Improved Path Planning of Mobile Robot using Atom Search Optimization Algorithm

Shahad M.Majeed[1], Issa Ahmed Abed[2], and Alaa A. Al-Saffar[3]

shahadmohammed@stu.edu.iq[1], issaahmedabed@stu.edu.iq[2], alaaasaffar@stu.edu.iq[3]

Basrah Engineering Technical College, Southern Technical University, Iraq

Abstract. Robots have found applications in a variety of settings, ranging from speeding up manufacturing processes for completing complex jobs in hazardous and hostile areas. Path planning for mobile robots is one of the most difficult challenges in robotics. Robot route planning is a difficult optimization issue that must be tackled in a variety of applications. In this paper, we proposed the Atom search optimization (ASO) algorithm to improve the mobile robot path with free, static and dynamic obstacles environments and compare the results with the particle swarm optimization (PSO) algorithm. The "atom search optimization" algorithm is a novel algorithm that was used for the first time in this study to plan the route of a mobile robot. This algorithm has improved the path and the execution time(time required for the robot to reach the destination) was reduced. The authors checked the proposed method with several cases using MATLAB2020a software. It has been demonstrated that the suggested strategy, rather than PSO, found better, more consistent, and smoother pathways with a shorter computation time.

Keywords. Mobile robot route planning, Atom search optimization algorithm, particle swarm optimization.

1 Introduction

Rapid technological advancements open up new and better prospects in a variety of areas. Robotics is now one of the most exciting developments [1]. The primary aim of robotic technology is to develop machines that can perform such human tasks. They are commonly used in a variety of circumstances to speed up a process or work in and exploring aggressive, hazardous, and inaccessible environments [1]. Forward kinematics and, later, inverse kinematics is necessary for the design of any work in robotics systems [2]. In this work Forward kinematics is considered. One issue is how the mobile robot determines the shortest path to the target while avoiding collisions. In mobile robot work, the path planning issue is crucial. To plan the course of the robot from one point to another, robotic systems employ intelligence algorithms. Path planning's main objective is to determine a robot's permissible movement in a given setting with obstacles. From the start position to the target position, these motions require a collision-free direction [2]. A smart mobile robot route planning system can save a lot of time while also lowering the robot's wear and capital expenditure

[3]. Researchers all around the world have developed a variety of meta-heuristic search algorithms to address complicated global optimization issues and approaches to improve them during the last few decades [5]. A constrained optimization problem can be used to overcome the difficulty of route planning. Depending on the environment and tasks that the robot must do, the goal or fitness function of the problem can be presented in a variety of ways [1].

In this paper, For tackling mobile robot path planning in two-dimensional space with static and dynamic impediments, the (ASO) technique was introduced. The results are compared with a PSO algorithm in the same manner. The suggested technique is a novel type of atom dynamics-based meta-heuristic global optimization methodology. ASO is a population-based iterative heuristic global optimization algorithm that that can be applied to a wide range of optimization issues. ASO is a simple and straightforward method that mathematically describes and replicates nature's atomic motion concept, where atom interaction forces emerging from the Lennard-Jones potential and constraint forces arising from the bond-length potential interact with each other. To investigate the problem of mobile robot path planning, the ASO method is utilized. In path planning issues, The findings of the experiments indicate that ASO can outperform other well-known techniques such as (PSO) and that it is competitive with its rivals..

2 Robot path planning

The fundamental purpose of robot route planning using optimization criteria is to identify a path from start to finish that is free of collisions [5]. In its most basic form, path planning attempts to build a collision-free path for the robot from its starting place to the destination location while staying away from the strewn about the workspace, using some knowledge about the environment [6]. There are some assumptions made in this work before exploring and proposing a solution to this issue:

• The proposed environment was designed with 200 x 200 pixels.
• The proposed environment was designed with three scenarios free, static &dynamic obstacles.
• The obstacles are represented by an equally size circular shape.
• The speed of the dynamic obstacles was fixed.

• The mobile robot movement was omnidirectional at any time.
• Minimizing the distance function yields the shortest distance. $f(x, y)$[7].

$$f(x,y) = \sqrt{(x_d - x_s)^2 + (y_d - y_s)^2} \tag{1}$$

where: The next position is represented by x_d, y_d , and the current position is represented by x_s, y_s.
• The shortest path planning was obtained by applying PSO &ASO algorithms, then compared the results between them.

