Nexus : Journal of Innovative Studies of Engineering Science (JISES)
Volume: 02 Issue: 04 | 2023 ISSN: 2751-7578
http://innosci.org/

Surveying Path-Planning Algorithms for Reliable Navigation of Mobile Robots: A Comprehensive Review of Four Common Approaches

Zahid Jawad Jbren, Jasim Khawwaf
Mechanical Eng., Eng. College, University Kufa, Iraq

Annotation: Robots have become ubiquitous in our daily lives, serving a variety of purposes in various forms. As their prevalence increases, so too does the need for reliable route planning algorithms to ensure safe and efficient navigation without collisions. This has led to a surge of interest in the study and refinement of path planning algorithms, with efforts focused on improving their effectiveness and performance. To shed light on this important topic, this paper aims to provide a comprehensive survey of four common path-planning algorithms for robots. Beginning with an overview of mobile robots and route-planning algorithms, we delve into the fundamental principles of each algorithm, exploring their relative advantages and disadvantages, and their respective applications. This survey aims to provide a comprehensive understanding of the state-of-the-art in path-planning algorithms for mobile robots and assist researchers and practitioners in selecting the most appropriate algorithm for their applications.

Key words: Robots, path-planning algorithms, PSO, optimization, ant colony, bee colony, genetic

1. Introduction

Algorithms have proven to be the most effective tools for addressing a wide range of issues related to robot mobility, particularly in the realm of path planning and navigation. In diverse environments, whether stationary or mobile, with or without fixed obstacles, algorithms have been instrumental in improving and optimizing robot paths. These algorithms have been employed in standalone form or in conjunction with other algorithms, resulting in remarkable performance improvements across a variety of scenarios. Notably, a number of studies have demonstrated the successful optimization of robot path, energy consumption, and processing time using a variety of algorithms. Among the most popular algorithms are the particle swarm algorithm, the ant colony algorithm, the bee colony algorithm, and the genetic algorithm. These algorithms are highly favored due to their ease of use, rapid convergence, and numerous other advantages. As such, they are widely utilized in practice and continue to drive innovation in the field of robotics. Numerous studies have been conducted on path planning using particle swarm optimization (PSO). Zhang et al. [1] presented a more effective PSO for path planning for mobile robots. The efficacy of the method was demonstrated through simulations. Gong et al. [2] proposed a global path planning strategy using a multi-objective PSO, which was also confirmed to be effective through simulations. To address path planning for unmanned aerial vehicles, an enhanced chaos-based PSO was proposed [3], which outperformed conventional PSO, particularly in a three-dimensional setting. Fitness-scaling adaptive Chaotic PSO was suggested [4] to address path planning for unmanned combat aerial vehicles. Liu et al. [5] introduced important technology for radiation environment path planning based on PSO, which was shown to be effective through experiments. Yusof et al. [6] proposed a predetermined waypoints method for mobile robot navigation, while Tang X. et al. [7] used a multi-agent particle filter to handle mapping and localization problems in uncharted areas. PSO was used to reduce computation and maintain more consistent convergence characteristics. Atyabi et al. [8] developed the Area Extended PSO (AEPSO) to deal with time-dependent and dynamic constraint issues in mobile robot navigation, which was effectively used in bomb defusing and search and rescue of survivors. Tang et al. [9] used PSO to address cooperative motion route planning in complex environments, taking into consideration fault tolerance of the suggested approach with multibody system dynamics. Chen et al. [10] used a multi-category classifier to construct a human expert control approach with the ability to learn in uncertain environments. The PSO was employed to achieve higher precision quickly. Local search PSO algorithm [12], fusion of different intelligent optimization algorithms [13], and adjusting the plan [11] were also used to improve the PSO method, resulting in benefits such as quick convergence and global optimization. Back-propagation neural network models based on global best adaptive mutation PSO were developed [14] to estimate welding penetration based on welding characteristic parameters, while the discrete elite PSO technique was researched to successfully find the shortest collision-free welding path [15]. Wen et al. [16] altered ant colony optimization (ACO) to improve overall path planning, while Wang et al. [17] studied global path planning using ACO. Zhu et al. [18] developed an algorithm to improve ACO performance in mobile robot path planning. Gao et al. [19] presented an enhanced ACO for the three-dimensional path planning of mobile robots, while a chaotic ant colony system was proposed to address the problem of mobile robot path planning, which was shown to be superior to the conventional ant colony system according to simulation data. The artificial bee colony algorithm has also been used for feature selection and has been effectively used to solve practical issues in various fields [20-36]. Despite its success, there is still room for improvement, particularly with regard to dimension-dependent problems. To address this, a coevolution framework was suggested [37, 38] that can successfully differentiate the dependent and independent dimensions groups. Several studies have been conducted to demonstrate the efficiency of genetic algorithms in enhancing the trajectory and performance of robots. For instance, Maine et al. [39] modified the route planning for a mobile manipulator using GA. Liu et al. [40] introduced a GA with two layers of encoding to flatten the route layout and improve the capacity to express symbols. Pehlivanoglu et al. [41] proposed a vibrating genetic route planning method. Xu et al. [42] developed an Optimizer to design the path for unmanned aerial vehicles, and simulation results indicate that the proposed method meets the criteria for computing efficiency and solution precision. Tsai et al. [42] proposed PEGA (parallel elite genetic algorithm) for navigation by autonomous robots, and the outcomes demonstrate its efficacy. Tuncer et al. [43] presented an enhanced GA for the dynamic route planning of mobile robots. Qu et al. [44] suggested a better GA with co-evolutionary technique to address the global path planning issue for numerous mobile robots, and simulations show that the strategy is effective. Fei et al. [45] proposed a customized GA for the ideal path planning of mobile robots, while Shorakaei et al. [46] employed a parallel GA to plan the best cooperative path for unmanned aerial vehicles, and numerous simulations demonstrated its efficacy. Genetic algorithms are now widely used in mobile robots [47-51], scheduling issues [52-55], sensor networks [56], building trade systems [57], logistics [58, 59], automobile industry [60], and cloud computing [61-64]. The references already mentioned frequently highlight the rise in GA convergence speed and application adjustments that aid the case study at hand.

2. Partial swarm optimization (PSO)

In 1995, PSO was proposed by Kennedy and Eberhart as a new idea that belongs to the evolutionary computing methods [65]. Researchers have been inspired by the social behavior of certain creatures such as bird migration and fish schooling to investigate how inter-species cooperation can affect group objectives. This has led to the study of bird dynamics drawings. The problem space of a PSO system is initially populated with a random collection of solutions, which coexist and collaborate simultaneously in order to find the best solution. Each potential solution candidate explores the problem space in search of a "specific" solution (similar to how a flock of birds would search for food). Over time, the particle is influenced by its neighboring particles.

Nexus : Journal of Innovative Studies of Engineering Science (JISES)
Volume: 02 Issue: 04 | 2023 ISSN: 2751-7578
http://innosci.org/

Swarm optimization is employed to reduce the total route planning time while avoiding obstacles. Swarm robots were originally developed to enable a group of mobile robots to achieve a common goal. The concept of optimizing to achieve a goal by overcoming a challenging condition is introduced in order to tackle more complex challenges. Recently, these methods have been applied to autonomous mobile robot applications to address issues with parameter estimates, machine learning tasks, work scheduling, and reduced responsiveness during travel. To provide an optimal outcome that is more efficient, there are several obstacles in the workspace that must be overcome for robot motion planning.

2.1. The following key terms are used in particle swarm optimization (PSO):

- Particles: Refers to any potential solution.
- Population (N): The group of all particles.
- Search Space ([a,b]): Represents all possible solutions to the problem.
- Each particle is characterized by two properties: position (xi) and velocity (vi).
- Additionally, each particle keeps track of its personal best (pbest) and the global best (gbest).

Note: pbest refers to the individual's best position or location achieved thus far, whereas gbest pertains to the best position attained by any individual in the entire population during the search process in the solution space. Global best g_{best}

2.2 Three stages are considered in PSO:

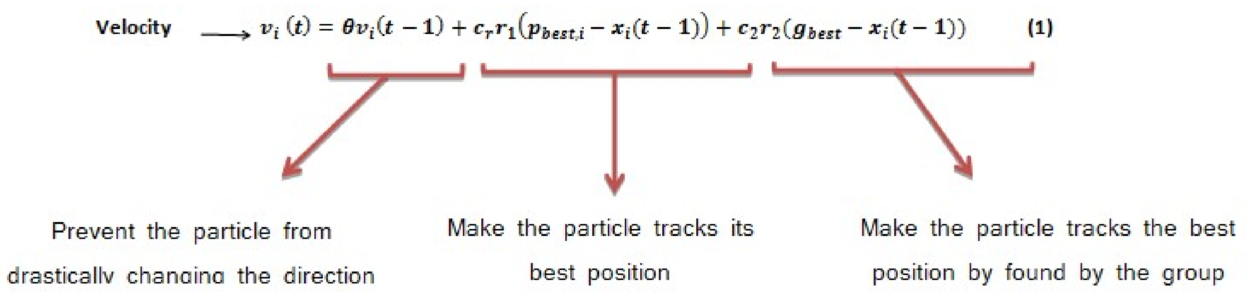
✓ **Initialization**

- Initial population (number of particles N)
- Initial position (x) and initial velocity(v)
- Assign p_{best}, and g_{best}(based on the objective function)

✓ **Update**

- Velocity and position of each particle are updated.

