Spectrum of Operational Research

Journal homepage: www.sor-journal.org
ISSN: 3042-1470

Gain-based Green Ant Colony Optimization for 3D Path Planning on Remote Sensing Images

Sangeetha Viswanathan[1] and Kattur Soundarapandian Ravichandran[2,*]

[1] School of Computer Science, BITS Pilani University, Karnataka, India
[2] Department of Mathematics, Amrita School of Physical Sciences, Amrita Vishwa Vidyapeetham, Coimbatore, India

ARTICLE INFO

Article history:
Received 11 September 2024
Received in revised form 6 October 2024
Accepted 19 October 2024
Available online 29 October 2024

Keywords:
3D path planning; Unmanned ground vehicle; Ant colony optimization; Pheromone enhancement; Remote sensing images.

ABSTRACT

Metaheuristic algorithms are powerful methods for handling complexities in 3D environments because of their adaptability property. This paper proposes a gain-based, green-ant colony optimization (GGACO) method for 3D path planning on remote sensing images. Shortest paths do not always imply minimum energy consumption. Moreover, computational complexity tends to increase in the case of higher-dimensional data. A novel method is proposed to alleviate this issue, one that provides an efficient path with minimum energy consumption by adding a gain quantity during its search. The results are validated using performance measures, viz., path length, time, and energy cost. Real-time images, along with their corresponding ground truth and "digital surface models (DSM)", have been sourced from the "International Society for Photogrammetry and Remote Sensing (ISPRS)". Comparisons have been made against state-of-the-art algorithms and analyzed. Finally, the convergence and stability of the proposed method have been verified; it has been found that the proposed method outperforms the existing method by 6%, 11% and 5% regarding length, computation time, and energy, respectively.

1. Introduction

The advent of intelligent optimization methods and many Geographic Information System (GIS) technologies is causing a daily rise in research interest in three-dimensional (3D) path planning. The applications of 3D path planning had become widespread with the proliferation of many unmanned and automated systems. 3D path planning requires finding an efficient path between a start and a destination location. Unlike 2D path planning, 3D introduces the details of an additional height axis. Compared to 2D path planning, a plethora of challenges exist in 3D path planning.

For example, more information must be processed, because elevation information should also be considered during computation. This, however, increases both the system's computation time and its storage complexity. The densely populated environment still increases the system's complexity. Factors like uncertainty, dynamicity, data acquisition and data preprocessing also increase the overhead during 3D path planning [1]. Thus, efficient and intelligent methods must be used to solve

* *Corresponding author.*
E-mail address: *ravichandran20962@gmail.com*

https://doi.org/10.31181/sor21202510

a 3D path planning problem. Many algorithms have been proposed over past decades to solve 3D path planning efficiently.

These methods can be classified as *Exact and Approximation* methods. Traditional exact methods [2] like Dijkstra, Bellman-ford, and A* are widely used in a sparsely populated environment, while approaches like Rapidly Exploring Random Trees (RRT) have also been used by reducing the environment's complexity. Yang *et al.* [3] used RRT to compute the 3D path by reducing the number of points to be visited, which led to reduced complexity; however, they lost vital information. Many heuristic methods have also been investigated in this area over the past decade. Genetic algorithm-based 3D path planning was developed by Aybars *et al.* [4] on a set of predetermined surface points. The scheme was combined with the Voronoi diagram to reduce the computational cost. The Bacterial Foraging Optimization-based path planning method was investigated by Montinel *et al.* [5] in an unknown environment with moving obstacles.

Table 1 investigates some of the recent works from the literature regarding 3D path planning. Regarding the literature, it has been found that Classical exact methods tend to give an efficient 3D path when the environment is sparsely populated. On the one hand, the algorithm fails to provide an efficient solution as the environment's complexity increases. On the other hand, heuristics tend to produce near-optimal solutions in most cases; however, computational complexity increases due to the increase in the solution space while searching. Increasing the randomness of the solution space may result in optimality at certain times; however, this may affect robustness. A significant shift towards metaheuristics has been observed recently due to their ability to handle uncertain and complex environments efficiently [6]. They focus on improving the solution over a number of iterations, thus leading to quasi-optimal solutions. Therefore, methodology is highly recommended when the environment is highly complex and uncertain [6]. *Exploration and exploitation* are the two phases of metaheuristic-based searching. Exploration refers to globally searching the workspace to find good and better solution areas. Once the workspace is explored and the right solution areas are found, they are exploited to find promising solutions from those areas. This can be related to searching for the explored solution area. Correct exploitation may avoid bigger jumps in the search space and lead to faster convergence. Near-optimal solutions can be achieved by correctly tuning the exploration and exploitation phases of metaheuristics

Table 1
Recent literature analysis on 3D path planning

Ref.	Year	Method Adopted	Dataset used	Application	Optimality achieved
[7]	2018	Elevation mapping algorithm	Simulated datasets through real-world experiments	Mobile Robot navigation	Reduced localization drift
[8]	2018	Markov Random Field, Variational Feature Projection	Robot-simulated dataset	4WD Robot Navigation	Accurate modeling of complex features
[9]	2018	Fast Genetic Algorithm	Landscapes from Mt. Albert, British Colombia	Fixed-Wing Military UAV Navigation	Parallelization resulting in increasing speedup up to 280x
[10]	2018	Neural Activity-based Model	Artificial Terrains	Four-legged Robot Navigation	Reduced space complexity with an increase in accuracy of the path in uncertain environments
[11]	2017	D* Path Finding	Real-time experiments	Robot Navigation in disaster scenarios	Reduced response time

Ref.	Year	Method Adopted	Dataset used	Application	Optimality achieved
[12]	2016	3D Navigation Mesh Generation	Real-time experiments on Osnabruck University Botanical Garden	Mobile Robot Navigation	Reliable terrain classification with enriched local connectivity
[13]	2018	UAV path generator	Google Earth	DTM Construction-Debris Fan example	Reduced overlapping
[14]	2019	Critical Obstacles and Surrounding Point set (COSPS)	Simulated Environments	Applicable in any scenario	Increase in efficiency of the path with high robustness
[15]	2018	ACO-A*	New Zealand Bathymetric dataset	Underwater vehicles	Optimized path with less time complexity
[16]	2015	Energy-Efficient A* algorithm	Real-time experiments	Mobile Robot navigation	Minimized energy consumption with verified completeness, robustness, search efficiency
[17]	2017	Constrained MOPSO algorithm	Simulated environments	Car-Robot navigation	Faster convergence

Table 1 presents a discussion of different methods for 3D path planning, and we identify some potential challenges from the analysis:

i. Most methods in the literature are validated based on a simulated dataset or real-case, small-sized, low point density dataset. Validation using real-case, large-sized data with high point density is an open challenge to address.
ii. Methods discussed earlier suffer from high energy consumption problems, making them ineffective for onboard systems. Proposing a method that could be readily applied onboard is a significant challenge to focus on.
iii. Another key challenge that could be encountered from the analysis is the faster convergence of methods with little deviation among their independent runs. Thus, the less the deviation (among independent runs), the higher the stability of the method.
iv. Generally, when dealing with large-sized, high point density data, multi-objective optimization methods consume much energy and time in finding all non-dominated points of interest, which must also be addressed considerably for effective path planning. We infer from the analysis that some of these methods [18-21] tend to choose unnecessary paths and create path expansion, causing high energy consumption.

Motivated by these challenges and to address them effectively, this paper makes the following novel contributions.

i. Two new mechanisms viz., *surpassing* and *searching* is proposed for the construction of *traversable map* from the real time data obtained from “International Society for Photogrammetry and Remote Sensing (ISPRS)”.
ii. A “gain- based, green ant colony optimization (GGACO)” method has been proposed to find a cost and energy effective path on 3D terrains. The “pheromone enhancement” of “Ant System (AS)” is modified to exploit the solution found and help ants in their search towards high quality solution. The pheromone-enhancement mechanism is calculated through a *gain* function. This enhancement will reduce the number of ant traversals during ant activities, thus significantly *reducing the total energy consumption* (Kindly refer to Table 2).

iii. Moreover, Han Huang *et al*., [20], claimed that pheromone rate has a significant role in the convergence of "Ant Colony Optimization (ACO)". Motivated by the claim in this paper, we adopt relative Euclidean norm for pheromone enhancement, which promotes *faster convergence* by avoiding local optima (readers may refer Figure 10).

iv. The proposed method has been compared with state-of-the-art *multi-objective evolutionary algorithms* (MOEA); the results are analyzed for *stability*. We infer from the analysis that the proposed GGACO provides a substantial improvement with a considerable standard deviation (readers may refer Table 2, Figure 9) at optimal parameter settings (readers may refer Figure 7) over the state-of-the-art methods and is a better candidate for 3D path planning.

The rest of the paper is organized as follows: Section II presents the preliminary idea of Ant Colony Optimization and Input construction. Section III presents the proposed core idea in which the problem statement is formulated, the proposed GGACO algorithm is developed, and a detailed discussion of the algorithm is provided. Section IV demonstrates the practicality of the algorithm proposed by considering the 3D real- time data from ISPRS. The experiment is conducted, the results are compared with the extant methods, and discussions of the strength and weaknesses of GGACO are presented. Section V provides the concluding remarks and suggests future directions of the research

2. Preliminaries

2.1. Terrain representation

The recent advancements in the many "Geographic Information Systems (GIS)" have made "digital elevation models (DEM)" and "digital surface models (DSM)" available for many of the earth's locations. These models contain surface elevation information of geographic locations. They are standard digital datasets consisting of a matrix of terrain elevation data. Figure 1 shows DEM data with a surface model.

