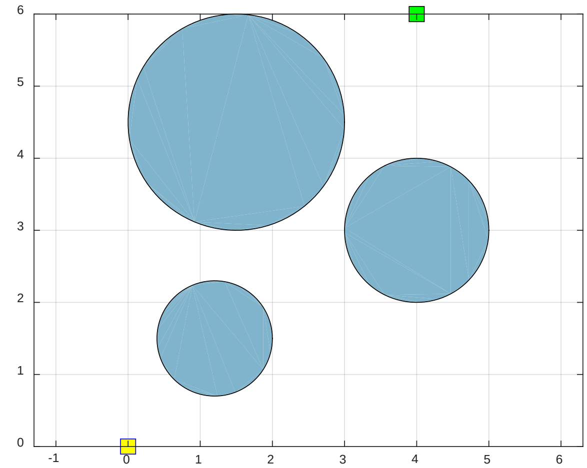

Figure 3: Path planning problem used in this study.

The codes of GWO and other meta-heuristic algorithms have been run on PC with Intel(R) Core(TM) i5-3230M CPU@2.60GHz RAM/8. Population size is 50, maximum number of iterations is 1000. The parameters of meta-heuristic algorithms used for robot path planning problem are summarized in Table 1. Fig.4 shows the best path planning solution obtained at the end of one-time run by the GWO and other meta-heuristic algorithms. As can be seen from this figure, the results of all algorithms are quite close together.

Table 1: Parameters of meta-heuristic algorithms.

Algorithm	Parameters
DE	Lower Bound of Scaling Factor : 0.5 Upper Bound of Scaling Factor : 1.0 Crossover Probability : 0.7 Strategy : rand2bin
PSO	Inertia Weight : 1.0 Inertia Weight Damping Ratio : 0.99 Personal Learning Coefficient : 1.5 Global Learning Coefficient : 2.0
ABC	Number of Onlooker Bees : 50 Abandonment Limit Parameter : round(0.6*NumberOfVar*PopSize)
FOA	Light Absorption Coefficient : 1.0 Initial Attraction Coefficient: 2.0 Mutation Coefficient : 0.2 Mutation Coefficient Damping R. : 0.98
GWO	Number of Antlions : 50

