Safe and Optimum Navigation of Wheeled Mobile Robot using Grey Wolf Optimization Algorithm

Tahseen Fadhel Abaas\textsuperscript{1,2} and Alaa Hassan Shabeeb\textsuperscript{1,3}

\textsuperscript{1} Department of Production Engineering and Metallurgy, University of Technology, Baghdad, Iraq
\textsuperscript{2} Email: 70047@uotechnology.edu.iq
\textsuperscript{3} Email: alaa.h.shabeeb@uotechnology.edu.iq

Abstract. This paper discusses our research in developing a track planner for a mobile robot using a swarm intelligence technique for optimal track planning in a short computational time to achieve better results in track planning. Through this technique, we proposed grey wolf optimization (GWO) for generating fastest and optimal path planning. this paper introduces an algorithm for rapid and global motion preparation for a mobile robot in a complex environment with static obstacles. the performing analysis for GWO algorithm was evaluated in two different maps. Finally, A comparative study was evaluated between the algorithm built and the other algorithm exist, the simulation results, which carried out using Matlab 2018a showed that the GWO algorithm made results for generating optimal path planning and efficiently in terms of path distance, execution-time.

Keywords: Nature-Inspired Optimization Algorithm, Grey Wolf Optimizer, Path Planning, Obstacle Avoidance, Mobile Robot.

1. Introduction
Autonomous navigation mobile robots (ANMR) are electro-mechanic and programmable devices with the ability to acclimate and interact with the surrounding environment without direct human intervention. An autonomous robotics device attempts to replicate human activities in a real manner with an appropriate degree of intelligence and precision. Examples of these behaviours include objects detection and recognition, obstacles avoiding, generating a collision-free path to the target. To find a free-obstacle path, the ground mobile robot (GMR) must use sensors to perceive the environment and to gather various information in the environment to identify obstacles, the robot starts location and target destination, calculate distances, and compute the shortest path. GMR applications are oriented to different areas that interact with humans in different fields\textsuperscript{[1]}: (i) industrial: combine between robots and artificial intelligence to develop the product through the use of mobile robots. (ii) health: as it requires robots to be assisted in tasks for the elderly and people with paraplegia, (iii) military: since it focuses on the supervision of autonomous and intelligent, remote-controlled robots. For a GMR to be able to perform various types of tasks in these application areas, it must autonomous navigation through intelligence in making decisions about its movement\textsuperscript{[2]}. Therefore, ANMR is one of the most difficult robotic tasks. This challenge can be split up by the following criteria into two different evaluation categories\textsuperscript{[3]}. One is based on GMR environment path planning, it refers to the area the robot works in, which may be broadly classified as static and dynamic, in the static environment the details about the environment and the location of all obstacles do not alter over time, in this type, global route planning algorithms can be implemented effectively to solve the problem due to obstacles...
of a static nature. On the contrary, in the dynamic environment, the target location and obstacle can change partially or entirely unknown. Because the same area needs to be explored, global path planning cannot be implemented, thus local path planning is the only option. The other is focused on the perception of GMR about the problem of robot navigation that can be categorized based on acquired information on a global and local path. Global path planning means GMR can obtain complete global environmental model details, including all obstacle information, destination information, but because of the slowness caused by their complexity, this technique is suitable only in the case of a static area where obstacles not moving, another hand, local route planning can only get information about local areas. The biggest difference between local and global strategies lies in the preparation of path planning [4]. The whole process of route planning can be considered as a series of continuous local route planning processes, thus should the combining local path planning and global path planning thus, the combination of local path planning and global path planning for ensuring that the developed hybrid system makes use of its advantages and avoids the weakness of each method. According to the above, the basic autonomous GMR navigation can be divided into three tasks: environment modelling in the form map, Path Planning, and Control system. So, designing a fast and efficient procedure for navigation is an essential step in the path planning of mobile robots Thus, the path planning problem of the navigation path is classified as an optimization problem. In any task, there are multiple paths for MR to reach the target point, but the recognition the best path depends on adapted some guidelines criteria such as shortest path, time and energy consumed. Many optimization algorithms have been used to control the path planning problem and collision avoidance [5]. Over the last years, researchers have introduced numerous algorithms to deal with the issue of GMR route planning. These algorithms are divided into main two approaches including the classical optimization techniques and nature-inspired optimization techniques, the classical techniques are can be subdivided into three main sub-classes, namely the grid-based methods such as (A*, D*), the potential-field methods and sampling methods such as (probabilistic roadmap. The nature-inspired optimization techniques are further classified into the evolutionary algorithm such as (genetic algorithm, differential algorithm) and swarm-based optimization such as (ant colony optimization, Bat Algorithm, particle swarm optimization). In general, The Classical approaches are required high computational time and costly to solve path planning compares with the nature-inspired approaches which easy and uses random solutions to achieve its goal [6,7]. Thus, select the appropriate algorithm of the path planning process is a vital issue to ensure that MR navigation optimally in the environment. This paper proposed methodology based on a grey wolf optimization (GWO) algorithm. GWO is one of the latest algorithms in swarm intelligence (SI) algorithms Which caught the attention of many researchers in various fields of optimization. GWO was distinguished from other SI algorithms by several characteristics. It has few tuning parameters, a good balance can be simply achieved between exploration and exploitation, and a favourable convergence can be achieved, GWO is simple, user-friendly, versatile, and scalable [8].