(a) (b)

Fig. 1. (a) A DEM's data (b) surface model of (a)

To enable path planning in this study, the model is represented as graph *G*, with the eight-connected neighborhood, as Figure 2 shows. For each node *V*, (x,y,z) are its terrain surface coordinates. The distance between the node V_i and node V_j (current node and it's neighboring node), in a plane with x-axis and y-axis, is given as

$$d(V_i, V_j) = \sqrt{\left(V_i(x) - V_j(x)\right)^2 + \left(V_i(y) - V_j(y)\right)^2} \tag{1}$$

The difference in elevation between the discussed nodes is given as,

$$h(V_i, V_j) = V_i(z) - V_j(z) \tag{2}$$

The slope between them given as,

$$\theta\left(V_i, V_j\right) = \frac{h\left(V_i, V_j\right)}{d\left(V_i, V_j\right)} \tag{3}$$

The Euclidean distance between them on a 3D space is thus given as,

$$d\left(V_i, V_j\right) = \sqrt{\left(\left(V_i(x) - V_j(x)\right)^2 + \left(V_i(y) - V_j(y)\right)^2\right) * \theta(V_i, V_j)} \tag{4}$$

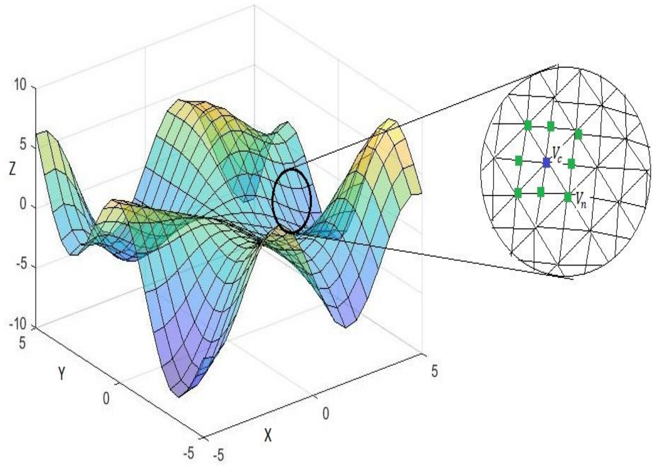


Fig. 2. DEM's data, eight-neighborhood representation

Various methods are used to optimize the distance cost of edges between nodes. A gain-based, energy-efficient, modified "ant colony optimization" method [22-24] has been developed in this study to optimize the cost. The work focuses on minimum energy consumption, so the energy cost of edges must be determined

2.2. Problem formulation

Let *R* be the area spread over the Cartesian coordinates *x, y, z*. Let *S* be the starting point and *D* be the destination point. Region *R* is structured into a graph; *G= (V, H)*. Let V_f be the set of vertices (traversable) such that $V_f \subset V$ and *H* be the edges. Inspiration is gained from [25,26]

Let $S_{x\times y\times z}$ be the subwindow in R with (s, u) being the current point. Then the neighbors of (s, t) in $S_{x\times y\times z}$are $(s, u-1)*\theta$, $(s, u+1)*\theta$, $(s+1, u)*\theta$, $(s-1, u)*\theta$, $(s-1, u-1)*\theta$, $(s-1, u+1)*\theta$, $(s+1, u-1)*\theta$, $(s+1, u+1)*\theta$. The final path is the set of traversable line segments between a set of traversable vertices from S to D. θ is the slope at the location. The entire workspace R consists of all the configurations (CN_{space}) of the UGV CN_{free} is the set of available configurations that avoid collision with obstacles. CN_{obs} is the complement of CN_{free}. CN_{target} is the subspace of CN_{target}, where the destination will reach.

The problem can be thus formulated as shown in (5) and (6). Given *S,D,* specifications of the UGV geometry of the workspace, the problem can be formulated as a multi-objective optimization problem such as,

$$Minimize\ h(x) = \sum_{i=1}^{r}\sum_{j=1}^{r} x_{ij}d_{ij} \tag{5}$$

$$Minimize\ g(x) = \sum_{i=1}^{r}\sum_{j=1}^{r} E_{ij}^{(t)} \tag{6}$$

Subject to

$$\sum_{i=1, i\neq j}^{r} x_{ij} = 1\ ,\ \ j = 1 \ldots\ldots r, \forall j\ \in \Lambda\backslash\{S\} \tag{7}$$

$$\sum_{j=1, j\neq i}^{r} x_{ij} = 1\ ,\ \ i = 1 \ldots\ldots r, \forall i\ \in \Lambda\backslash\{D\} \tag{8}$$

$$d_{ij} = \sqrt{(x_i - x_{i+1})^2 + (y_i - y_{i+1})^2} \tag{9}$$

$$x_{ij} = \begin{cases} 1 & for\ path\ between\ i\ and\ j \\ 0 & otherwise \end{cases} \tag{10}$$

Where the size of input is r.

Equation (10) represents the objective function to minimize the total distance. Equation (6) represents the objective function to minimize the energy consumed during traversal. Equations (7) and (8) indicate that there can be only one arrival (except start *S*) and one departure (except Destination *D*) at a time. Equation (9) represents the distance function to calculate distance. Equation (10) indicates that the node is either included or discarded from the list.

2.3. Dataset description

Satellite images, along with their corresponding ground truth and DSM are obtained from the "International Society for Photogrammetry and Remote Sensing (ISPRS) [27]". Elevation data of 30-90m resolution can also be obtained freely from the "Shuttle Radar Topography Mission (SRTM)". Furthermore, data can be obtained from www.usgs.gov.in. Figure 4 shows a sample height map and its corresponding DEM. Landscapes of Vaihingen and Potsdam, which are different sizes, as mentioned in Section IV, are considered for the analysis.

2.4. Ant Colony Optimization

"Ant Colony Optimization (ACO)", proposed by Marco Dorigo *et al* [22], is a swarm-based, nature-inspired algorithm based on the foraging behavior of ants when finding a path between their nest and a food source. Ants deposit a chemical liquid called pheromone when they traverse between points. This helps communication between them so they can follow each other. The intensity of the pheromone deposited on a path is proportional to the number of ants following the path. The path with high pheromone intensity will be chosen for traversal. Artificial ants, the counterparts of real-world ants are like real-world ants except that they have their own memory and internal state. Two processes, pheromone evaporation and pheromone accrual, explain the variation in the pheromone intensity on the paths. Positive feedback on the trail indicates that a higher number of ants shall take the path relating to pheromone accrual. Negative feedback on the trail shows that a lower number of ants has considered the path, and the path shall be avoided relating to pheromone evaporation. The pseudo code for ACO is given in section 2 of the supplementary file. Table A1 gives the notations and parameters used in ACO (refer to the supplementary file). The behaviors of ants are simulated into various types of models, such as Min-Max Ant System (MMAS), Ant Colony System (ACS), Ant System (AS), Elitist Ants, Rank-based Ants and like [23] From the investigation of these mechanisms, it is inferred that (i) AS is simple and straightforward in nature. (ii) Unlike the variants of ACO, AS reduces the overhead in search mechanism by avoiding artificial parameter settings. Motivated by these inferences, this work enhances the pheromone-update mechanism of the AS to exploit the current strong solution and direct the ants in their search towards high quality solution. Use of meta-heuristic procedures for path planning is gaining attention in the present decade [28].

3. Proposed gain-based green ant colony optimization (GGACO)

The ants traverse around the workspace to build solutions. During traversal, the path with high pheromone intensity is preferred by the ants. A new gain-based, pheromone-enhancing function has been introduced based on this Ant system principle. The idea of GGACO has been adapted from [24]. For curious readers, the core differences between the proposed GGACO & ACO are listed here:

Unlike ACO, GGACO considers the number of neighbors of the next node in the node transition probability as given in equation (15). The greater the number of neighbors for a node, the greater the chances for its selection during Node Transition Probability (NTP) NTP calculation. When the number of neighbors is greater for a node, the amount of exploration around it increases, thus leading to a larger current search space.

Unlike ACO, GGACO's convergence is faster because of the pheromone enhancement on the currently found best path. The sigmoid function determines the amount of pheromone added or subtracted on the available trail based on the relative distance of the current node. Thus, in addition to the NTP values of the nodes, the pheromone gain of each node determines the next node more accurately.

Unlike ACO, GGACO provides paths with smooth turns because of the sigmoid function-based pheromone enhancement [26]. Furthermore, GGACO is scalable irrespective of the obstacles' shape and the environment size, as shown in section IV-D.

The core uniqueness of this paper is:

GGACO has been formulated for real-time remote sensing images (3D) by integrating DSM with remote sensing images obtained from the ISPRS repository.

Unlike [24], traversable maps are constructed from the visualized terrain data using two new mechanisms viz., Surpassing and Searching based on θ values.

Moreover, the proposed GGACO considers relative distance to decide the next points, which enables intuitive selection of the next reversible point rather than considering the total distance between S and D [24].

Furthermore, unlike [24], the proposed GGACO identifies an optimal path in the 3D complex environment that provides adequate information for efficient path planning. GGACO retains the information without potential loss of generality.

