Artificial Potential Field-Based Path Planning for Cluttered Environments

Diab, Mosab ; Mohammadkarimi, Mostafa; Rajan, Raj Thilak

DOI
10.1109/AERO55745.2023.10115857

Publication date
2023

Document Version
Final published version

Published in
Proceedings of the 2023 IEEE Aerospace Conference

Artificial Potential Field-Based Path Planning for Cluttered Environments

Mosab Diab
SPS, Faculty of EEMCS, Mekelweg 4
Delft University of Technology (TUD)
2628CD Delft, The Netherlands

Mostafa Mohammadkarimi
SPS, Faculty of EEMCS, Mekelweg 4
Delft University of Technology (TUD)
2628CD Delft, The Netherlands

Raj Thilak Rajan
SPS, Faculty of EEMCS, Mekelweg 4
Delft University of Technology (TUD)
2628CD Delft, The Netherlands

***Abstract*—In this paper, we study path planning algorithms of resource constrained mobile agents in unknown cluttered environments, which include but are not limited to various terrestrial missions e.g., search and rescue missions by drones in jungles, and space missions e.g., navigation of rovers on the Moon. In particular, we focus our attention on artificial potential field (APF) based methods, in which the target is attractive while the obstacles are repulsive to the mobile agent. In this paper, we propose two major updates to the classical APF algorithm which significantly improve the performance of path planning using APF. First, we propose to improve an existing classical method that replaces the gradient descent optimization of the potential field cost function on a continuous domain with a combinatorial optimization on a set of predefined points (called bacteria points) around the agent's current location. Our proposition includes an adaptive hyperparameter that changes the value of the potential function associated to each bacteria point based on the current environmental measurements. Our proposed solution improves the navigation performance in terms of convergence to the target at the expense of minimal increase in computational complexity. Second, we propose an improved potential field cost function of the bacteria points by introducing a new branching cost function which further improves the navigation performance. The algorithms were tested on a set of Monte Carlo simulation trials where the environment changes for each trial. Our simulation results show 25% lower navigation time and around 300% higher success rate compared to the conventional potential field method, and we present future directions for research.**

TABLE OF CONTENTS

1. INTRODUCTION

Autonomous agents such as drones and rovers are the future of delivery systems, search and rescue missions, and surveillance systems. Autonomous agents can be used in manufacturing plants, such as pick and place robots. They can also be used for disinfection in medical facilities. Moreover, they can be used in transport of both people and goods and many more different applications.

The navigation systems for drones in unknown and cluttered environments like jungles, oceans, and sea floors are in their early development stages and still do not have the sufficient performance levels to be deployed into the field. Specifically, low complexity path planning and collision avoidance algorithms are needed to be developed for autonomous agents in cluttered environments. Path planning is defined as finding a geometrical path from the current location of the agent to the target location such that it avoids the obstacles in the environment [1]. The goal is to reach the target location through a safe path in the shortest time possible.

In this paper, we propose a path planning algorithm optimized for cluttered environments that outperforms existing classical path planning algorithms in terms of convergence and time it takes to navigate. We first start by analyzing the existing methods and deciding which one is the most suitable for use in cluttered environments. We then define our system model and describe the artificial potential field (APF) method. After that, the path planning system developed for cluttered environments is described along with the contributions we made. We then discuss the local minima trap which is the biggest drawback of APF methods. We conclude the paper with the simulation setup and results.

Related Research

The working principle of existing collision avoidance systems employed for path planning can be explained by either reactive control or deliberative planning. Reactive control is when the agent gathers information using on-board sensors and reacts based on the gathered data. Reactive control allows for rapid response but can lead to a corner case in the path and might get the agent stuck so it may need another technique combined with it in order to avoid that. Deliberative planning on the other hand is when there is an environmental map that is constantly updated by the agent. Then, the optimal collision free navigation path is calculated. The latter method needs an accurate map of the environment and that is computationally extensive specially for a dynamic environment. A hybrid approach between the two is considered more suitable for dynamic environments [2].

Existing collision avoidance methods can be divided into four groups:

1. Geometric Methods
2. Force-Field Methods
3. Optimisation-Based Methods

978-1-6654-9032-0/23/$31.00 ©2023 IEEE

2023 IEEE Aerospace Conference | 978-1-6654-9032-0/23/$31.00 ©2023 IEEE | DOI: 10.1109/AERO55745.2023.10115857

4. Sense-and-Avoid Methods

Comparison Between Methods— Multiple parameters can be used for performance comparison of collision avoidance methods employed for path planning. In this section, we will review some of these methods and try to indicate the discrepancy.

The real time performance of the sense and avoid and geometric methods is better than the force-field and optimisation methods [2, p. 11]. Sense and avoid does not increase computational complexity if a change happens in the environment such as an obstacle moving [3]. Geometric methods are also computationally light but are highly dependent on the algorithm implementation in terms of computation time [4]. Optimisation methods are of medium complexity while the force-field methods are considered low-complex in implementation [2, p. 11]. The velocity constraint is a metric that takes the velocity of the obstacles in consideration. Sense and avoid and geometric approaches are capable to handle this constraint well [5],[6]. However, force-field and optimisation methods are better for predefined planning which does not take changing dynamics into consideration [2, p. 11]. The third metric is static and dynamic environment suitability. For dynamic environments, sense and avoid is the easiest and lightest due to the local computations being performed on board the agent through the changes observed by the sensory system on the agent itself [7]. Geometric methods give an acceptable performance but less optimal than sense and avoid [8]. Force-field methods do not perform well in narrow passages and can lead to a local minima trap in dynamic environments [9]. Optimisation methods are more suitable for static environments as they require pre-planning and have to optimise the whole route again in case a change is detected [2, pp. 11-12].

