Swarm Intelligence Optimization Techniques for Obstacle-Avoidance Mobility-Assisted Localization in Wireless Sensor Networks

Abdullah Alomari, *Student Member, IEEE,* William Phillips, *Member, IEEE,*
Nauman Aslam, *Member, IEEE,* and Frank Comeau, *Member, IEEE*

***Abstract*—In many applications of wireless sensor networks (WSNs), node location is required to locate the monitored event once occurs. Mobility-Assisted Localization has emerged as an efficient technique for node localization. It works on optimizing a path planning of a location-aware mobile node, called mobile anchor (MA). The task of the MA is to traverse the area of interest (network) in a way that minimizes the localization error while maximizing the number of successful localized nodes. For simplicity, many path planning models assume that the MA has a sufficient source of energy and time, and the network area is obstacle-free. However, in many real-life applications such assumptions are rare. When the network area includes many obstacles, which need to be avoided, and the MA itself has a limited movement distance that cannot be exceeded, a dynamic movement approach is needed. In this paper, we propose two novel dynamic movement techniques that offer obstacle-avoidance path planning for mobility-assisted localization in WSNs. The movement planning is designed in a real-time using two swarm intelligence based algorithms, namely Grey Wolf Optimizer (GWO) and Whale Optimization Algorithm (WOA). Both of our proposed models, Grey Wolf optimizer based Path Planning (GWPP) and Whale Optimization algorithm based Path Planning (WOPP), provide superior outcomes in comparison to other existing works in several metrics including both localization ratio and localization error rate.**

***Index Terms*—wireless sensor networks, path planning; mobility models; localization models; optimization; grey wolf optimizer, whale optimization algorithm; obstacle-avoidance path planning**

I. Introduction

A WIRELESS sensor network (WSN) is a network consisting of a large number of sensor nodes that are linked together wirelessly to monitor, sense and gather required data from a physical area of interest [1], [2]. In the recent few years, WSNs have been engaged in many applications due to their size, cost, and simplicity of use [3]. Such applications include a wide range of health, military, agricultural, object tracking, underwater and many other applications [4], [5]. In many of these applications, the sensed data is significant and the location of this data is required to take a further action [6], [7]. A simple example is the location of pollution or the exact position of a fire source [8]. Since most of WSNs deployment processes are done randomly, it is difficult to provide each node with its own location manually. Thus, other efficient methods are needed to have such process done.

The Global positioning system (GPS) is one of the easiest location techniques to implement in WSNs; however, it is impractical to attach every single node with a GPS device because of the financial cost and other reasons including its size and inability to work in some applications like indoor applications [9]. For such reasons, alternatives methods have arisen to provide the deployed nodes with their locations. One simple solution is to provide only a portion of the deployed nodes with their own locations, and let those location-aware nodes, called anchors, spread their locations through single or multi-hops communications. Based on the received information from anchors, the unknown nodes (UNs), which represent the nodes that are not location-aware, can estimate their locations [10]. However, this solution also faces many challenges to be addressed and solved. Examples of these challenges include the cost of using a large number of anchor nodes, the uncertainty of completing the localization process in case some of the anchors fail, and the computation load that may affect the energy of anchors and their surround nodes.

An interesting area of research suggests taking advantages of such method with an improvement of it by using the mobility of anchors [11]. This idea aims to decrease the need of using a large number of nodes by letting one mobile anchor (MA) moves around the network and provides its own location to the nearby nodes. Based on the ability of receiving the MA signals, some UNs will be able to estimate their locations. Moreover, the MA typically can move freely in the network, which enables it to reach an extensive quantity of nodes. Unfortunately, even this solution has challenges, such as path distance minimization, the effects of the designed path on both localization ratio and accuracy, energy

A. Alomari is with the Department of Engineering Mathematics and Internetworking, Dalhousie University, Halifax, NS, Canada and the Faculty of Computer Science and Information Technology, Albaha University, Albaha, Saudi Arabia e-mail: (alomari@dal.ca).

W. Phillips is with the Department of Engineering Mathematics and Internetworking, Dalhousie University, Halifax, NS, Canada e-mail: (william.phillips@dal.ca).

N. Aslam is with the Department of Computer and Information Sciences, Northumbria University, Newcastle upon Tyne, UK e-mail: (nauman.aslam@northumbria.ac.uk).