Like other meta-heuristic algorithms, ACO undergoes both exploration and exploitation during its foraging process. Achieving an optimal convergence requires maintaining an equilibrium between exploration and exploitation. Ant system is based on two factors: pheromone trail τ and heuristic η. Exploration is performed by the extensive search of the heuristic η and pheromone trail update performs exploitation. The traditional update rule operates based on the amount of pheromone on the links. In addition to the amount of pheromone on the link, a gain quantity (proposed pheromone enhancement) is added or subtracted during the pheromone update (equation 13). This gain quantity is obtained by performing a local search based on the relative distance to destination with respect to the neighbor. The proposed method leads to a faster convergence by enhancing the pheromone on the currently best path, thus leading to better exploitation. The proposed Gain function will help to avoid unnecessary traversals, thus resulting in minimum energy consumption.

3.1. Calculating Gain

On to the best path found so far; in order to quicken the process of pathfinding an amount of pheromone is added. This is called a pheromone gain. Thus, the new quantity added will enable quicker pathfinding by reducing the energy calculation in all paths around the obstacle. Pheromone gain is given by (12).

$$progress_{jDest} = \frac{\delta_{jDest}}{d_{iDest}} \tag{11}$$

$$Gain_{ij} = \frac{1}{(1+e^{-\lambda * progress})} \tag{12}$$

where V_{dest} = Destination vertex;

Where V_j is the neighbor node, V_i is current node, $0 \leq \lambda \leq 1$, and λ decides the degree of smoothness.

$\delta_{jDest} = dj_{Dest}$; where $d_{jDest} = \sqrt{\left(V_j - V_{Dest}\right)^2}$ for j = 1, 2, 3….8.

For all values of λ and progress, the value of the gain always lies between 0 and 1. Values obtained from the Sigmoidal function will help to transit the UGV to the destination position smoothly.

Consider a configuration space of 20X20, as Figure 3 shows.

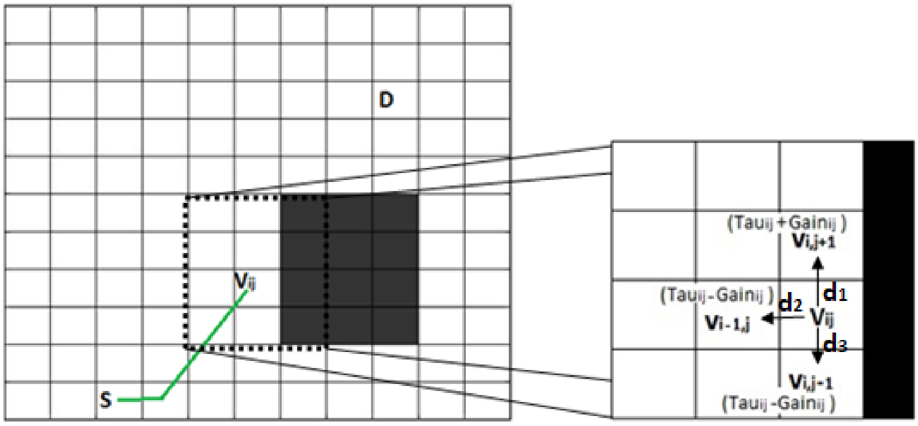


Fig. 3. Abstract representation of Pheromone gain process

The algorithm starts from the known start position. The node to be visited next is searched upon reaching V_{ij} (current position). The node with the shortest distance ($V_{i,j+1}$) is selected as the next node to be visited, and the corresponding pheromone trail is updated accordingly. During the pheromone update, $Gain_{ij}$is added to τ_{ij}^{new} of $V_{i,j+1}$, and subtracted from the τ_{ij}^{new} of $V_{i,j-1}$,and $V_{i-1,j}$. If all the current nodes' neighbors have the same distance, then gain is added to all nodes. Mathematically the procedure can be written as,

δ_{jDest} = d_{jDest}; where $d_{jDest} = \sqrt{\left(V_j - V_{Dest}\right)^2}$ for j = 1, 2, 3….8. If the path provides minimum δ_{jDest} , then the gain will be added along with the current pheromone quantity, otherwise subtracted.

$$\tau_{ij}^{new} = \begin{cases} (1-\rho)\tau_{ij}^{old} + \sum_{k=1}^{N} \Delta\tau_{ij}^{k} + Gain_{ij}^{k} & for\ min\left(\delta_{jDest}\right) \\ (1-\rho)\tau_{ij}^{old} + \sum_{k=1}^{N} \Delta\tau_{ij}^{k} - Gain_{ij}^{k} & otherwise \end{cases} \tag{13}$$

Using equation (13), τ_{ij}^{new} is calculated and used in equation (15).

According to equation (13), $Gain_{ij}^{k}$ of k^{th} ant is added to the trail of the node with min δ_{jDest} and subtracted from other trails. As a result, the amount of pheromone on the current best path will accumulate, whereas it evaporates on the other trails. This will enable the ants to settle down in the current best trail with less latency, thereby leading to faster convergence. Table A4 gives the performance of the proposed GGACO with other optimization techniques.

3.2. Methodology for GGACO

Path planning using GGACO on uneven terrain (3D) consists of two stages. Its detailed explanation is given as follows:

i. Terrain Data Preparation
ii. Gain-based Green Ant Colony Optimization

3.2.1. Terrain Data Preparation

The acquired satellite images from ISPRS and its corresponding DSM are processed using Open Source QGIS 3.4.3®, and the terrain model is visualized. Figure 4 shows a sample DSM, Surface plot, and the terrain model.

The slope (θ) is calculated for all points, from the terrain matrix, and a traversable map is generated using these slope values. Two mechanisms, Surpassing and Searching, are considered.

i. Surpassing: For $\underline{z} < \theta < \overline{z}$, the corresponding points of θ are to be considered traversable, allowing the ants to surpass those points and continue their traversal.
ii. Searching: For $\theta \notin [\underline{z}, \overline{z}]$, the corresponding points of θ are to be considered non-traversable, treating them as obstacles for the ants.

The parameters $\overline{z}, \underline{z}$ are tunable based on the specific the UGV. Here, $\overline{z}, \underline{z}$ represent the upper and lower bounds of θ that can be surpassed by the UGV.

3.2.2 Gain based Green Ant Colony Optimization

3.2.2.1 Environment understanding

Map creation is carried out in this phase. First, the traversable map constructed from 3D terrain is modeled into a grid map suitable for simulation. Many methods, such as occupancy grid, Voronoi diagram, and probabilistic road maps, are available for this transformation. This paper uses the occupancy grid method because of its low storage requirements and fast, accurate discretization [25]. Each unit- sized grid has two probabilities of being either an obstacle or a free space. A single point is associated with each grid, and two points are connected only if there is an adjacent grid associated with them. Nodes are represented as points in the occupancy grid map. Each grid is connected to others through a link representing the edges in a graph. Thus, the entire configuration space is designed as *G = (V, H),* where *V* is the point in the grid and *H* is the link. The configuration space, free space, and obstacle space are defined. Start and destination/target positions are configured.

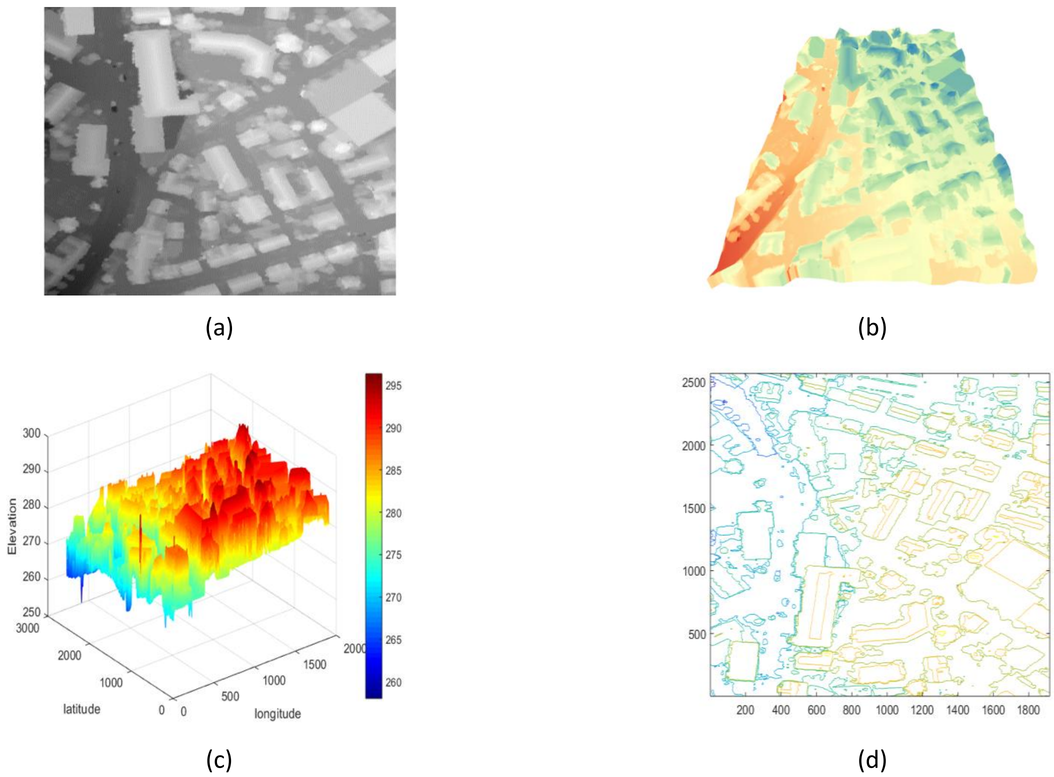


Fig. 4. (a) Height map (b) Terrain model (c) Surface plot (d) Contour plot

3.2.2.2 ACO initialization

In this phase, the parameters of the Ant systems are initialized. The initial value of τ_o is set to 0. The parameters α, β, ρ, *start node, and destination node* are also initialized. The parameter values used for the simulation are provided in Table A1 (refer to the supplementary file). Figure 5 gives the flow diagram for GGACO working.