When it comes to deadlock (local minima trap), the geometrical and optimisation methods do not have this issue due to the fact that they are designed for known structured environments. Force-field methods can get stuck in a local minima. Sense and avoid methods can reach a deadlock and require another methodology for handling this issue as it cannot be solved locally [2, p. 12]. When it comes to swarm compatibility all the approaches can be applied to a swarm of agents but might need an additional algorithm for handling communication between the agents. When it comes to dimensionality, all the mentioned methods need some minor modifications when it comes to scaling the algorithms developed for 2D to handle 3D environments. However, there is current research on the feasibility of force-field methods for 3D dynamic environments [2],[10]. For pre-mission path planning, sense and avoid and force field methods do not require that because the plan is made when the obstacle is detected in-flight. Optimisation and geometric methods on the other hand, require path planning beforehand as the obstacle locations need to be known to the UAV before encountering the obstacles.

The takeaway point here is that for unknown cluttered environments, the force-field method is the best one to use. The reason for that is that the geometric and optimisation methods are designed for known structured environments and do not perform well in unknown cluttered environments while the sense and avoid methods do not take multiple obstacles into account which is necessary for cluttered environments.

2. System Model

We consider an agent with a known position $\mathbf{r} = [x, y]^T$ which aims to navigate through a set of N cluttered static obstacles in a 2D environment[2] to a target of a known location $\mathbf{r}_\mathrm{t} = [x_\mathrm{t}, y_\mathrm{t}]^T$. The agent uses dead reckoning technique to obtain its location during navigation [11]. It is assumed that the number and location of the obstacles are unknown to the agent. The location of the obstacles are considered to be a random vector uniformly distributed in a rectangular environment with length L_x and L_y. The agent starts the navigation from a known location $\mathbf{r}_\mathrm{s} = [x_\mathrm{s}, y_\mathrm{s}]^T$ and it is equipped with a range measurement sensor that measures the distances to the obstacles.

The detection range of the agent's sensor is denoted by ρ_rn. The agent can detect only the obstacles within its detection range in each navigation step. Thus, the number of detected obstacles varies during navigation. The navigation is composed of a measurement step followed by path planning, where it is finding a geometrical path from the current location of the agent to its next location while avoiding obstacles in the environment.

3. Artificial Potential Fields (APF)

In this section, we briefly overview classical APF methods for unknown cluttered environments. In general, the APF methods works on the basis that a target is attractive while an obstacle is repulsive. The agent is to be attracted by the goal point and hence moves towards it while changing direction whenever an obstacle is encountered. An example for this is a mobile electrical charge moving in a field of static similar charges while being attracted by a static opposite charge. The advantage of this method is that it takes into account the effect of multiple obstacles at once which makes it perfect for cluttered environments which are the scope of this research.

Classical APF (CAPF) Algorithm

The original APF method proposed in 1985 [12] uses geometric calculations at each iteration to determine the direction and speed the agent should move on in order to avoid the obstacles and reach the target safely. It was originally proposed for robotic arm manipulators but has been extended to mobile agents afterwards [13].

The CAPF method was originally designed for structured environments. It can still work in cluttered environments but is far from optimal and often gets stuck in local minimas and collides with obstacles. Hence, the need arises for a new method that is dedicated for cluttered environments and that is the purpose of this paper.

Bacteria APF (BAPF) Algorithm

The principle idea of the BAPF method for autonomous navigation is illustrated in Figure 1 and is adopted from [14]. As seen, it is based on the idea of having a point agent surrounded by a circle of possible future position points called bacteria points. There are potential functions that are calculated for those points based on the distance from the target and from the detected obstacles. Based on those potential functions a certain bacteria point is selected to be the next position of the agent, and the agent moves to it. This process is repeated during navigation [14]. This method shall

[2]Extension to 3D case is straightforward.

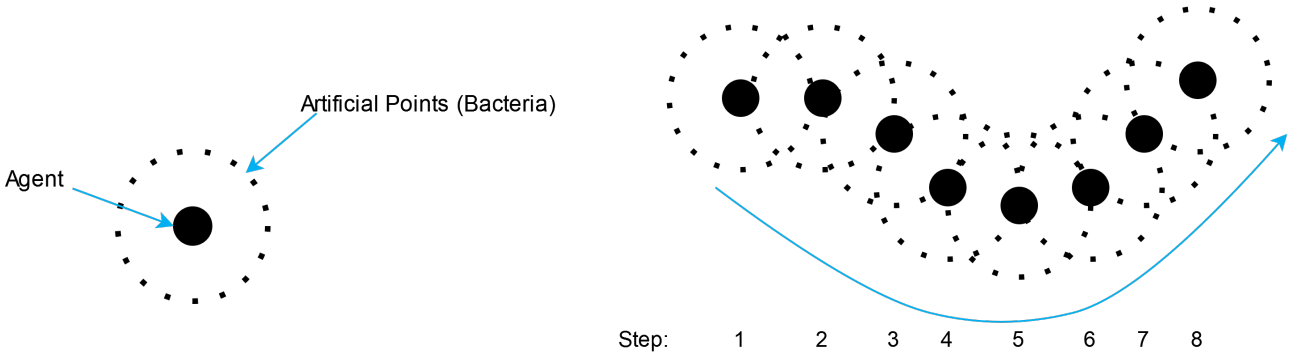


Figure 1: Agent and bacteria points around it (left) Agent movement by selecting different bacteria points (right).

overcome the limitations that the CAPF method faces when it comes to unknown cluttered environments in terms of the agent getting stuck between the scattered obstacles [13].

In the BAPF algorithm, the potential function from the target located at $\mathbf{r}_\mathrm{t} = [x_\mathrm{t}, y_\mathrm{t}]^T$ to the agent located at $\mathbf{r} = [x, y]^T$ is given by

$$J_\mathrm{t}(\mathbf{r}) \triangleq -\alpha_\mathrm{t} \exp(-\mu_\mathrm{t} \rho_\mathrm{t}(\mathbf{r})), \tag{1}$$

where $\rho_\mathrm{t}(\mathbf{r}) \triangleq \|\mathbf{r} - \mathbf{r}_\mathrm{t}\|_2^2$ is the square distance between the agent and target, and α_t and μ_t are constant parameters empirically determined based on the size of the environment and the initial distance between the agent and the target.