F. Comeau is with the Engineering Department, St. Francis Xavier University, Antigonish, NS, Canada e-mail: (fcomeau@stfx.ca).

efficiency, localization time issues, and many others [9].

Typically, the mobile path planning is formed in advance, which works when the MA has sufficient sources of time and energy, and the area of interest is obstacles-free. However, in many real-time scenarios of WSNs, this assumption is uncommon. When the MA has a limited movement distance that cannot be exceeded and the network area includes many obstacles that also need to be avoided, a sufficient and dynamic path planning is needed.

This paper introduces two novel dynamic meta-heuristic optimization techniques for mobility-assisted localization in WSNs. The suggested path planning models are based on two new optimization algorithms, namely the Grey Wolf Optimizer (GWO) [12] and the Whale Optimization Algorithm (WOA) [13]. The proposed models are respectively called Grey Wolf optimizer based obstacle-avoidance Path Planning (GWPP) and Whale Optimization algorithm based obstacle-avoidance Path Planning (WOPP). The novelty of our proposed models lies in employing an optimization algorithms to direct the path formation of the MA, which helps to maximize the localization ratio and minimize the localization error. By using the optimization algorithms, the MA movement is formed in real-time; it also avoids the obstacles, takes into account the maximum distance constraint, and simultaneously achieves the objectives of the entire localization process. To the best of our knowledge, we are the first to use swarm based optimization techniques assuming such scenarios in path planning for localization in WSNs. The proposed models provide outstanding results in several metrics in comparison to some existing works.

We work on designing an obstacle-avoidance path for mobility-assisted localization in WSNs. We summarize our contribution in the following points:

1) For the first time, the MA path is dynamically formed based on meta-heuristic optimization models. Using either GWO or WOA in the movement decision helps to increase the number of localized nodes and more importantly minimize the localization error.
2) While considering all of the area's and nodes' constraints, the proposed models ensure that a larger number of unknown nodes can receive the MA's localization information compared to other models. This number increases when the maximum distance increases. In comparison to other existing models, our proposed models offer better localization ratios.
3) The objective function comes first. In every movement step, the MA will make its decision for next movement based on the fitness of the objective function. Therefore, both models show a competitive accuracy.
4) Regardless of the number of obstacles, locations, or dimensions, the optimized MA movement can sense and find them. Thus, the MA can consequently act by ignoring the direction of the obstacles and consider the alternative directions while also taking other constrains into account.
5) Unlike the other models, in which the MA has to go around the obstacles and keep moving in the same movement pattern, the MA in our proposed models is free to change its own direction based on the applied optimization model. This freedom is important for avoiding to have the MA being trapped in a small region.

The rest of this paper is organized as follows: Section II provides a brief review about the related work. Section III provides an overview of swarm intelligence based works in WSNs, and the two optimization models used in this work, GWO and WOA. Section IV states the system model assumptions. We introduce our proposed models starting with the constraints and the objective analysis, then present the GWPP and WOPP approaches in details, and ending with describing the localization process all in Section V. In Section VI and VII respectively, we show the simulation and performance setting, and discuss the evaluation results. Section VIII offers a discussion on both proposed models based on the shown results, and we conclude our work by Section IX and ending by stating the future works.

II. Related Work

Localization in WSNs has become a active area of research, resulting in several models and algorithms in the last few years [9]. In general, these models differ based on their objectives in several ways such as the localization processing, localization method, application area, deployment area, or anchor types [8]. Figure 1 summarizes the objectives of the localization models.

In the last classification, anchor types, the localization models are categorized into four different types, (static nodes and static anchors), (mobile nodes and static anchors), (static nodes and mobile anchors), and (mobile nodes and mobile anchors) [14], [15]. Since our proposed models are located under the third area (static nodes and mobile anchors), we limit our discussion to this category. In this area, the path planning is formed following one of three types, random, static or dynamic. A movement type can be chosen based on the available sources of time or energy, and on the objective of the localization technique. As this work focuses on static and dynamic path models, the related work review will be limited to these types, specifically to [8], [14]–[22].

