

## Article

# Autonomous Terrestrial Image Segmentation and Sensor Node Localization for Disaster Management using Plant Growth Simulation Algorithm

Deblina Bhattacharjee <sup>1</sup>, Anand Paul <sup>1,\*</sup>, Won-Hwa Hong <sup>2</sup>, HyunCheol Seo <sup>2</sup> and S. Karthik <sup>3</sup>

<sup>1</sup> Department of Computer Science and Engineering, Kyungpook National University, Daegu, South Korea 702701; deblina021091@gmail.com

<sup>2</sup> School of Architectural, Civil, Environmental and Energy Engineering, Kyungpook National University, Daegu, South Korea; hongwh@knu.ac.kr (W.-H.H.); notsools@gmail.com (H.C.S.)

<sup>3</sup> Department of Computer Science and Engineering, SNS College of Technology, Coimbatore, India; profskarthik@gmail.com

\* Correspondence: paul.editor@gmail.com; Tel.: +82-(0)10-7930-9696

**Abstract:** The use of unmanned aerial vehicle (UAV) during emergency response of a disaster has been widespread in recent years and the terrain images captured by the cameras on board these vehicles are significant sources of information for such disaster monitoring operations. Thus, analyzing such images are important for assessing the terrain of interest during such emergency response operations. Further, these UAVs are mainly used in disaster monitoring systems for the automated deployment of sensor nodes in real time. Therefore, deploying and localizing the wireless sensor nodes optimally, only in the regions of interest that are identified by segmenting the images captured by UAVs, hold paramount significance thereby effecting their performance. In this paper, the highly effective nature-inspired Plant Growth Simulation Algorithm (PGSA) has been applied for the segmentation of such terrestrial images and also for the localization of the deployed sensor nodes. The problem is formulated as a multi-dimensional optimization problem and PGSA has been used to solve it. Furthermore, the proposed method has been compared to other existing evolutionary methods and simulation results show that PGSA gives better performance with respect to both speed and accuracy unlike other techniques in literature.

**Keywords:** image segmentation; wireless sensor node deployment; plant growth simulation algorithm; disaster management.

## 1. Introduction

Wireless sensor networks (WSNs) are effective in disaster monitoring and management systems as they can sense environmental as well as physical conditions being a network of distributed autonomous nodes. The widespread use of sensor nodes deployed by UAVs (like drones) flying over disaster-prone or post disaster sites has brought the problem of their optimal and automated deployment as well as localization into limelight. Needless to say, it is an area of active research and recently, bio-inspired algorithms have gained a strong foothold in solving this problem giving a new direction to the ongoing research works and showing promising results [1, 2]. As the localization of the nodes directly effects their performance, it becomes imperative that the nodes are deployed only in the terrains of importance. These terrains/regions of interests can be achieved by analysing the terrestrial images captured by the UAVs. Modern UAVs have the capabilities to control and perceive terrestrial information that aids in the coordinated deployment of the sensor nodes. Using motion and position estimation, detection and monitoring, tracking and terrain mapping, the UAVs can provide useful images of the terrain which when segmented, based on optimal thresholds, can output regions of interest for the autonomous deployment of the sensor nodes. Therefore, the wireless sensor nodes can be prevented from being dropped by such UAVs into useless terrains like water or an area with fire. The uses of this are threefold. First, it reduces the unnecessary deployment of sensor nodes

into such areas maintaining the cost. Second, it reduces the loss of the nodes that are dropped in unimportant areas thereby ensuring that the sensing performance and coverage can be achieved with minimal nodes that are deployed where necessary. Third, the problem of traffic congestion and delay due to redundant traffic from nodes from unimportant regions is solved.

While processing the image to analyse the relevant terrains, the images should be pre-processed and this is done with segmentation. For segmentation, the commonly used method is that of thresholding, which segments the image into two or more classes. Generally, a 2-level thresholding process is to find a threshold  $t$  such that an operation as shown in equation (1) for an image  $im(x, y)$  can be performed, where  $im(x, y)$  is a grayscale image of size  $l \times b$  having  $L$  pixel intensity levels and  $x=1,2,\dots,l$  and  $y=1,2,\dots,b$ .

$$I(x,y) = \begin{cases} 0, & \text{if } im(x,y) \leq t \\ L, & \text{if } im(x,y) > t \end{cases} \quad (1)$$

Therefore, this thresholding scheme can be generalized to an  $n$ -level thresholding where  $n-1$  threshold levels,  $t_1, t_2, \dots, t_{n-1}$  are used. For the intermediate threshold levels, the average of the boundary thresholds are found i.e.

$$I(x,y) = \begin{cases} 0, & \text{if } im(x,y) \leq t \\ \frac{1}{2}(t_1 + t_2), & \text{if } t_1 < im(x,y) \leq t_2 \\ L, & \text{if } im(x,y) > t \end{cases} \quad (2)$$

Thus, the multi-level segmentation is largely dependent on its threshold levels and finding these levels is the problem to be addressed. There have been many methods including Otsu method [3], and many other shown in [4]. To find such threshold levels to segment an image, the popular Otsu method evaluates an objective function based on the variance of pixel distribution in every class and the best feasible threshold solution among all the candidate thresholds is reported for segmentation, making the problem a real time optimization problem. In this paper, we solve this optimization problem by employing PGSA to find the best threshold corresponding to the best objective function value.

Wireless sensor nodes are mainly employed for detecting and tracking events and for that location information becomes quite significant. For this, it is critical that the node's location is traced beforehand. Many location algorithms till date have used a priori location estimate of reference nodes known as beacons or anchors [5]. However, as the nodes are deployed into unknown territory, the priori knowledge about location is unavailable. Hence, estimating the space co-ordinates of the sensor nodes is required which is the task of localization. Although, each node can be accompanied by GPS for deciphering their coordinates, this solution is unfeasible because of cost incurred, size constraints and energy requirements for maintenance. A WSN can comprise  $N$  nodes, having a communication range of  $R$  units, distributed in a 2-D site field. An overview of the various components of the localization systems for sensor nodes in a WSN is presented in [6]. The WSN is represented by the Euclidean graph  $G = (V, E)$ , where  $V = \{v_1, v_2, \dots, v_m\}$  is the set of sensor nodes and  $E$  is the edge such that  $\{i, j\} \in E$  if the distance between  $v_i$  and  $v_j$  is  $d_{ij} \leq R$ . Also,  $U$  represents the set of unknown nodes which have to be localized based on the location of the reference beacons. The set  $A$  represents the anchor nodes which has prior knowledge about its location. The localization problem is defined as transforming the unknown nodes into anchor nodes, by iteratively determining the coordinates  $(x_u, y_u)$  of as many  $u \in U$  as possible by referring to the set of anchor nodes  $A$  and their positions  $(x_a, y_a)$ , for all  $a \in A$ . The many parameters based on which localization can be solved include inter-node angle based on range, distance estimation methods of arrival time difference, angle of arrival, time of arrival, received signal strength and range-free techniques [7]. Sensor node localization includes determining the distance between the anchor nodes and the unknown nodes using the received signal strength and based on this estimated distance the position of unknown nodes is derived. This is done

by iteratively refining the node positions using range information and positions of neighbouring anchor nodes. This can in turn be achieved by applying an optimization algorithm like PGSA, used in this paper, which increases the accuracy of localization. The motivation behind using PGSA is that it has a very well balanced exploration to exploitation ratio [8], does not depend on parameter setting, has no restrictions on the target function [9] and has been seen to outperform other state-of-the-art algorithms to solve the following two main problems addressed in this paper. This is further explained and supported in the upcoming sections.

The highlights of this paper are as follows:

- 1) PGSA is used for segmentation of the terrestrial images captured by a UAV which has been used for the automated deployment of the wireless sensor nodes.
- 2) PGSA is used to solve the distributed localization problem of wireless sensor nodes and finally a comparative performance analysis is presented with respect to speed and accuracy.

The results show that PGSA is able to achieve localization accuracy of wireless sensor nodes in minimal computing time, thereby giving a fast and effective method of optimal localization of wireless sensor nodes in disaster management. The rest of the paper presents related work in Section 2, outlines the PGSA and states the proposed scheme for terrestrial image segmentation and wireless sensor node localization in Section 3. The results are presented followed by a comparative discussion in section 4 and the paper concludes in Section 5.