The main contribution of our work is developed with parameters effect study of GWO is a complete algorithm for path planning in multiple scenarios in the environment, contain static obstacle obstacles through the following.

1. The study on performance for the GWO algorithm to find the optimal path through various environments.
2. we compared the strategy effective effect on GMR movement path planning in an environment contained fixed locations for obstacles and target point to reduce the total time for the path-finding.
3. Prove the effectiveness of our methods by conducting simulations with various scenarios.
2. The Map Building Description
The GMR navigation in the 2D environment requires existence a map for located start and destination points, it's used to design GMR optimal movement, GMR represented in this map as point by a set of Cartesian coordinate positions (x,y) in the map, that contains many static obstacles. Each boundary obstacle is extended by the amount of the GMR radius (R-GMR) depending on the studied GMR shape, to assure the safety of the robot while trying movement in the environment, as shown in Figure 1. The GMR has no prior knowledge of environmental parameters such as locations, shapes, and sizes of the obstacles, and assume existence equipping overhead sensor (camera) for on-line detection to the environment, located the GMR, and build the map in the same time.

![Figure 1. Extend obstacle size corresponding to GMR.](image)

3. GREY WOLF OPTIMIZER (GWO) for Path Planning
The GWO algorithm is a newcomer in nature-inspired optimization algorithms by grey wolf conduct, originally put forward by Mirjalili et al. [9]. It stimulates both the hunting mechanisms and the social leadership observed in grey wolves in the wild. Grey wolves often prefer to live in a group, commonly called a pack with a well-defined hierarchy, as presented in Figure 2.

![Figure 2. Grey wolves’ hierarchy and their responsibilities source.](image)

To get on modelled and efficient optimization to the GWO algorithm, the wolf pack (population) is divided four ranks according to the social hierarchy, are alpha (α), beta (β), delta (δ), and omega (ω). Alpha wolves are leader wolves, and the alpha wolf (α) occupies the first rank in the pack hierarchy at the top of the hierarchy, the alpha wolf follows the other wolves in the group. They are responsible for making decisions that are followed by all other wolves. These decisions are related to hunting time, prey (target) selection, sleeping place, etc [10]. In the second rank in the pack hierarchy is beta wolves (β). They sub-coordinate the lower-level wolves and help the α leader to make decisions or other tasks, the third rank in the pack hierarchy are delta wolves (δ). Their main function in the pack is to execute the decisions of the leaders. They are responsible for performing different tasks such as watching the
territory, helping the alphas and betas during a hunt, taking care of weak wolves, among others. The four ranks (last) in the pack hierarchy is the omega wolves (ω). Omega wolves respond to orders the other wolves dominant. The top important point in the social hierarchy is that just leading wolves known the location of the target (prey) and steer omegas to perform the search. The GWO algorithm is modelled to optimize based on the social hierarchy and the hunting strategy, as presented in Figure 3. In this strategy assumed, the wolves are recognizing the location of the target and encircle it under the leadership of the alfa wolf, a group of candidate solutions for the optimization problem under consideration is randomly generated in the search space, α, β, δ provide better knowledge (best solution) about the location of the target. Then, the positions of the wolves are updated as a function of their distances to the α, β, and δ wolves, aiming at getting closer to the target and encircle it, as explained in the following sections.