3 Atom Search Optimization

ASO approach is fundamental molecular dynamics. The notion of molecular dynamics was developed in theoretical physics and has since been applied to domains like chemistry, biology, and materials science; where atomic motion is governed by classical mechanics. Newton's second law is that, if F_i is the interaction force and G_i is the constraint force acting on the ith atom and the atom has mass m_i then the acceleration of the atom is [8], [4]:

$$a_i = \frac{F_i + G_i}{m_i} \tag{2}$$

The interaction force operating on the Lennard-Jones potential (L-J potential) is employed on i^{th} atom from the jth atom in the dth dimension at t time, which can be represented as[8], [4]:

$$F_{ij}^d(t) = \frac{24\varepsilon(t)}{\sigma(t)}\left[2\left(\frac{\sigma(t)}{r_{ij}(t)}\right)^{13} - \left(\frac{\sigma(t)}{r_{ij}(t)}\right)^{7}\right]\frac{r_{ij}(t)}{r_{ij}^d(t)} \tag{3}$$

And

$$F_{ij}'(t) = \frac{24\varepsilon(t)}{\sigma(t)}\left[2\left(\frac{\sigma(t)}{r_{ij}(t)}\right)^{13} - \left(\frac{\sigma(t)}{r_{ij}(t)}\right)^{7}\right] \tag{4}$$
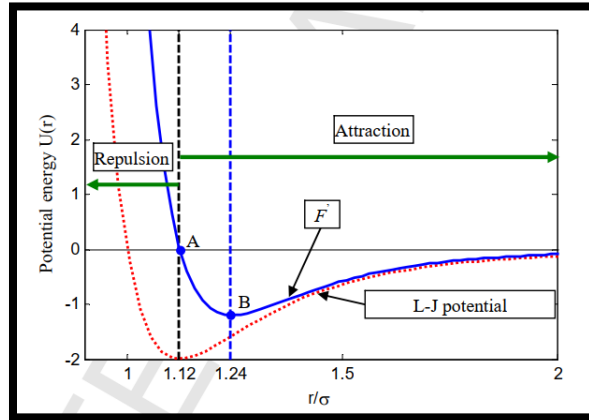

Fig. 1. Atoms force curve

Figure 1 depicts the atoms' force curve for molecular dynamics [4]. It can be demonstrated that due to repulsion or attraction, atoms retain a constant relative distance that fluctuates within a given range, and the repulsion shift relative to the equilibration distance ($r=1.12\sigma$) is substantially larger than the attractiveness. As a result, the repulsion is positive, while the attraction is negative. thus,

atoms will not converge to a single spot. As a result, equation (4) cannot be directly used to the solution of optimization problems. This equation has been updated to the following as: [8], [4]:

$$F'_{ij}(t) = -\eta(t)\left[2\left(h_{ij}(t)\right)^{13} - \left(h_{ij}(t)\right)^{7}\right] \tag{5}$$

Where the depth function is $\eta(t)$ that can be used to modify the repulsion or attraction zone, defined as [8], [4]:

$$\eta(t) = \alpha\left(1 - \frac{t-1}{T}\right)^{3} e^{-\frac{20t}{T}} \tag{6}$$

Where the maximum number of iterations is T and a is the depth weight and the function behaviors of F' with different values of η corresponding to the values of h ranging from 0.9 to 2 are depicted in Figure 2.
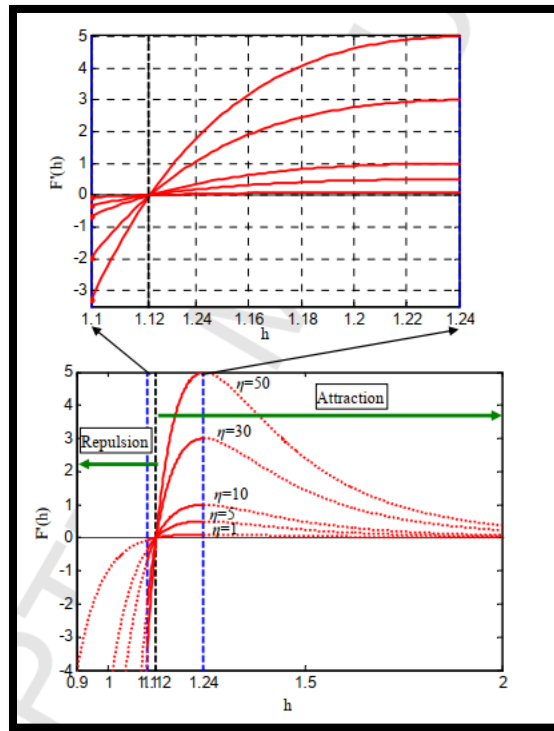

Fig. 2. F' function behavior with various values of η

According to Figure 2, repulsion develops when h values are in a certain range of 0.9 to 1.12, attraction takes place when h values range from "1.12 to 2".The equilibration takes place when h values range from 1.12 to 2. The attraction steadily grows h from equilibration " h=1.12 " to maximum " h=1.24 " and then starts to reduce. If h is greater than or equal to 2, the attraction is about equivalent to zero. As a result, in ASO, to encourage exploration, with a smaller function value, the lower limit of repulsion is set to h= 1.1, while with a bigger function value, the higher limit of attraction is set to h= 1.2. With a larger function value, the upper limit of attraction is set at "h =2.4", therefore h is expressed as [4]:

$$h_{ij}(t) = \begin{cases} h_{min} \text{ if } \frac{r_{ij}(t)}{\sigma(t)} < h_{min} \\ \frac{r_{ij}(t)}{\sigma(t)} \text{ if } h_{min} \le \frac{r_{ij}(t)}{\sigma(t)} \le h_{max} \\ h_{max} \text{ if } \frac{r_{ij}(t)}{\sigma(t)} > h_{max} \end{cases} \tag{7}$$

h_{max} and h_{min} are the upper and lower limits of h_{ij} respectively. The length scale $\sigma(t)$ is described as follows [8], [4]:

$$\sigma(t) = \left\| x_{ij}(t), \frac{\sum_{j \in KBest} x_{ij}(t)}{K(t)} \right\| \tag{8}$$