Velocity $v_i(t) = \theta v_i(t-1) + c_r r_1(p_{best,i} - x_i(t-1)) + c_2 r_2(g_{best} - x_i(t-1))$ (1)



Inertia weight is a proportional agent that is related with the speed of last inertia weight θ is a proportional agent that related with the speed of last improvement; the value of θ is assumed to vary linearly from 0.9 to 0.4. c_1And $c_2$are the cognitive (individual) and social (group) learning rates. r_1 And r_2 are uniformly distributed random numbers in the range 0 and 1 x_i

If $x_i(t) > x_{ub} \longrightarrow x_i(t) = x_{ub}$

$x_i(t) > x_{lb} \longrightarrow x_i(t) = x_{lb}$

Position $\rightarrow x_i(t) = x_i(t-1) + v_i(t)$ (2)

- Fitness function f_{xi}
- The $p_{best.i}$ and g_{best} are updated.

$p_{best.i} = x_i$ if f_{xi} better $f_{p\ best.i}$

$g_{best} = p_{best.i}$ if $f_{pbest.i}$ better g_{best}

Case 1: Min

x = 5 $\rightarrow$ $f(x)$ = 50

p_{best}3 $\rightarrow$ $f(Poest)$ = 30

g_{best}= 7 $\rightarrow$ $f(best)$ = 20

Case 2: Min

x = 2 $\rightarrow$ $f(x)$ = 25

p_{best} =3 $\rightarrow$ $f(p_{best})$ = 30

$p_{best} = x$

g_{best}= 7 $\rightarrow$ $f(g_{best})$ = 20

$X = 9 \rightarrow f(x) = 10$

$p_{best} = 3 \rightarrow f(p_{best}) = 30$

$p_{best} = x$

$g_{best} = 7 \rightarrow f(g_{best}) = 20$

$g_{best} = p_{best}$

$\rightarrow$ $f(x)$

p_{best} = $\rightarrow$ $f(p_{best})$

$p_{best} = x$

g_{best} $\rightarrow$ $f(g_{best})$

$g_{best} = p_{best}$

Table (1) cases of $p_{beast\ and\ g_{beast}}$

Cases	Better than p_{best}	Better than g_{best}	Remark
1	×	×	Not update p_{best} and g_{best}
2	√	×	update p_{best} and not update g_{best}
3	√	√	update p_{best} and g_{best}

✓ **Termination**

- The steps of PSO algorithm are iteratively repeated until the maximum number of generations is reached or a termination criterion is met.
- Convergence: is the case where the positions of all particles converge to the same Set of values, the method is assumed to have converged.

2.3. Parameters required from user

- Population size (N)
- Initial position (x) and initial velocity (v)
- Inertia weight (θ)
- The individual and social cognitive ($c_1$andc_2)
- Uniformly distributed random numbers ($r_1$andr_2) in the range (0.1)
- Termination criteria (i.e. number of iteration T)

2.4. Pseudo code

2.4.1. input

- ✓ Objective function (fitness function). Upper bound (ub) and lower bound (lb), population size (N), inertia weight(θ) individual and social captive (c_1 and c_2). Number of iteration T

2.4.2. Initialization

- ✓ Initialize random position(x) mod velocity(v) within search space boundary
- ✓ Assign p_{beast} and g_{best}(based on the objective function)

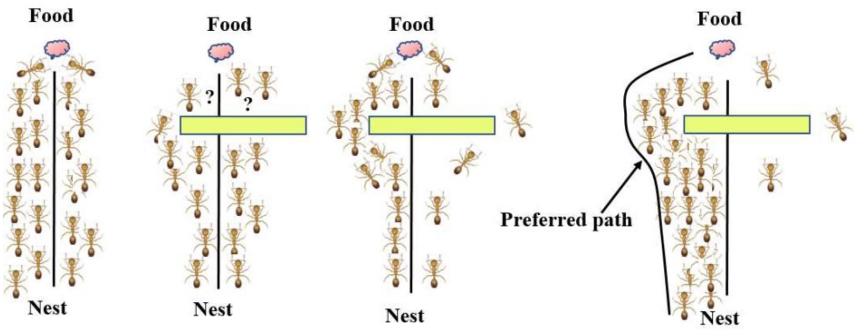
2.4.3. Loop

- ✓ For t = 1:T
- ✓ For i = 1:N
- ✓ Update velocity $v_i\,(t) = \theta v_i(t-1) + c_\tau r_1(p_{best,i} - x_i(t-1)) + c_2 r_2(g_{best} - x_i(t-1))$
- ✓ Update position $x_i\,(t) = x_i(t-1) + v_i(t)$
- ✓ Check $x_i(t)$ within boundary. If $x_i(t) > x_{ub} \rightarrow x_i(t) = x_{ub}$ &if $x_i(t) < x_{lb} \rightarrow x_i(t) = x_{lb}$
- ✓ Evaluate the objective function f_{xi} .
- ✓ Update p_{best} and g_{best} ::$p_{best\ i} = x_i$ if f_{xi} better $f_{p\ best}$ & $g_{best} = p_{best\ i}$ if $f_{p_{best\ i}}$ better $f_{g\ best}$
- ✓ If there is no convergence of the current solution &if $t > T$ go to Loo

2.4.4. Print g_{best} and $f_{g_{best}}$

3. Ant colony optimization algorithm (ACO)

The ACO is proposed by Marco Dorigo in 1992[67-69]. The basic principle of the ACO is each ant will release a secretion on the path it walked as a reference and will also perceive the secretions released by other ants while it is searching for food. This secretion is usually called pheromone. Under the action of pheromones, the ant colony can communicate with each other and choose paths. When the pheromone on a path is more than other paths, the ant colony will spontaneously move to this path, and release more secretions during the movement, so that the concentration of the pheromone becomes higher to attract the latter ants which forms a mechanism of positive feedback. After a period of time, the concentration of pheromone on the shorter path is getting higher and higher, then the ants that choose it are gradually increasing, while the pheromones on other paths are gradually reduced until there is no. Finally the whole ant colony is concentrated in the optimal path. The process of ant foraging is similar to the path planning of robots. As long as there are enough ants in the nest, these ants will find the shortest path from the nest to the food to avoid obstacles. The principle of ant colony searching for food is shown in Figure1. The ACO has not only the global search ability of the population, but also has synergy between individuals. It can find a better path, even if the complete information of the environment is not known. However, in the early stage of the algorithm, the convergence speed is slow and it takes a lot of computation time. It is prone to prematurity. When an ant finds the obstacles in the middle it neglects and find any alternate path for reaching a goal, likewise an autonomous mobile robot involves in searching a path when any interruption occurs in the middle. The artificial ant mobile robot involves in finding a path faster by following a previous ant which produces more pheromone in order to find a shortest path. Likewise, in autonomous navigational mobile robots, each robot communicates with each other using signal strength instead of a pheromone. The mobile robot senses the signal from the previously travelled robot and follows that path for reaching a goal by using a sensor. So, the time consuming is reduced and also obstacles are detected as soon as possible.



The principle of ant colony searching for food)(Fig. 1

The path planning algorithm starts from a Source point X_s (starting point) and ends at Xg (Goal point). If a robot moves from a source to an adjacent or new point is denoted as Xn (new point) and it is calculated by summing up the current position and step size with dimension angle (Θ) as follows:

$$Xn = Xp + step * cos(\theta) \text{ and } Xn = Xp + step * sin(\theta) \qquad (3)$$

The flag value is set for encountering obstacles while travelling and if it so moves three steps back. The process of bypassing the obstacles in the navigational robot is achieved, which is far better using an ACO algorithm by overcoming the failures like allocation of tasks over time.

3.1. The principal method of ACO is as per the following:

- Produce ant(s) and looping for each ant until and unless the whole task is completed.
- Store pheromone on visited states/sites covered by ant(s).
- Daemon exercises and dissipation of pheromone.

For taking care of multi-objective arranging issues within the sight of obstruction, the ant colony is joined with a sample-based location-to-location path planning method. Execution is assessed by quantitative examination with two existing sample-based methods. The ACO method, offering a trade-off between arrangement quality value and speediness. It is the prevalent decision given the physical parameters of robotics path planning.