Similarly, the repulsive potential function from the nth detected obstacle located at $\mathbf{r}_\mathrm{o}^n = [x_\mathrm{o}^n, y_\mathrm{o}^n]^T$ to the agent located at $\mathbf{r} = [x, y]^T$ is defined as

$$J_\mathrm{o}^n(\mathbf{r}) \triangleq \alpha_\mathrm{o} \exp(-\mu_\mathrm{o} \rho_\mathrm{o}^n(\mathbf{r})), \tag{2}$$

where $\rho_\mathrm{o}^n(\mathbf{r}) \triangleq \|\mathbf{r} - \mathbf{r}_\mathrm{o}^n\|_2^2$ is the square distance between the agent and the nth obstacle, and α_o and μ_o are constant parameters determined through experimentation based on the estimated density of obstacles in the navigation region.

Let N_d denote the total number of detected obstacles by the agent at the location of $\mathbf{r} = [x, y]^T$. The total repulsive potential function from the N_d detected obstacles at the location of the agent $\mathbf{r} = [x, y]^T$ is given by

$$J_\mathrm{o}(\mathbf{r}) = \sum_{n=1}^{N_\mathrm{d}} J_\mathrm{o}^n(\mathbf{r}), \tag{3}$$

where $J_\mathrm{o}^n(\mathbf{r})$ is given in (2). By using (1) and (3), the total potential function at the agent's location $\mathbf{r} = [x, y]^T$ can be written as

$$J_\mathrm{a}(\mathbf{r}) = J_\mathrm{t}(\mathbf{r}) + J_\mathrm{o}(\mathbf{r}). \tag{4}$$

Let N_b and $\mathbf{r}_{\mathrm{b},k} = [x_{\mathrm{b},k}, y_{\mathrm{b},k}]^T$, $k = 1, 2, \ldots, N_\mathrm{b}$, denote the total number of bacteria points and the location of the kth bacteria point around $\mathbf{r} = [x, y]^T$ respectively. The total potential function at the location of the kth bacteria point around $\mathbf{r} = [x, y]^T$ is expressed as

$$J_k(\mathbf{r}_{\mathrm{b},k}) = J_\mathrm{t}(\mathbf{r}_{\mathrm{b},k}) + J_\mathrm{o}(\mathbf{r}_{\mathrm{b},k}). \tag{5}$$

The criterion for the movement of the agent located at $\mathbf{r} = [x, y]^T$ to the kth bacteria point at $\mathbf{r}_{\mathrm{b},k} = [x_{\mathrm{b},k}, y_{\mathrm{b},k}]^T$ is given by

$$J_k(\mathbf{r}_{\mathrm{b},k}) - J_\mathrm{a}(\mathbf{r}) < 0, \tag{6}$$

which means that the kth bacteria point is at a lower total potential towards the target than the agent. Recall from (1) and (2) that the target potential is negative while the repulsive potential from the obstacles is positive.

Figure 2 illustrates the flow graph of the BAPF algorithm. As seen, a linear search algorithm on the bacteria points is performed to select the best bacteria point to move to. The flow graph of the search algorithm is shown in Figure 3. As seen, the search starts with testing the bacteria point closest to the target. If that point meets the potential criteria in (6), then the agent moves to it without testing other bacteria points. If the bacteria point does not meet the potential criteria then the second closest bacteria point to the target is tested. This procedure is repeated until a bacteria point meets the potential criteria in (6). When a bacteria point is selected it becomes the next point that the agent should move to. The distance between the agent and the bacteria points is constant and is defined as the movement step size of the agent. It is determined empirically given the type of environment and the detection range of the agent's sensor.

Local Minima Trap—The major drawback of the APF methods is that they can get stuck in a local minima where the agent cannot find a solution to the problem on a certain area on the navigation region. The local minima trap is sometimes inevitable in the case of unknown environments and especially unknown cluttered environments. For the BAPF algorithm, the local minima trap occurs if none of the N_b bacteria points meets the criteria in (6). In this case, the agent cannot move.

To escape the local minima trap, a random walk can be employed where a bacteria point is selected randomly from the N_b bacteria points, and the agent to move to it [15]. A strict condition for this randomly selected bacteria point is that it should not violate a minimum safety distance to a detected obstacle which will be further discussed later on. Let $q \in \{1, 2, \ldots, N_\mathrm{b}\}$, denote the index of the selected bacteria point for the next movement through the random walk. The mathematical formulation of the random walk movement can

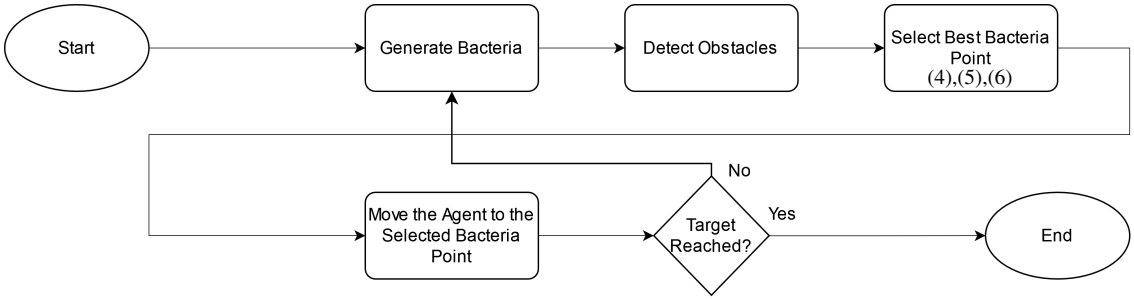


Figure 2: BAPF algorithm flow graph for agent navigation.