Where: K_{Best} denotes the subset of K atoms with the best function fitness values [8], [4].

$$\begin{cases} h_{min} = g_0 + g(t) \\ h_{max} = u \end{cases} \tag{9}$$

where u is the upper limit and g is a drift function that aids the algorithm in drifting from exploration to exploitation [8], [4]:

$$g(t) = 0.1 \times \sin\left(\frac{\pi}{2} \times \frac{t}{T}\right) \tag{10}$$

The weighted sum of components of the forces in the dth dimension is now the total force acting on the ith atom from all other atoms. [8], [4]:

$$F_i^d(t) = \sum_{j \in KBest} \text{random}_j F_{ij}^d(t) \tag{11}$$

where random j is a number in the range [0,1] [8], [4].

The geometric restriction plays a crucial role in atom motion in molecular dynamics. For the sake of simplicity, each atom in ASO is assumed to have a covalent link with the best atom, the best atom

exerts a restriction force on each atom. The constraint force operating on the ith atom in the dth dimension is defined as [4]:

$$G_i^d = \lambda(t)\left(x_{best}^d(t) - x_i^d(t)\right) \tag{12}$$

where, the Lagrangian multiplier is $\lambda(t)$, which is described as:

$$\lambda(t) = \beta e^{-\frac{20t}{T}} \tag{13}$$

Where: the multiplier weight is β.

As a result, the ith acceleration of an atom at time t can be expressed as [8]:

$$a_i^d(t) = \frac{F_i^d(t) + G_i^d(t)}{m_i}\alpha\left(1 - -\frac{t-1}{T}\right)^3 e^{\frac{-20t}{T}}$$

$$* \sum_{j \in KBest} \frac{random_j\left[2\left(h_{ij}(t)\right)^{13} - \left(h_{ij}(t)\right)^7\right]}{m_i(t)}$$

$$\frac{\left(x_j^d(t) - x_i^d(t)\right)}{\|x_i(t), x_j(t)\|_2} \tag{14}$$

$$+ \beta e^{-\frac{20t}{T}}\frac{x_{best}^d(t) - x_i^d(t)}{m_i(t)}$$

At the tth repetition, the mass of the ith atom is $m_i(t)$. It is determined as follows [8]::

$$M_i(t) = e^{-\frac{Fit_i(t) - Fit_{best}(t)}{Fit_{worst}(t) - Fit_{best}(t)}} \tag{15}$$

$$m_i(t) = \frac{M_i(t)}{\sum_{j=1}^{N} M_j(t)} \tag{16}$$

Where: $Fit_{best}(t) = \min_{i=\{1,2,...,n\}} Fit_i(t)$ and $Fit_{worst}(t) = \max_{i=\{1,2,...,n\}} Fit_i(t)$

The ith atom's location and velocity at time$(t + 1)$ are given by [8]:

$$v_i^d(t + 1) = random_i^d v_i^d(t) + a_i^d(t) \tag{17}$$

$$x_i^d(t + 1) = x_i^d(t) + v_i^d(t + 1) \tag{18}$$

Each atom must communicate with as many atoms as feasible that have a better fitness value than its K neighbors to improve the exploration in the first stage of the ASO algorithm. The atoms must associate with as few atoms with a higher fitness value as feasible as their K neighbors to improve exploitation in the last stage of iterations[4]. The distance ratio r_{ij} to the length scale σ controls

whether Attraction or repulsion is the attraction or repulsion interaction force acting on each atom from its neighbors. The length scale σ reflects the distance between each atom and its K neighbors' average location. As a result, K is a time-dependent function that decreases with the number of repetitions, and K may be determined [4], [8]:

$$K(t) = N - (N - 2) \times \sqrt{\frac{t}{T}} \tag{19}$$

4 Simulation Results

TO plan the path of a "mobile robot", the proposed ASO algorithm was implemented utilizing MATLAB R2020a. Experiments were carried out On the platform with Intel® Core™ i7-8550U CPU at 1.99GHz, Windows 10 Home OS. A number of pre-tests were used to identify the parameters for the suggested procedures as in **Table** 1. In order to discover the shortest pathways for free, static and dynamic obstacle environments 5 random points were generated. The maximal number of iterations was 20. To resolve the path planning issue, ASO was proposed and compared with a particle swarm optimization algorithm. The study was testing our proposed ASO in three environments (free, Static & Dynamic obstacles) as in figures3,4,5 & 6. Search space was 200 × 200 grid with three obstacles. The maximal number of iterations was 20. The proposed method was run by taking four cases in each environment as in **Tables** 2,3 & 4. From the obtained result, it can be concluded that the execution time(the time required to travel the robot from the current point to the goal) and the minimum path(minimum cost) in ASO were reduced as compared with PSO algorithms. This demonstrates the Atom Search optimization algorithm's strength.