Any discussion of mobility assisted-localization in WSNs must include the SCAN and Hilbert models. SCAN and Hilbert [15] are two models that are considered as the inspiring paths to many other following models. Although they opened a wide area of research, SCAN and Hilbert suffer from the collinearity problem. Such problem affects both the localization accuracy and the coverage of the localized nodes. A subsequent work, called Circles, inspired by a circling movement to overcome the collinearity problem is proposed in [16]. However, Circles

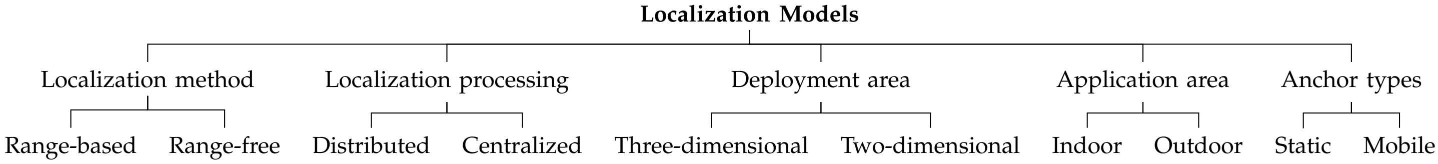


Figure 1: Classification of localization models in WSNs based on their objectives

has difficulty to reach the corners of the network, which also affects negatively the localization ratio and coverage. To overcome the problem of collinearity, a newer model that presents a mobile anchor-assisted localization algorithm based on a regular hexagon (MAALRH) is proposed in [17]. MAALRH forms the path planning following a hexagon-like movement, which succeeds in solving the collinearity problem. But similar to Circles, MAALRH in its first design cannot reach the corners of the network. LMAT, which is stand for Localization algorithm with a Mobile Anchor node based on Trilateration, is proposed in [18]. LMAT path planning is designed based on a balanced set of triangles, which increases the localization accuracy by covering the collinearity. In addition, LMAT is able to cover the entire network area, which improves the localization ratio. H-Curve in [14] proposed another models that works on both solving the collinearity problem and reaching the entire deployed nodes. The proposed model guarantees that all nodes inside the network can be reached by the MA, thus, localized. H-Curves comes with a shorter path length in comparison to that in LMAT. A three-dimensional path planning based on H-Curve concepts is designed in [19]. Z-Curves is a static model that composes the movement of the MA following a number of Z-shapes [21]. Node Localization Algorithm with Mobile Beacon (NLA-MB) is a novel dynamic model proposed in [20]. NLA-MB assumes that the MA has a limited movement distance, hence, the path formation should be designed based on such constraint. Fuzzy-Logic based Path Planning for mobile anchor-assisted Localization (FLPPL) is proposed in [8]. FLPPL has the same constraints as NLA-MB; however, it employs multiple individual inputs in a fuzzy-logic system for path planning that succeeds in minimizing the localization error and to maximizing the localization ratio in comparison to other similar models.

All of the above mentioned papers, except Z-Curves, consider the area of interest to be plain and empty of obstacles. However, in many applications of WSNs, some kinds of obstacles and objects can be found in the area. Z-Curves proposes that when the MA faces an obstacle on its way, it simply turns around the corner of that obstacle and goes to the obverse point of the obstacle to continue the movement. However, such a solution leaves several drawbacks of path length and collinearity. Another proposed model, called Snake-Like is proposed for obstacle-avoidance in [22]. The proposed movement formation is similar to that proposed in SCAN; however it follows a horizontal approach. When an obstacle is faced, similar to Z-Curves, the MA will move on the border of the obstacle and reach the other point on the same line to the last point before the obstacle, and keeps its movement. The main disadvantage of the Snake-Like proposal is the collinearity. Snake-Like does not provide a solution for the collinear points and how to deal with them.

As NLA-MB and FLPPL assume that the MA has a limited movement distance that cannot be exceeded, and since Z-Curves and Snake-Like consider the obstacle-existence scenarios, and there is no work that takes into account both problems, we decided to investigate consider both constraints and developed the two dynamic optimized models, GWPP and WOPP.