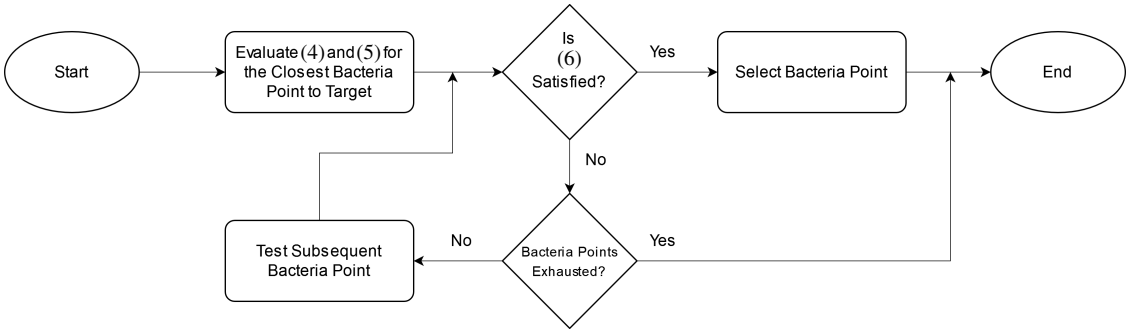


Figure 3: BAPF algorithm flow graph for best bacteria point selection.

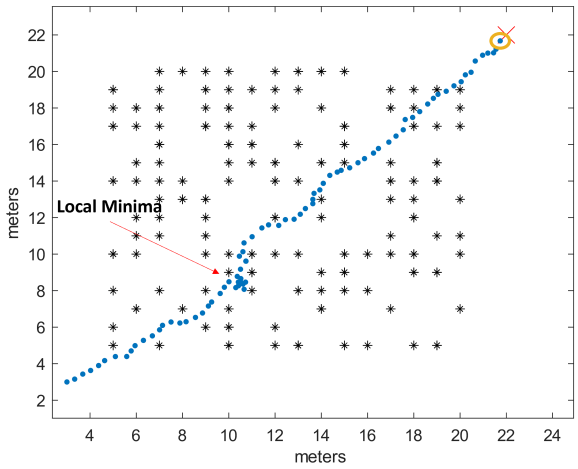


Figure 4: Local minima escape through random walk

be expressed as

$$q \sim \mathcal{U}\{\mathcal{Q}\}, \tag{7}$$

where

$$\mathcal{Q} = \left\{ k \in \{1, 2, \ldots, N_\mathrm{b}\} \,|\, \sqrt{\rho_\mathrm{o}^n(\mathbf{r}_{\mathrm{b},k})} \geqslant \rho_\mathrm{l} \right\} \tag{8}$$

and $\mathcal{U}\{\mathcal{Q}\}$ returns an element of set $\mathcal{Q}$ randomly, $\sqrt{\rho_\mathrm{o}^n(\mathbf{r}_{\mathrm{b},k})} = \|\mathbf{r}_{\mathrm{b},k} - \mathbf{r}_\mathrm{o}^n\|_2$, N_b is the number of bacteria points, and ρ_l denotes the safety distance as it is explained in the next section. Figure 4 shows the agent escaping a local minima trap via random walk.

4. Improved APF Algorithms

In this section, we propose the Adaptive BAPF (A-BAPF) and the Changing Radii BAPF (CR-BAPF) algorithms to improve the performance of the BAPF algorithm for unknown cluttered environments.

A-BAPF Algorithm

The BAPF algorithm cannot achieve the best performance in unknown cluttered environments when it comes to the convergence to the target (successful navigation). Performance improvement in terms of convergence to the target can be achieved by employing our proposed A-BAPF method that is explained in this section.

The idea of A-BAPF is to adaptively choose the value of μ_o for each bacteria point given a predetermined value of α_o. We define the modified total bacteria function as

$$\bar{J}_k(\mathbf{r}_{\mathrm{b},k}, \mu_{\mathrm{o},k}) \triangleq J_\mathrm{t}(\mathbf{r}_{\mathrm{b},k}) + \bar{J}_\mathrm{o}(\mathbf{r}_{\mathrm{b},k}, \mu_{\mathrm{o},k}), \tag{9}$$

where

$$\bar{J}_\mathrm{o}(\mathbf{r}_{\mathrm{b},k}, \mu_{\mathrm{o},k}) \triangleq \sum_{n=1}^{N_\mathrm{d}} \alpha_\mathrm{o} \exp(-\mu_{\mathrm{o},k} \rho_\mathrm{o}^n(\mathbf{r}_{\mathrm{b},k})), \tag{10}$$

and

$$\begin{aligned} \hat{\mu}_{\mathrm{o},k} = \arg\min_{\mu_{\mathrm{o},k}} \quad & \bar{J}_k(\mathbf{r}_{\mathrm{b},k}, \mu_{\mathrm{o},k}) \\ \text{subject to} \quad & \mu_{\mathrm{o},k} \in [\mu_\mathrm{min} : \mu_\mathrm{max}] \end{aligned} \tag{11}$$

for $k = 1, 2, \ldots, N_\mathrm{b}$, and μ_min and μ_max are the lower and upper values of $\mu_{\mathrm{o},k}$. Now, the criterion for the movement in (6) is replaced by

$$\bar{J}_k(\mathbf{r}_{\mathrm{b},k}, \hat{\mu}_{\mathrm{o},k}) - J_\mathrm{a}(\mathbf{r}) < 0. \tag{12}$$

One can easily show that $\bar{J}_k(\mathbf{r}_{\mathrm{b},k}, \mu_{\mathrm{o},k})$ in (11) is a convex function of $\mu_{\mathrm{o},k}$. The solution of the minimization in (11)

can be obtained by using grid search or golden-section search methods. Intuitively, when the value of $\mu_{\mathrm{o},k}$ approaches $\mu_{\max}$, the repulsive potential, $\bar{J}_{\mathrm{o}}(\mathbf{r}_{\mathrm{b},k}, \mu_{\mathrm{o},k})$ in (10), approaches zero and the obstacles' influence on the navigation decreases. The minimization of the potential cost function for the bacteria points can almost guarantee a solution for (12) at each navigation step and prevent the agent from ending up in a local minima. This is due to the fact that the optimization in (11) increases the chance of that at least one bacteria point meets the condition in (12). Our proposed A-BAPF algorithm can be considered as an upper bound for the BAPF algorithm in terms of successful navigation performance. In the A-BAPF algorithm, the bacteria point closest to the target can be selected even if it does not initially meet the condition in (6). This is done by changing the value of $\mu_{\mathrm{o},k}$ through (11).