## 2. Background

### 2.1. Related Work on Terrestrial Image Segmentation

It has already been discussed that in order to improve the performance of wireless sensor nodes, they should be deployed in effected regions for which image segmentation of the remotely captured image is necessary. In literature, there have been many suggested methods for the classification of such geographic information system (GIS) images into different pixel classes [10, 11] using machine learning classifiers like maximum likelihood [12], support vector machines [13, 14] fuzzy inference rule with support vector machines [1] etc. However, due to increase in the spatiotemporal dimensions for such data such algorithms require longer training time for classification and as in the case of maximum likelihood classifiers give lower precision and higher noise susceptibility [15]. To overcome this problem, metaheuristic methods inspired from evolutionary processes such as, artificial neural networks (ANN) [16], genetic algorithms (GA) [17, 18], artificial immune system (AIS) [19-21], particle swarm optimization (PSO) and bacteria foraging algorithm (BFA) [22] have been recently used to solve this image segmentation problem. Metaheuristic methods are highly robust with a self-learning ability when compared to traditional classifiers. The advantages of such biomimicry metaheuristic methods are their self-adaptivity to the data without predefined functional or distributional specifications for the model on which they are based, their real world applicability and flexibility being non-linear models, and their ability to approximate functions with some accuracy [23-25]. In this paper, we have used a highly effective metaheuristic technique for analysis of UAV captured images. To obtain the regions of interest, the captured image is segmented based on various threshold levels which is found using PGSA in this paper. The use of PGSA over other methods to find the optimal threshold levels for segmenting the image can be justified which has been explained in detail in Section 3.1.

### 2.1. Related Work on Wireless Sensor Node Localization

Similarly, PGSA has been applied to find the optimal locations of wireless sensor nodes thereby solving the localization problem in this paper. Extensive work has been done in the field of wireless sensor node localization. Surveys on node localization are presented in [26] with overview of localization systems in [27]. The various measurement techniques in sensor network localization is presented in [28]. Further, localization techniques inspired from GPS based methods are applied to unknown wireless sensor nodes in GPS free mode [29, 30]. In [6], issues in sensor localization are formulated as metaheuristics or multi-dimensional optimization problems, and is solved using PSO.

The localization problem can be treated with a centralized approach where every node relays its parameters to a central managing node which then computes the global solution. A centralized localization method using simulated annealing and GA is presented in [31]. However, a centralized approach has low scalability and high relay cost as the nodes need to relay information to the central node. Therefore, distributed solutions are more beneficial for networks requiring high scalability. Bio-inspired stochastic optimization methods like GA, PSO [32], bat algorithm [33], gravitational search algorithm [34] and simulated annealing [35] have also been used for localization problems to overcome the computational complexity and resource requirements of conventional methods [36]. However, in all the above methods the localization solution is optimal with respect to either the speed of finding the solutions or the accuracy of the solutions but not both. Specifically, simulated annealing has a slow global search speed in the initial iterations, GA tends towards premature convergence and all these heuristic algorithms are parameter critical and heavily rely on the setting values. Thus, the selection of proper parameters becomes essential. However, there is no specific standard for their selection and are mostly dependent on conditions of the problem domain. Hence, the parameters selected strongly influence the optimal solutions. On the other hand, PGSA does not depend on parameter setting and has no restrictions on the target function [9]. Moreover, PGSA has a well-balanced exploration to exploitation ratio [8] and hence is able to give high quality solution which is both accurate and achieved in minimal time.

### 3. Materials and Methods: Proposed Scheme

#### 3.1. Plant Growth Simulation Algorithm (PGSA)

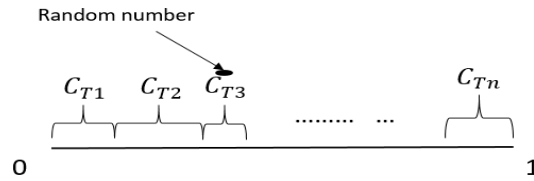
The PGSA is a nature-inspired simple bionic random algorithm that functions on the basis of plant phototropism which is defined as the ability of a plant to search for the light source and bend towards it. It was first introduced by Li Tong [37] and regards the feasible region of integer programming as the plant growth environment. This algorithm simulates the growth process of the plant where the search starts from the initial solution which is the seed and iteratively proceeds by assessing the concentration of the plant growth hormone, morphactin on all the growth nodes until the entire growth space is covered [38]. Here, the global optimal solution is the source of light and the algorithm proceeds by calculating the fitness function defined on the basis of morphactin concentrations on the possible growth points of the plant. This morphactin concentration which decides the growth process is in turn dependent on the light intensity. PGSA emphasises on the decision making process of a plant which is based on the biological growth rules and probability models. According to the biological laws for plant growth, 1) the node on the plant with a higher morphactin concentration has a greater probability to grow into a branch 2) the morphactin concentrations on these nodes vary according to the environmental information and the relative positions of these nodes on the plant. If a node has the highest morphactin concentration it is saved as the best solution and will branch out. Also, the morphactin concentrations of all the remaining plant nodes will be recalculated and reallocated as per the new environment while the just branched node will be assigned a concentration equal to zero.

##### 3.1.1. Mathematical Model for Plant Growth

PGSA has an objective function  $f(i)$  for each node  $i$  that is evaluated according to the probability model of plant growth in the given environment. According to the biological mechanism of phototropism,  $f(i)$  should be minimized for the node  $i$  to have a better growth environment to branch out. According to the mathematical model of PGSA, let there be a root  $R_0$  such that a trunk  $T$  stems out from the root. The morphactin concentrations of the nodes present on the trunk is calculated according to equation (3) where it is assumed that there are  $n$  nodes on the trunk having a better environment than the root which is the initial solution. Thus, if  $f(R_{Ti}) < f(R_0)$  ( $i = 1, 2, \dots, n$ ), the morphactin concentrations  $C_{Ti}$  is calculated as follows:

$$C_{Ti} = \frac{f(R_0) - f(R_{Ti})}{\sum_{i=1}^n (f(R_0) - f(R_{Ti}))} \quad (i=1, 2, \dots, n) \quad (3)$$

Per equation (3), the growth environment of all the nodes on the plant contribute towards the change in morphactin concentrations of a particular node in a plant. Hence, if the concentration in one node changes, it effects the remaining nodes. Given the set of concentrations of all the nodes, it can be seen that  $\sum C_{Ti}=1$ , as it is based on probability. Hence, the state space of the concentrations has a range of  $[0, 1]$  from which a number is randomly selected that is associated with a concentration  $C_{TM}$  shown in Figure 1.



**Figure 1.** Morphactin concentration space

Let  $C_{TM}$  be the concentration of the node  $R_{TM}$  which now takes preference over other nodes to branch out in the next iteration. However, the node  $R_{TM}$  will grow into a branch only if the random number  $\beta$  satisfies the following two conditions outlined in equations (4) and (5).

$$0 \leq \beta \leq \sum_{i=1}^M C_{Ti} \quad (M = 1) \quad (4)$$

$$\sum_{i=1}^{M-1} C_{Ti} < \beta \leq \sum_{i=1}^M C_{Ti} \quad (M = 2, 3, \dots, n) \quad (5)$$

After the node grows into a branch, the morphactin concentration of all the remaining non-preferential nodes are recalculated as seen in equations (6) and (7), while  $R_{TM}$  has a concentration set to zero.