3.1.1. Key Terms

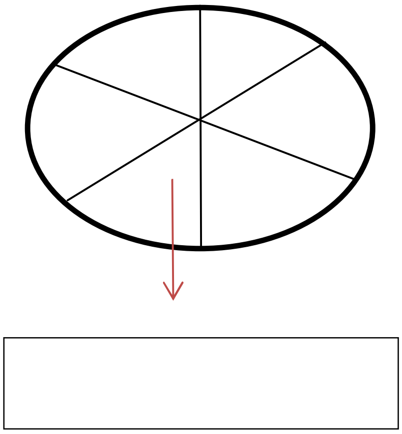
- Ants k: Any possible solution.
- Population N-Group of all ants.
- Search Space $[lb, ub]$- All possible solutions to the problem.
- Search Space is divided by step size h
- Pheromone trail τ
- Scaling parameter ζ
- Evaporate rate ρ

3.1.2. Four phases are considered in ACO

- Build tours

Nexus : Journal of Innovative Studies of Engineering Science (JISES)
Volume: 02 Issue: 04 | 2023 ISSN: 2751-7578
http://innosci.org/

- From the home node, ants start travelling through the various paths and end at the destination node in each iteration (discrete values of design variables)
- Find the cumulative probability ranges associated with different discrete values based on its probabilities.
- The specific discrete values chosen by ant k will be determined using the roulette-wheel selection.
- Generate N random numbers r in the range (0,1), one for



Determine the discrete value by ant k for variable as the one for which the cumulative probability range includes the random numbers r.

3.1.3. Four phases are considered in ACO

- ✓ Deposit and update trail - Once the path is complete, the ant deposits some pheromone on the path.
- Evaluate the objective function values of each ant
- Determine the best $fbest$ and worst $fworst$ objective function of the discrete value among ants
- Update the pheromone
- ✓ Best ants: reinforcement the pheromone of the best path by:

$$\tau_j^{new} \leftarrow \tau_j^{old} + \textstyle\sum_k \Delta\tau_j^{(k)}$$ f_{best} →best objective function

$$\Delta\tau_j^{(k)} = \frac{\zeta f_{best}}{f_{worst}}$$ f_{worst} →worst best objective function

- ✓ Other ants: evaporates the pheromone of other paths by:

$$\tau_j^{new} \leftarrow (1-\rho)\tau_j^{old}$$ ζ → scaling parameter

ρ⟶ evaporate rate

3.1.4. Four phases are considered in ACO

- ✓ Termination
- The steps of ACO algorithm are iteratively repeated until the maximum number of iteration is reached or a termination criterion is met.
- Convergence: is the case where the positions of all particles converge to the same set of values, the method is assumed to have converged.

3.2. Pseudo code

3.2.1. Input

- ✓ Objective function (fitness function), upper bound (ub) and lower bound (lb), population size (N), number of iteration T, scaling parameter ζ, evaporate rate ρ, step size h (or number of discrete value m)

3.2.2. Initialization

- ✓ Initialize all discrete values m of design variables equal amounts of pheromone τ

3.2.3. loop:

Nexus : Journal of Innovative Studies of Engineering Science (JISES)
Volume: 02 Issue: 04 | 2023 ISSN: 2751-7578
http://innosci.org/

- ✓ For $t = 1{:}\,T \rightarrow$
- ✓ Find probability to select discrete values of design variables is:

$$p_j^k = \frac{\tau_j}{\sum_{j=1}^{m} \tau_j} \quad (4)$$

- ✓ Find the cumulative probability ranges associated with different discrete values based on its probabilities.(design roulette-wheel)
- ✓ For $i = 1{:}\,N$
- ➢ Generate a random numbers r
- ➢ Find corresponding discrete value
- ➢ Evaluate the objective function f_{x_j}
- ✓ Determine the best f_{best} and worst f_{worst} objective function of the discrete value among ants
- ✓ Update best path by. $\tau_j^{new} \leftarrow \tau_j^{old} + \sum_k \Delta\tau_j^{(k)}$ (5)

and other paths by.

$$\tau_j^{new} \leftarrow (1-\rho)\tau_j^{old} \quad (6)$$

- ✓ If there is no convergence of the current solution & if $t>T$ go to Loop

3.2.4. Print x_{fbest} and f_{best}

```
Pseudo code of ACO
Begin
  Initialize
  While stopping criterion not satisfied do
      Position each ant in a starting node
      Repeat
          For each ant do
              Choose next node by applying the state transition rule
              Apply step by step pheromone update
          End for
      Until every ant has built a solution
      Update best solution
      Apply offline pheromone update
  End While
```

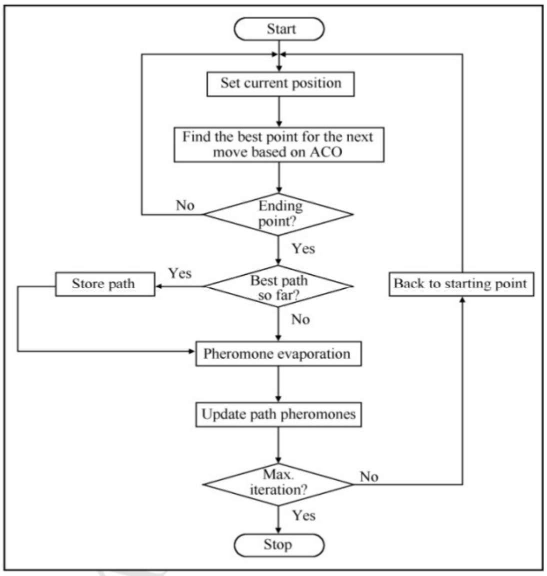


Figure (3)ACO pseudo code and flowchart for path planning of robot\ by Brand et al. [70]

4. Bee Colony

4.1. Bees in the Nature

The ABC algorithm is a swarm-based intelligent approach inspired by the activities of honey bees (Figure 20) in search of food and is proposed by Kharaboga [71] .Self-organization of bees is based on a few relatively simple rules of individual insect's behavior. In spite of the existence of a large number of different social insect species, and variation in their behavioral patterns, it is possible to describe individual insects' as capable of performing a variety of complex tasks [72]. The best example is the collection and processing of nectar, the practice of which is highly organized. Each bee decides to reach the nectar source by following a nestmates who has already discovered a patch of flowers. Each hive has a so called dance floor area in which the bees that have discovered nectar sources dance, in that way trying to convince their nestmates to follow them. If a bee decides to leave the hive to get nectar, she follows one of the bee dancers to one of the nectar areas. Upon arrival, the foraging bee takes a load of nectar and returns to the hive relinquishing the nectar to a food storer bee. After she relinquishes the food, the bee can (a) abandon the food source and become again uncommitted follower, (b) continue to forage at the food source without recruiting the nestmates, or (c) dance and thus recruit the nestmates before the return to the food source. The bee opts for one of the above alternatives with a certain probability. Within the dance area, the bee dancers "advertise" different food areas. The mechanisms by which the bee decides to follow a specific dancer are not well understood, but it is considered that "the recruitment among bees is always a function of the quality of the food source" [72]. It is also noted that not all bees start foraging simultaneously. The experiments confirmed, "new bees begin foraging at a rate proportional to the difference between the eventual total and the number presently foraging".

4.2. Bee Colony Optimization (BCO)

The basic idea of designing BCO is to compose the multi-agent system (colony of artificial bees) that will search for good solutions of a variety of combinatorial optimization problems. The artificial bees explore the principles used by honey bees for the period of nectar collection process. In other words, BCO principles are gathered from natural systems. Artificial bees explore through the search space, looking for the feasible solutions. In order to discover better and better solutions, artificial bees collaborate and exchange information. via collective knowledge and sharing information among themselves, artificial bees focus on more promising areas, and gradually discard solutions from the less promising ones. Little by little, artificial bees jointly generate and/or improve their solutions. The BCO search is running in iterations until some predefined stopping criteria is satisfied. Population of agents (artificialbees) consisting of B bees collaboratively searches for the optimal solution. Every artificial bee generates one solution to the problem. There are constructive [73-76] and improvement version [77](submitted for publication) of the BCO algorithm. In constructive BCO each bee adds a (different) new component to the previously generated partial solution,

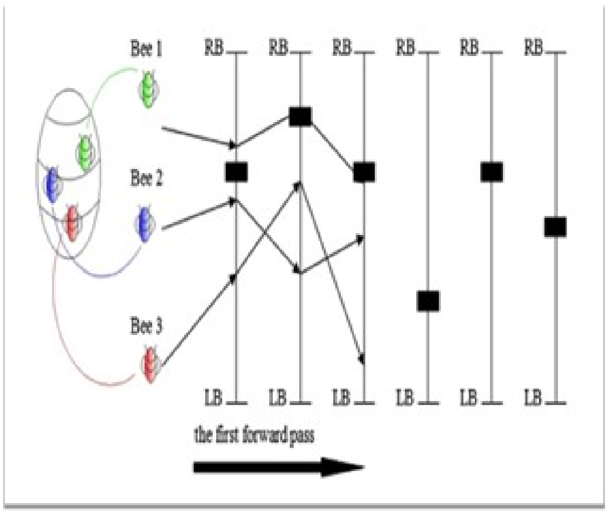


Fig.(4)An example of partial solutions after the first forward pass, NC = 3, B = 3

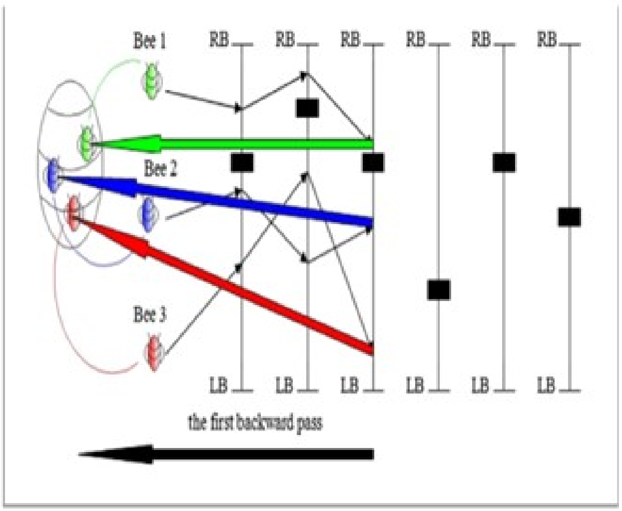


Fig.(5)The first backward pass, NC = 3, B = 3.