CR-BAPF Algorithm

In the previously discussed APF methods, the potential from a detected obstacle is always affecting the agent during navigation. In this section, we propose the CR-BAPF algorithm where the repulsive potentials from obstacles affecting the agent change based on regions around each obstacle.

The CR-BAPF algorithm introduces two radii around each obstacle as shown in Figure 5. For the CR-BAPF algorithm, we define the repulsive potential function from the nth detected obstacle located at $\mathbf{r}_{\mathrm{o}}^n = [x_{\mathrm{o}}^n, y_{\mathrm{o}}^n]^T$, $n = 1, 2, \ldots, N_{\mathrm{d}}$, to the agent located at $\mathbf{r} = [x, y]^T$ as

$$J_{\mathrm{o}}^n(\mathbf{r}) = \begin{cases} 0 & \sqrt{\rho_{\mathrm{o}}^n(\mathbf{r})} > \rho_{\mathrm{u}} \\ \alpha_{\mathrm{o}} \exp(-\mu_{\mathrm{o}} \rho_{\mathrm{o}}^n(\mathbf{r})) & \rho_{\mathrm{l}} \leqslant \sqrt{\rho_{\mathrm{o}}^n(\mathbf{r})} \leqslant \rho_{\mathrm{u}}, \\ \infty & \sqrt{\rho_{\mathrm{o}}^n(\mathbf{r})} < \rho_{\mathrm{l}} \end{cases} \quad (13)$$

where $\rho_{\mathrm{o}}^n(\mathbf{r}) \triangleq \|\mathbf{r} - \mathbf{r}_{\mathrm{o}}^n\|_2^2$ is the square distance between the agent and the nth obstacle, and ρ_{l} and ρ_{u} denote the lower and upper radii, respectively. As seen in (13), if the agent's distance to the nth detected obstacle is greater than ρ_{u}, then the repulsive potential is zero. On the other hand, if the agent's distance to the nth obstacle is lower than ρ_{l}, the repulsive potential from the obstacle to the agent becomes infinity. This means that the agent does not select a bacteria point that its distance to the nth obstacle is lower than ρ_{l} because it does not satisfy the criteria in (6). Moreover, this also implies that the agent is not affected by obstacles that are at a greater distance than ρ_{u}. This way, the agent has more motion freedom in the map without risking its safety which improves the overall performance. The lower radius is meant as a safety perimeter around the obstacles to prevent collisions. The value of ρ_{l} is determined based on the error margin in the motion of the agent in order to prevent the agent from colliding with the obstacle. Moreover, the value of ρ_{u} is determined based on the estimated density of the obstacles in the navigation region. It should be mentioned that the attractive potential function to the target for the CR-BAPF is given in (1).

CR-BAPF Algorithm*

To further improve the performance of the proposed CR-BAPF algorithm in terms of avoiding the local minima trap discussed in 3, we propose to employ the CR-BAPF algorithm in combination with the random walk technique. We call this new algorithm CR-BAPF*.

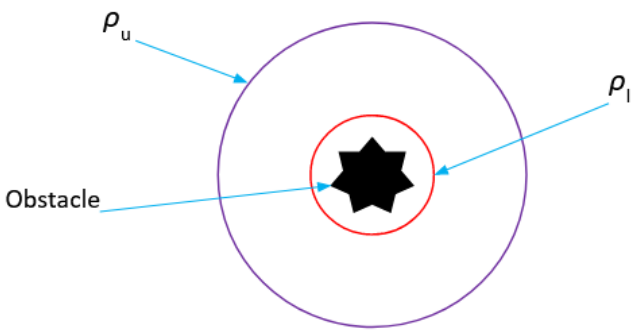


Figure 5: The upper and lower radii around a point obstacle.

Table 1: **Simulation hyperparameter values.**

Hyperparameter	Value
ρ_{rn}	8 m
α_t	10^4
μ_{t}	1
α_{o}	1
μ_{o}	1000
ρ_{l}	0.4 m
ρ_{u}	4.5 m