![Figure 3. Behaviour grey wolves in hunting: (A) Chasing, getting closer and tracking target (B-D) Convince, threaten, and encircle (E) Fixed situation and attack.](image)

### 3.1 Mathematical modelling

The mathematical modelled of GWO algorithm is modelled in four steps, as follows:

i- **Social Hierarchy Structure**

The GWO algorithm is applied to optimize based on the social hierarchy of the wolves that including follow three main steps (i) tracking the target, (ii) encircling the target, and (iii) attacking to the target. According to the three steps, GWO simulates these three crucial steps to optimize. The best fitness solution is defined as alpha (α), while the second and third best fitness solution is beta (β) and delta (δ), remaining wolves are assumed to follow α, β and δ wolves. Which are in the hierarchy above them.

ii- **Encircling the target (prey)**

During target hunting, each wolf has the same inherent function of encircling a goal, the mathematical equation modelled in the GWO algorithm to mimic encircling behaviour of the wolves are formulation in Eq. (1) and Eq. (2).

\[
\begin{align*}
\vec{D} & = |\vec{C} \cdot \vec{X}_{P(t)} - \vec{X}_t| \\
\vec{X}(t+1) & = \vec{X}_{P(t)} - \vec{A} \cdot \vec{D}
\end{align*}
\]

where, \( \vec{D} \) is the distance vector between the target and the wolf. \( \vec{X} \) and \( \vec{X}_{P(t)} \) indicates position vector of the given grey wolf and the target, \( t \) is current iteration. \( \vec{A} \) and \( \vec{C} \) are indicate the coefficient vectors of wolves, it uses the following formulations [13].
\[ A = 2\hat{a}\cdot \hat{r}_1 - \hat{a} \]  
\[ C = 2\cdot \hat{r}_2 \]  
where, \( \hat{r}_1 \) and \( \hat{r}_2 \) are the vectors randomly generated within the [0,1] range. The above formulations allow wolves to arrive at any point between the target and the wolf. It controlled the activity of the GWO algorithm and is used in the calculation of \( \hat{A} \). Vector variable values of \( \hat{a} \) vector decreases linearly throughout iterations from 2 to 0 [9].

iii. Hunting the Target

In this phase, wolves can identify where the target is located, it can encircle target easily under the leadership of the alfa wolf to guide whole wolves during hunting processing. The positions of the wolves are updated as a function of their distances of the \( \alpha, \beta, \) and \( \delta \) wolves, it is formulations in the following equations [9].

\[ \vec{D}_1 = |C_1 \cdot X_\alpha - X_t| \]  
\[ \vec{D}_2 = |C_2 \cdot X_\beta - X_t| \]  
\[ \vec{D}_3 = |C_3 \cdot X_\delta - X_t| \]  
\[ X_1 = X_\alpha - A_1 \cdot \vec{D}_\alpha \]  
\[ X_2 = X_\beta - A_2 \cdot \vec{D}_\beta \]  
\[ X_3 = X_\delta - A_3 \cdot \vec{D}_\delta \]  
The updating location of the grey wolf (best position) can be calculated using the Eq (11).

\[ \hat{X}_{(t+1)} = \frac{X_1 + X_2 + X_3}{3} \]  

iv. Search (exploration) and Attacking (exploitation) the Target

Grey wolves attack the target and try to stop the movement of the target, during a hunting mechanism. This mechanism is modelled done by declining the value of \( \hat{a} \) used in Eq. (3). \( |\hat{A}| \) is a random vector, its value is within the range of \([-\hat{a}, \hat{a}]\), where the value of \( \hat{a} \) is decreases linearly throughout iterations and it is in the range of [2, 0] as shown in Eq. (12).

\[ \hat{a} = 2 - \left( \frac{2 \cdot \text{iter}}{\text{iter}_{\text{max}}} \right) \]  

So, (exploitation) if \( |\hat{A}| < 1 \), This means that the wolf would be forced to strike the target by approaching it.