Table 1. parameters for ASO and PSO

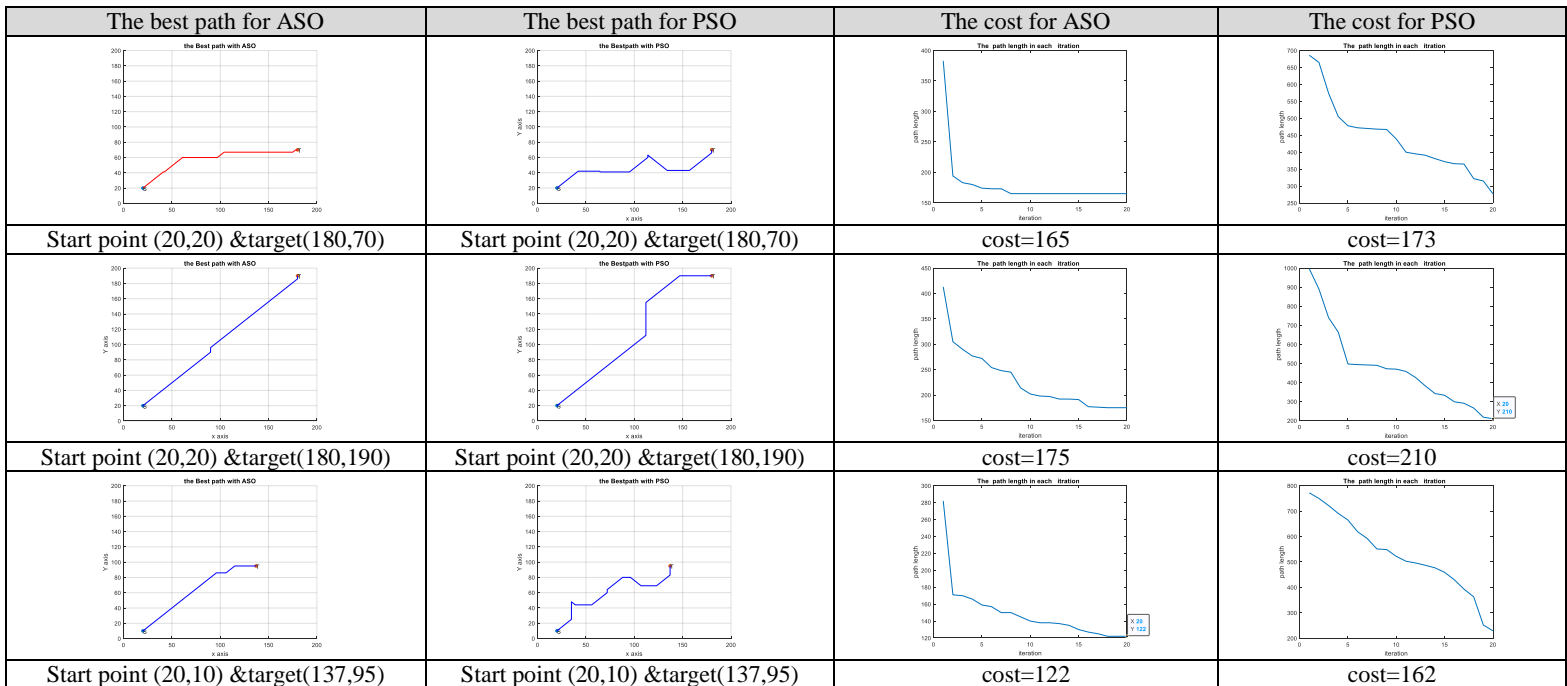| Optimization algorithm	parameters				
	Maximum iteration	alpha(α)	Beta(β)	Limda(λ)	m
ASO	20	20	0.5	1	1
	parameters				
PSO	Maximum iteration	W	C1	C2	Population size
	20	1	1.5	1.5	50

5 conclusion

This work looked at the challenge of mobile robot path planning in a two-dimensional workspace. Atom Search optimization algorithm and Particle Swarm Optimization search methods were proposed for finding the shortest path in the space were employed to select the best path with free

static and dynamic obstacles. Only the path length was used as objective by using Euclidian distance. The method that has been proposed is ASO was tested in three different settings and shown to be capable of determining the best path with minimum time as compared with PSO. In further research, instead of using ASO, the Butterfly algorithm can be used to optimize the robotic path.