5. SIMULATIONS

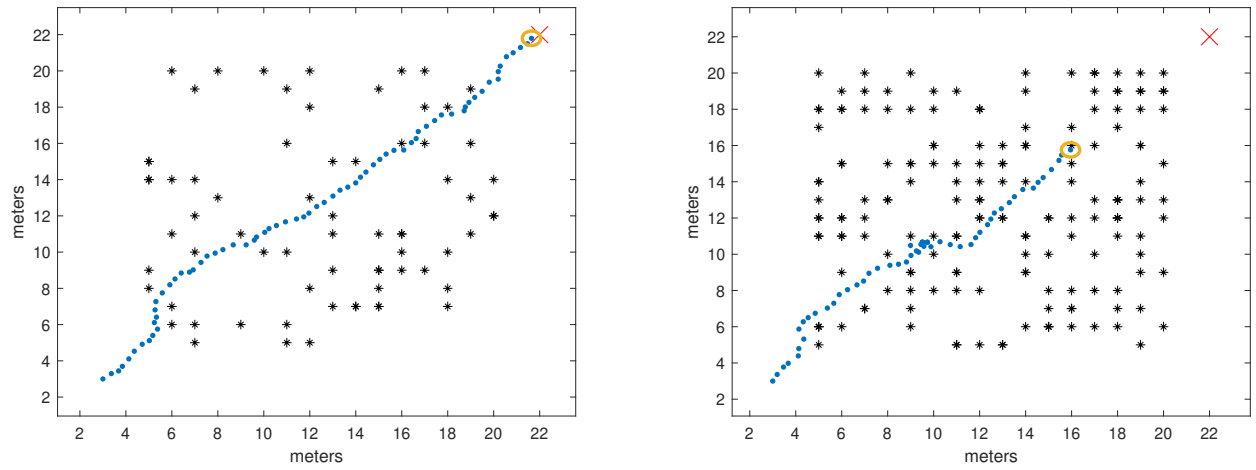
Simulation Setup

The simulation environment is setup in MATLAB with a fixed starting position for the agent at $\mathbf{r}_{\mathrm{s}} = [3, 3]^T$ meters and the target location at $\mathbf{r}_{\mathrm{t}} = [22, 22]^T$ meters. The location of the obstacles is considered to be a random vector $\mathbf{r}_{\mathrm{o}}^n = [x_{\mathrm{o}}^n, y_{\mathrm{o}}^n]^T$, $n = 1, 2, \ldots, N$, uniformly distributed in a square region with length $L_{\mathrm{x}} = 30$ m and width $L_{\mathrm{y}} = 30$ m. For each Monte Carlo trial, the number of obstacles in the navigation region N is modeled by the discrete uniform random variable $N \in \mathcal{U}[N_{\mathrm{l}}, N_{\mathrm{u}}]$, where N_{l} and N_{u} are the lower and upper limit for the number of obstacles, respectively. The considered simulation setup emulates the navigation through a set of randomly uniformly distributed obstacles such as trees or craters. The detection range is set $\rho_{\mathrm{rn}} = 8$ m, and the number of bacteria points is $N_{\mathrm{b}} = 60$ to provide a rotation accuracy of $6°$.

The movement step size of the agent is $\Delta r \triangleq \|\mathbf{r} - \mathbf{r}_{\mathrm{b},k}\|_2 = 0.4$ m, $k = 1, 2, \ldots, N_{\mathrm{b}}$, and the position errors of the agent Δx and Δy are modeled by a zero mean Gaussian distribution with variance 0.01. The values of the hyperparameters for $N_{\mathrm{m}} = 4000$ Monte Carlo trials are given in Table 1. The upper and lower radii are considered the same for all the obstacles in the navigation region.

Performance Metrics

The following terms and definitions will be used in order to compare different algorithms.



(a) Successful navigation

(b) Unsuccessful navigation

Figure 6: Successful and unsuccessful navigation of an agent employing CR-BAPF* algorithm.

Rate of Success ($R_{\rm s}$)—For $N_{\rm m}$ Monte Carlo trials, the rate of successful navigation is defined as

$$R_{\rm s} \triangleq \frac{N_{\rm s}}{N_{\rm m}}, \tag{14}$$

where $N_{\rm s}$ denotes the number of successful navigation without getting stuck in a local minima or colliding with an obstacle.

Average Navigation Steps ($\bar{M}_{\rm s}$)—For $N_{\rm s}$ successful navigation trials, the average navigation steps is defined as

$$\bar{M}_{\rm s} = \frac{1}{N_{\rm s}} \sum_{m=1}^{N_{\rm s}} M_{\rm s}(m), \tag{15}$$

where $M_{\rm s}(m)$ is the number of navigation steps at the mth successful navigation trial.

Safety Parameter (S)—For a successful navigation of $M_{\rm s}$ steps at locations $\mathbf{r}_1, \mathbf{r}_2, \ldots, \mathbf{r}_{M_{\rm s}}$ with a total number of N unique detected obstacles during navigation, the safety parameter is the average minimum distance that the agent maintained from the detected obstacles during the navigation and is defined as

$$S \triangleq \frac{\sum_{m=1}^{N} \rho_{\min}(\mathbf{r}_{\rm o}^m)}{N}, \tag{16}$$

where $\rho_{\min}(\mathbf{r}_{\rm o}^m) \triangleq \min\left\{\|\mathbf{r}_1 - \mathbf{r}_{\rm o}^m\|_2, \|\mathbf{r}_2 - \mathbf{r}_{\rm o}^m\|_2, \ldots, \|\mathbf{r}_{M_{\rm s}} - \mathbf{r}_{\rm o}^m\|_2\right\}$ with $\mathbf{r}_{\rm o}^m$ is the location of the mth obstacle detected throughout the navigation.

Algorithm Complexity ($T_{\rm a}$)—The computational complexity of the algorithms is evaluated in terms of average run time of the algorithm for $N_{\rm m}$ Monte Carlo trials, and it is denoted by $T_{\rm a}$.

Simulation Results

Figure 6 shows an example of a successful navigation of the agent employing the proposed CR-BAPF* algorithm. An unsuccessful navigation where the agent gets stuck in a local minima is also shown in Figure 6. As expected, by increasing the density of the obstacles in the navigation region, the chance of successful navigation decreases.

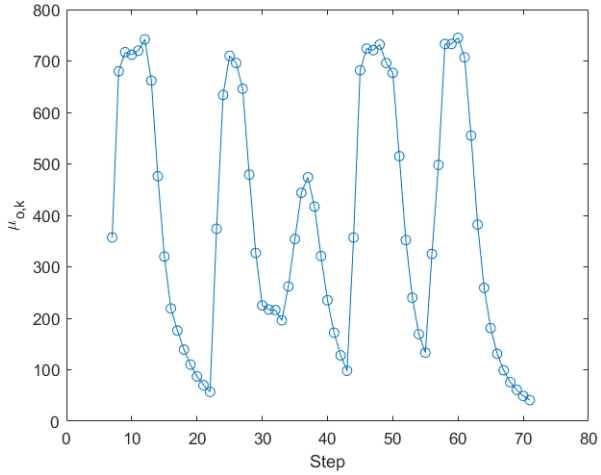


Figure 7: Optimized $\mu_{{\rm o},k}$ for the selected bacteria point versus navigation step for the proposed A-BAPF algorithm for a single successful run.