(Exploration) if \( |\hat{A}| > 1 \), The wolf gets diversified from the target and seeks a fitter target. 

The searching mechanism by grey wolves searching for the target is done according to the location influence of the wolves \( \alpha, \beta \) and \( \delta \) wolves. The exploitation and exploration depend only on the \( \hat{A} \) and \( \hat{C} \) vector values. We can make use of \( \hat{A} \)'s random values to get on diverge or converge from the target. \( \hat{C} \) is another element factor, Which influences the phase of exploration in the GWO algorithm. The \( \hat{C} \) values are within the range [0, 2], this value plays a significant role in preventing (avoiding) stagnation at the local optima. It applies some random weight to the target to make it harder for grey wolves, to determine the gap between the target and the wolf. 

So, 

if \( \hat{C} > 1 \) means \( \hat{C} \) is Emphasizing the target impact, and if \( \hat{C} < 1 \), the impact of \( \hat{C} \)’s effect deemphasized stochastically. During all process steps, the \( \hat{A} \) and \( \hat{C} \) vectors shall be carefully balanced. Both the \( \hat{A} \) and \( \hat{C} \) vectors will reinforce or Do not reinforce exploitation or exploration. In the end, 

When the final criterion is met, then the GWO Algorithm is finished and the best location is the determined, the diagrammatic version of the effect these parameters is represented during updating the location of the wolves in Table 1, while Figure 4, illustrates the way the wolves update their position.
Table 1: General Pseudo_code for Proposed Techniques (GWO).

1- Input: initialize the parameters (population size, maximum iteration, C, a, A).

2- Calculate cos value (fitness) of the search agent to define $\tilde{X}_\alpha$, $\tilde{X}_\beta$ and $\tilde{X}_\delta$.
   - Save the best fitness search _agent (wolf) (wolf) is $\tilde{X}_\alpha$.
   - Save the second-best search _agent (wolf) is $\tilde{X}_\beta$.
   - Save the third best search _agent (wolf) is $\tilde{X}_\delta$.

   for each gent I (i=1 to N) do
     Random initialized position
   end for

   iter $\leftarrow$ 0
   while (iter < iter_max) do
     for each agent i (i=1 to N) do
       Evaluation of the fitness function $f_{final}$ for each agent.
       If grey wolf < alpha wolf
         Update alpha wolf
       If grey wolf < beta wolf
         Update beta wolf
       If grey wolf < delta wolf
         Update delta wolf

       Update location of the present search agent using Eq. (11).
     End for

     Update parameter a, A, and C
     Calculate the fitness of all search_agents.
     Update $\tilde{X}_\alpha$, $\tilde{X}_\beta$, and $\tilde{X}_\delta$.
     Iter $\leftarrow$ iter+1
   end while
Figure 4. Other wolf movement mechanism in GWO algorithm

4 Evaluation Function

The evaluation function is fundamental in the algorithm, it should be tested with accuracy. The algorithm GWO is based on the behavior of swarm to find the food (target) source with minimum time and length of the route, taking into account the obstacle avoidance. So, the evaluation Function in this research based on two estimation functions to optimize the objective function, as described below.

4.1 Minimization of the Length path between start and Target Point

Euclidean distance is the criteria to find minimal path length (minimum optimization) is defined as the distance between the GMR and the goal position in each iteration. The evaluation process based in this study based on location which given the minimum objective function value must be selected for the next move of a GMR, hence for tracking the minimum length of the path from the initial point \((R_x_s, R_y_s)\) to target point \((G_x, G_y)\) in the 2D environment, via way-points by the following equation.

\[
F_1(i) = \sqrt{(R_x_i - G_x)^2 + (R_y_i - G_y)^2} \quad \text{......... (13)}
\]

where:
- \(F_1(i)\): distance function between GMR and goal.
- \(i\): current iteration
- \((R_x_s, R_y_s)\): coordinates of the start position.
- \((G_x, G_y)\): coordinates of the goal position.
- \((R_x_i, R_y_i)\): coordinates of the GMR at the current position.