Table 2. comparative between ASO and PSO for free obstacles environment

cases	Start point	target	minimum cost (Shortest path in pixels for ASO)	minimum cost (Shortest path in pixels for PSO)	Execution time for ASO (second)	Execution time for PSO (second)
Case1	(20,20)	(180,70)	165	173	36.0142	38.7519
Case2	(20,20)	(180,190)	175	210	41.2211	42.7239
Case3	(20,10)	(137,95)	122	162	25.5928	32.4814
Case4	(20,20)	(30,160)	145	146	29.9424	32.0411

The best path for ASO	The best path for PSO	The cost for ASO	The cost for PSO
Start point (20,20) &target(180,70)	Start point (20,20) &target(180,70)	cost=165	cost=173
Start point (20,20) &target(180,190)	Start point (20,20) &target(180,190)	cost=175	cost=210
Start point (20,10) &target(137,95)	Start point (20,10) &target(137,95)	cost=122	cost=162

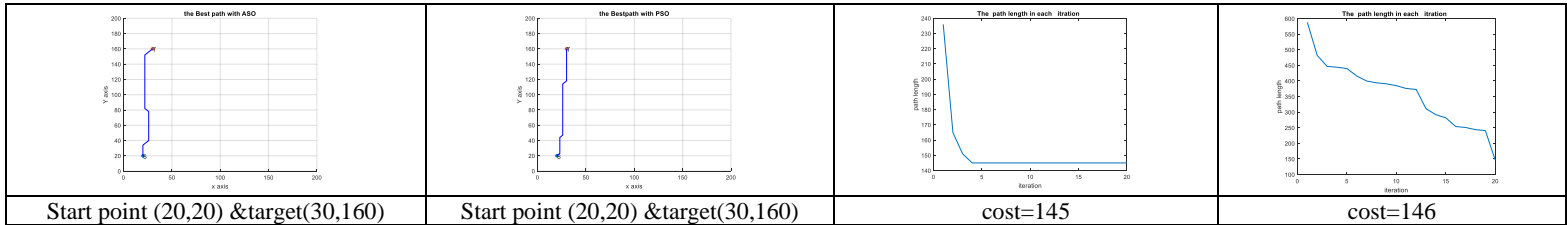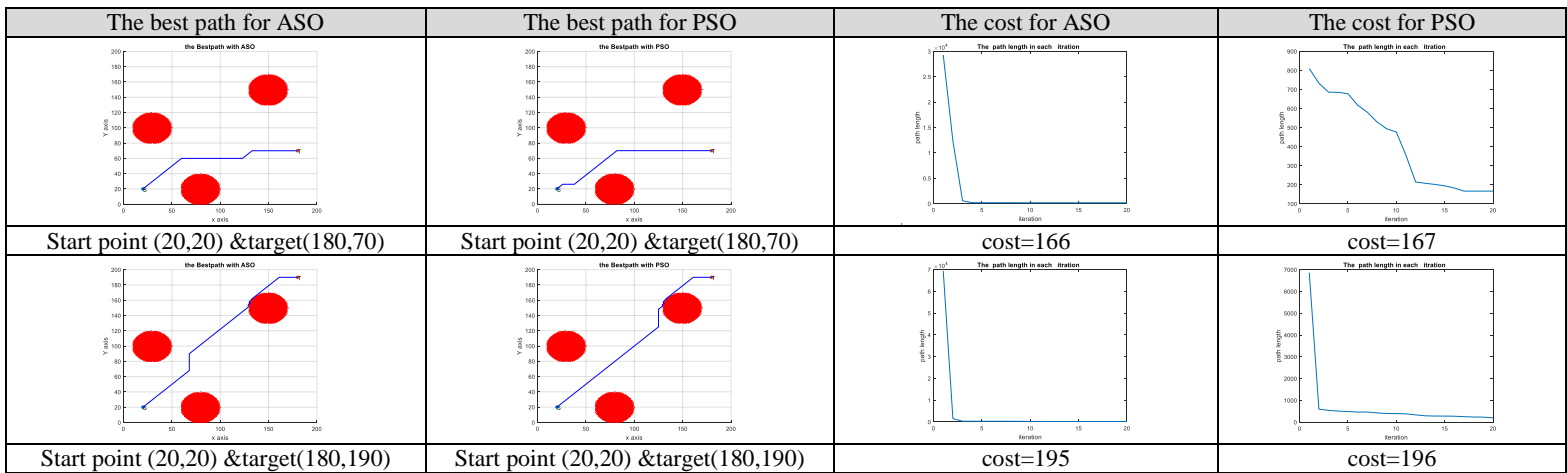|			
Start point (20,20) &target(30,160)	Start point (20,20) &target(30,160)	cost=145	cost=146

Fig.3. the best path ,the cost in each iteration for ASO & PSO in free obstacles environment