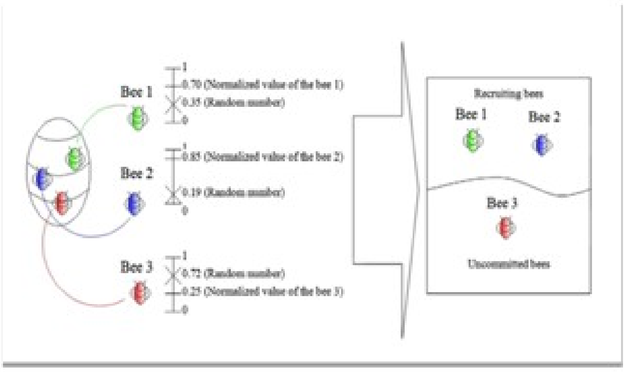


Fig. (6) Dividing bees into two groups (B = 3)

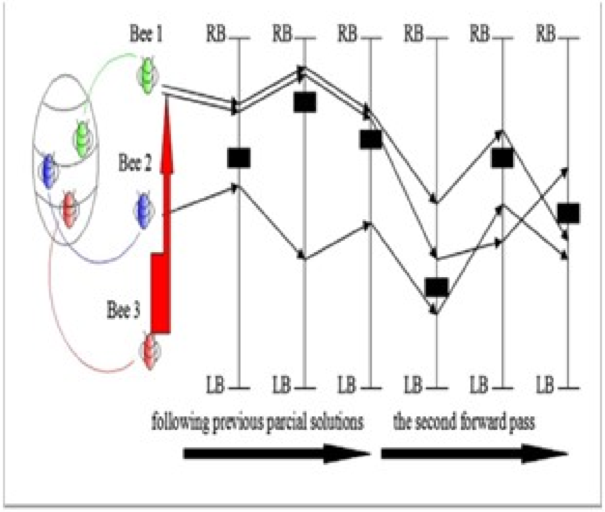


Fig. (7) Recruiting of uncommitted followers and the second forward pass NC = 3, B = 3.

while in the improvement version of the BCO bees modify some components of the complete solution in orderto enhance them. The algorithm consists of two alternating phases: forward pass and backward pass. During each forward pass, every artificial bee explores the search space. It applies a predefined number of moves (NC), which construct and/or improve the solution, yielding a new solution. NC is a parameter used to define the frequency of information exchange between bees. Its value depends on each particular problem characteristics. If NC takes small values, then the search process is intensified, since each newly generated part consists of only a few components. The difference between solutions, generated by different bees, is minor. On the other hand, if NC is large, each bee adds more components to its partial solution, thus introducing variety among different solutions. Suppose we have B bees, namely Bee 1,Bee 2,. . .,Bee B which participate in the decision-making process on n entities. One of the possible situations which may arise after the first forward pass in the case NC = 3 and B = 3 is illustrated in (Fig. 7). Upon obtaining new partial solutions for each bee, the second phase, the so-called backward pass, starts (Fig. 5). During the backward pass, all bees share information about their solutions. In nature, bees would perform a dancing ritual, which would inform other bees about the amount of food they have found, and the proximity of the patch to the hive. In the search algorithm, the quality of each generated solution is determined, i.e. the current value of the objective function is calculated. During the backward pass, every bee decides, with a certain probability, whether it will stay loyal to its solution or not. Contrary to bees in nature, artificial bees that are loyal to their generated solutions are at the same time recruiters, i.e. their solutions are considered by other bees. Once the solution is abandoned the bee becomes uncommitted and has to select one of the advertised solutions. This decision is taken with a probability, such that better advertised solutions have greater opportunities to be chosen for further exploration. In such a way, within each backward pass all bees are divided into two groups (R recruiters, and the remaining B R uncommitted bees) as shown in (Fig. 6). Values for R and B R change from one backward pass to another. Let us assume that after comparing all generated partial solutions Bee 3 from the previous example decided to abandon its solution, and join Bee 1. The resulting situation is presented in (Fig. 4). Bee 1 and Bee 3 ‘‘fly together’’ along the path already generated by Bee 1. In practice, this means that the partial solution generated by Bee 1 is associated (copied) to Bee 3. When they ‘‘reach the end of the path’’, they are free to make an individual decision about the next constructive step. This actually means that each of them will add different components to the same partial solution. Bee 2 will keep its partial solution without being chosen by any hive-mates and will perform a new constructive step independently. The two phases of the search algorithm, namely the forward and backward passes, alternate in order to generate all required complete solutions (one for each bee). At that stage the best solution is determined and an iteration of BCO is completed. The BCO algorithm runs iteration by iteration until a stopping condition is met. A possible stopping condition could be, for example, the maximum number of iterations, the maximum number of iterations without the improvement of the objective function, the maximum allowed CPU time, etc. In the end, the best solution found is reported as the final one. In this paper we apply the improvement version of the BCO algorithm. The BCO algorithm parameters whose values need to be set prior the algorithm execution are as follows: B – the number of bees involved in the search, IT – the number of iteration, NP – the number of forward and backward passes in a single iteration, NC – the number of changes in one forward pass, S – the best known solution[78] As shown in the fig.(8)

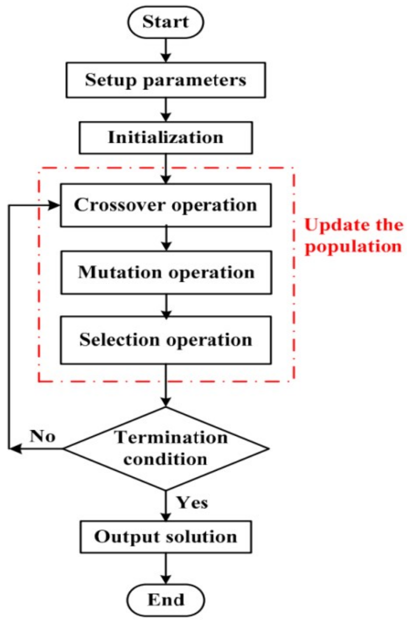


Figure (8) proses flowchart of basic GA

4.3. pseudo code of the BCO algorithm:

procedure BCOi (in B, IT, NP, NC, out S)

for i = 1 to B do

Determine an initial solution for the i-th bee.

Evaluate the solution of the i-th bee.

S the best solution of the bees.

for j = 1 to IT do

for i = 1 to B do

the bee i Set an initial solution.

for k = 1 to NP do

for i = 1 to B do

for r = 1 to NC do

Evaluate modified solutions generated by possible changes of the i-th bee solution.

By roulette wheel selection choose one of the modified solutions.

for i = 1 to B do

Evaluate solution of the i-th bee.

for i = 1 to B do

Make a decision whether the i-th bee is loyal.

for i = 1 to B do

if the bee i doesn’t loyal then

Choice one of the loyal bees to be followed by the i-th bee.

if the best solution of the bees better then solution S

S the best bee’s solution.