III. Swarm Intelligence in WSNs

In recent years, Meta-heuristics have gained an attention and have been applied in many fields. The term Meta-heuristic denotes an area of general algorithms and frameworks that are designed to deal with complex optimization problems [23]. Simplicity of concept, ease of implementation, and its applicability to be used in different problems are few reasons behind its successful spread [13]. Their inspiration, typically, is based on mimicking a natural phenomenon [13], [23]. Generally speaking, the Meta-heuristics can be categorized into three main classification, evolution-based, physics-based, and swarm-based methods [12]. Genetic Algorithms (GA) [24] is the most popular example of the first category, the evolution-based algorithms. In physics-based category, we can mention the Simulated Annealing (SA) [25] as a popular example, while in the swarm-based methods, there is a list of existing models that includes Particle Swarm Optimization (PSO) [26], Ant Colony Optimization (ACO) [27], Artificial Bee Colony (ABC) algorithm [28], and many other existing algorithms. Since our techniques focuses on swarm-based optimization, we limit our discussion only to such methods. Swarm-based, or Swarm Intelligence (SI), optimization is a relatively new field that showed a novel direction in optimization research [29]. Simply, swarm-based algorithms are optimization models that try to solve the researched problem by imitating the social behavior of creatures, especially animals [12]. In WSNs, swarm-based optimization models have been used for many purposes including routing [30], [31], energy efficiency [32]–[34], reliability [35] and other applications. For instance, in [32], the authors introduce a hybrid swarm intelligence energy efficient algorithm that works to enhance the clustering and routing processes using both ABC and ACO algorithms. In [33], a new energy efficient cluster head selection algorithm based on the PSO, called PSO-ECHS, is proposed. It consists of two phases, a cluster head selection phase that is based on PSO, and a cluster formation phase, which depends on the residual energy of nodes. In node localization in WSNs, several works have been done considering the SI algorithms. In [36], another SI based model is proposed for node localization this time. The work introduces two different localization models that use PSO and ABC together. The localization algorithms are evaluated in both single-stage and multi-stage localization. The evaluation results show that the PSO-based localization algorithm performs better than the one that uses ABC. However, no comparison to other existing localization works was performed. A multi-objective PSO localization algorithm, MOPSOLA, is presented in [37] to enhance the localization in WSNs. The investigated objective functions consist of the space distance constraint and the geometric topology constraint. The proposed model shows better results in terms of localization error compared to other similar models. Another direction of localization is considered in [38], where the unknown nodes are assumed to be moving and distributed in underwater WSNs. The proposed localization model is based on mobility prediction and PSO. The results show that the nodes locations can be estimated along with their velocity and movement can be predicted. However, to best of our knowledge, no study of using SI in controlling the MA movement and path planning for obstacle-exist networks in localization assistance in WSNs has been proposed. Therefore, we propose our GWPP and WOPP models. More details about the proposed models will be shown in Section V, but an overview about GWO and WOA will be first presented in the following sections III-A and III-B respectively.

A. Grey Wolf Optimization

Proposed by Mirjalili et *al.* in [12], Grey Wolf Optimizer (GWO) is a new metaheuristic algorithm that mimics the natural leadership hierarchy system of the grey wolves. Grey wolves live in small groups of members. They have a special social dominant hierarchy that divides the group into four hierarchal parts starting of the top leaders called alphas (α), then betas (β), delta (δ), and the lowest ranking of the hierarchy is omega (ω). Each of these kinds leads the subgroup that is located in its lower ranking. For example, Delta wolves are followers of alphas and betas but they can lead the omegas. Figure 2 depicts the social pyramid of the Grey Wolf in nature.

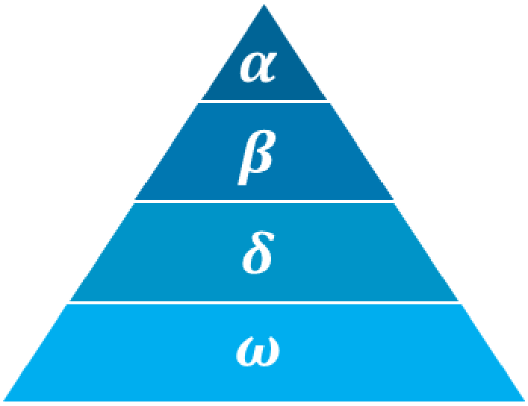


Figure 2: Hierarchy of grey wolf (adopted from [12])