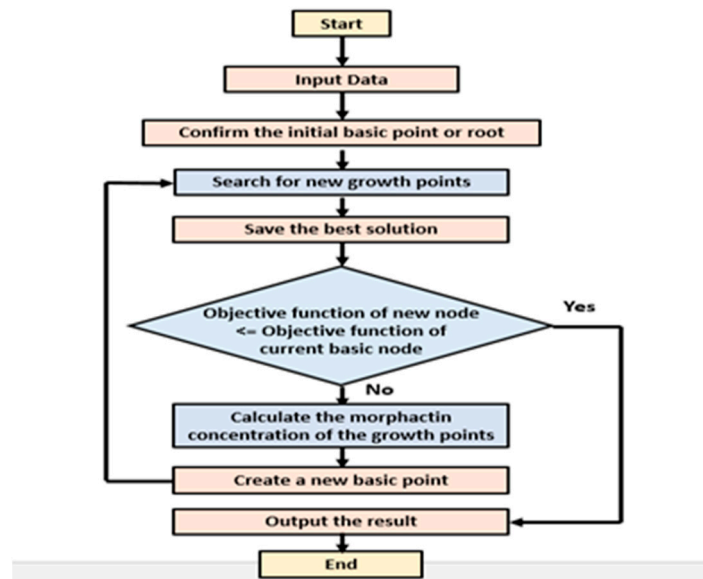
$$C_{Ti} = \frac{f(R_0) - f(R_{Ti})}{\sum_{i=1, i \neq M}^n (f(R_0) - f(R_{Ti})) + \sum_{j=1}^p (f(R_0) - f(R_{bj}))} \quad (6)$$

$$C_{bj} = \frac{f(R_0) - f(R_{bj})}{\sum_{i=1, i \neq M}^n (f(R_0) - f(R_{Ti})) + \sum_{j=1}^p (f(R_0) - f(R_{bj}))} \quad (7)$$

Now when the reallocation of the fresh concentrations are done for all nodes except the previous preferential node, the state space of concentrations is again formed  $\in [0, 1]$ . If the newly grown branch  $b$  has  $p$  nodes, such that  $f(R_{bi}) < f(R_0)$  ( $i = 1, 2, \dots, p$ ), again a number  $\beta$  is randomly chosen from the state space and the process continues. This growth process stops in the bionic world when the plant has reached its maturity and cannot further branch out. In the simulation world, PGSA stops when the same solution is repeated for a set number of times. The PGSA is a highly effective solution to optimization problems. The parameters of PGSA are as follows:  $f(i)$  is the objective function,  $R_0$  is the initial solution or root, length of the trunk and branches is the search space of candidate solutions, the nodes on the plants are the feasible solutions. And the maximum number of times for which the best solution can be repeated is set to 5.

A comment on the exploration to exploitation ratio of PGSA can be found in [8, 39] where it has been seen that PGSA keeps exploring the entire search interval in  $[0, 1]$  which remains constant. Moreover, the morphactin reallocation is done to the nodes on the new branch by exploitation and to the previous nodes on the trunk by exploration in a single iteration. This is mainly because the objective function (growth environment) is dependent on the concentration of all the nodes on the plant. Thus PGSA has been applied to solve the optimization problems in this paper. Figure 2 shows the overall working of PGSA.





**Figure 2.** Overview of the plant growth simulation algorithm

### 3.1. PGSA Based Image Thresholding

Till date there have quite a few image segmenting techniques which can be classified into the following categories per the type of information they process on the images. In [4], a review of the different thresholding algorithms along with their performance measures is presented which can be categorised as histogram thresholding, spatial segmentation, segmentation based on entropy thresholding, clustering, object oriented methods using attributes and local segmentation techniques.

The PGSA method that has been proposed in this paper can be categorized as a new form of histogram thresholding. Suppose there's an image of pixel size  $l \times b$  pixels with  $L$  intensity levels represented as  $L$  gray levels  $[1, 2, \dots, L]$ . If  $p_i$  be the number of pixels having intensity level  $i$  then the total number of pixels satisfied by  $P$  is given as  $P = l \times b = p_1 + p_2 + \dots + p_L$ . This is based on the unsupervised classification technique for thresholding by Otsu. It is obvious that  $p_i$  represents the image histogram. The histogram is thereafter normalized to form a probability distribution function (pdf) as shown in equation (8).

$$pdf_i = \frac{p_i}{N}, \quad pdf_i > 0 \quad \text{and} \quad \sum_{i=1}^L pdf_i = 1 \quad (8)$$

To classify the pixels into two classes  $M_1$  and  $M_2$  representing the background pixels and object pixels with a threshold level  $t$  we follow equation (1) where  $M_1$  contains the pixels with intensities  $\leq t$  and the pixels having an intensity level higher than the threshold  $t$  are classified into class  $M_2$ . Equations (9) and (10) give the probabilities of occurrences of classes  $M_1$  and  $M_2$  and their respective averages are shown in equations (11) and (12).

$$\omega_1(t) = \text{Probability}(M_1) = \sum_{i=1}^t pdf_i \quad (9)$$

$$\omega_2(t) = \text{Probability}(M_2) = \sum_{i=t+1}^L pdf_i \quad (10)$$

$$\mu_1(t) = \sum_{i=1}^t \frac{i * pdf_i}{\omega_1} \quad (11)$$

$$\mu_2(t) = \sum_{i=t+1}^L \frac{i * pdf_i}{\omega_2} \quad (12)$$

The above equations are subjected to the conditions

$\omega_1(t) + \omega_2(t) = 1$ ;  $\omega_1 \mu_1 + \omega_2 \mu_2 = \mu_L = \sum_{i=1}^L i * pdf_i$ . Note  $\mu(L)$  is the average for the total intensity levels in the image. As determining the image threshold levels requires the knowledge of the variance of the pixel distribution in the two classes of pixels, we now calculate the pixel variances in the respective classes, as seen in equation (13) and (14).

$$\sigma_1^2(t) = \sum_{i=1}^t (i - \mu_1(t))^2 \frac{pdf_i}{\omega_1} \quad (13)$$

$$\sigma_2^2(t) = \sum_{i=t+1}^L (i - \mu_2(t))^2 \frac{pdf_i}{\omega_2} \quad (14)$$

The next step is to formulate the objective function for finding optimal threshold levels, Otsu mentioned the following objectives.

$$\alpha = \frac{\sigma_{inter}^2}{\sigma_{intra}^2}, \beta = \frac{\sigma_{total}^2}{\sigma_{intra}^2} \text{ and } \gamma = \frac{\sigma_{inter}^2}{\sigma_{total}^2} \quad (15)$$

Here the variances are defined as intraclass variance ( $\sigma_{intra}^2$ ), interclass variance ( $\sigma_{inter}^2$ ) and the total variance of pixel intensities ( $\sigma_{total}^2$ ). The definition of the variances are further given as follows:

$$\sigma_{intra}^2 = \omega_1 \sigma_1^2 + \omega_2 \sigma_2^2 \quad (16)$$

$$\sigma_{inter}^2 = \omega_1 \omega_2 + (\mu_2 - \mu_1)^2 \quad (17)$$

$$\sigma_{total}^2 = \sum_{i=1}^L (i - \mu_L)^2 pdf_i \quad (18)$$

The task thus reduces to find the optimal threshold levels such that either  $\alpha$ ,  $\beta$  or  $\gamma$  maximizes. Note that the optimization problem that has been defined, maximizes the interclass variance while minimizing the intraclass variance. The optimization problem when extended to an  $n$ -level thresholding scheme can be assumed as a multi-dimensional optimization problem with  $n-1$  candidate thresholds which has been solved using PGSA here. The proposed PGSA applied to finding the threshold levels is as follows.