3.2.2.3 Cost and energy calculation

Upon reaching a vertex, ants check in the list $visit_k$ for the next node to be visited. $visit_k$ contains all the nodes that are allowed to be visited. This is the forward move of map exploration. The pheromone values of all links are updated initially based on equation (13). The gain calculation for pheromone enhancement on the current best link is performed using equation (14)

$$Gain_{ij}^{k} = \frac{1}{(1+e^{-\lambda * progress_{jdest}^{k}})} \tag{14}$$

where $progress_{jdest}^{k} = \frac{\delta_{jDest}^{k}}{d_{iDest}^{k}}$.

Here $\delta_{jDest}^{k} = d_{jDest}^{k}$; $d_{jDest}^{k} = \sqrt{\left(V_j^k - V_{Dest}^{k}\right)^2}$ and $d_{iDest}^{k} = \sqrt{\left(V_i^k - V_{Dest}^{k}\right)^2}$. The sigmoid function is utilized to compute the pheromone gain on the current optimal link. This function features a '**S**' shaped curve, with values ranging from 0 to 1. Upon closely examining the sigmoid cure, it resembles a path with smooth turns, which is the ideal for most UGV path planning problems [26]. The function's inherent tendency towards smoothness and bounded intervals helps identify the current best path. Also, $progress_{jdest}^{k}$ in equation (14) calculates the relative distance between the current point and destination with respect to the current neighbor. A measure of relative distance rather than total distance will enable finding the nearest neighbor node, through which an optimal path can be found when traversed.

(Note: For convenience throughout this paper, (V_i, V_j) and (i,j) are used interchangeably.)

Node transition probability is used to determine the node to be visited next. The equation (15) is used for the forward move.

$$NTP_{ij}^{k}(t) = \begin{cases} \dfrac{\left(\tau_{ij}^{k}\right)^{\alpha}\left(\eta_{ij}^{k}\right)^{\beta}}{\sum_{h \in visit_k}\left(\tau_{ih}^{k}\right)^{\alpha}\left(\eta_{ih}^{k}\right)^{\beta}} * \dfrac{n_j}{n_j+1} & if\ j \in visit_k \\ 0 & otherwise \end{cases} \tag{15}$$

Where, the cost of edge (i,j) is η_{ij}, τ_{ij} calculated by (18) through the backward move is the pheromone intensity in the edge (i,j), and is *h* denotes the nodes that do not belong to the tabu list, the possible unvisited neighbor nodes. n_j is used to calculate components connected to the next vertex *j*. n_j denotes the number of neighbors of node *j*. The more the number of neighbors for *j*, the more the chances are for it to be selected for exploration. η_{ij} is calculated using (16).

$$\eta_{ij}^{k} = \frac{1}{d_{ij}^{k}} + \frac{v_{ij}}{\varphi} \tag{16}$$

$$d_{ij}^{k} = \sqrt{\left(V_i^k - V_j^k\right)^2} * \theta_{ij} \tag{17}$$

where v_{ij} denotes the speed at that instant over (i,j), φ indicates the maximum speed of UGV and has been set to 2 m/s [21,22] to normalize the value. In equation (17), θ_{ij} is the slope between *i* and *j*.

The cost of traversal between *i* and *j* is calculated using (16). Equation (16) represents that the choice of the next node has a direct relation with UGV speed and an inverse relationship with distance.

3.2.2.4 Path planning

In this phase, as the forward movement is completed, the backward move is initiated by the ants. The pheromone update rule is used to find the current pheromone quantity, during this move. The pheromone's intensity is updated using (13). The usability of the path is either increased or decreased through pheromone reinforcement or pheromone evaporation, respectively.
$\Delta\tau_{ij}^{k}$ is given by (18),

$$\Delta\tau_{ij}^{k} = \begin{cases} \frac{1}{d_{ij}^{k}} + \frac{1}{E_{ij}^{k}} & if\ k^{th} ant\ passes\ i\ and\ j \\ 0 & otherwise \end{cases} \tag{18}$$

where d_{ij}^{k} is the distance covered by k[th] ant over the edge (i, j), and similarly E_{ij}^{k} is energy consumed.

Algorithm 1 provides the pseudocode for complete GGACO. The algorithm begins on line 1, followed by initialization of ant system parameters τ_o , α, β, ρ,S,D, N. S, and D are the source and destination. N is the total number of ants. The procedure between lines 3-29 runs a predetermined number of iterations. The steps between lines 4-21 are repeated for all ants. For each ant, $Gain_{ij}^{k}$is calculated according to lines 6-8. in line 6, δ_{jDest} is taken as the distance between the current neighbor and the destination. The value of $Gain_{ij}^{k}$ is calculated according to line 8 and 9. The relative distance between the destination-current node and destination-neighbor is calculated as Progress and passed on to the sigmoidal function under $Gain_{ij}^{k}$ in line 9.

Once $Gain_{ij}^{k}$ is calculated, the pheromone values of all links are updated according to line 11. $Gain_{ij}^{k}$ is added for the node whose δ_{jDest}^{k} is minimum. $Gain_{ij}^{k}$ is subtracted from other links, thus performing exploitation at the current search level. This operation also helps to channel the ants towards the optimal path. With the updated pheromone values, the next node to be traversed is found in lines 13-15 for all ants.

The list $visit_k$ of all ants is updated based on the NTP calculation in line 16. The best path found so far is updated in line 17. If the solution converges (target node reached), the tabu list of other ants is updated, and pheromone update is performed using line 29. Once the ants reach the next nodes, their pheromone trail has to be updated. The pheromone update is based on distance and energy according to the objective in line 28. During the δ_{jDest}^{k} calculation, if two neighbors have the same distance, $Gain_{ij}^{k}$ is added to the E calculation during energy calculation and pheromone trail is updated according to line 24. The process ends when the predetermined number of iterations is reached. The algorithm returns the best path according to line 26. The algorithm continues until either of the following conditions is achieved; namely, the shortest path is obtained, or a predetermined number of iterations has been completed.

Algorithm 1

Begin
Initialize τ_o , α,β,ρ,S,D,N
while (max_number_of_iterations_not_reached) do
for i=1:N do
// Compute 〖Gain〗_ij^k
δ_jDest=d_jDest, j=1,2…8
for all V_j of V_i do

$$progress_{jdest}^{k} = \frac{\delta_{jDest}^{k}}{d_{iDest}^{k}}$$

$$Gain_{ij}^{k} = \frac{1}{\left(1+e^{-\lambda * progress_{jdest}^{k}}\right)}$$

end for
//Compute new pheromone quantity:

$$\tau_{ij}^{new} = \begin{cases}(1-\rho)\tau_{ij}^{old} + \sum_{k=1}^{N}\Delta\tau_{ij}^{k} + Gain_{ij}^{k} & for\ min(\delta_{ij}) \\ (1-\rho)\tau_{ij}^{old} + \sum_{k=1}^{N}\Delta\tau_{ij}^{k} - Gain_{ij}^{k} & otherwise\end{cases}$$

// Compute Node Transition Probability (NTP) :
for each kϵN do

$$NTP_{ij}^{k}(t) = \frac{\left(\tau_{ij}^{k}\right)^{\alpha}\left(\eta_{ij}^{k}\right)^{\beta}}{\sum_{h\in visit_k}\left(\tau_{ih}^{k}\right)^{\alpha}\left(\eta_{ih}^{k}\right)^{\beta}} * \frac{n_j}{n_j+1}$$

end for
Update $visit_k$;
Update the best_ path among N;
If (target_reached)
Update $visit_k$ for all N
break
end if
end for
//Update pheromone trail:

$$\Delta\tau_{ij}^{k} = \begin{cases}\frac{1}{d_{ij}^{k}} + \frac{1}{E_{ij}^{k}} & if\ k^{th} ant\ passes\ i\ and\ j \\ 0 & otherwise\end{cases}$$

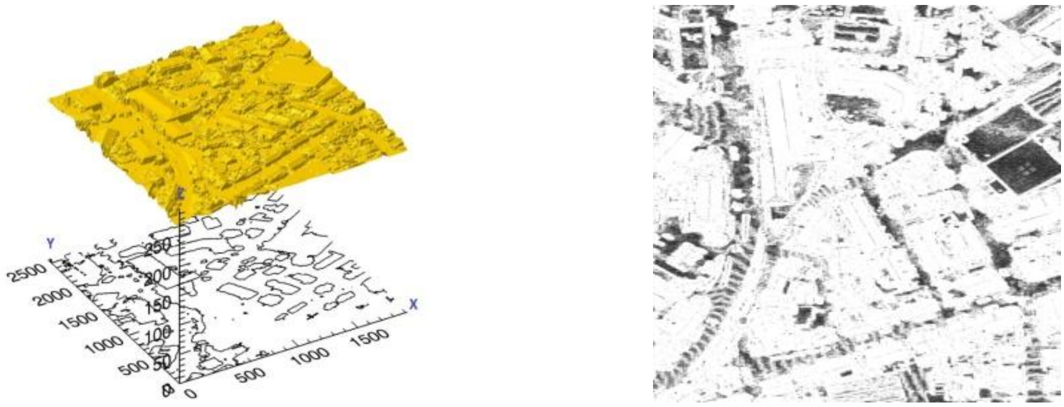
end while
return best_ path
end begin

4. Experimental analysis and discussion

This section evaluated GGACO against existing algorithms for energy-efficient path planning on remote sensing images. Extensive tests were performed to check for optimality, scalability, and search for the effectiveness of the algorithms. Tests were also carried out for optimally choosing GGACO parameters

4.1. Experimental setup

The proposed algorithm was implemented in MATLAB R2018b®. DSM data obtained from ISPRS [27] was modeled using QGIS. Images from the Vaihingen, Potsdam landscapes dataset have been taken to test the algorithm on different dimensions. Figure 6 shows one such image and its slope value. Figure 4 shows the modeled terrain and corresponding height map. Five different test cases/images of sizes 2569 x1919, 3018 x 2178, 1789 x 1512, 2789 x 2003, 1568 x 1800 have been considered for simulations.