Figure 7 illustrates the optimized value of $\mu_{{\rm o},k} \in [1:1000]$ for the selected bacteria point versus navigation step for the proposed A-BAPF algorithm in a single successful run. As seen, there are sharp changes in the curve because the environment the agent is moving in is a cluttered environment and thus the number of detected obstacles, the distance from the detected obstacles, and the distance from the target significantly changes during the navigation. Our simulation result show that the A-BAPF algorithm can achieve a navigation success rate of more than 95%.

Figure 8 compares the rate of success $R_{\rm s}$ versus $\mu_{\rm o}$ for the proposed CR-BAPF* and the BAPF algorithm. As seen, the CR-BAPF* algorithm outperforms the BAPF algorithm for

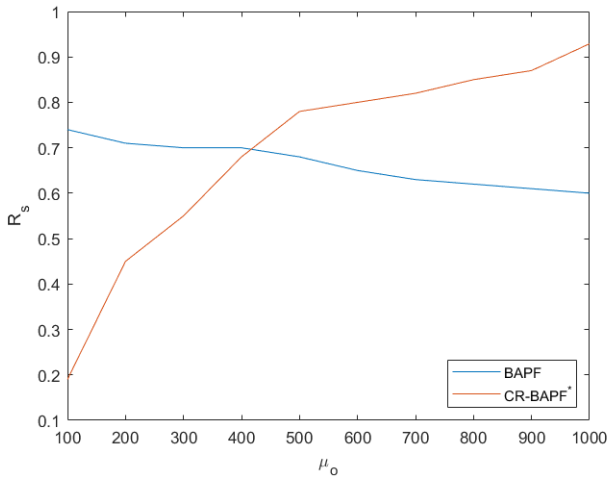


Figure 8: Rate of success R_{s} versus μ_{o} for the proposed CR-BAPF* and the BAPF algorithms.

$\mu_{\mathrm{o}} \in [416.67, 1000]$. The reason is that as μ_{o} increases, the repulsive potential from the obstacles decreases. In the BAPF algorithm, the repulsive potential depends only on the distance between the obstacle and the agent. When μ_{o} increases, the risk of the agent getting close to an obstacle and collision increase. However, in the CR-BAPF* algorithm there are defined perimeters around each obstacle which control the repulsive potential value and prevent the agent from getting dangerously close to an obstacle. On the other hand, for $\mu_{\mathrm{o}} \in [1, 416.67)$, the repulsive potential is higher for the CR-BAPF*. This results in a higher success rate for the BAPF algorithm because it becomes more likely for the CR-BAPF* algorithm to reach a local minima due to the high regional repulsive potential around each obstacle.

The comparison between the CAPF, the BAPF, and the proposed CR-BAPF and the CR-BAPF* algorithms for different performance metrics including R_{s}, $\bar{M}_{\mathrm{s}}$, S, and T_{a} is shown in Table 2 for different obstacle densities. As seen, the proposed CR-BAPF* algorithm outperforms the other algorithms in terms of success rate R_{s}. The reason is that the proposed branching repulsive potential function and implementation of the random walk reduces the local minima trap and the chance of collision. Moreover, it can be seen that the success rate decreases as the number of obstacles in the environment increases which is a straightforward result. In terms of average navigation steps $\bar{M}_{\mathrm{s}}$, the number of steps generally increases for all methods when the number of obstacles in the environment increases. All the four algorithms perform well when it comes to the safety criteria S because they offer $S > 2$ m overall. The CR-BAPF* algorithm has a longer execution time compared to the others due to the random walk implementation which imposes extra steps to get out of a local minima but it yields a much higher success rate.

Table 2: **Performance comparison between the four APF methods for different obstacle densities**

(a) $N \in \mathcal{U}[20, 45]$

Algorithm	R_{s}	$\bar{M}_{\mathrm{s}}$	S (m)	T_{a} (ms)
CAPF [13]	0.333	91.26	2.71	26.80
BAPF [14]	0.739	68.25	2.37	26.30
CR-BAPF	0.770	68.39	2.38	27.05
CR-BAPF*	**0.935**	70.47	2.35	31.63

(b) $N \in \mathcal{U}[45, 70]$

Algorithm	R_{s}	$\bar{M}_{\mathrm{s}}$	S (m)	T_{a} (ms)
CAPF [13]	0.170	108.49	3.22	52.72
BAPF [14]	0.552	77.82	2.40	66.87
CR-BAPF	0.490	69.24	2.42	44.18
CR-BAPF*	**0.873**	76.54	2.38	65.95

(c) $N \in \mathcal{U}[70, 95]$

Algorithm	R_{s}	$\bar{M}_{\mathrm{s}}$	S (m)	T_{a} (ms)
CAPF [13]	0.157	111.93	3.53	71.64
BAPF [14]	0.407	70.62	2.43	59.84
CR-BAPF	0.270	70.22	2.52	60.43
CR-BAPF*	**0.812**	83.36	2.44	108.65

6. Summary

We proposed the CR-BAPF algorithm to solve the local minima trap problem in the classical APF algorithms. The CR-BAPF takes the advantage of a branching repulsive potential function, which resulted in a navigation success rate of more than 75% in lightly cluttered environments. This value increased to more than 90% when the CR-BAPF algorithm was combined with the random walk technique (CR-BAPF*). Our simulation results showed that the BAPF, the CR-BAPF, and the CR-BAPF* algorithms all take less navigation steps to converge compared to the CAPF algorithm. Furthermore, the simulation results showed that all four algorithms maintain a good safety distance from the obstacles and that the corresponding execution times of the algorithms are acceptable. With the implementation of the random walk, the CR-BAPF* outperforms the other algorithms at the expense of a higher execution time. In addition to the CR-BAPF algorithm, we also proposed the A-BAPF algorithm that is an adaptive version of the BAPF algorithm and achieves the upper bound performance of the BAPF. In the future, we aim to implement these algorithms in field agents to obtain experimental validation of the proposed solutions.