- 1) The PGSA starts by taking the initial root which is just a starting pixel intensity level out of  $L$  pixel intensity levels and can be anywhere on the image. Note that the overall difference in the results due to the selection of a good initial candidate is not visible due to the randomization involved in such bionic algorithms.
- 2) The set of possible  $n-1$  threshold solutions are analogous to the  $n-1$  growth nodes on the trunk  $T$  with each node having a morphactin concentration as  $C_{Ti}$  where  $i = (1, 2, \dots, n-1)$ , forming the state space  $[0, 1]$  as seen in Figure 1. Let the candidate threshold solutions be  $t_i$  where  $i = (1, 2, \dots, n-1)$  with the ranges of each threshold being,  $1 \leq t_1 < L - n + 1$ ,  $t_1 + 1 \leq t_2 < L - n + 2$ , and  $t_{n-2} + 1 \leq t_{n-1} < L - 1$ .
- 3) The objective for PGSA is to maximize the interclass variance. Thus, the objective function given as  $\sigma_{inter}^2(\hat{t}_1, \hat{t}_2, \dots, \hat{t}_{n-1}) = \max_{1 \leq t_1 < t_2 < \dots < t_{n-1} < L} \sigma_{inter}^2(t_1, t_2, \dots, t_{n-1})$  is evaluated to find the best candidate node like  $R_{TM}$  mentioned previously in the PGSA discussion.
- 4) As per the PGSA discussed, the morphactin concentration of every growth node is checked as per equation (3). Higher the concentration of the threshold level, higher is its probability to proceed as the best feasible solution in next iteration.
- 5) The best solution from the previous iteration ( $R_{TM}$ ) branches out after storing its value and the morphactin concentrations of all the remaining nodes are reallocated as per equation (6) and (7) except for  $R_{TM}$  as it is the current local optimal solution whose concentration is assigned 0.
- 6) For every new best solution (preferential node), a maximum number of  $p-1$  thresholds are generated analogous to  $p-1$  nodes on the branch  $b$  and based on the newly calculated morphactin concentration in Step 5, the new best candidate is found on the branch generated in the previous step. This process continues up to a maximum of 20 iterations.
- 7) The set of all nodes that have branched out are the possible candidate solution with the final nodes being the global best threshold solutions and others the local optimal solutions.
- 8) The optimal threshold levels determined by PGSA (assumed to be integers else rounded off to the nearest integer) is used for segmenting the image and this result is used by the UAV to deploy the sensor nodes.

### 3.1. PGSA Based Localization

The main objective of the localization algorithm is to estimate the position of  $U$  unknown nodes, given there is a set of  $A$  anchor nodes whose position is assumed to be known as they are the reference nodes which keep broadcasting a beacon message containing their coordinates. The node localization problem, thus becomes an optimization problem where the optimal positions of the unknown nodes (assumed to be deployed by a UAV in an unknown terrain) is to be estimated. The node localization problem has the following domain parameters. There are maximum of  $U$  unsettled/unknown nodes and  $A$  anchor/beacon nodes. The beacon nodes have their position coordinates stored in them and hence, are used as reference. Moreover, the nodes whose positions are found at the end of each iteration become new reference/ anchor nodes for the next iteration where they too broadcast the beacon message. The range of transmission for all the nodes (unknown and anchor) is  $R$ . The approach of sensor localization is as follows.

1) The unsettled nodes check if there are a minimum of three non-collinear anchor nodes within their transmission range as only then they can be localizable. Upon finding such reference nodes, the unsettled node computes the distance between the respective pair of  $\{U, A\}$  nodes using the simple Euclidean distance formula in equation (19).

$$d_i = \sqrt{(x - x_i)^2 + (y - y_i)^2} \quad (19)$$

2) However, the estimated distance by the localizable unsettled node  $u \in U$  from a surrounding anchor  $a_i \in A$  is not equal to the actual distance  $d_i$  shown above, as there is noise present in such measurements. Hence, a Gaussian noise  $\phi_i$  (being the most common noise in such scenario) is added to  $d_i$  to find  $d'_i = d_i + \phi_i$ , where  $d'_i$  is the distance estimated by the unsettled node from the surrounding anchor nodes.

3) If  $(x, y)$  is the coordinate of the unsettled node that requires localization and  $(x_i, y_i)$  is the coordinate of the  $i^{th}$  beacon surrounding the unsettled node  $u$ . The Gaussian noise  $\phi_i$  is randomly selected from the range  $d_i \pm d_i(E_n/100)$  where  $E_n$  is the error introduced. Thus, the noise directly impacts the distance calculation and hence the localization result.

4) Thus, the objective function can be formulated for the localization problem using the error parameter that should be minimized for effective position estimation. The problem hence, is assumed as an optimization problem where the cost function based on the error parameter is as follows which has to be minimized.

$$f(x, y) = \frac{1}{A} \sum_{i=1}^A (\sqrt{(x - x_i)^2 + (y - y_i)^2} - d'_i)^2 \quad (20)$$

Where  $A$  is the number of anchors for the nodes to be localized (having transmission range  $R$ ), given that there are a minimum of 3 non-collinear anchors within  $R$ . Thus, it is a two-dimensional position optimization problem because the error needs to be minimized.

5) PGSA searches for the best optimal coordinates  $(x, y)$  in the two-dimensional search space for the unknown nodes thereby localizing it. For this, a set of all coordinate positions within the range of minimum 3 non-collinear neighbouring anchors are taken as initial candidate solutions for the position of the target node, that behave as growth points on the plant in PGSA. Thereafter, the morphactin concentrations are assigned to these growth points forming a concentration state space as seen in Figure 1. The methodology for checking each growth point on the plant for its morphactin concentration is similar to the PGSA model discussed earlier. Higher the morphactin concentration associated with a coordinate position (behaving like a plant growth point), greater is its branching probability. As the morphactin concentration in a plant node depends on its environment that in turn depends on the objective function for localization which has to be minimized, all the candidate solutions are evaluated based on their cost function value. The candidate with the minimum cost is the best solution and becomes the preferential coordinate position and branches out after storing its



position value and the morphactin concentrations of all the remaining nodes are reallocated as per equation (6) and (7) except for preferential node as it is the current local optimal solution whose concentration is assigned to 0. The best candidate solution in the previous iteration becomes one of the anchor nodes for the next iteration as its coordinates have now been estimated. Now, the nearby search space is exploited.

6) After all  $\tilde{U}$  localizable nodes have been localized, the total error in localization is computed as the mean absolute error given by the average of the square of the distances between actual nodes coordinates  $(x_i, y_i)$  and the coordinates returned by running the PGSA  $(\tilde{x}_i, \tilde{y}_i)$  shown in equation (21).

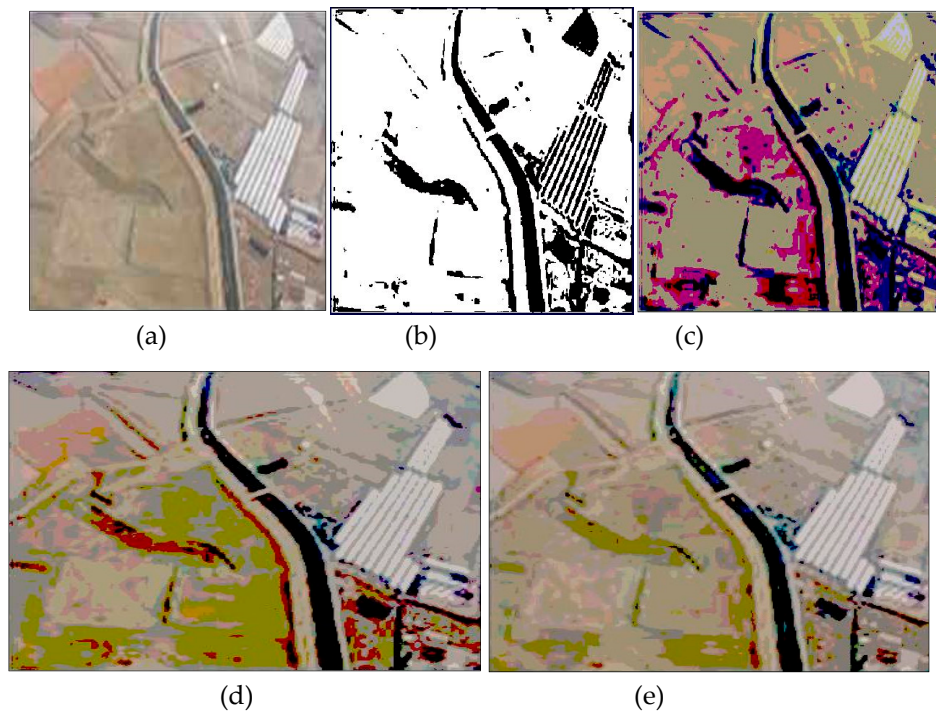
$$E_{loc} = \frac{1}{\tilde{U}} \sum_{i=1}^{\tilde{U}} (x_i - \tilde{x}_i)^2 + (y_i - \tilde{y}_i)^2, (i=1, 2, \dots, \tilde{U}) \quad (21)$$

7) The steps 2 through 6 are repeated till either all unknown nodes are localized or till it is not possible to localize any further nodes.