4.2 Optimum Safe Distance from Obstacles.

A second part to completely optimize the objective function is optimum avoiding of the collision during path planning, the GMR should have the safety distance from the obstacles and it should be equal or larger than the radius of GMR \((R_{GMR})\). Therefore, we used penalty value, Where the actual
distance between GMR and obstacle is equal to or less than the distance to protection ($\epsilon$), the penalty value will be applied to the main function ($F_1$) and path become unfeasible as the issue of optimisation is minimisation, otherwise penalty value is equal to zero, The distance from the GMR to the obstacle can be determined using the following formula.

$$\text{Dis}(i) = \sqrt{(Rx_i - Ox_n)^2 + (Ry_i - Oy_n)^2} \quad \ldots \ldots \quad (14)$$

Where:

- $\text{Dis}(i)$: Distance function between GMR and obstacle.
- $(Ox_n, Oy_n)$: Coordinates of the obstacle, $n$ is the number of an obstacle.

$$F_2(i) = \begin{cases} 1, & \text{if } \text{Dis} \leq \epsilon \\ 0, & \text{otherwise} \end{cases}$$

Hence, the total fitness function is

$$Z_i = F_1(i) + F_2(i) \quad \ldots \ldots \quad (15)$$

Where $\epsilon$ is a minimum distance between the points of track and the obstacle. Based on Eq (14), the global best position and the new position for GMR is determined. This cycle repeats for locations interpolations (intermediate points) until the GMR is reaching the final destination point.

5. performance study and Simulation Analysis

5.1. Setting and Environment

To getting simulation results for the performance of GWO algorithm to find the optimal path through two environments with taking different numbers of obstacles, the proposed methods in this study are coded in MatLab R2018a and tested on Intel(R) core i7, 2.2 GHz CPU, 8.00 GB RAM system. The GWO is used in the test for each environment with 80 agents and 100 iterations. During simulation setting, we created two maps with dimension (10x10) units, each the map is evaluated by the proposed method, the GMR has a circular dot shape and red colour and its move from the start point (0,0) to the goal point (10,10) in Presence the different number and shapes of obstacles for each map, and location for each obstacle for each map is given in table.2.

| Environment | No of obstacles | Obstacle positions in (unit) (x,y, radius) |
|-------------|----------------|------------------------------------------|
| Map 1       | 5              | (2,1,1), (3,3,1.8), (2.5,1), (6,2,2), (5,5,1.8) |
| Map 2       | 6              | (2,2,0.5), (6,2,0.5), (2,4,0.5) (8,4,0.5), (5,5,1.1) (3,7,0.5) |

5.2. Simulation Results

This section evaluates the performance of our GWO algorithm strategy in a simulation environment, each map and its fitness function are modelled base on the GMR sensory information, the result of simulation proved that proposed algorithm is effective in finding the optimal path in all of the maps, as shown in figures (5, 6). The table 3, shows the optimal path length and the algorithm execution time in each map, after performance analysis, results indicated that GWO is effective for generate the optimal travelled path with collision avoidance.

| Environment | Algorithm | Optimal path length (in pixels) | GWO execution time (in seconds) | Number of best generations |
|-------------|-----------|---------------------------------|---------------------------------|---------------------------|
| MAP1        | GWO       | 14.4654                         | 1.6286                          | 34                        |
| MAP2        | GWO       | 14.4696                         | 1.8339                          | 96                        |
Figure 5: Path length optimization based on GWO algorithm in MAP1
(a): Best path found by GWO algorithm.  (b): Variation of optimal path length with iterations
Figure 6: Path length optimization based on GWO algorithm in MAP1
(a): Best path found by GWO algorithm. (b): Variation of optimal path length with iterations