(a) (b)

Fig. 6. (a) DSM data modeled as terrain (b) Slope of (a)

Performance Measures:

The proposed method's performance is assessed using the following performance metrics. Each algorithm is executed 20 times, and the results are analyzed as follows:

i. *The mean and standard deviation* of the solutions are calculated and compared to assess the stability of the proposed method.

ii. *The average, minimum, and maximum* computation times across the 20 runs are compared to evaluating the algorithm's computational efficiency.

iii. *The energy* consumed during traversal is measured to analyze the overall efficiency of the algorithm.

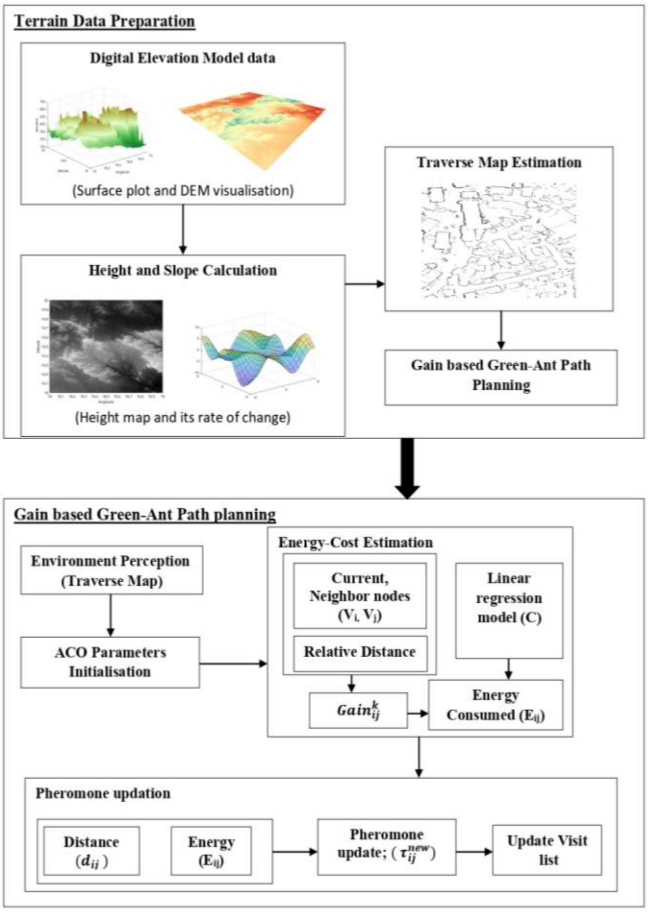


Fig. 5. Flow Diagram of proposed GGACO

Parameter Setting

The simulations were conducted under different conditions, such as a different number of iterations and varying values of ρ. Simulations were done 20 times, and the average has been considered. Determining an optimal value of ρ is essential, because the strength of GGACO is mainly based on the pheromone-enhancing mechanism. Simulations were performed that varied the value of ρ from 0.1 to 0.9, and Figure 7 shows the results. It can be seen from the box plots in Figure 7 that the distribution range of values when ρ=0.4 is the least when compared with other values. The algorithm exhibits more stability and minimum variation with no outliers when at ρ=0.4. Table A2 (refer to supplementary file) gives all the parameters, and their values used for simulations [28]. From Figure 7, it can be found that the value of fitness evaluation is minimum when ρ=0.4. Thus ρ=0.4 is set for all further simulations

Performance Evaluation

The proposed GGACO method has been compared for its performance against LRO-GA [29], MOSFLA [30], ACO-A* [15], MOVNS [31], Z* [16], and MOPSO [17] and Table 2 tabulates the results. The following *drawbacks* are inferred from the literature analysis:

Reduction in energy consumption is not considered. Although [15] deals with a reduction in energy consumption, sensitivity analysis over the size of data is not performed, which gauges the scalability factor of the algorithm.

As the size of data increases, methods [17,29,30] performs weakly in terms of the convergence rate, i.e., these methods have slower convergence when compared with GGACO. Conversely, methods [15,16,31] perform closely with GGACO, but these methods do not consider the reduction of energy consumption for different sizes of input. The main advantage of GGACO is its ability to deal with large- sized data wherein discussions regarding input size are not often made in existing works. Table A3 (refer supplementary file) gives the parameter values of the state-of-the-art methods and their description. Each algorithm is executed 20 times, and the parameters (provided in Section IV-C) are analyzed. It is generally observed from the Literature that energy consumption and cost of the path can be related directly to the number of turns in paths. The more the number of turns in a path, the more cost and energy are consumed to traverse the path. The proposed GGACO method reaches D with a fewer number of turns when compared with other algorithms. Figure 8 shows the path obtained by GGACO, LRO-GA, MOSFLA, ACO-A*, MOVNS, Z*, and MOPSO. It can be observed from Figure 8 (a-g) that GGACO has the path with the fewest number of turns. The paths obtained through MOSFLA and LRO-GA (Figure 8 (c) and (e)) take the longest length and have more turns. GGACO was able to overcome the local optima trap by efficient searching through the added Gain function. The path obtained through Z* and ACO-A* are lengthier than GGACO with more turns. Moreover, the path obtained in Figure 8 (b) & (d) has more of its traversable points on high θ space, thus leading to more energy consumption. Table 2 shows the comparisons between all the previously mentioned algorithms. The path in Figure 8 (f) is not smooth and has more turns than that in Figure 8 (a). Moreover, it is longer than the proposed method. It can be inferred from Table 2 that the length and time taken for the path obtained through GGACO is the least when compared with other methods in case 1,2,4. The energy cost of GGACO is the least when compared with the existing techniques in these cases. During traversal, ants look for a path with a higher pheromone value. The Gain function enhances the value of pheromone on the shortest path at time t. Since optimization is done while determining the path segments of the final path, additional overhead in traversing around all available paths during searching is avoided. Additionally, the extra energy consumed will also be avoided, thus leading to minimum energy consumption. The fitness values of GGACO is slightly less than that of ACO-A* in cases 3 and 5. The sizes of cases 3 and 5 are relatively smaller than the others. The proposed method produces more significant results when compared with other methods in the cases of larger environments. The proposed GGACO has been compared with existing methods based on descriptive parameters like mean, maximum value, minimum value, standard deviation. The box plot analysis of the comparison of the proposed method against other methods is shown in Figure 9.

It can be inferred from Table 2 that the standard deviation of GGACO is less than all the other methods. Standard Deviation is the measure of an algorithm's stability. For the optimal value of ρ, the algorithm converges and is stable. The gain-based, pheromone- enhancing mechanism and the optimal value of ρ helps to escape the local optima trap and in faster convergence of the algorithm. Although the standard deviation of GGACO is not optimal in cases 3 and 5, it is agreeable. Table A4 (refer supplementary file) gives the minimum and maximum values of performance comparison statistically verified through one-tailed ANOVA at 95% confidence.

Figure 9 illustrates the box plot analysis of the simulations. A box plot is a way of demonstrating the spread of data values within the range. It indicates the solution space of the problem. It may consist of outliers represented as dots outside the solution space. The median line represents the average fitness at the run. From Figure 9, in almost all cases, the value distribution of GGACO (Blue box plot) is the same as that of ACO-A*; however, GGACO exhibits lower fitness evaluation values in cases 1, 2 and 4 (Figure 9 a,b,d). The fitness value for ACO-A* is the least with the least minimum value in case 3 (Figure 9 (c)). The proposed GGACO outperforms other algorithms with the least minimum and less variation in the case of large environments. Figure 10 shows the convergence analysis of the simulations illustrated in Table 2. In the two-dimensional coordinate system of the convergence graph, the number of iterations is expressed by the abscissa, and the ordinate represents the value of fitness evaluation as obtained from pheromone update. Figure 10 (a,c) shows the convergence with respect to average fitness and number of iterations, and Figure 10 (f,h) shows the convergence with respect to minimum fitness and number of iterations. In larger test cases (1,2,4), GGACO converges for 1000 iterations approximately, which is earlier than the other methods.