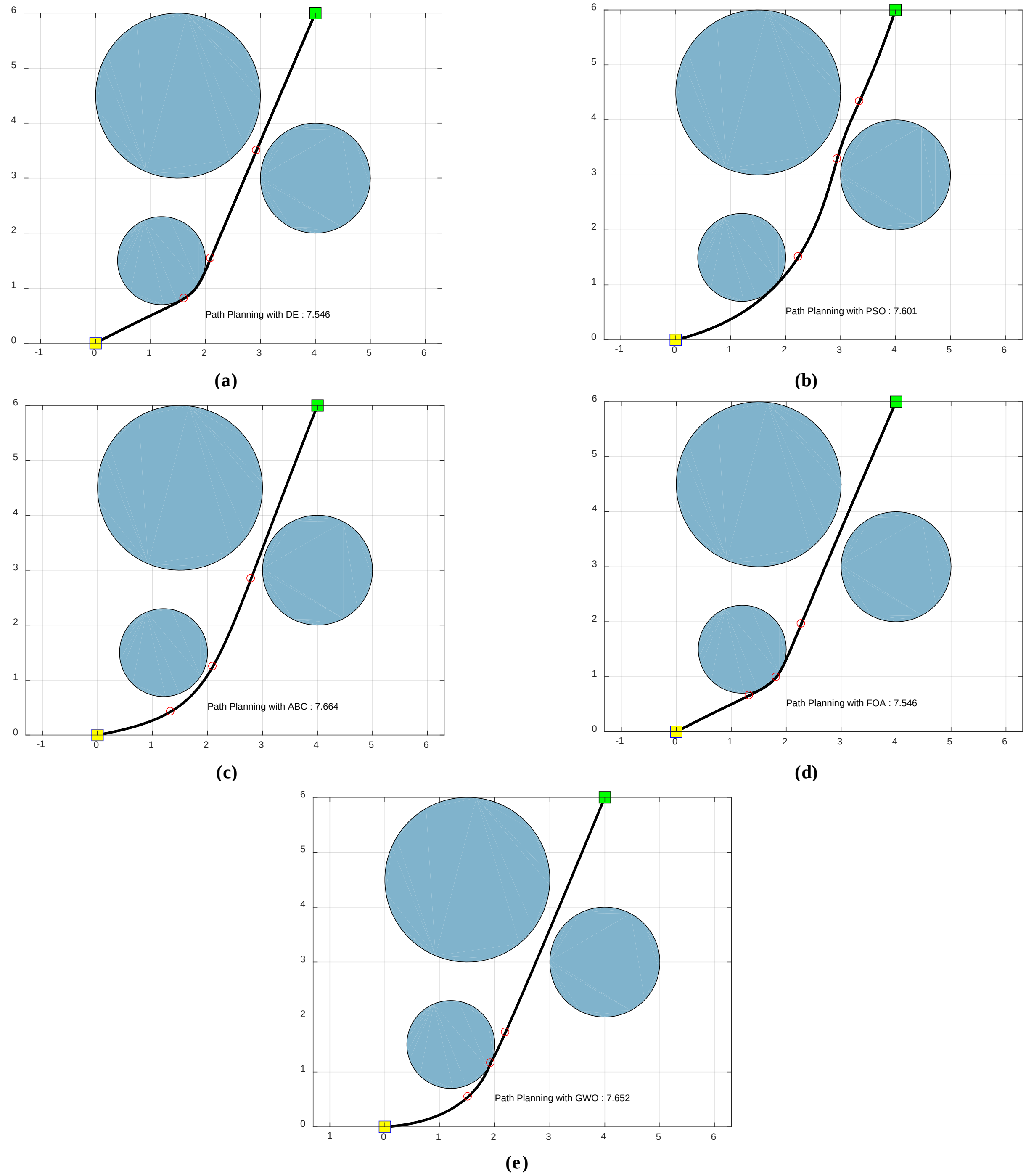

Figure 4: The best solutions obtained by meta-heuristic algorithms, (a) DE, (b) PSO, (c) ABC, (d) FOA, (e) GWO.

The cost value of GWO algorithm is found as 7.652. Fig. 5 shows all solutions of the path planning problem obtained by GWO algorithm. According to this figure, GWO algorithm deals with finding the suitable path with the minimum distance between start and target locations during optimization. Moreover, the paths found by the best current solution at each iteration have very little violation. The convergence curves of GWO and the other algorithms are shown in Fig.6. This figure show that the performance of GWO provides the competitive result and it can be an alternative algorithm for path planning.

V. CONCLUSION

In this study, robot path planning problem was discussed and GWO algorithm was proposed for solving this problem. To evaluate the algorithm's performance on solving path planning problem, four well-known meta-heuristic algorithms (DE, PSO, ABC and FOA) were used. The comparison results show that the proposed GWO algorithm is able to provide very competitive results. In future, we will add opposition learning to the GWO algorithm to increase its performance.

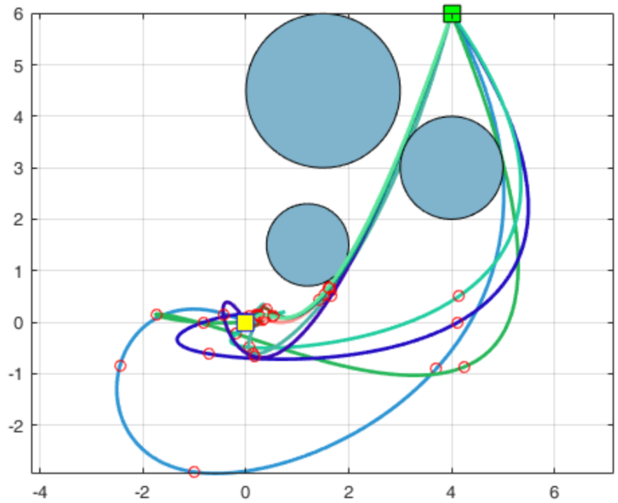

Figure 5: Solutions of path planning problem using GWO algorithm for all iterations.