5.3 Comparison with previously Chosen algorithm technique

Execute the simulation process of any algorithm technique in different environments is not sufficient to assert that it is best. It should provide some proof through compared with previously applied strategies to sure that the proposed technique is better. So, we compared the current technique with the technique previously chosen to determine the response in the selected environment. To get the best outcome possible, will reproduce the previous design environment. Nizar and Farah [11], The
result was observed, as shown in the figure. 7.a, after the AT-BFO technique, was implemented. Figure 7.b is describing the best result which is obtained by embedding the GWO technique. The results we have obtained are far better than the technique used previously. Table 4 is enough to find out which strategy is best. The length of travel in the current technique is less than the length of travel in the technique chosen previously.

![Graph](image)

**Figure 7:** (a): Best path length found by ATBFO algorithm [11]. (b): Best path length found by GWO algorithm.

| Figure no | Algorithm | Optimal path length |
|-----------|-----------|---------------------|
| 7.a       | ATBFO     | 14.5346             |
| 7.b       | GWO       | 14.3574             |

*ATBFO: Adaptive Tumble Bacterial Foraging Optimization

6. Discussion and Conclusion

Modelling and simulation of an autonomous mobile robot to produce the fastest and optimal route planning are the main objectives of this paper. Path planning problem using GWO algorithm was discussed, we evaluated the performance of the GWO algorithm to get better path planning results and reducing the total time for the pathfinding, and also we compared GWO algorithm with previously selected algorithm technique, the simulation result showed that the proposed technique for the robot path planning task is more effective with avoiding the collision than previously selected algorithm technique.

In future works, we would like to study our work through a comparison with another algorithm.
References

[1] Segarra, D., Caballeros, J. and Aguilar, W.G., 2018, August. Visual Based Autonomous Navigation for Legged Robots. In International Conference on Intelligent Science and Big Data Engineering (pp. 22-23). Springer, Cham.

[1] Hliwa, H. and Atieh, B., 2020, March. Multi Objective Path Planning in Static Environment using Region of Sight. In 2020 International Youth Conference on Radio Electronics, Electrical and Power Engineering (REEPE) (pp. 1-5). IEEE.

[3] Patle, B. K., Pandey, A., Parhi, D. R. K., & Jagadeesh, A, 2019. A review: On path planning strategies for navigation of mobile robot Defence Technology, vol. 15, pp. 582–585.

[4] Zafar, M.N. and Mohanta, J.C., 2018. Methodology for path planning and optimization of mobile robots: A review. Procedia computer science, 133, pp.141-152.

[5] Bayat, F., Najafinia, S. and Aliyari, M., 2018. Mobile robots path planning: Electrostatic potential field approach. Expert Systems with Applications, 100, pp.68-78.

[6] Mac, T.T., Copot, C., Tran, D.T. and De Keyser, R., 2017. A hierarchical global path planning approach for mobile robots based on multi-objective particle swarm optimization. Applied Soft Computing, 59, pp.68-76

[7] Lei, X., Wang, F. and Tan, Y., 2019. Swarm intelligent optimization algorithms and its application in mobile robot path planning. In Rapid Automation: Concepts, Methodologies, Tools, and Applications (pp. 609-648). IGI Global.

[8] Faris, H., Aljarah, I., Al-Betar, M.A. and Mirjalili, S., 2018. Grey wolf optimizer: a review of recent variants and applications. Neural computing and applications, 30(2), pp.413-435.

[9] Mirjalili, S., Mirjalili, S.M. and Lewis, A., 2014. Grey wolf optimizer. Advances in engineering software, 69, pp.46-61.

[10] Muangkote, N., Sunat, K. and Chiewchanwattana, S., 2014, July. An improved grey wolf optimizer for training q-Gaussian Radial Basis Functional-link nets. In 2014 international computer science and engineering conference (ICSEC) (pp. 209-214). IEEE.

[11] Abbas, N.H. and Ali, F.M., 2016. Path planning of an autonomous mobile robot using enhanced bacterial foraging optimization algorithm. Al-Khwarizmi Engineering Journal, 12(4), pp.26-35.