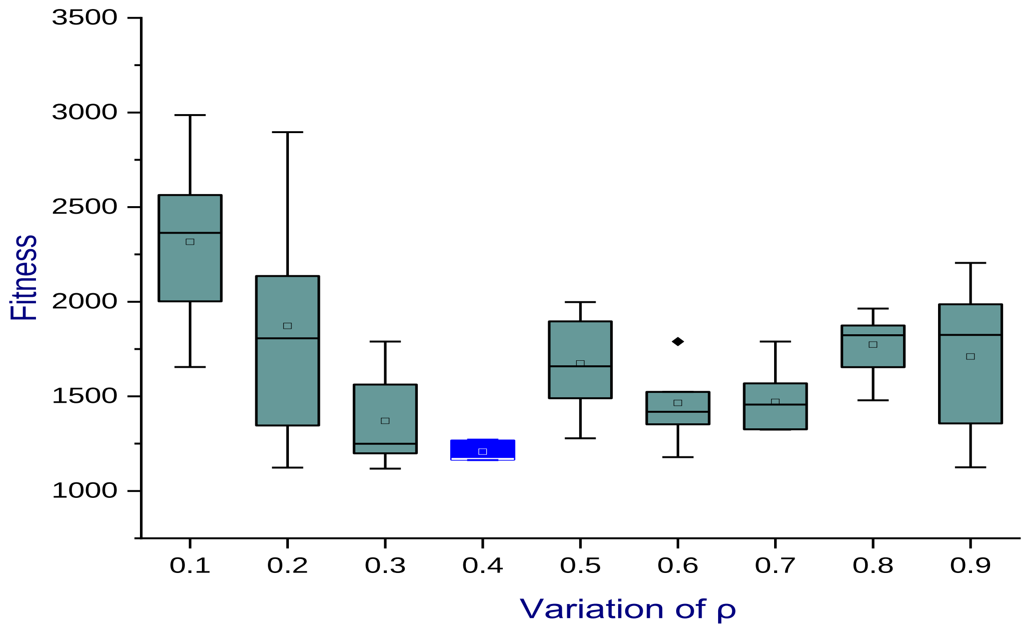


Fig. 7. Variation of ρ

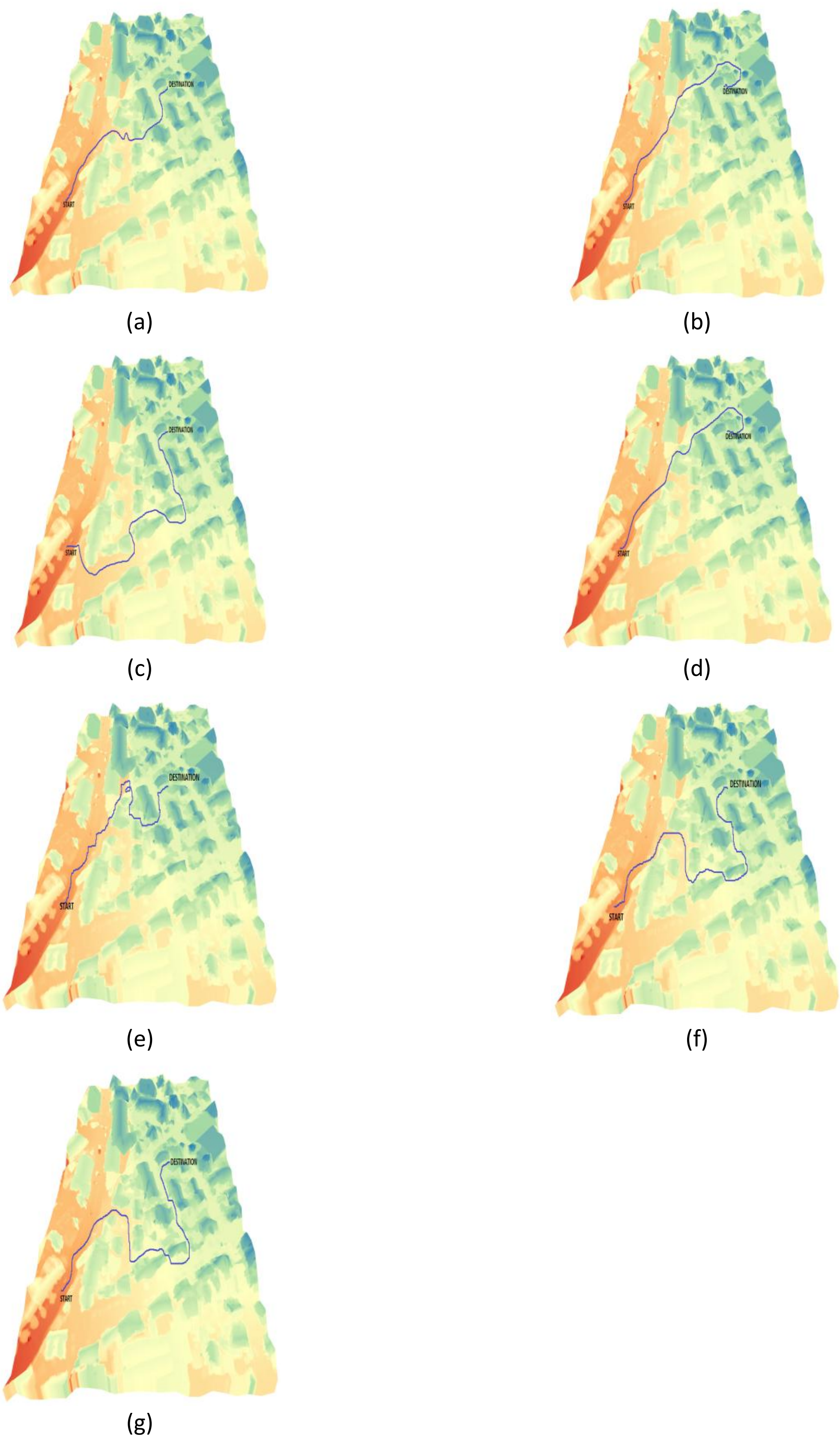


Fig. 8. Path obtained through (a) GGACO (proposed), (b) Z* (c) MOSFLA (d) ACO*A* (e) MOVNS (f) LRO-GA (g) MOPSO

Table 2
Performance Analysis of Proposed Algorithm against Existing Methods. Medians from 20 independent runs

Case	Method	Length (No. of points)		Time (sec.)		Energy (kJ)	
		Median	SD	Median	SD	Median	SD
1	1	1167.9	52.11	2950.5	82.01	454.2	8.42
	2	2203.8↑	499.36↑	3708.4↑	118.12↑	702.5↑	29.46↑
	3	1236.1↑	109.85↑	3156.5↑	159.22↑	472.9↑	12.28↑
	4	2105.4↑	83.79↑	3196.5↑	258.86↑	612.1↑	22.48↑
	5	1172.9↑	70.834↑	3069.1↑	397.02↑	450.9↓	12.34↑
	6	1166.4↓	42.08↓	3000.5↑	92.63↑	474.2↑	10.84↑
	7	2010.6↑	165.65↑	3188.6↑	226.80↑	598.8↑	24.79↑
2	1	1211.1	20.79	3267.7	44.14	533.7	14.91
	2	2466.6↑	321.37↑	3721.6↑	189.28↑	798.8↑	27.59↑
	3	1321.6↑	111.16↑	3421.1↑	161.28↑	538.6↑	23.89↑
	4	1345.7↑	68.64↑	3677.8↑	185.08↑	641.7↑	34.56↑
	5	1240.9↑	195.86↑	3678.4↑	215.50↑	533.7≈	19.01↑
	6	1252.4↑	38.45↑	3266.7↓	55.91↑	515.9↓	13.45↓
	7	1700.6↑	61.23↑	3325.9↑	65.89↑	661.3↑	26.84↑
3	1	984.3	6.16	1922.3	82.76	384.8	7.89
	2	1116.1↑	77.07↑	2196.16↑	262.89↑	549.3↑	24.56↑
	3	1009.6↑	34.95↑	1992.5↑	117.24↑	400.2↑	21.87↑
	4	1068.5↑	64.42↑	2103.5↑	127.27↑	399.9↑	21.54↑
	5	1007.7↑	67.74↑	2014.1↑	151.69↑	451.6↑	18.71↑
	6	979.5↓	5.48↓	1912.7↓	79.56↓	384.8≈	7.89≈
	7	996.6↑	10.26↑	1915.1↓	104.56↑	385.2↑	15.47↑
4	1	1142.1	27.89	2569.1	113.51	420.5	9.56
	2	2029.2↑	57.21↑	3011.4↑	281.33↑	431.4↑	26.16↑
	3	1181.4↑	33.24↑	2648.9↑	226.54↑	428.6↑	25.38↑
	4	2005.2↑	37.89↑	3066.6↑	414.19↑	426.7↑	26.49↑
	5	1201.6↑	34.56↑	2757.3↑	348.56↑	422.5↑	21.56↑
	6	1164.1↑	28.74↑	2541.8↓	110.48↓	421.1↑	11.45↑
	7	1165.8↑	30.48↑	2795.8↑	301.47↑	419.9↑	13.88↑
5	1	871.2	12.56	1631.1	104.5	361.7	6.51
	2	642.7↓	34.37↑	1587.2↓	138.78↑	387.41↑	19.81↑
	3	881.3↑	24.45↑	1701.6↑	117.41↑	369.1↑	18.53↑
	4	929.7↑	27.89↑	1703.2↑	120.45↑	382.3↑	17.45↑
	5	921.9↑	26.48↑	1659.5↑	119.45↑	367.5↑	18.03↑
	6	871.2≈	12.56≈	1631.1≈	104.5≈	361.7≈	6.51≈
	7	896.2↑	17.89↑	1637.2↑	115.89↑	364.1↑	7.56↑

Time (w/t/1)	Proposed	MOSFLA	MOVNS	LRO-GA	Z*	ACO-A*	MOPSO
Proposed	----	4/0/1	5/0/0	5/0/0	5/0/0	2/1/2	4/0/1