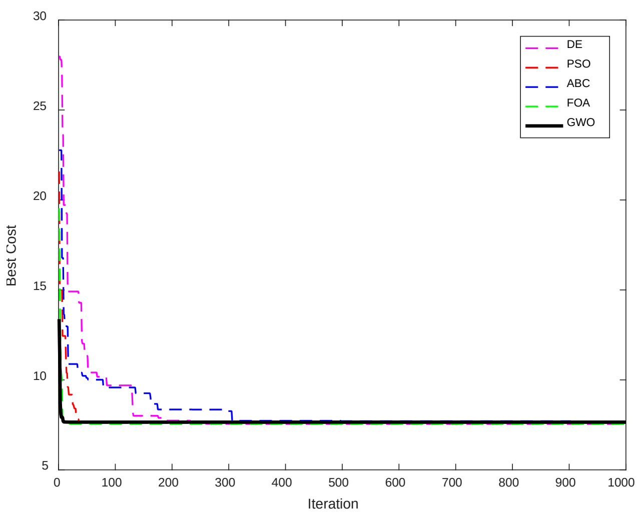

Figure 6: Comparison results of GWO and other meta-heuristic algorithms for robot path planning problem.

REFERENCES

[1] Z. Beheshti and S. M. H. Shamsuddin, "A review of population-based meta-heuristic algorithm," *International Journal of Advances in Soft Computing and its Applications*, vol. 5, no. 1. 2013.

[2] C. Blum, J. Puchinger, G. R. Raidl, and A. Roli, "A brief survey on hybrid metaheuristics," in *Bioinspired Optimization Methods and their Applications - Proceedings of the 4th International Conference on Bioinspired Optimization Methods and their Applications, BIOMA 2010*, 2010, pp. 3–16.

[3] E. Nowicki and C. Smutnicki, "A fast tabu search algorithm for the permutation flow-shop problem," *Eur. J. Oper. Res.*, vol. 91, no. 1, pp. 160–175, 1996.

[4] M. Gendreau, M. Iori, G. Laporte, and S. Martello, "A Tabu Search Algorithm for a Routing and Container Loading Problem," *Transp. Sci.*, vol. 40, no. 3, pp. 342–350, 2006.

[5] S. Kirkpatrick, C. D. Gelatt, and M. P. Vecchi, "Optimization by Simulated Annealing," *Science (80-.).*, vol. 220, no. 4598, pp. 671–680, 1983.

[6] T. D. Mavridou and P. M. Pardalos, "Simulated Annealing and Genetic Algorithms for the Facility Layout Problem: A Survey," *Comput. Optim. Appl.*, vol. 7, pp. 111–126, 1997.

[7] D. E. Goldberg, *Genetic Algorithms in Search, Optimization, and Machine Learning*, vol. Addison-We. 1989.

[8] M. Noraini and J. Geraghty, "Genetic algorithm performance with different selection strategies in solving TSP," *World Congr. Eng.*, vol. II, no. 978-988-19251-4-5, pp. 4–9, 2011.

[9] R. Storn and K. Price, "Differential Evolution – A Simple and Efficient Heuristic for global Optimization over Continuous Spaces," *J. Glob. Optim.*, vol. 11, no. 4, pp. 341–359, 1997.

[10] K. V Price, R. M. Storn, and J. A. Lampinen, *Differential Evolution: A Practical Approach to Global Optimization*, vol. 28. 2005.

[11] U. Yüzgeç, "Performance comparison of differential evolution techniques on optimization of feeding profile for an industrial scale baker's yeast fermentation process," *ISA Trans.*, vol. 49, no. 1, pp. 167–176, 2010.