The above proposed scheme is validated via simulations on MATLAB R2016b which is explained in the following section.

#### 4. Results and Discussion

To validate the above methodology, PGSA has first been applied to find the optimal threshold levels for multilevel segmentation of an aerial image as shown in Figure 3. This image has 256 levels of intensity. PGSA runs for 30 trial runs with a maximum candidate plant node population as 25. The thresholding result seen in Figure 3 is for varying levels achieved by PGSA.



**Figure 3.** Results of the multi-level thresholding of an image using PGSA. a) The original aerial image. b) Two-level thresholding. c) Five-level thresholding. d) Six-level thresholding. e) Seven-level thresholding.

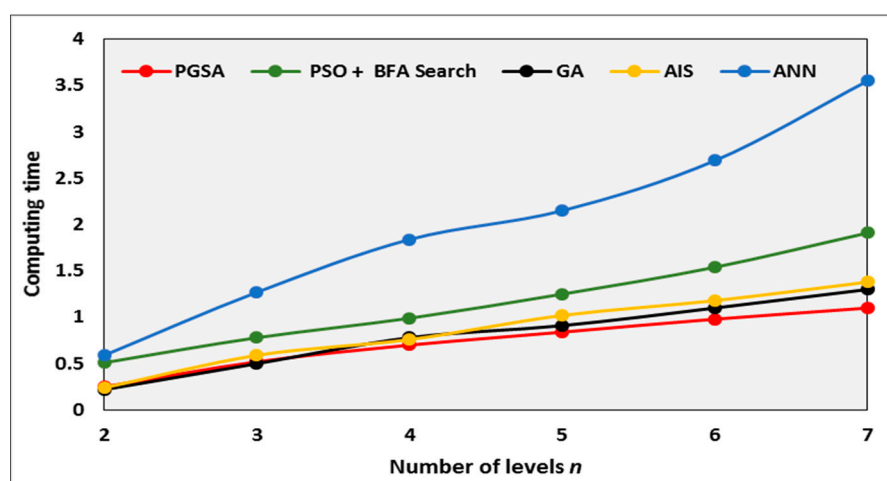
The above simulation results show that the PGSA is effectively able to segment the images with optimal threshold levels. Now, the time taken by PGSA in 30 trial runs is compared with other metaheuristic techniques like genetic algorithm (GA), particle swarm optimization (PSO) and bacterial foraging algorithm (BFA), artificial immune system (AIS) and artificial neural network (ANN) for performance evaluation of the proposed scheme. All the implementations were performed on the same system using the original parameter settings for all the algorithms in the cited literature.

Table 1 presents the computational time for different levels as found by PGSA, PSO+BFA, GA, AIS and ANN.

**Table 1.** Comparative analysis of the computational time with respect to the optimal thresholds produced for varying levels by the different algorithms and PGSA.

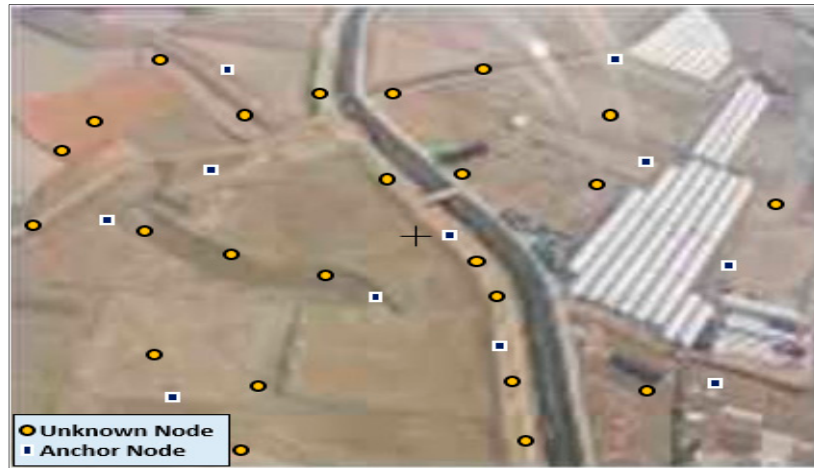
Number of levels	PGSA		PSO + BFA Search	GA	AIS	ANN
	Optimal thresholds	Computing time	Computing time	Computing time	Computing time	Computing time
2	112	0.25 s	0.51 s	0.22 s	0.24 s	1.27 s
3	103146	0.52 s	0.78 s	0.5 s	0.59 s	2.01 s
4	94125155	0.70 s	0.99 s	0.78 s	0.76 s	3.93 s
5	9.0119E+10	0.84 s	1.25 s	0.91 s	1.02 s	5.17 s
6	6.15E+13	1.02 s	1.54 s	1.1 s	1.34 s	8.43 s
7	4.13E+15	1.24 s	1.91 s	1.3 s	1.53 s	10.94 s

It can be seen that the number of optimal thresholds between levels increase exponentially. Figure 4 shows the computational time for finding optimal threshold levels with increasing number of levels for the 5 analysed algorithms. While the optimal thresholds found by all the algorithms except GA were same, the time taken for the search was significantly different, especially between PGSA and ANN. This was because of the training time required by ANN while PGSA operates as a single objective optimization algorithm over a multi-dimensional search space. Thus, it has no training time to handle. Moreover, the computation time of GA is found to be close to PGSA and even better for initial levels. However, GA converges prematurely for the subsequent levels failing to give optimal thresholds. Also, GA requires specifications like mutation, cross-over rates and functional coefficients which is not the case with PGSA. Further, PSO+BFA due to large number of decision parameters take more time. AIS, based on the auto immune mechanism has similar computation time as GA but it does not suffer premature convergence and gives better solutions for the subsequent levels.



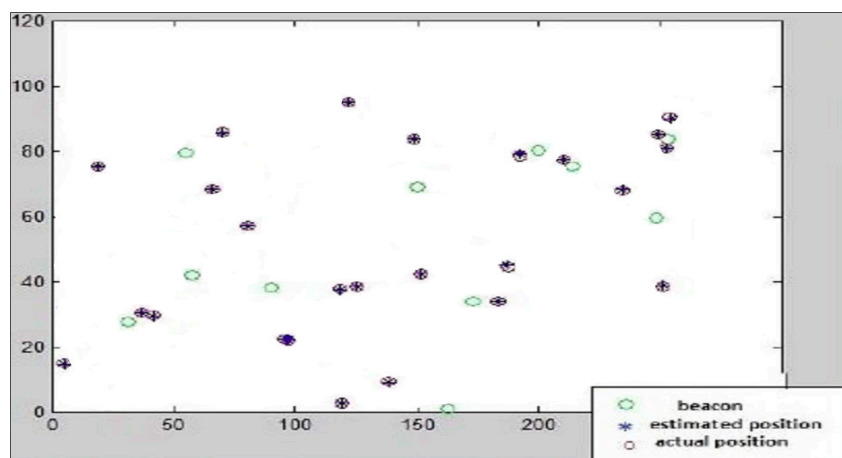
**Figure 4.** Computing time vs. Thresholding levels of 5 algorithms.

Therefore, during the process of autonomous node deployment the terrain image is segmented using the PGSA based thresholding and the threshold information is used to decide whether the node should be deployed in that terrain or not. Figure 5 shows the deployment scenario after PGSA based thresholding information was used. The aim was to avoid deploying the sensor on the highway or main road and areas with high vegetation or water. Here, the sensors have been deployed in such regions of interest and the average number of nodes saved from falling on the unwanted area (highway here) is 9.56.