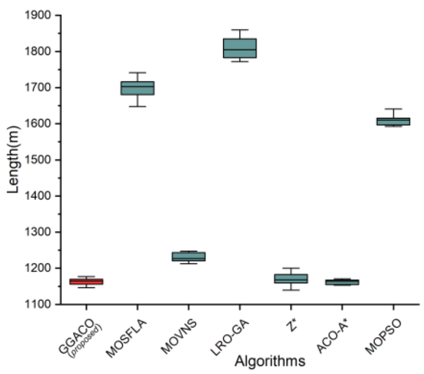
Energy (w/t/1)	Proposed	MOSFLA	MOVNS	LRO-GA	Z*	ACO-A*	MOPSO
Proposed	----	4/0/1	5/0/0	5/0/0	5/0/0	2/1/2	5/0/0

Length (w/t/1)	Proposed	MOSFLA	MOVNS	LRO-GA	Z*	ACO-A*	MOPSO
Proposed	----	4/0/1	5/0/0	5/0/0	5/0/0	2/1/2	5/0/0

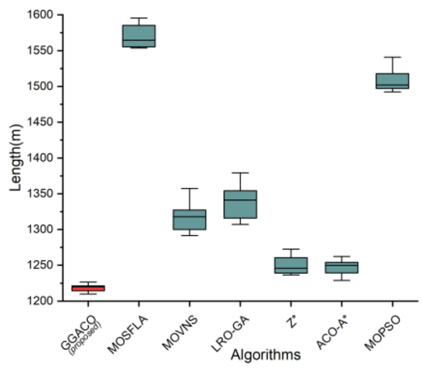
Note:
SD- Standard Deviation
The Wilcoxon signed-rank test at 95% confidence is performed to compare GGACO with others.
w/t/1: number of results where GGACO statistically wins/ties/loses with other methods for all cases
Here, from length analysis, proposed method wins in 4 cases, loses in 0 cases and ties in 1 case with MOSFLA statistically
↑, ↓, ≈: Proposed method statistically performs better, worse, equal

In cases 3 and 5, GGACO converges a bit slower than the other methods because the size of the environment is considerably smaller than the other test cases. It is inferred from all the preceding analyses that the proposed GGACO method is more stable than the existing methods when handling larger environments. It converges faster than the other methods because of the "gain-based, pheromone-enhancing mechanism". The efficiency of the algorithm is improved through the pheromone-enhancing mechanism, thus improving the quality of the solution.

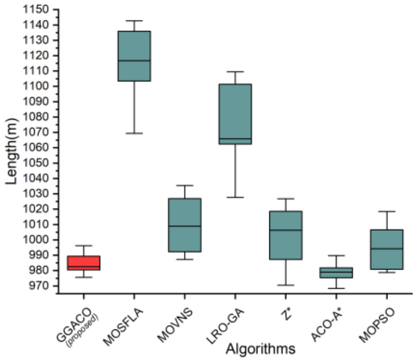
4.2. Discussions
Using Table 2 and based on the experiments conducted, the following observations have been made. Unlike [16], GGACO handles input of varying sizes, thus proving to be 7%, 8%, and 7% efficient in terms of length, time, and energy, respectively, under smaller input size (case 3,5).Also, GGACO is 4%, 7%, and 6% efficient in terms of length, time, energy, respectively, under larger input sizes (case 1, 2, 4).Upon comparison with [15],GGACO was found to be 4%, 11%, and 11% efficient with higher input size (case 1,2,4) and the same results with smaller input sizes (case 3,5) in terms of length, time, energy, respectively. GGACO outperforms [29] with a greater margin of 49%, 20%, and 33% in terms of length, time, energy, respectively, under both input sizes. When compared with [17,30,31], GGACO was found to be more efficient by 10%, 7%, and 10% in terms of length, time, energy, respectively.



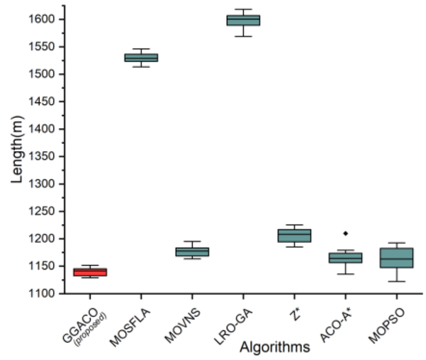
(a)



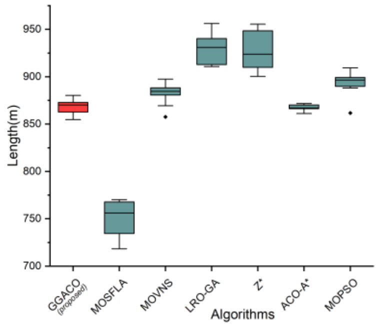
(b)



(c)

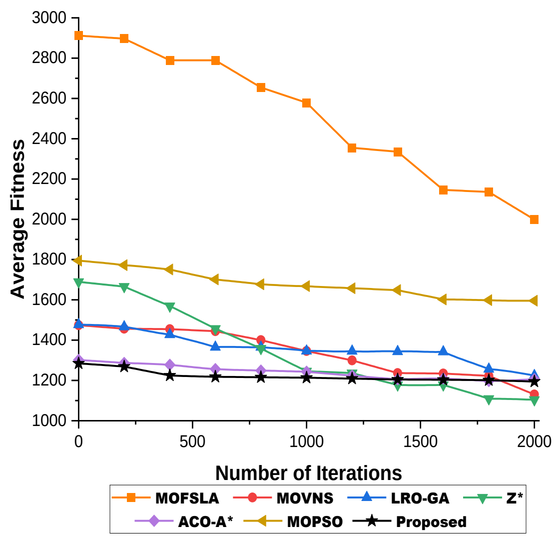


(d)

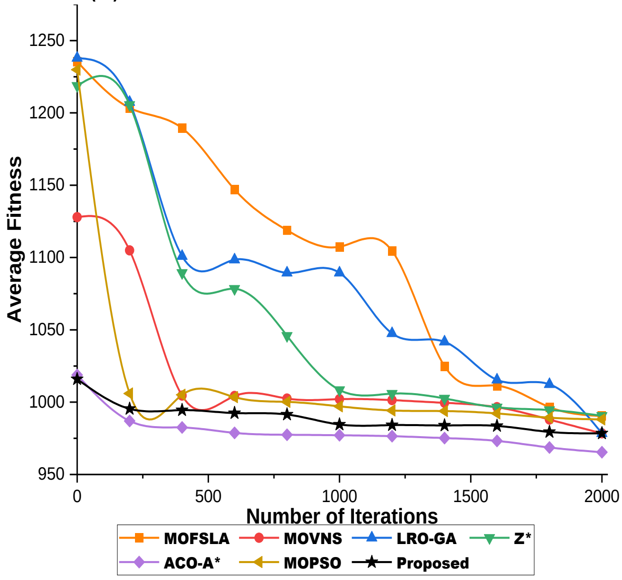


(e)

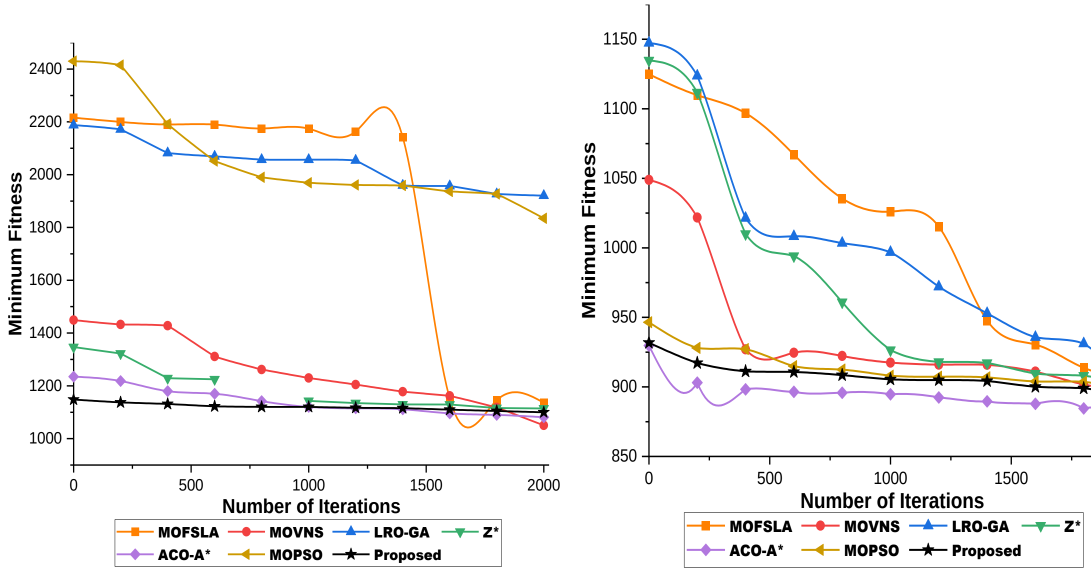
Fig. 9. Box plot analysis of case 1 to case 5 (a-e) for different algorithms: (1) GGACO (proposed) (2) Z* (3) MOSFLA (4) ACO*A* (5) MOVNS



(a)



(b)



(c) **(d)**

Fig. 10. Convergence analysis (a and b) Average Fitness Vs. Number of iterations (c and d) Minimum Fitness Vs. Number of Iterations

5. Conclusion

This paper proposes the GGACO method, a novel extension to ACO that can effectively plan paths on uneven terrains (3D) in remote sensing images. The proposed method determines an efficient path in the *least time with minimum energy consumption* (refer to Table 2). A new gain-based, pheromone-enhancing mechanism for AS has been proposed for the updating pheromone to *enable faster convergence* (Figure 10). Pheromone enhancement enables ACS to channel the ants towards the optimal path at the exploitation stage. This helps to reduce energy consumption by avoiding unwanted traversals during ant activities. GGACO is compared with state-of-the-art methods, viz., Z*, MOSFLA, MOPSO, ACO-A*, MOVNS, and LRO-GA; results have been analyzed for performance measures like path length, time and energy cost (refer to Table 2). Additionally, the proposed method’s convergence analysis and stability have been investigated (refer to Table 2, Figs. 10, 10), and GGACO is found to be *stable and energy efficient*. It is inferred from the experimental investigation that GGACO resolves the drawbacks mentioned in section IV-D.