5. Genetic Algorithms (GA)

GA is an optimization tool that is most commonly used to generate high quality solutions for combination optimization problems and search problems. The GA is inspired from the process of natural selection and relies on evolutionary operators like mutation, crossover and selection. The GA starts with no knowledge of correct solution and entirely depends on the responses of the environment and the above mention evolutionary operators and arrive at the best solution[79]. are parallel and global search technique that emulates natural genetic operators. As it simultaneously evaluates many points in the parameter space, it is more likely to converge to the global optimal[80] . Many path planning methods use a grid-based model to represent the environment space, leading to two possible representations: (*i*) through an orderly numbered grid, as shown in Fig.(9), or (*ii*) through the (*x*,*y*) coordinates plane[80]. A chromosome represents a candidate solution for the path planning problem. A chromosome representing a path encodes a starting node, a target node and the nodes through which the mobile robot travels. These nodes, or steps, in the path are called genes of the chromosome. A valid path consists of a sequence of grid labels which begins at the starting node and ends at the target node, as shown in Fig. (10) .The initial population is generally generated randomly. Thus, some of the generated chromosomes may include infeasible paths intersecting an obstacle. An optimal, or near optimal, solution can be found by genetic operators, even though the initial population includes infeasible paths. However, this reduces the search capability of the algorithm and increases the time to find the solution. Additionally, crossover of two infeasible chromosomes may generate new infeasible paths. To solve this problem, each chromosome must be checked whether it intersects an obstacle, when generating the initial population. If it does, the intersected gene on the chromosome is changed randomly, until a feasible one is found . The optimal path may be the shortest one, or the one requiring the least time or less energy to be traversed. Generally, in path planning problems, the objective function is considered as the shortest path. In ,the objective function value of a chromosome used in the GA is given by Equations 7 and 8:

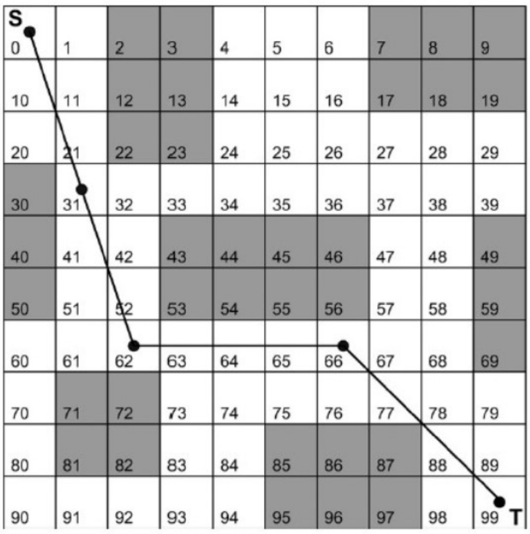


Fig. 9. Example of the orderly numbered grid environment representation [80].

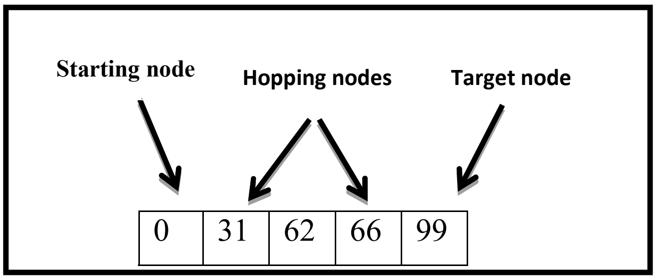


Fig. 10. Decimal coded genes of a chromosome [80]

Nexus : Journal of Innovative Studies of Engineering Science (JISES)
Volume: 02 Issue: 04 | 2023 ISSN: 2751-7578
http://innosci.org/

$$f = \begin{cases} \sum_{i=1}^{n-1} d(p_i, p_i + 1) & \text{for feasible paths} \\ \sum_{i=1}^{n-1} d(p_i, p_i + 1) + penalty; & \text{for infeasible paths} \end{cases} \tag{7}$$

$$d(p_i ; p_i + 1) = \sqrt{\left(x_{(i+1)} - x_i\right)^2 + \left(y_{(i+1)} - y_i\right)^2} \tag{8}$$

being, f the fitness function, *pi* the *i*th gene of the chromosome, n the length of the chromosome, *d* the distance between two nodes, *xi* and *yi* the robot current position, and $x_{(i+1)}$ and $y_{(i+1)}$ the robot next position. The direction of the robot path is given by equation 9 [80]:

$$\alpha = tan^{-1} \frac{(y_{(i+1)} - y_i)}{(x_{(i+1)} - x_i)} \tag{9}$$

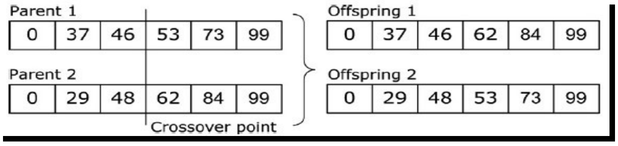


Fig.11. Single-point

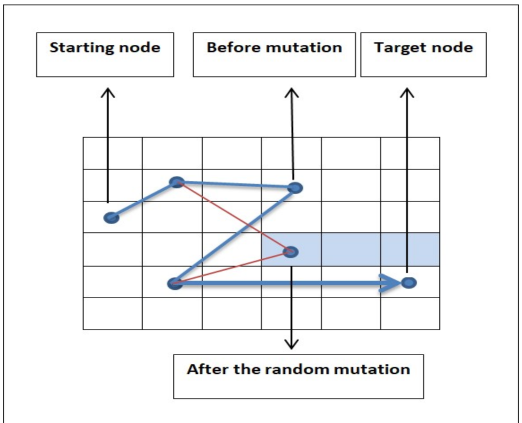


Fig.12 Infeasible path [80]

The objective function value is defined as the sum of distances between each node in a path. If there is an obstacle in the robot path, a penalty is added to the objective function value. The penalty value should be greater than the maximum path length on the environment. In order to find an optimal path, the algorithm searches for the chromosome with the least value for the objective function [80]. The main principle of the GA is that the best genes on the chromosomes should survive and be transferred to new generations. A selection procedure needs to be done to determine the best chromosomes. This process consists in the following three steps [80]:

- Objective function values of all chromosomes are computed
- Fitness values are assigned to chromosomes according to their objective function values. In[80], the rank based fitness assignment is used instead of the proportional assignment method. This prevents a few better chromosomes to be dominant in the population.
- Chromosomes are selected according to their fitness values and then

placed into a mating pool to produce new chromosomes. Normally, crossover combines the features of two parent chromosomes to form two offsprings. In Fig.11, single-point crossover operator is illustrated, and the genes of the two "parent" chromosomes after the crossover point are changed [80]. All candidate chromosomes in the population are subjected to the random mutation after the crossover operation. This is a random bit-wise binary complement operation or a random small change in a gene, depending on the coding of chromosomes, applied uniformly to all genes of all individuals in the population, with a probability equal to the mutation rate. The mutation operation increases the diversity of the population and avoids the premature convergence. It expands the search space to regions that may not be close to the current population, thus ensuring a global search [80]. In conventional GA, random mutation is the most commonly used operator. However, random mutation can cause generation of infeasible paths. Even though a chromosome is feasible before the mutation operation, the new node changed by the mutation may have an obstacle and therefore it constitutes an infeasible path (see Fig. 12). This makes the optimization slower and increases the number of generations . To overcome this problem, several studies concerned with the improvement of mutation operation have been done in the literature. The authors of those studies, as well as the method proposed by each author, are described in[80].

Table (2) The advantages and disadvantages for ACO, PSO,ABC and GA algorithms

Algorithm	Advantages	Disadvantages
ACO[81, 82]	➢ Able to cluster and build routes ➢ Simple implementation ➢ Easily parallelized for concurrent processing ➢ Derivative free ➢ Efficient for TSP and similar problems.	➢ It is time-consuming to lay pheromone on trails used by ants as a communication medium ➢ Able to fall easily into the trap of local optimum Probability distribution changes by iteration. ➢ Time to convergence uncertain (but convergence is Guaranteed)
PSO[83-87]	➢ Very easy to implement ➢ The training speed is fast ➢ the efficiency is high, ➢ the algorithm is simple	➢ Has problems in parameter selection due to its poor exploration ➢ It is easy to fall into local optimal solution ➢ poor handling of discrete optimization problems
GA [88 ,87]	➢ It has an ability to prevent from fall into a local optimum with the help of mutation ➢ The convergence speed is fast and the versatility is strong ➢ parallelism ➢ travelling in a search space with more individuals so they are less likely to get stuck in a local extreme like some other methods ➢ easy to implement ➢ have some GA, just write new chromosome to solve another problem ➢ same encoding - change the fitness function	➢ The best solution very hard to obtain because GA easily falls into premature convergence. ➢ It is more complex, and depends on the initial population ➢ Disadvantage ➢ computational time ➢ slower than some other methods ➢ choosing encoding and fitness function can be difficult ➢ But with today's computers it is not so big problem
ABC[87]	➢ The global search ability is strong and the convergence speed is fast ➢ Few control parameters ➢ Fast convergence ➢ Both exploration & exploitation	➢ it is easy to fall into the local optimum, and the search speed slows down later ➢ Search space limited by initial solution (normal distribution sample should use in initialize step