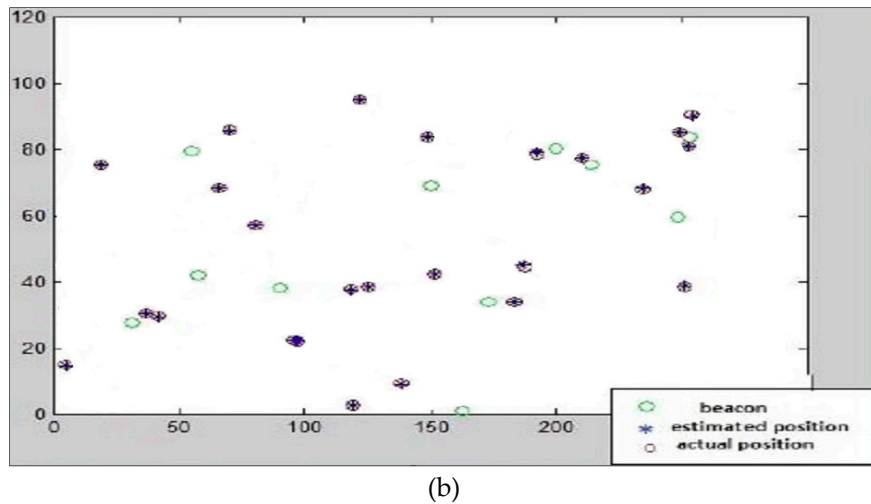


**Figure 5.** Sensor node deployment scenario after PGSA based thresholding

Further, with the application of PGSA, the deployed sensor nodes are localized. The simulation was done in MATLAB with 11 anchor nodes and 24 target/unknown nodes randomly deployed over an area of  $120 \times 300$  square units. The selection of the number of unknown nodes, anchors and area of deployment field is arbitrary, where transmission radius of each node is 40 units. Every unknown node that is localizable employs PGSA over this two-dimensional space to localize itself for a maximum of 20 iterations and maximum candidate plant node population as 25. Thirty trial runs of the experiment was performed for our proposed scheme where the error parameter  $E_n$  was selected as 3 and 6 on which the fitness of the solutions depend as already discussed while formulating equation(20). The results of a particular trial run of the experiment is shown in Figure 6, where Figure 6(a) shows the localization by PGSA and Figure 6(b) shows the localization by PSO. For a comparative analysis of results and performance metrics, five algorithms including our proposed scheme of PGSA, GA [32], AIS [2], ANN [40] and PSO [22] were implemented on the same system using the original parameters of the algorithms. As PSO has been widely used for sensor node localization problems, the localization result is shown in Figure 6 for the same trial run with  $E_n = 3$  and same initial deployment of anchors and unknown nodes as PGSA. Note that, all these algorithms being stochastic in nature do not produce the same result in all trials even with exactly identical initial settings. The results therefore, need to be averaged. Moreover as the initial deployment of unknown nodes while running these algorithms is random, the number of localizable nodes differ between iterations thereby effecting the computing time.

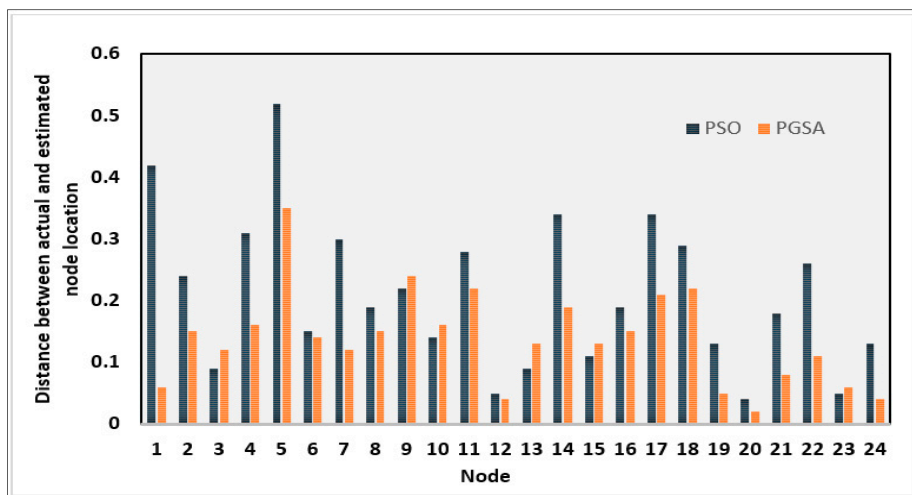


(a)



**Figure 6.** Results of the a)PGSA based localization b) PSO based localization of a trial run with  $A = 11, U = 24, R = 40, E_n = 3$  and sensor search space size =  $120 \times 300$  simulated using MATLAB.

It can be seen from Figure 6 that both PGSA and PSO perform well in localizing the deployed nodes. To analyse the results further, the difference between the actual locations of the target nodes and the estimated locations after applying both these algorithms can be seen in Figure 7.



**Figure 7.** Results of a trial run of PSO and PGSA showing the distance between actual and estimated node locations for  $A = 11, U = 24, R = 40, E_n = 3$  and sensor field size =  $120 \times 300$ .

From Figure 7, it is evident that the average distance between the actual coordinates and estimated coordinates for 24 nodes is lesser for PGSA than that of PSO. Further, the effect of the error  $E_n$ , added in form of Gaussian noise while estimating the distance, on localization accuracy can be seen from Table 2 where the performance of all 5 algorithms are compared. The performance of PGSA based localization is evaluated from  $(\tilde{N}\tilde{U}, E_{loc})$ , in Tables 2 and 3, where  $\tilde{N}\tilde{U} = U - \tilde{U}$  is the number of nodes that could not be localized,  $U$  is the total unknown nodes and  $\tilde{U}$  is the number of nodes that were localized. Also, the problem of flip ambiguity in sensor networks as shown in [41] is solved by using PGSA with the increasing number of anchor nodes in each iteration because of the good exploration and exploitation ratio of PGSA which finds accurate locations in lesser time. It is seen that the performance metric  $(\tilde{N}\tilde{U}, E_{loc})$  of PGSA is lesser than that of the other 4 methods, therefore it can be concluded both from Figure 7 and Table 2 that PGSA has a superior performance with respect to both accuracy and computational time.



**Table 2.** Comparative performance analysis of 30 trial runs of PGSA, PSO, GA, AIS and ANN based sensor node localization

Method	$E_n = 3$			$E_n = 6$		
	Mean $\bar{NU}$	Mean $E_{loc}$	Computing Time (s)	Mean $\bar{NU}$	Mean $E_{loc}$	Computing Time (s)
PGSA	0.641	0.0246	133.2232	0.812	0.1040	154.6479
PSO	1.514	0.0412	154.7412	1.645	0.2213	161.4267
GA	1.906	0.1037	148.7841	2.484	0.3184	157.1245
AIS	0.836	0.0389	146.3333	1.042	0.1927	157.2457
ANN	1.722	0.0571	198.1440	1.954	0.2556	245.4512

While flip ambiguity is solved with increasing iterations, the number of anchor nodes also increase as a result of localization in a sensor network which means that an unknown node has more references in the succeeding iteration, thereby the time taken for localization increases. So solving flip ambiguity increases computing time for localization. Table 3 presents the detailed observations over the first 5 iterations for the first three trials out of total 30 trials for node localization using PGSA. However, it has been observed from experimentation that when the number of reference anchor nodes were restricted to 5 in PGSA not only the time constraint was achieved but also the flip ambiguity problem was solved.

**Table 3.** Results of first 3 trial runs of PGSA based sensor node localization over first 5 iterations with  $A = 11, U = 24, R = 40, E_n = 3$  and sensor field size =  $120 \times 300$ .