The *time complexity* of GGACO is found to be $O(NMk)\ for\ all\ iterations$ where N is the number of ants, M is the input size (m x n) , k is the dimension. Here since the dimension is 3, k is 3. GGACO requires $\mathrm{O}(ph^2)$ number of operations to update the pheromone at all traversals. Constructing a complete solution requires $\mathrm{O}(N * |V_f|)$ space. Thus, the *space complexity* of GGACO will be $\mathrm{O}(N * |V_f|) + \mathrm{O}(ph^2)$.

Some of the societal perspectives that could be addressed by GGACO are:

Urban Traffic Planning: With the increase in autonomous driving technologies, intelligent path planning strategies are needed in urban traffic scenarios. Efficient and robust path planning strategies will improve the safety and efficiency of path planning.

Emergency Scenarios: The level of uncertainty is huge in emergency scenarios. An efficient path planning system will greatly support humanitarian systems by offering reliable and accountable assistance.

Battlefield Environment: A path planned on a battlefield must satisfy criteria such as requiring low fuel consumption, providing safety, and being the least time-consuming option. An efficient path planning system will be of great assistance to the main force on a battlefield.

As a part of its future scope, the proposed framework can be extended by creating an AI system to help with the autonomous navigation system. The convergence rate can still be increased in GGACO by analyzing different ant models. More parallelized versions of the algorithm can also be developed to handle transportation problems.

Acknowledgement
This research was not funded by any grant

Conflicts of Interest
The authors declare no conflicts of interest.

References
[1] Hota, S.; Ghose, D. Optimal path planning for an aerial vehicle in 3D space. 49th IEEE Conference on Decision and Control (CDC) 2010, Atlanta, GA, pp. 4902-4907.
[2] Qi, Z,; Shao ,Z.; Ping, Y .S.; Hiot,L .M.; Leong, Y.K. An Improved Heuristic Algorithm for UAV Path Planning in 3D Environment. Second International Conference on Intelligent Human-Machine Systems and Cybernetics 2010 , Nanjing, Jiangsu, pp. 258-261.
[3] Yang,K.; Sukkarieh, S.; 3D smooth path planning for a UAV in cluttered natural environments. IEEE/RSJ International Conference on Intelligent Robots and Systems 2008, IROS 2008, IEEE, pp. 794–800.
[4] Aybars, U. Ğ. U. R. Path planning on a cuboid using genetic algorithms. Information Sciences 2008, 178.16, pp: 3275-3287.
[5] Montiel.; Oscar.; Ulises Orozco-Rosas.; Roberto Sepúlveda. Path planning for mobile robots using Bacterial Potential Field for avoiding static and dynamic obstacles. Expert Systems with Applications 2015, 42.12, pp: 5177-5191.
[6] BoussaïD, Ilhem, Julien Lepagnot, and Patrick Siarry. "A survey on optimization metaheuristics." Information sciences237 (2013): 82-117.
[7] Fankhauser.; Péter.; Michael Bloesch.; Marco Hutter. Probabilistic terrain mapping for mobile robots with uncertain localization. IEEE Robotics and Automation Letters 2018, 3.4,pp:3019-3026.
[8] Yu,H. W.; Lee, B.H. MRF-based terrain map inference using variational feature projection. Electronics Letters 2018, vol. 54, pp. 595-597.
[9] Roberge.; Vincent.; Mohammed Tarbouchi.;Gilles Labonté. Fast Genetic Algorithm Path Planner for Fixed-Wing Military UAV Using GPU. IEEE Transactions on Aerospace and Electronic Systems 2018.
[10] Saputra, A.A.; Toda, A.;Botzheim, J.;Kubota, N. Neuro-Activity-Based Dynamic Path Planner for 3-D Rough Terrain. IEEE Transactions on Cognitive and Developmental Systems 2018, vol. 10, pp. 138-150.
[11] Delmerico, J.; E. Mueggler.; Nitsch, J.; Scaramuzza, D. Active Autonomous Aerial Exploration for Ground Robot Path Planning. IEEE Robotics and Automation Letters 2017, vol. 2, pp. 664-671.
[12] Pütz, S.; Wiemann, T.; Sprickerhof, J.; Hertzberg, J. 3d navigation mesh generation for path planning in uneven terrain. IFAC-PapersOnLine 2016, 49(15), 212-217.
[13] Yang, Cheng-Hsuan, et al. "UAV path planning method for digital terrain model reconstruction–A debris fan example." Automation in Construction 2018,93, pp: 214-230.
[14] Han, J. An efficient approach to 3D path planning. Information Sciences 2019, 478, 318-330.
[15] X. Yu, W. Chen, T. Gu, H. Yuan, H. Zhang and J. Zhang, "ACO-A*: Ant Colony Optimization Plus A* for 3D Traveling in Environments with Dense Obstacles," in IEEE Transactions on Evolutionary Computation.
[16] N. Ganganath, C. Cheng and C. K. Tse, "A Constraint-Aware Heuristic Path Planner for Finding Energy-Efficient Paths on Uneven Terrains," in IEEE Transactions on Industrial Informatics, vol. 11, no. 3, pp. 601-611, June 2015.
[17] Wang, Baofang, et al. "Car-like mobile robot path planning in rough terrain using multi-objective particle swarm optimization algorithm." Neurocomputing 282 (2018): 42-a
[18] Ganganath, N.; Cheng, C.; Ts, C, K. A Constraint-Aware Heuristic Path Planner for Finding Energy-Efficient Paths on Uneven Terrains. IEEE Transactions on Industrial Informatics 2015, vol. 11, pp. 601-611.
[19] Wang, B.; Li, S.; Guo, J.; Chen, Q. Car-like mobile robot path planning in rough terrain using multi-objective particle swarm optimization algorithm. Neurocomputing 2018, 282, 42-51.
[20] Huang, Han, Chun-Guo Wu, and Zhi-Feng Hao. "A pheromone-rate-based analysis on the convergence time of ACO algorithm." IEEE Transactions on Systems, Man, and Cybernetics, Part B (Cybernetics) 39, no. 4 (2009): 910-923.
[21] Sadrpour, A.; Jin, J.; Ulsoy, A.G. ``Mission energy prediction for unmanned ground vehicles using real-time measurements and prior knowledge,'' J. Field Robot. 2013, vol. 30, no. 3, pp. 399_414.

[22] Dorigo, Marco, Vittorio Maniezzo, Alberto Colorni. "Ant system: optimization by a colony of cooperating agents." IEEE Transactions on Systems, man, and cybernetics, Part B: Cybernetics 26, no. 1 (1996): 29-41.
[23] Padhy, N. P. Artificial intelligence and intelligent systems. Oxford University Press, 2005.
[24] Viswanathan, S., Ravichandran, K., Tapas, A., & Shekhar, S. (2019). An Intelligent Gain based Ant Colony Optimisation Method for Path Planning of Unmanned Ground Vehicles. Defence Science Journal, 69(2), 167-172. https://doi.org/10.14429/dsj.69.12509
[25] P. Zhao, J. Chen, T. Mei, and H. Liang, ‘‘Dynamic motion planning for autonomous vehicle in unknown environments,’’ in Proc. IEEE Intell. Vehicles Symp. (IV), Jun. 2011, pp. 284–289
[26] Ravankar, A., Ravankar, A. A., Kobayashi, Y., Hoshino, Y., & Peng, C. C. (2018). Path smoothing techniques in robot navigation: State-of-the-art, current and future challenges. Sensors, 18(9), 3170.
[27] http://www2.isprs.org/commissions/comm3/wg4/tests.html, data retrieved 26/08/2016.
[28] Jabbarpour, M, R.; Zarrab, H.; Jung, J.J.; Kim, P. A green ant-based method for path planning of unmanned ground vehicles. IEEE Access 2015, vol. 5, pp. 1820–1832.
[29] J. Li, G. Deng, C. Luo, Q. Lin, Q. Yan and Z. Ming, "A Hybrid Path Planning Method in Unmanned Air/Ground Vehicle (UAV/UGV) Cooperative Systems," in IEEE Transactions on Vehicular Technology, vol. 65, no. 12, pp. 9585-9596, Dec. 2016.
[30] Hidalgo-Paniagua, A., Vega-Rodríguez, M. A., Ferruz, J., & Pavón, N. (2015). MOSFLA-MRPP: multi-objective shuffled frog-leaping algorithm applied to mobile robot path planning. Engineering Applications of Artificial Intelligence, 44, 123-136.
[31] Hidalgo-Paniagua, A., Vega-Rodríguez, M. A., & Ferruz, J. (2016). Applying the MOVNS (multi-objective variable neighborhood search) algorithm to solve the path planning problem in mobile robotics. Expert Systems with Applications, 58, 20-35.