Table 3. comparative between ASO and PSO for static obstacles environment

cases	Start point	target	minimum cost (Shortest path in pixels for ASO)	minimum cost (Shortest path in pixels for PSO)	Execution time for ASO (second)	Execution time for PSO (second)
Case1	(20,20)	(180,70)	166	167	31.367	35.5854
Case2	(20,20)	(180,190)	195	196	36.5991	37.2071
Case3	(20,10)	(137,95)	123	139	23.5234	26.3468
Case4	(20,20)	(30,160)	147	149	27.9884	28.9725

The best path for ASO	The best path for PSO	The cost for ASO	The cost for PSO
Start point (20,20) &target(180,70)	Start point (20,20) &target(180,70)	cost=166	cost=167
Start point (20,20) &target(180,190)	Start point (20,20) &target(180,190)	cost=195	cost=196

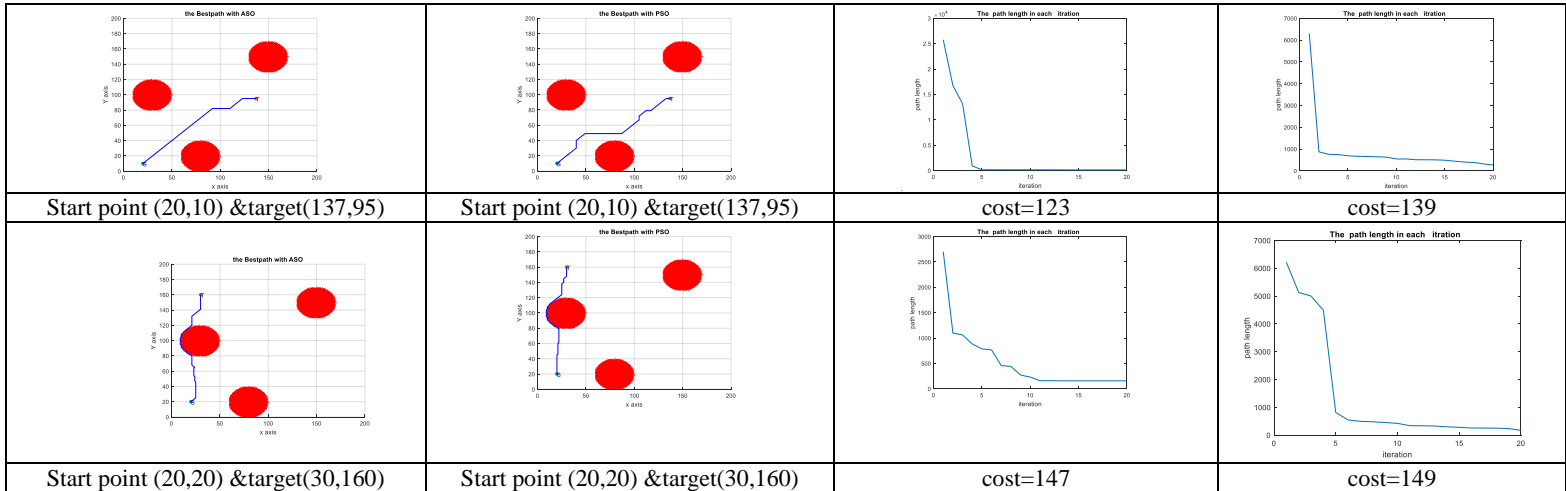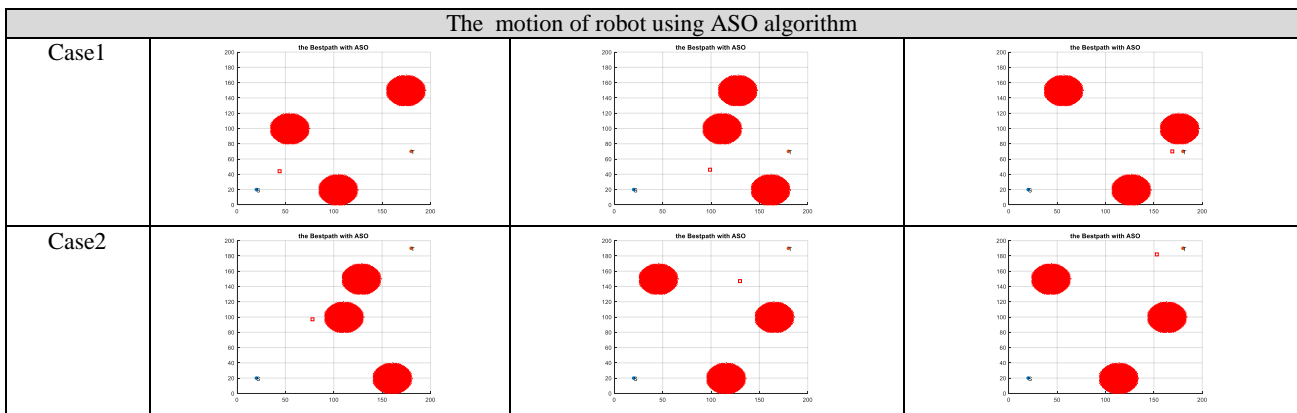|			
Start point (20,10) &target(137,95)	Start point (20,10) &target(137,95)	cost=123	cost=139
Start point (20,20) &target(30,160)	Start point (20,20) &target(30,160)	cost=147	cost=149