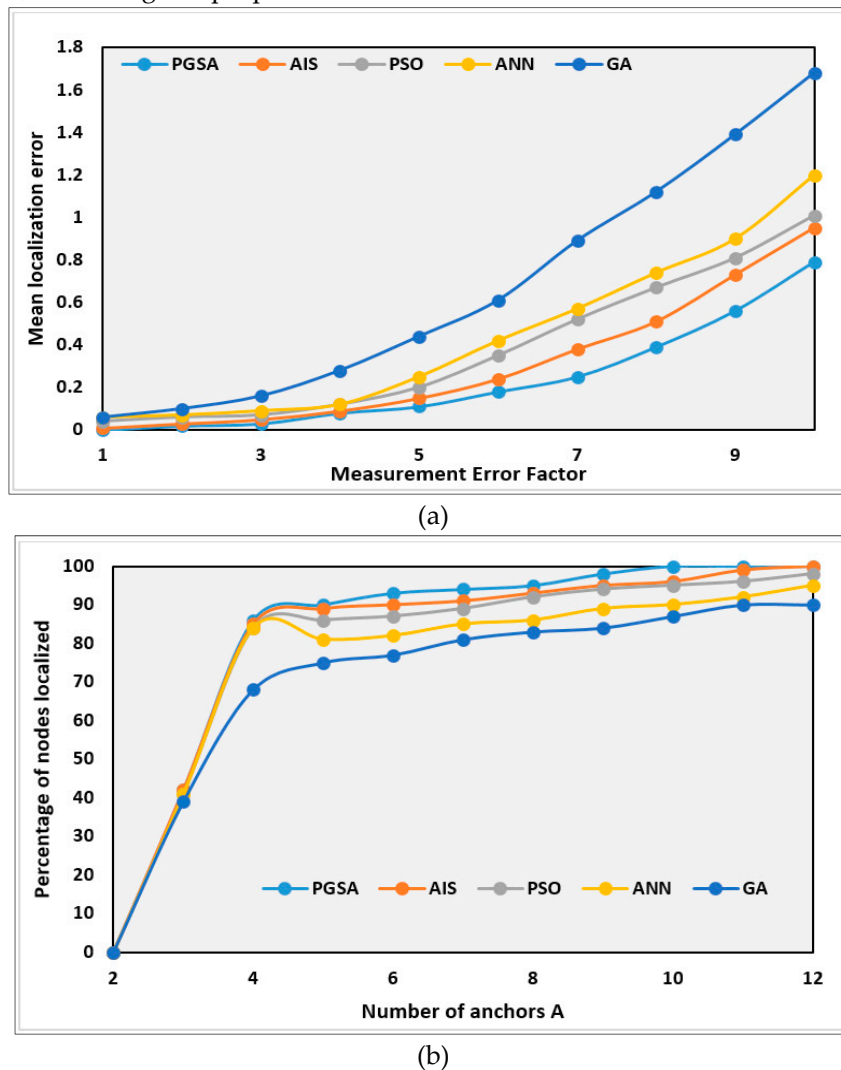
Trial		Iteration1	Iteration2	Iteration3	Iteration4	Iteration5
1	$\bar{U}$	12	18	20	23	24
	$E_{loc}$	0.0898	0.0919	0.0432	0.0511	0.5412
	Time(s)	8.46	12.84	15.46	20.84	21.11
2	$\bar{U}$	12	15	18	24	-
	$E_{loc}$	16.3762*	1.3977	0.0298	0.0184	-
	Time	7.14	25.0150	31.210	47.457	-
3	$\bar{U}$	13	14	16	20	23
	$E_{loc}$	0.0243	0.0412	0.0344	0.0646	0.0311
	Time	7.66	9.35	10.81	12	15.41

\*Effect of flip ambiguity which is removed in Iteration2 but takes longer computing time.

From table 2 above, it can be seen that increase in  $E_n$  leads to a decrease in the localization accuracy thereby increasing the mean absolute error  $E_{loc}$ . Figure 8(a) shows the increase in  $E_{loc}$  with  $E_n$  over 10 trials of PGSA. Moreover, as more target nodes get settled and behave as anchor nodes for other target nodes over successive iterations, the network suffers a problem of error propagation limiting its scalability. But as the number of anchor nodes and  $E_n$  were controlled in our study, we could restrict the error propagation during node localization. Therefore, the value of  $E_n$  should be kept in control as much as possible with very limited range of fluctuation in order to increase the accuracy of localization. This problem can be handled further by using the mechanism of node neighbour selection, error characterization and update criteria [42] which is beyond the scope of this paper. By now, it is clear that the number of nodes that are localized increase with iterations thereby giving more reference nodes which facilitates further localization of remaining target nodes. Figure 8(b) shows how the percentage of nodes that get localized depend on the node density or number of nodes serving as reference over 10 trials. Interestingly, from this figure we can infer that a wireless sensor network requires only its anchor nodes to be GPS enabled instead of having all the nodes GPS abled, thereby saving cost considerably. The result of 10 trials of PGSA with  $A = 11, U = 24, R = 40, E_n = 3$  and sensor field size =  $120 \times 300$  square units show that 10 GPS enabled anchor nodes are sufficient to localize all the 24 target nodes, making it 69% more cost efficient. The performance of



GA is not surprising in all these analyses as it suffers through premature convergence, thereby taking lesser time compared to PSO and ANN while giving less accurate solutions. This can be seen in Figure 8 (a) and (b). Thus, we can conclude that PGSA gives efficient optimal solutions with high accuracy and in lesser time, validating our proposed scheme.



**Figure 8.** Results of a) Mean localization error ( $E_{loc}$ ) vs. Measurement error factor ( $E_n$ ) over 10 trial runs of the 5 algorithms. b) Percentage of nodes localized with increasing number of anchors over 10 trials.

## 5. Conclusions

In this paper, a bionic random algorithm has been presented for the effective segmentation of terrestrial images for the autonomous deployment of sensor nodes from a UAV and also for the task of node localization deployed into that terrain. This is done via an iterative approach using PGSA where the solutions evolve to give globally optimal results while treating the problem as a multi-dimensional optimization problem. The PGSA is discussed in detail and the simulations and statistical results are reported. The proposed scheme is seen to be faster and more accurate than four state-of-the-art bioinspired algorithms that have already been used to solve such tasks. The importance of image segmentation is stressed in this paper for autonomous deployment of sensor nodes giving terrains of interest for node distribution, thereby reducing node loss. Further, the PGSA based localization discussed facilitates energy retention in the nodes due to fewer transmissions to the base station which is of paramount significance in such networks. The performance of nodes thus increases in such disaster monitoring scenarios. The results obtained have shown the effectiveness of the proposed method over existing methods, giving higher accuracy and faster performance. The

study can be extended in the direction of energy efficient sensor node deployment, PGSA based centralized localization and error propagation control methods.

**Acknowledgments:** This work is also supported by the National Research Foundation of Korea (NRF) grant funded by the Korea government (MSIP) (NRF- 2016R1A2A1A05005459). The grant covers the research study as well as the publication costs for Open Access Articles.

**Author Contributions:** D.B. and A.P. conceived and designed the experiments; D.B. performed the experiments; D.B., H.S. and W.H. analyzed the data; A.P. and S.K. contributed materials and analysis tools; D.B. wrote the paper.

**Conflicts of Interest:** The authors declare no conflict of interest. The founding sponsors had no role in the design of the study; in the collection, analyses, or interpretation of data; in the writing of the manuscript, and in the decision to publish the results.