For the mathematical modelling, alpha (α), beta (β), and delta (δ) represent the best three candidate solutions within the search space respectively. The optimization process is guided by these candidates. All other candidate solutions are considered as omegas (ω). Each candidate solution is represented as a vector in

$$\vec{X} = x_1, x_2, ..., x_n \tag{1}$$

Where x_i is the current position of the grey wolf, and n is the dimension of the search space [39]. Mathematically, the hunting process represents the optimization, while searching for the prey represents the available solutions. The grey wolf hunting behavior consists of three main phases starting with encircling the prey, hunting, and attacking the prey. The first phase, prey encircling, is mathematically modelled as

$$\vec{D} = |\vec{C}.\vec{X}_P(t) - \vec{X}(t)| \tag{2}$$

$$\vec{X}(t+1) = \vec{X}_P(t) - \vec{A}.\vec{D} \tag{3}$$

Where t indicates the current iteration, $\vec{A}$ and $\vec{C}$ are two coefficient vectors, $\vec{X}_P$ is the position vector of the prey, and $\vec{X}$ indicates the position vector of a grey wolf.

The two coefficient vectors of $\vec{A}$ and $\vec{C}$ can be calculated as

$$\vec{A} = 2\,\vec{a}.\vec{r}_1 - \vec{a} \tag{4}$$

$$\vec{C} = 2\,\vec{r}_2 \tag{5}$$

Where components of $\vec{a}$ are linearly decreased from 2 to 0 over the iterations course, and $\vec{r}_1$, $\vec{r}_2$ are vectors chosen randomly in the range of [0, 1]. Initially, the grey wolves are able to locate potentially the prey positions in order to hunt them. This localization is guided by the first best solutions, namely (α), beta (β), and delta (δ). Thus, these three best candidate solutions so far will be saved and updated over the iteration times in order to support other wolves (ω) finding their own positions. This process of hunting is represented using the following formulas

$$\vec{D}_\alpha = |\vec{C}_1.\vec{X}_\alpha - \vec{X}| \tag{6}$$

$$\vec{D}_\beta = |\vec{C}_2.\vec{X}_\beta - \vec{X}| \tag{7}$$

$$\vec{D}_\delta = |\vec{C}_3.\vec{X}_\delta - \vec{X}| \tag{8}$$

Where $\vec{D}_\alpha$, $\vec{D}_\beta$, and $\vec{D}_\delta$ are the updated distance vectors between the position of each leader wolf and the other wolves. $\vec{C}_i$ is the required coefficient vector that is calculated using the formula in equation 5, and $\vec{X}$ is the position of other wolves. Each $\vec{X}_i$ represent an estimated position calculated based on the distance vector between the omega wolf and each leader wolf of $\vec{D}_\alpha$, $\vec{D}_\beta$, and $\vec{D}_\delta$ respectively. They are calculated as

$$\vec{X}_1 = \vec{X}_\alpha - \vec{A}_1.(\vec{D}_\alpha) \tag{9}$$

$$\vec{X}_2 = \vec{X}_\beta - \vec{A}_2.(\vec{D}_\beta) \tag{10}$$

$$\vec{X}_3 = \vec{X}_\delta - \vec{A}_3.(\vec{D}_\delta) \tag{11}$$

The updated new position vectors are given as $\vec{X}_i$ where $\vec{X}_1$ is the new position based on alpha position $\vec{X}_\alpha$ and the distance vector $\vec{D}_\alpha$, $\vec{X}_2$ is the new position based on alpha position $\vec{X}_\beta$ and the distance vector $\vec{D}_\beta$, and $\vec{X}_3$ is the new position based on alpha position $\vec{X}_\delta$ and the distance vector $\vec{D}_\delta$. The coefficient vectors of $\vec{A}_i$ are calculated as in equation 4. Therefore, using the average sum of all previous positions, the new position vector is calculated as

$$\vec{X}(t+1) = \frac{\vec{X}_1 + \vec{X}_2 + \vec{X}_3}{3} \tag{12}$$

The third phase, the prey attacking, comes after the hunting phase. In this phase, the value of $\vec{a}$ is decreased, which therefore decreases the value of $\vec{A}$. The value of $\vec{A}$ is limited by the range $(-2a, 2a)$. In order to find a better solution, $\vec{A}$ value has to be more than 1.

Since its first appearance, GWO has caught growing attention. It has been used in tremendous engineering and optimization problems. In WSNs, GWO is used varying from routing [40], energy efficiency and clustering [41], and localization [39].