7. Acknowledgements

Mosab Diab is partially funded by the TU Delft | Global Initiative. Mostafa Mohammadkarimi and R.T.Rajan are partially funded by the European Leadership Joint Undertaking (ECSEL JU), under grant agreement No. 876019, the ADACORSA (Airborne Data Collection on Resilient System Architectures) project. The code developed is provided in a repository at `https://github.com/M-DIAB/APF`.

References

[1] S. Azadi, R. Kazemi, and H. R. Nedamani, "Chapter 10 - trajectory planning of tractor semitrailers," in *Vehicle Dynamics and Control*, S. Azadi, R. Kazemi, and H. R. Nedamani, Eds. Elsevier, 2021, pp. 429–478.

[Online]. Available: https://www.sciencedirect.com/science/article/pii/B9780323856591000100

[2] J. N. Yasin, S. A. S. Mohamed, M.-H. Haghbayan, J. Heikkonen, H. Tenhunen, and J. Plosila, "Unmanned aerial vehicles (uavs): Collision avoidance systems and approaches," *IEEE Access*, vol. 8, 2020, pp. 105 139–105 155, June 2020.

[3] N. Gageik, P. Benz, and S. Montenegro, "Obstacle detection and collision avoidance for a uav with complementary low-cost sensors," *IEEE Access*, vol. 3, pp. 599–609, 2015.

[4] J. Seo, Y. Kim, S. Kim, and A. Tsourdos, "Collision avoidance strategies for unmanned aerial vehicles in formation flight," *IEEE Transactions on Aerospace and Electronic Systems*, vol. 53, no. 6, pp. 2718–2734, 2017.

[5] M. Wang, H. Voos, and D. Su, "Robust online obstacle detection and tracking for collision-free navigation of multirotor uavs in complex environments," *Proc. 15th Int. Conf. Control, Automat., Robot. Vis. (ICARCV)*, pp. 1228–1234, November 2018.

[6] D. Bareiss and J. van den Berg, "Reciprocal collision avoidance for robots with linear dynamics using lqr-obstacles," in *2013 IEEE International Conference on Robotics and Automation*, 2013, pp. 3847–3853.

[7] S. Hrabar, "Reactive obstacle avoidance for rotorcraft uavs," in *2011 IEEE/RSJ International Conference on Intelligent Robots and Systems*, 2011, pp. 4967–4974.

[8] P. Conroy, D. Bareiss, M. Beall, and J. Berg, "3-d reciprocal collision avoidance on physical quadrotor helicopters with on-board sensing for relative positioning," 11 2014.

[9] S. Roelofsen, D. Gillet, and A. Martinoli, "Reciprocal collision avoidance for quadrotors using on-board visual detection," in *2015 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, 2015, pp. 4810–4817.

[10] J. Sun, J. Tang, and S. Lao, "Collision avoidance for cooperative uavs with optimized artificial potential field algorithm," *IEEE Access*, vol. 5, pp. 18 382–18 390, 2017.

[11] C. Randell, C. Djiallis, and H. Muller, "Personal position measurement using dead reckoning," 11 2005, pp. 166– 173.

[12] O. Khatib, "Real-time obstacle avoidance for manipulators and mobile robots," *Int. Conf. Robot. Automat.*, vol. 2, pp. 500–505, March 1985.

[13] J. C. Mohanta, D. R. Parhi, S. K. Patel, and S. K. Pradhan, "Real-time motion planning of multiple mobile robots using artificial potential field method," *Journal of Advance Computational Research*, vol. 01, 01 2016.

[14] Y. Ahmad. (2020, 05) Matlab implementation of artificial potential field. [Online]. Available: https://github.com/Yaaximus/artificial-potential-field-matlab

[15] F. Xia, J. Liu, H. Nie, Y. Fu, L. Wan, and X. Kong, "Random walks: A review of algorithms and applications," *IEEE Transactions on Emerging Topics in Computational Intelligence*, vol. PP, pp. 1–13, 11 2019.

BIOGRAPHY

Mosab Diab *received his B.Sc. degree from the University of Khartoum in Sudan in 2018 and his M.Sc. degree from the Delft University of Technology (TUD) in the Netherlands in 2022. Developed an interest for control systems and robotics early on, which culminated in him conducting research on myoelectric prosthetic hands for his bachelor's graduation project at the Center for Neuroscience and Biomedical Engineering at the University of Electro-Communications in Tokyo, Japan. For his master's thesis, he worked on path planning algorithms for autonomous agents in unknown cluttered environments.*

Mostafa Mohammadkarimi *received the M.Sc. degree in electrical engineering from K. N. Toosi University of Technology, Tehran, Iran, in 2011, and the Ph.D. degree in computer engineering from Memorial University, Canada, in 2017. From March 2016 to April 2017, he was a visiting Ph.D. student at Massachusetts Institute of Technology (MIT). From February 2018 to June 2021, he was a Post-Doctoral Fellow with the University of Alberta, Edmonton, AB, Canada, The University of British Columbia, Vancouver, BC, Canada, Friedrich-Alexander-Universität Erlangen-Nürnberg, Erlangen, Germany. Since September 2021, he has been with the Signal Processing Systems Group, Department of Microelectronics, Delft University of Technology (TUD), Delft, The Netherlands. His research interests include wireless communications, statistical signal processing information theory, and localization & navigation.*

Raj Thilak Rajan *(S'11, M'17, SM'22) is an Assistant Professor with the Signal Processing Systems (SPS) group at the Faculty of Electrical Engineering, Mathematics and Computer Science (EEMCS) in the Delft University of Technology (TUD), and the Co-director of the Delft Sensor AI Lab. He is an elected member of the IAF SCAN (Space communications and Navigation) committee, the IEEE ASI (Autonomous Systems Initiative), and is the Associate Editor of the IEEE OJSP (Open Journal on Signal Processing). His research interests lie in statistical machine learning and distributed optimisation, with applications to autonomous systems.*