[12] J. Kennedy and R. Eberhart, "Particle swarm optimization," *Neural Networks, 1995. Proceedings., IEEE Int. Conf.*, vol. 4, pp. 1942–1948 vol.4, 1995.

[13] M. Clerc and J. Kennedy, "The particle swarm - explosion, stability, and convergence in a multidimensional complex space," *IEEE Trans. Evol. Comput.*, vol. 6, no. 1, pp. 58–73, 2002.

[14] D. Karaboga, "An idea based on Honey Bee Swarm for Numerical Optimization," *Tech. Rep. TR06, Erciyes Univ.*, no. TR06, p. 10, 2005.

[15] D. Karaboga and B. Akay, "A comparative study of Artificial Bee Colony algorithm," *Appl. Math. Comput.*, vol. 214, no. 1, pp. 108–132, 2009.

[16] M. Dorigo, V. Maniezzo, and A. Colorni, "Ant system: Optimization by a colony of cooperating agents," *IEEE Trans. Syst. Man, Cybern. Part B Cybern.*, vol. 26, no. 1, pp. 29–41, 1996.

[17] M. Dorigo, M. Birattari, and T. Stutzle, "Ant colony optimization," *IEEE Comput. Intell. Mag.*, vol. 1, no. 4, pp. 28–39, 2006.

[18] X. Li, Z. Shao, and J. Qian, "An optimizing method based on autonomous animats: fish-swarm algorithm," *Syst. Eng. Theory Pract.*, vol. 22, no. 11, pp. 32–38, 2002.

[19] M. Neshat, G. Sepidnam, M. Sargolzaei, and A. N. Toosi, "Artificial fish swarm algorithm: a survey of the state-of-the-art, hybridization, combinatorial and indicative applications," *Artif. Intell. Rev.*, vol. 42, no. 4, pp. 965–997, 2014.

[20] S. A. Hofmeyr and S. Forrest, "Architecture for an Artificial Immune System," *Evol. Comput.*, vol. 8, no. 4, pp. 443–473, 2000.

[21] J. Timmis, P. Andrews, and E. Hart, "On artificial immune systems and swarm intelligence," *Swarm Intell.*, vol. 4, no. 4, pp. 247–273, 2010.

[22] S. Das, A. Biswas, S. Dasgupta, and A. Abraham, "Bacterial Foraging Optimization Algorithm : Theoretical Foundations , Analysis , and Applications," *Found. Comput. Intell. Vol. 3*, vol. 3, pp. 23–55, 2009.

[23] K. M. Passino, "Biomimicry of bacterial foraging for distributed optimization and control," *Control Syst. IEEE*, vol. 22, no. 3, pp. 52–67, 2002.

[24] K. M. Passino, "Bacterial Foraging Optimization," *Int. J. Swarm Intell. Res.*, vol. 1, no. 1, pp. 1–16, 2010.

[25] X. S. Yang and S. Deb, "Cuckoo search: Recent advances and applications," *Neural Computing and Applications*, vol. 24, no. 1. pp. 169–174, 2014.

[26] X.-S. Yang, "Cuckoo Search via Lévy flights," in *2009 World Congress on Nature & Biologically Inspired Computing (NaBIC)*, 2009, pp. 210–214.

[27] X. S. Yang, "Firefly Algorithm," *Nature-Inspired Metaheuristic Algorithms*, pp. 79–90, 2007.

[28] X. S. Yang, "Firefly algorithms for multimodal optimization," in *Lecture Notes in Computer Science (including subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics)*, 2009, vol. 5792 LNCS, pp. 169–178.

[29] H. Iscan and M. Gunduz, "Parameter Analysis on Fruit Fly Optimization Algorithm," *J. Comput. Commun.*, vol. 2, no. March, pp. 137–141, 2014.

[30] E. Rashedi, H. Nezamabadi-pour, and S. Saryazdi, "GSA: A Gravitational Search Algorithm," *Inf. Sci. (Ny).*, vol. 179, no. 13, pp. 2232–2248, 2009.