Fig. 4. The best path, the cost in each iteration for ASO & PSO in static obstacles environment

Table 4. comparative between ASO and PSO for dynamic obstacles environmen

cases	Start point	target	minimum cost (Shortest path in pixels for ASO)	minimum cost (Shortest path in pixels for PSO)	Execution time for ASO (second)	Execution time for PSO (second)
Case1	(20,20)	(180,70)	165	169	24.5218	27.9011
Case2	(20,20)	(180,190)	195	268	20.2752	22.4291
Case3	(20,10)	(137,95)	121	161	6.6995	15.7852
Case4	(20,20)	(30,160)	145	151	27.5628	50.9255

The motion of robot using ASO algorithm			
Case1			
Case2			

Case3	the Bestpath with ASO	the Bestpath with ASO	the Bestpath with ASO
Case4	the Bestpath with ASO	the Bestpath with ASO	the Bestpath with ASO
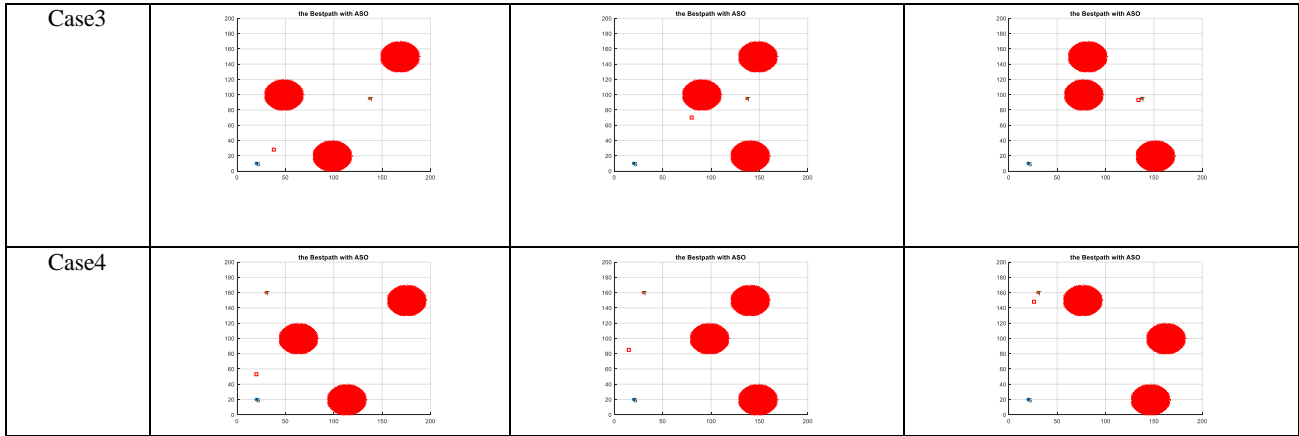
Fig. 5. The robot's motion from start point to end point in each case using ASO

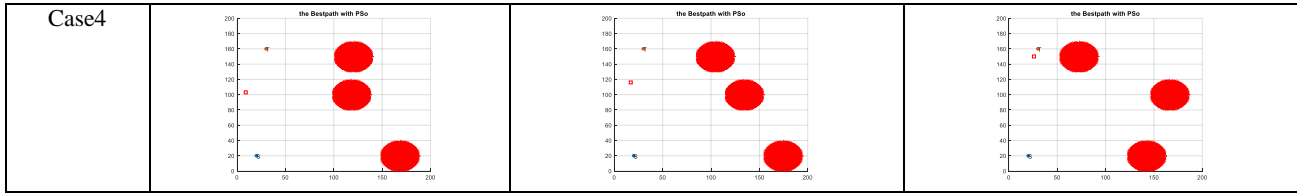|			
The motion of robot using PSO algorithm			
Case1	the Bestpath with PSo	the Bestpath with PSo	the Bestpath with PSo
The motion of robot using PSO algorithm			
Case2	the Bestpath with PSo	the Bestpath with PSo	the Bestpath with PSo
Case3	the Bestpath with PSo	the Bestpath with PSo	the Bestpath with PSo