Table (3) application areas for GA,ABC,ACO and PSO algorithms

algorithms	application areas
ACO [69, 89-97]	➢ Routing/ Traveling salesman ➢ Scheduling/ Single machine ➢ Subset / Maximum independent set ➢ Assignment and layout / Quadratic assignment] ➢ Machine learning / Bayesian networks ➢ Bioinformatics / Shortest common super sequence ➢ Multiobjective/ Portfolio selection ➢ Continuous / Test problems ➢ Dynamic/ Dynamic TSP
GA[98]	➢ Optimization ➢ Economics ➢ Neural Networks ➢ Image Processing ➢ Vehicle routing problems ➢ Scheduling applications ➢ Machine Learning ➢ Robot Trajectory Generation ➢ Parametric Design of Aircraft ➢ DNA Analysis ➢ Multimodal ➢ Traveling salesman problem and its applications
PSO[98]	➢ Scheduling ➢ Shortest Common Sequence ➢ Constraint Satisfaction ➢ 2D-HP protein folding Bin Packing Machine Learning ➢ Classification Rules ➢ Bayesian networks ➢ Fuzzy systems ➢ Network Routing ➢ Connection oriented network routing Connection network routing ➢ Optical network routing
ABC[94]	➢ comparative analysis ➢ modified versions ➢ electric power systems ➢ parallel and grid computing ➢ data clustering and image analysis ➢ computer science applications ➢ signal processing and communication system

Table (4) outlining the common oprators and control parameters for genetic algorithms (GA),artificial bee colony (ABC), ant colony optimization (ACO), and particle swarm optimization (PSO) algorithms [99]:

Operator/Control Parameter	GA	ABC	ACO	PSO
Objective Function	✓	✓	✓	✓
Population Size	✓	✓	✓	✓
Fitness Function	✓			
Parent Selection	✓			
Crossover Operator	✓			
Mutation Operator	✓			
Elitism	✓			
Employed Bees		✓		
Onlooker Bees		✓		
Scout Bees		✓		
Food Source Selection		✓		
Waggle Dance Communication		✓	✓	
Pheromone Update			✓	
Swarm Size				✓
Velocity Update				✓
Personal Best Update				✓
Global Best Update				✓

6. Hybrid and improved algorithms

There are improved algorithms and others called hybrids that have high effectiveness as well and have proven their worth in solving difficult problems, as more algorithms can be integrated to improve the quality and efficiency of the solution in[100] , And the optimization strategy can be applied to optimize the trajectory of the welding robot. The shortest path length, collision-free path, and welding deformation control were considered to improve the welding path. In [101](PSO_ALS algorithm) feature selection problem, a global best solution within the search space and an adaptive local search method was presented in exploring the local search space. Using the adaptive local search technique improves the results of the proposed algorithm. PSO_ALS shows superior performance over other similar methods. And in [102]it was found that constrained particle swarming optimization (CPSO) works better than other meta-methods in unknown environments. As well as in [103] the experimental results show the promising behavior of the proposed method in increasing classification accuracy and optimal selection of traits using the hybrid algorithm (AC-ABC). And in [104] the BCO algorithms are described which we call Bee System (BS) and Fuzzy Bee System (FBS). In the case of FBS, agents (artificial bees) use approximate thinking and fuzzy logic rules in their communication and behavior. In this way, FBS is able to solve Deterministic combinatorial problems, as well as combinatorial problems that are characterized by uncertainty.in[105] the shortest collision-free paths are considered as the criteria, genetic algorithm and particle swarm optimization algorithm are combined to realize welding robot path optimization. In [106]for navigation of multiple mobile robots in the real world. They modified the form of PSO and Darwinian PSO (DPSO) system for obstacle avoidance and mutual communication issues. They found that in a system of 12 physical robots the efficiency achieved was up to 90% in a sense of maximum communication distance and global optimum.In literature, ACO and ABC have been widely used for optimizing the selection of features in problems like face recognition, high dimensional gene expression, speech segments classification, texture classification, medical diagnosis, text mining and other data mining applications [21-24, 107-113]. In applications that involve ACO, either the actual form or variant forms of ACO like ACO with Support Vector Machine, ACO with Neural Networks ACO with Fuzzy Logic have been attempted[22-24]. Also, ACO hybridized with other SI algorithms such as ACO-PSO Hybrid, ACO-GA Hybrid and ACO-Cuckoo Hybrid have been experimented for FS optimization [108-111]. Same is the case for ABC, it has also been applied in its actual and modified forms [114-116]. So far, only one hybrid form of ABC with other SI algorithm, ABC-DE Hybrid has been tried out for FS [113] , Also, both the algorithms show boosted performance, when hybridized with other SI algorithms than being standalone [108-111, 113].In [117]The double global optimum GA-PSO algorithm it can obtain the shortest collision-free welding path effectively and improve the welding efficiency as a result. The diversity of particles and global search ability were enhanced after another global solution from the GA was considered. Simulation results confirmed that the algorithm is better than the basic GA and PSO algorithm, and it can be applied to welding robot path planning.

7. Discussion

It is clear from the research that there is a high efficiency of these algorithms in many areas that were mentioned according to their applications, and the most important thing is to use the algorithm according to its characteristics and in the applications that were mentioned to get high efficiency. This paper presented a review of the four algorithms (ABC, ANO, PSO, GA). The article discussed these algorithms and explained the advantages and disadvantages of each of them, as well as reviewed the applications in which they were highly successful as basic, improved, or hybrid algorithms. From this review paper, we can learn about the most appropriate and best algorithms for future work, and all mentioned algorithms can improve performance by hybridizing algorithms by combining two or more of them. These algorithms are able to solve the most difficult problems facing robots path planning in particular and other problems in general.

8. Reference

1. Amoshahy, M.J., M. Shamsi, and M.H. Sedaaghi, *A novel flexible inertia weight particle swarm optimization algorithm.* PloS one, 2016. **11**(8): p. e0161558.
2. Zhang, Y., D.-w. Gong, and J.-h. Zhang, *Robot path planning in uncertain environment using multi-objective particle swarm optimization.* Neurocomputing, 2013. **103**: p. 172-185.
3. Cheng, Z., Y.X. Tang, and Y.L. Liu. *3-D path planning for UAV based on chaos particle swarm optimization*. in *Applied Mechanics and Materials*. 2012. Trans Tech Publ.
4. Zhang ,Y., L. Wu, and S. Wang, *UCAV path planning by fitness-scaling adaptive chaotic particle swarm optimization.* Mathematical Problems in Engineering, 2013. **2013**.
5. Liu, Y.-k., et al., *Path-planning research in radioactive environment based on particle swarm algorithm.* Progress in Nuclear Energy, 2014. **74**: p. 184-192.
6. Yusof, T., S.F. Toha, and H.M. Yusof, *Path planning for visually impaired people in an unfamiliar environment using particle swarm optimization.* Procedia Computer Science, 2015. **76**: p. 80-8.6
7. Tang, X.-l., L.-m. Li, and B.-j. Jiang, *Mobile robot SLAM method based on multi-agent particle swarm optimized particle filter.* The Journal of China Universities of Posts and Telecommunications, 2014. **21**(6): p. 78-86.
8. Atyabi, A., S. Phon-Amnuaisuk, and C.K. Ho, *Applying area extension PSO in robotic swarm.* Journal of Intelligent and Robotic Systems, 2010. **58**(3): p. 253-285.

9. Tang, Q. and P. Eberhard, *Cooperative motion of swarm mobile robots based on particle swarm optimization and multibody system dynamics.* Mechanics based design of structures and machines, 2011. **39**(2): p. 179-193.
10. Chen, Y.-L., et al., *Classification-based learning by particle swarm optimization for wall-following robot navigation.* Neurocomputing, 2013. **113**: p. 27-35.
11. Guedria, N.B., *Improved accelerated PSO algorithm for mechanical engineering optimization problems.* Applied Soft Computing, 2016. **40**: p. 455-467.
12. Ye, Y., et al., *Position control of nonlinear hydraulic system using an improved PSO based PID controller* .Mechanical Systems and Signal Processing, 2017. **83**: p. 241-259.
13. Benvidi, A., et al., *Spectrophotometric determination of synthetic colorants using PSO–GA-ANN.* Food chemistry, 2017. **220**: p. 377-384.
14. Wang, X. and R. Li, *Intelligent modelling of back-side weld bead geometry using weld pool surface characteristic parameters.* Journal of Intelligent Manufacturing, 2014. **25**(6): p. 1301-1313.
15. Wang, X., et al., *Intelligent welding robot path optimization based on discrete elite PSO.* Soft Computing, 2 :(20)21 .017p. 5869-5881.
16. Wen, Z.-q. and Z.-x. Cai, *Global path planning approach based on ant colony optimization algorithm.* Journal of Central South University of Technology, 2006. **13**(6): p. 707-712.
17. Wang, H.-j. and W. Xiong, *Research on global path planning based on ant colony optimization for AUV.* Journal of Marine Science and Application, 2009. **8**(1): p. 58-64.
18. Zhu, X.G., Q.Y. Han, and Z.Q. Wang. *The application of an improved ant colony algorithm in mobile robot path planning*. in *Key Engineering Materials*. 2011. Trans Tech Publ.
19. Gao, M.-k., et al., *Three-dimensional path planning and guidance of leg vascular based on improved ant colony algorithm in augmented reality.* Journal of medical systems, 2015. **39**(11): p. 1-11.
20. Abd-Alsabour, N. and M. Randall. *Feature selection for classification using an ant colony system*. in *2010 Sixth IEEE International Conference on e-Science Workshops*. 2010. IEEE.
21. Robbins, K., et al., *The ant colony algorithm for feature selection in high-dimension gene expression data for disease classification.* 2007. **24**(4): p. 413-426.
22. Aghdam, M.H., N. Ghasem-Aghaee, and M.E.J.E.s.w.a. Basiri, *Text feature selection using ant colony optimization.* 2009. **36**(3): p. 6843-6853.
23. Sivagaminathan, R.K. and S.J.E.s.w.a. Ramakrishnan, *A hybrid approach for feature subset selection using neural networks and ant colony optimization.* 2007. **33**(1): p. 49-60.
24. Kabir, M.M., M. Shahjahan, and K.J.E.S.w.A. Murase, *A new hybrid ant colony optimization algorithm for feature selection.* 2012. **39**(3): p. 3747-3763.
25. Ali, M., et al., *An image watermarking scheme in wavelet domain with optimized compensation of singular value decomposition via artificial bee colony.* 2015. **301**: p. 44-60.
26. Gao, H., C.-M. Pun, and S.J.I.S. Kwong, *An efficient image segmentation method based on a hybrid particle swarm algorithm with learning strategy.* 2016. **369**: p. 500-521.
27. Zhang, X., et al., *A modification of artificial bee colony algorithm applied to loudspeaker design problem.* 2014. **50** :(2)p. 737-740.