[31] N. M. Sabri, M. Puteh, and M. R. Mahmood, "A review of gravitational search algorithm," *Int. J. Adv. Soft Comput. its Appl.*, vol. 5, no. 3, 2013.

[32] E. Atashpaz-Gargari and C. Lucas, "Imperialist competitive

algorithm: An algorithm for optimization inspired by imperialistic competition," in *2007 IEEE Congress on Evolutionary Computation, CEC 2007*, 2007, pp. 4661–4667.

[33] S. Hosseini and A. Al Khaled, "A survey on the Imperialist Competitive Algorithm metaheuristic: Implementation in engineering domain and directions for future research," *Applied Soft Computing Journal*, vol. 24. pp. 1078–1094, 2014.

[34] S. Mirjalili, "The ant lion optimizer," *Adv. Eng. Softw.*, vol. 83, pp. 80–98, 2015.

[35] H. Kiliç and U. Yüzgeç, "Improved antlion optimization algorithm," in *2nd International Conference on Computer Science and Engineering, UBMK 2017*, 2017.

[36] S. Mirjalili, "Dragonfly algorithm: a new meta-heuristic optimization technique for solving single-objective, discrete, and multi-objective problems," *Neural Comput. Appl.*, vol. 27, no. 4, pp. 1053–1073, 2016.

[37] S. R. Sree Ranjini and S. Murugan, "Memory based Hybrid Dragonfly Algorithm for numerical optimization problems," *Expert Syst. Appl.*, vol. 83, pp. 63–78, 2017.

[38] S. Mirjalili and A. Lewis, "The Whale Optimization Algorithm," *Adv. Eng. Softw.*, vol. 95, pp. 51–67, 2016.

[39] S. Mirjalili, S. M. Mirjalili, and A. Lewis, "Grey Wolf Optimizer," *Adv. Eng. Softw.*, vol. 69, pp. 46–61, 2014.

[40] Q. Luo, S. Zhang, Z. Li, and Y. Zhou, "A Novel Complex-Valued Encoding Grey Wolf Optimization Algorithm," *Algorithms*, vol. 9, no. 1, p. 4, 2015.

[41] M. Kohli and S. Arora, "Chaotic grey wolf optimization algorithm for constrained optimization problems," *J. Comput. Des. Eng.*, 2017.

[42] S. Amirsadri, S. J. Mousavirad, and H. Ebrahimpour-Komleh, "A Levy flight-based grey wolf optimizer combined with back-propagation algorithm for neural network training," *Neural Computing and Applications*, pp. 1–14, 2017.

[43] A. Madadi and M. M. Motlagh, "Optimal Control of DC motor using Grey Wolf Optimizer Algorithm," *Tech. J. Eng. Appl.*, pp. 373–379, 2014.

[44] S. Mohanty, B. Subudhi, and P. K. Ray, "A Grey Wolf-Assisted Perturb & Observe MPPT Algorithm for a PV System," *IEEE Trans. Energy Convers.*, vol. 32, no. 1, pp. 340–347, 2017.

[45] L. Li, L. Sun, J. Guo, J. Qi, B. Xu, and S. Li, "Modified Discrete Grey Wolf Optimizer Algorithm for Multilevel Image Thresholding," *Comput. Intell. Neurosci.*, vol. 2017, 2017.

[46] C. Alexopoulos and P. M. Griffin, "Path planning for a mobile robot," *IEEE Trans. Syst. Man. Cybern.*, vol. 22, no. 2, pp. 1132–1139, 1992.

[47] N. Sariff and N. Buniyamin, "An overview of autonomous mobile robot path planning algorithms," in *Research and Development, 2006. SCOReD 2006. 4th Student Conference on*, 2006, pp. 183–188.

[48] C. Hofner and G. Schmidt, "Path planning and guidance techniques for an autonomous mobile cleaning robot," *Rob. Auton. Syst.*, vol. 14, no. 2–3, pp. 199–212, 1995.

[49] Yarpiz, Optimal Robot Path Planning using PSO. http://yarpiz.com/403/ypap115-path-planning, Accessed 28 March 2018.