Case4			
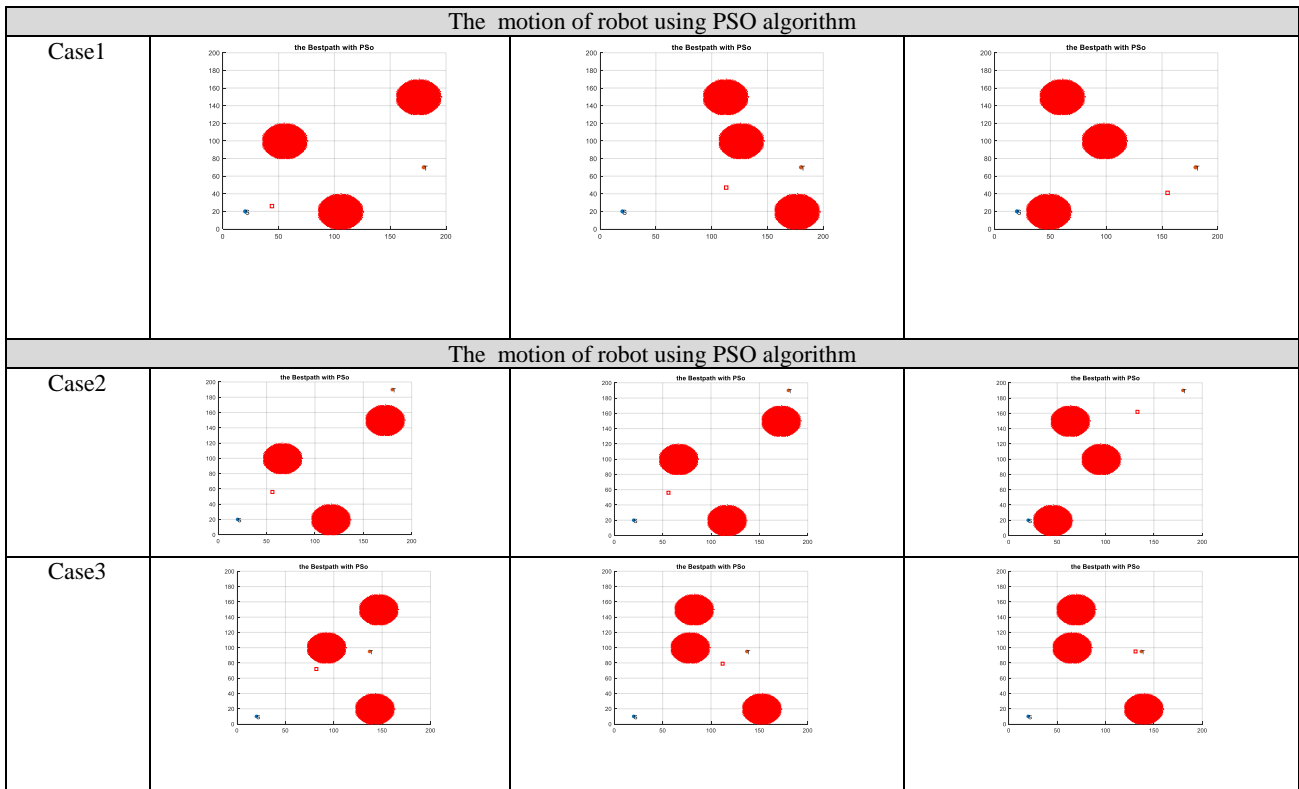
Fig. 6. The robot's motion from start point to end point in each case using PSO

References:

[1]   Tuba E, Strumberger I, Zivkovic D, Bacanin N, Tuba M. Mobile Robot Path Planning by Improved Brain Storm Optimization Algorithm. 2018 IEEE Congr Evol Comput CEC 2018 - Proc. 2018; (July): 1–8.

[2]   Abed, I. A., Koh, S. P., Sahari, K. S. M., Tiong, S. K., & Yap, D. F. (2012). Comparison between genetic algorithm and electromagnetism-like algorithm for solving inverse kinematics. *World Applied Sciences Journal*, *20*(7), 946-954.

[3]   Yüzgeç U, Doğan L. Robot Path Planning using Gray Wolf Optimizer. 2019; (May 2018).

[4]   Zhang H, Lin W, Chen A. SS symmetry Path Planning for the Mobile Robot : A Review. 2018;

[5]   Tan, J. D., Dahari, M., Koh, S. P., Koay, Y. Y., & Abed, I. A. (2016). An improved electromagnetism-like algorithm for numerical optimization. *Theoretical Computer Science*, *641*, 75-84.

[6]   Zhao, W., Wang, L., & Zhang, Z. (2019). A novel atom search optimization for dispersion coefficient estimation in groundwater. *Future Generation Computer Systems*, *91*, 601-610.

[7]   Wang Y, Cai F, Wang Y. "Dynamic path planning for mobile robot based on particle swarm optimization Dynamic Path Planning for Mobile Robot based on Particle Swarm Optimization". 2017;020024(August).

[8]   Châari I, Koubâa A, Bennaceur H, Trigui S, Al-Shalfan K., SmartPATH: A hybrid ACO-GA algorithm for robot path planning. 2012 IEEE Congr Evol Comput CEC 2012. 2012;(1):10–5.

[9]   Ibraheem IK, Ajeil FH, Khan ZH. Path planning of an autonomous mobile robot in a dynamic environment using modified bat swarm optimization. arXiv. 2018; (July).

[9]   Ghosh KK, Guha R, Ghosh S, Bera SK, Sarkar R. Atom search optimization with simulated annealing- A hybrid metaheuristic approach for feature selection. arXiv. 2020.