## References

1. Soliman, O.S., Mahmoud, A.S., and Hassan, S.M. Remote Sensing Satellite Images Classification using Support Vector Machine and Particle Swarm Optimization. In proceedings of Third International Conference on Innovations in Bio-Inspired Computing and Applications, pp. 280- 285, 2012.
2. Mubarak, M., Reith, K.N., and Gopakumar, A. Node Localization in Wireless Sensor Networks by Artificial Immune System. In proceedings of Fifth International Conference on Advances in Computing and Communications, pp. 126-129, 2015.
3. Otsu, N. A threshold selection method from gray-level histograms. IEEE Transactions on Systems, Man, and Cybernetics, SMC-9(1): 62–66, Jan.1979.
4. Sezgin, M., and Sankur, B. Survey over image thresholding techniques and quantitative performance evaluation. Journal of Electronic Imaging, 13(1): 146–168, Jan. 2004.
5. Kuriakose, J., Amruth, V., and Nandhini, N.S. A survey on localization of Wireless Sensor nodes. In proceedings of the IEEE Conference on Information Communication and Embedded Systems (ICICES 2014), pp. 1-6, 2014.
6. Corke, P., Hrabar, S., Peterson, R., Rus, D., Sampalli, S., and Sukhatme, G. Autonomous deployment and repair of a sensor network using an unmanned aerial vehicle. In Proceedings of IEEE International Conference on Robotics Automation and Mechatronics (RAM 2004), vol. 4, pp. 3602–3608, May 2004.
7. Nazir, U., Arshad, M.A, Shahid, N., and Raza, S.H. Classification of localization algorithms for wireless sensor network: A survey. In proceedings of the IEEE International Conference on Open Source Systems and Technologies (ICOSST 2012), 2012.
8. Bhattacharjee, D., Paul, A., Kim, J.H., and Kim, M. SpringerPlus, 5(1784):1-20. 2016.
9. Ding, X., and You, J. Studies on Large-Scale Traveling Salesman Problem (LTSP) Based on Plant Growth Simulation Algorithm. In Proceedings of International Conference on Management and Service Science (MASS 2011), pp.12-14, August 2011.
10. Zhang, J., Fu, M., Qi, W., Yuan, C., and Tian, D. A comparison of land cover classification methods based on remote sensing and GIS technologies. In Proceedings of the International Conference on Information Engineering and Computer Science, pp.1-6, 2009.
11. Qiang, L., Jingfa, Z., and Na, W. Damaged road extraction from post-seismic remote sensing images based on GIS and Object-Oriented Method. In Proceedings of IEEE International Geoscience and Remote Sensing Symposium (IGARSS 2016), pp. 4247-4250, 2016.
12. Miner, A.S., Vamplew, P., Windle, D.J., Flentje, P., and Warner, P. A Comparative Study of Various Data Mining Techniques as applied to the Modeling of Landslide Susceptibility on the Bellarine Peninsula, Victoria, Australia. In Proceedings of the 11th IAEG Congress, Auckland, New Zealand, 2010.
13. Wang, X., and Niu, R. Spatial Forecast of Landslides in Three Gorges Based On Spatial Data Mining. Sensors, 9(3):2035-2061, 2009.
14. Hiremath, P.S., and Kodge, B.G. Visualization techniques for data mining of Latur district satellite Imagery. Journal of Advances in Computational Research, 2(1):21-24, 2010.
15. Nagaprasad, S., VinayaBabu, A., Madhukar, K., and Verghese, D. Spatial Data Mining Using Novel Neural Networks for Soil Image Classification and Processing. International Journal of Engineering Science and Technology, 2(10): 5621-5625, 2010.
16. Bishop, C. M. Neural Networks for Pattern Recognition. Oxford University Press, 1995.

17. Holland, J.H. *Adaptation in Natural and Artificial system*. Ann Arbor: The University of Michigan Press, 1975.
18. Siedlecki, W., and Sklansky, J. A note on genetic algorithms for large-scale feature selection. *Pattern Recognition Letters*, 10(5):335-347, 1989.
19. Timmis, J., and Knight, T. Artificial immune systems: Using the immune system as inspiration for data mining. In Abbass, H.A., Sarker, R.A. and Newton, C.S. (ed.), *Data Mining: A Heuristic Approach*, chapter XI, pp. 209–230. Group Idea Publishing, 2001.
20. Watkins, A., and Bogges, L. A new classifier based on resource limited artificial immune systems. In *proceedings of Congress on Evolutionary Computation. Part of the World Congress on Computational Intelligence (IEEE CEC 2002)*, pp.1546-1551, 2002.
21. Watkins, A., and Bogges, L. A resource limited artificial immune classifier. In *proceedings of Congress on Evolutionary Computation. Part of the World Congress on Computational Intelligence (IEEE CEC 2002)*, pp. 926-931, 2002.
22. Kulkarni, R.V., and Venayagamoorthy, G.K. Bio-inspired Algorithms for Autonomous Deployment and Localization of Sensor Nodes. *IEEE Transactions on Systems, Man and Cybernetics- Part C: Applications and Reviews*, 40(6):663-675, November 2010.
23. Gopakumar, A., and Jacob, L. Performance of some metaheuristic algorithms for localization in wireless sensor networks, *International Journal of Network Management*, 19(5):355-373, Oct. 2008.
24. Kheddad, R. and Belhadj-Aissa, A. Classification of Multispectral Images Using an Artificial Ant-based Algorithm. In *proceedings of the International Conference on Digital Information and Communication Technology and its Applications (DICTAP 2011)*, pp. 254-266, Dijon, France, 2011. Springer-Verlag Berlin Heidelberg 2011.
25. Kheddad, R., and Belhadj-Aissa, A. Contextual fusion by genetic approach applied to the classification of satellite images. In Goossens (ed.), *Remote Sensing in Transition*, chapter 3, pp.127-133. Millpress, Rotterdam, 2004.
26. Duda, R. O., Hart, P. E., and Stork, D. G. *Pattern classification*. Wiley Interscience. 2001.
27. Boukerche, A., Oliveira, H.A.B., Nakamura, E.F., and Loureiro, A.A.F. Localization systems for wireless sensor networks. *IEEE Wireless Communication Magazine*, 14(6): 6–12, Dec. 2007.
28. Mao, G., Fidan, B., and Anderson, B.D.O. Wireless sensor network localization techniques. *Computer Networks*, 51(10): 2529–2553, July 2007.
29. Niculescu, D., and Nath, B. Ad hoc positioning system (APS). In *Proceedings of IEEE Global Telecommunications Conference (GLOBECOM 2001)*. vol. 5, pp. 2926–2931. November 25–29, 2001.
30. Savarese, C., Rabaey, J., and Langendoen, K. Robust positioning algorithms for distributed ad hoc wireless sensor networks. In *Proceedings of USENIX Technical Annual Conference*, pp. 317–327. June 2002.
31. Dréo, A. Pétrowski, P. Siarry, E. Taillard. *Métaheuristiques pour l'optimisation difficile*. Edition Eyrolles, Chapitre 4, pp. 115-141, 2003.
32. Das, S., Abraham, A., and Konar, A. *Particle Swarm Optimization and Differential Evolution Algorithms: Technical Analysis, Applications and Hybridization Perspectives*. *Advances of Computational Intelligence in Industrial Systems* (Springer), 116: 1-38, 2008.
33. Goyal, S., and Patterh, M.S. Wireless sensor network localization based on BAT algorithm. *International Journal of Emerging Technologies in Computational and Applied Sciences (IJETCAS)*, pp. 507-512, 2013.
34. Guo, Z.W., Yang, S.P., Kui, L., and Ying, W.L. Gravitational Search Algorithm for Node Localization in Wireless Sensor Network. *The Journal of Information Technology*, 12: 5806-5811, 2013.
35. Kannan, A., Mao, G., and Vucetic, B. Simulated annealing based wireless sensor network localization. *Journal of Computers*, 1(6):15-22, 2006.
36. Zhang, Q., Wang, J., Jin, C., Ye, J., Ma, Ch., and Zhang, W. Genetic algorithm based wireless sensor network localization. In *Proceedings of 4th International Conference on Nature Computing (ICNC 2008)*, Jinan, China, pp. 608-613, 2008.
37. Tong L, Wang C, Wang W, Su W .A global optimization bionics algorithm for solving integer programming—plant growth simulation algorithm. *Syst Eng Theory Pract* 25:76–85. 2005.
38. Li, T., and Wang, Z-t. Application of plant growth simulation algorithm on solving facility location problem. *Systems Engineering Theory Practice*, 28(12):107–115. 2008.

39. Bhattacharjee, D., and Paul, A. A hybrid search optimization technique based on evolutionary learning in plants. In Proceedings of 2016 7th international conference of swarm intelligence (ICSI 2016), pp 271–279. Bali, Indonesia, 2016. Springer LNCS.
40. Ahmadi, H., and Bouallegue, R. Comparative study of learning-based localization algorithms for Wireless Sensor Networks: Support Vector Regression, Neural Network and Naïve Bayes. In Proceedings of 2015 International Wireless Communications and Mobile Computing Conference (IWCMC 2015), pp 1554–1558. 2015.
41. Dan, L., and Xian, W. An Improved PSO Algorithm for Distributed Localization in Wireless Sensor Networks. International Journal of Distributed Sensor Networks, 2015:1-9, 2015. Hindawi Publishing Corporation.
42. Liu, J., and Zhang, Y. Error control in distributed node self-localization. EURASIP Journal of Advanced Signal Processing, 2008:1–13, 2008.



© 2017 by the authors. Submitted for possible open access publication under the terms and conditions of the Creative Commons Attribution (CC BY) license (<http://creativecommons.org/licenses/by/4.0/>).