28. Zhang, X., et al., *An improved artificial bee colony algorithm for optimal design of electromagnetic devices.* 2013. **49**(8): p. 4811-4816.

29. Xia, Z., et al., *A privacy-preserving and copy-deterrence content-based image retrieval scheme in cloud computing.* 2016. **11**(11): p. 2594-2608.

30. Hsieh, T.-J., H.-F. Hsiao, and W.-C.J.A.s.c. Yeh, *Forecasting stock markets using wavelet transforms and recurrent neural networks: An integrated system based on artificial bee colony algorithm.* 2011. **1** :(2)1p. 2510-2525.

31. Bose, D., et al., *Optimal filter design using an improved artificial bee colony algorithm.* 2014. **281**: p. 443-461.

32. Gu, B., et al., *Incremental support vector learning for ordinal regression.* 2014. **26**(7): p. 1403-1416.

33. Habbi ,H., et al., *Self-generated fuzzy systems design using artificial bee colony optimization.* 2015. **295**: p. 145-159.

34. Horng, S.-C.J.I.T.o.S., Man, and C. Systems, *Combining artificial bee colony with ordinal optimization for stochastic economic lot scheduling problem.* 2014. **45**(3): p. 373-384.

35. Li, J.-q. and Q.-k.J.I.S. Pan, *Solving the large-scale hybrid flow shop scheduling problem with limited buffers by a hybrid artificial bee colony algorithm.* 2015. **316**: p. 487-502.

36. Pan, Q.-K., et al., *An effective artificial bee colony algorithm for a real-world hybrid flowshop problem in steelmaking process.* 2012. **10**(2): p. 307-322.

37. Liao, X., et al., *An adaptive artificial bee colony algorithm for long-term economic dispatch in cascaded hydropower systems* :(1)43 .2012 .p. 1340-1345.

38. Babaoglu, I.J.A.S.C., *Artificial bee colony algorithm with distribution-based update rule.* 2015. **34**: p. 851-861.

39. Zhao, M., N. Ansari, and E.S.J.J.o.R.S. Hou, *Mobile manipulator path planning by a genetic algorithm.* 199 :(3)11 .4p. 143-153.

40. Liu, Y., *Robot path planning based on genetic algorithms with two-layer encoding.* 2000.

41. Pehlivanoglu, Y.V., et al., *Path planning for autonomous UAV via vibrational genetic algorithm.* 2007.

42. Yao, P. and H.J.S.C. Wang, *Dynamic Adaptive Ant Lion Optimizer applied to route planning for unmanned aerial vehicle.* 2017. **21**(18): p. 5475-5488.

43. Tsai, C.-C., H.-C. Huang, and C.-K.J.I.T.o.I.E. Chan, *Parallel elite genetic algorithm and its application to global path planning for autonomous robot navigation.* 2011. **58**(10): p. 4813-4821.

44. Tuncer, A., M.J.C. Yildirim, and E. Engineering, *Dynamic path planning of mobile robots with improved genetic algorithm.* 2012. **38**(6): p. 1564-1572.

45. Qu, H., K. Xing, and T.J.N. Alexander, *An improved genetic algorithm with co-evolutionary strategy for global path planning of multiple mobile robots.* 2013. **120**: p. 509-517.

46. Liu, F., S. Liang, and D.X.J.T.I.J.o.E.E. Xian, *Optimal path planning for mobile robot using tailored genetic algorithm* :(1)12 .2014 .p. 1-9.

47. Shorakaei, H., et al., *Optimal cooperative path planning of unmanned aerial vehicles by a parallel genetic algorithm.* 2016. **34**(4): p. 823-836.

48. Wu, M., et al. *Path planning of mobile robot based on improved genetic algorithm* .in *2017 Chinese Automation Congress (CAC)*. 2017. IEEE.
49. Mohanta, J.C., et al., *Path planning strategy for autonomous mobile robot navigation using Petri-GA optimisation.* 2011. **37**(6): p. 1058-1070.
50. Naderan-Tahan, M. and M.T. Manzuri-Shalmani. *Efficient and safe path planning for a mobile robot using genetic algorithm*. in *2009 IEEE Congress on Evolutionary Computation*. 2009. IEEE.
51. Yao, Z. and L. Ma. *A static environment-based path planning method by using genetic algorithm*. in *2010 International Conference on Computing, Control and Industrial Engineering*. 2010. IEEE.
52. Fu, S.-Y., et al. *Path planning for unmanned aerial vehicle based on genetic algorithm*. in *2012 IEEE 11th International Conference on Cognitive Informatics and Cognitive Computing*. 2012. IEEE.
53. Liang, Y. and L. Xu, *Global path planning for mobile robot based genetic algorithm and modified simulated annealing algorithm*, in *Proceedings of the first ACM/SIGEVO Summit on Genetic and Evolutionary Computation*. 2009. p. 303-308.
54. Pillay, N. and W.J.A.S.C. Banzhaf, *An informed genetic algorithm for the examination timetabling problem.* 2010. **10**(2): p. 457-467.
55. Raghavjee, R. and N. Pillay. *Evolving solutions to the school timetabling problem*. in *2009 World Congress on Nature & Biologically Inspired Computing (NaBIC)*. 2009. IEEE.
56. Raghavjee, R. and N. Pillay. *An informed genetic algorithm for the high school timetabling problem*. in *Proceedings of the 2010 Annual Research Conference of the South African Institute of Computer Scientists and Information Technologists*. 2010.
57. Bari, A., et al., *A genetic algorithm based approach for energy efficient routing in two-tiered sensor networks.* 2009. **7**(4): p. 665-676.
58. Jiang, H. and L. Kang. *Building trade system by genetic algorithm* .in *International Symposium on Intelligence Computation and Applications*. 2009. Springer.
59. Shi, L., et al. *Network model and optimization of medical waste reverse logistics by improved genetic algorithm*. in *International symposium on intelligence computation and applications*. 2009. Springer.
60. Thamilselvan, R. and P.J.I.J.o.R.T.i.E. Balasubramanie, *A genetic algorithm with a Tabu search (GTA) for traveling salesman problem.* 2009. **1**(1): p. 607.
61. Ju, B., L. Xi, and X. Zhou. *New product design based target cost control with BP neural network and genetic Algorithm-A Case Study in Chinese Automobile Industry*. in *International Symposium on Intelligence Computation and Applications*. 2009. Springer.
62. Jang, S.H., et al., *The study of genetic algorithm-based task scheduling for cloud computing.* 2012. **5**(4): p. 157-162.
63. Kessaci, Y., N. Melab, and E.-G. Talbi. *A pareto-based GA for scheduling HPC applications on distributed cloud infrastructures*. in *2011 International Conference on High Performance Computing & Simulation*. 2011. IEEE.
64. Zhu, K., et al. *Hybrid genetic algorithm for cloud computing applications*. in *2011 IEEE Asia-Pacific Services Computing Conference*. 2011. IEEE.

65. Mocanu, E.M., et al. *Cloud computing—task scheduling based on genetic algorithms*. in *2012 IEEE International Systems Conference SysCon 2012*. 2012. IEEE.

66. Kennedy, J. and R. Eberhart. *Particle swarm optimization*. in *Proceedings of ICNN'95-international conference on neural networks*. 1995. IEEE.

67. Dorigo, M. and G. Di Caro. *Ant colony optimization: a new meta-heuristic*. in *Proceedings of the 1999 congress on evolutionary computation-CEC99 (Cat. No. 99TH8406)*. 1999. IEEE.

68. Dorigo, M., G. Di Caro, and L.M.J.A.l. Gambardella, *Ant algorithms for discrete optimization.* 199 :(2)5 .9p. 137-172.

69. Dorigo, M., M. Birattari, and T.J.I.c.i.m. Stutzle, *Ant colony optimization.* 2006. **1**(4): p. 28-39.

70. Brand, M., et al. *Ant colony optimization algorithm for robot path planning*. in *2010 international conference on computer design and applications*. 2010. IEEE.

71. Karaboga, D., *An idea based on honey bee swarm for numerical optimization*. 2005, Technical report-tr06, Erciyes university, engineering faculty, computer….

72. Camazine, S. and J.J.J.o.t.B. Sneyd, *A model of collective nectar source selection by honey bees: self-organization through simple rules.* 1991. **149**(4): p. 547-571.

73. Lucic, P. and D. Teodorovic. *Bee system: modeling combinatorial optimization transportation engineering problems by swarm intelligence*. in *Preprints of the TRISTAN IV triennial symposium on transportation analysis*. 2001.

74. Lucic, P. and D. Teodorovic. *Transportation modeling: an artificial life approach*. in *14th IEEE International Conference on Tools with Artificial Intelligence, 2002.(ICTAI 200 .(2 Proceedings.* 2002. IEEE.

75. Lučić, P. and D.J.I.J.o.A.I.T. Teodorović, *Computing with bees: attacking complex transportation engineering problems.* 2003. **12**(03): p. 375-394.

76. Lučić, P. and D. Teodorović, *Vehicle routing problem with uncertain demand at nodes: the bee system and fuzzy logic approach*, in *Fuzzy sets based heuristics for optimization*. 2003, Springer. p. 67-82.

77. Davidović, T.J.Y.J.o.O.R., *Bee colony optimization Part I: The algorithm overview.* 2016. **25**(1.(

78. Nikolić, M. and D.J.E.S.w.A. Teodorović, *Empirical study of the Bee Colony Optimization (BCO) algorithm.* 2013. **40**(11): p. 4609-4620.

79. Leena, N., K.J.I.J.o.M. Saju, and C. Engineering, *A survey on path planning techniques for autonomous mobile robots.* 2014. **8**: p. 76-79.

80. Tamilselvi, D., et al. *Optimal path selection for mobile robot navigation using genetic algorithm in an indoor environment*. in *International Conference on Advanced Computing, Networking and Security*. 2011. Springer.

81. Kao, Y., M.-H. Chen, and Y.-T.J.M.P.i.E. Huang, *A hybrid algorithm based on ACO and PSO for capacitated vehicle routing problems.* 2012. **2012**.

82. Ge, B., Y. Han, and C.J.J.o.C. Bian, *Hybrid ant colony optimization algorithm for solving the open vehicle routing problem.* 2016. **27**(4): p. 41-54.

83. Azadeh, A. and H.J.T.L. Farrokhi-Asl, *The close–open mixed multi depot vehicle routing problem considering internal and external fleet of vehicles.* 2019. **11**(2): p. 78-92.

84. Stodola, P. and J.J.I.J.o.B.-I.C. Mazal, *Applying the ant colony optimisation algorithm to the capacitated multi-depot vehicle routing problem.* 2016. **8**(4): p. 228-233.
85. Xu, S.-H., et al., *A combination of genetic algorithm and particle swarm optimization for vehicle routing problem with time windows.* 2015. **15**(9): p. 21033-21.053
86. Zhou, B.J.A.m.-d.t.-e.v.r.p.w.d.o.a.i.t.l.m.d., Eur. J. Oper. Res, *Zhou L., Baldacci R., Vigo D., Wang X.* 2018. **265**(2): p. 765-778.
87. Yang, L., et al., *Applied research on distributed generation optimal allocation based on improved estimation of distribution algorithm.* 2018. **11**(9): p. 2363.
88. Korde, V., C.N.J.I.J.o.A.I. Mahender, and Applications, *Text classification and classifiers: A survey.* 2012. **3**(2): p. 85.
89. Dorigo, M., V. Maniezzo, and A. Colorni, *Positive feedback as a search strategy.* 1991.
90. Dorigo, M. and L.M.J.I.T.o.e.c. Gambardella, *Ant colony system: a cooperative learning approach to the traveling salesman problem.* 1997. **1**(1): p. 53-66.
91. Merkle, D. and M. Middendorf. *An ant algorithm with a new pheromone evaluation rule for total tardiness problems*. in *Workshops on Real-World Applications of Evolutionary Computation*. 2000. Springer.
92. Merkle, D. and M.J.A.I. Middendorf, *Ant colony optimization with global pheromone evaluation for scheduling a single machine.* 2003. **18**(1 :(p. 105-111.
93. Leguizamon, G. and Z. Michalewicz. *A new version of ant system for subset problems*. in *Proceedings of the 1999 Congress on Evolutionary Computation-CEC99 (Cat. No. 99TH8406)*. 1999. IEEE.
94. Stützle, T., et al., *A concise overview of applications of ant colony optimization.* 2011. **2**: p. 896-911.
95. Maniezzo, V., A.J.I.T.o.k. Colorni, and d. engineering, *The ant system applied to the quadratic assignment problem.* 1999. **11**(5): p. 769-778.
96. De Campos, L.M., et al., *Ant colony optimization for learning Bayesian networks.* 2002. **31**(3): p. 291-311.
97. Michel, R. and M. Middendorf. *An island model based ant system with lookahead for the shortest supersequence problem*. in *International conference on parallel problem solving from nature*. 1998 .Springer.
98. Boussaïd, I., J. Lepagnot, and P.J.I.s. Siarry, *A survey on optimization metaheuristics.* 2013. **237**: p. 82-117.
99. Rajan, C., et al., *Investigation on bio-inspired population based metaheuristic algorithms for optimization problems in ad hoc networks.* 2015. **9**(3): p. 163-170.
100. Wang, X., et al., *Intelligent welding robot path optimization based on discrete elite PSO.* 2017. **21**(20): p. 5869-5881.
101. Alzaqebah, M., et al., *Hybrid feature selection method based on particle swarm optimization and adaptive local search method.* 2021. **11**(3): p. 2414.
102. Ab Wahab, M.N., S. Nefti-Meziani, and A.J.A.R.i.C. Atyabi, *A comparative review on mobile robot path planning: Classical or meta-heuristic methods?* 2020. **50**: p. 233-252.

103. Shunmugapriya, P ,.S.J.S. Kanmani, and E. Computation, *A hybrid algorithm using ant and bee colony optimization for feature selection and classification (AC-ABC Hybrid).* 2017. **36**: p. 27-36.

104. Teodorovic, D., et al. *Bee colony optimization: principles and applications* .in *2006 8th Seminar on Neural Network Applications in Electrical Engineering*. 2006. IEEE.

105. Wang, X., et al., *Double global optimum GA-PSO based welding robot path planning.* 2015. **10**: p. 1080.

106. Couceiro, M.S., R.P. Rocha, and N.M.J.A.R. Ferreira ,*A PSO multi-robot exploration approach over unreliable MANETs.* 2013. **27**(16): p. 1221-1234.

107. Kanan, H.R., K.J.A.M. Faez, and Computation, *An improved feature selection method based on ant colony optimization (ACO) evaluated on face recognition system* :(2)205 .2008 .p. 716-725.

108. Nemati, S., et al., *A novel ACO–GA hybrid algorithm for feature selection in protein function prediction.* 2009. **36**(10): p. 12086-12094.

109. Menghour, K. and L.J.I.J.I.E.S. Souici-Meslati, *Hybrid ACO-PSO based approaches for feature selection.* 2016. **9**(3): p. 65-79.

110. Jona, J. and N.J.P.j.o.b.s.P. Nagaveni, *Ant-cuckoo colony optimization for feature selection in digital mammogram.* 2014. **17**(2): p. 266-271.

111. Basiri, M.E. and S. Nemati. *A novel hybrid ACO-GA algorithm for text feature selection*. in *2009 IEEE Congress on Evolutionary Computation*. 2009. IEEE.

112. Marinaki, M., Y. Marinakis, and C.J.A.S.C. Zopounidis, *Honey bees mating optimization algorithm for financial classification problems.* 2010. **10**(3): p. 806-812.

113. Zorarpacı, E. and S.A.J.E.S.w.A. Özel, *A hybrid approach of differential evolution and artificial bee colony for feature selection.* 2016. **62**: p. 91-103.

114. Suguna, N., K.J.I.J.o.G.C. Thanushkodi, Rough Sets, and I. Systems, *A weighted bee colony optimisation hybrid with rough set reduct algorithm for feature selection in the medical domain.* 2011. **2**(2): p. 123-140.

115. Suguna, N. and K.J.a.p.a. Thanushkodi, *A novel rough set reduct algorithm for medical domain based on bee colony optimization.* 2010.

116. Palanisamy, S. and S.J.I.J.o.C.S.I. Kanmani, *Artificial bee colony approach for optimizing feature selection.* 2012. **9**(3): p. 432.

117. Wang, X., et al., *Double global optimum genetic algorithm–particle swarm optimization-based welding robot path planning.* 2016. **48**(2): p. 299-316.