B. Whale Optimization Algorithm

Whale Optimization Algorithm (WOA) is another new swarm intelligence optimization model that was recently proposed [13]. As indicated by its name, WOA simulates the social behavior of humpback whales. Whales have the ability to think, learn, communicate and have a higher level of smartness in comparison to many other creatures. An interesting social behavior of whales is their special strategy of hunting, so-called bubble-net feeding. This strategy consists of two main maneuvers, upward-spirals and double-loops. The proposed WOA is based only on the former one. In this maneuver, whales dive deeply in the water and start creating bubble around the prey in a spiral shape, and swim up toward the surface of the water. The spiral shape movement is similar to number '9'. The mathematical model of WOA consists of three phases, prey encircling, spiral bubble-net feeding maneuver, and search for prey. In the first phase, the prey encircling, the whales are assumed to recognize the location of the prey and encircle them. Initially, the WOA considers the target prey as the current best candidate solution, since the position of the optimal solution is not known a priori. This search candidate will be updated in case a better candidate solution is achieved. Similar to GWO, this behavior is formulated as

$$\vec{D} = |\vec{C}.\vec{X}^*(t) - \vec{X}(t)| \tag{13}$$

$$\vec{X}(t+1) = \vec{X}^*(t) - \vec{A}.\vec{D} \tag{14}$$

Where $\vec{C}$ and $\vec{A}$ are coefficient vectors, $\vec{X}^*$ is the so far obtained best solution of position vector, and $\vec{X}$ is the position vector. The value of $\vec{X}^*$ is updated continually with each iteration. The two coefficient vectors of $\vec{C}$ and $\vec{A}$ are calculated as

$$\vec{C} = 2\ \vec{r} \tag{15}$$

$$\vec{A} = 2\vec{a}.\vec{r} - \vec{a} \tag{16}$$

Where, similar to GWO, $\vec{r}$ is a random vector in the range of [0, 1], and $\vec{a}$ is a linearly decreased value from 2 to 0 over iterations course. Adjusting the values of $\vec{C}$ and $\vec{A}$ leads to give different places around the best candidate achieved. In the second phase, the bubble-net attacking method is represented mathematically as the exploitation phase. This behavior of bubbling is done following two approaches, the shrinking encircling mechanism and the spiral updating position. The former approach is achieved by decreasing the value of $\vec{a}$, which therefore decreases the value of $\vec{A}$. In the latter approach, the distance between the whale current location and the prey location is calculated. A spiral equation is formulated to mimic the whales' movement between the two locations as follows

$$\vec{X}(t+1) = \vec{D'}.e^{bl}.\cos(2\pi l) + \vec{X}^*(t) \tag{17}$$

$$\vec{D'} = |\vec{X}^*(t) - \vec{X}(t)| \tag{18}$$

Where $\vec{D'}$ indicates the best solution so far (the distance of the ith whale to the prey), b is a constant value that defines the logarithmic spiral, l is a random number in the range [-1, 1]. The whales swim simultaneously within a shrinking circle in a spiral-shaped path. Similarly, the WOA has a 50% of choosing the shrinking encircling mechanism or the spiral model and updates the new position based on that. This is mathematically modelled as

$$\vec{X}(t+1) = \begin{cases} \vec{X^*}(t) - \vec{A}.\vec{D}, & \text{if p} < 0.5 \\ \vec{D'}.e^{b1}.\cos(2\pi l) + \vec{X^*}(t), & \text{if p} \geq 0.5 \end{cases} \tag{19}$$

Where p is a random value in [0, 1]. The last phase, search for prey, is simulated as the exploration phase in WOA. Unlike the previous phase, the whales search for prey randomly according to the position of each other. For this reason, the value of $\vec{A}$ is chosen randomly. However, it has to be greater than 1. This is intended to let the whale exploration to perform a global search. This is formulated as

$$\vec{D} = |\vec{C}.\vec{X}_{rand} - \vec{X}| \tag{20}$$

$$\vec{X}(t+1) = \vec{X}_{rand} - \vec{A}.\vec{D} \tag{21}$$

Where $\vec{X}_{rand}$ is a random value representing a random position vector (a random whale) selected from the current population.

Generally, the WOA is initiated with a set of random solutions. With every iteration, the search agents update their location based on either the best solution obtained so far or a random search agent.

Although, it was published recently, the WOA is used in many engineering application including WSNs. The work introduced in [42] proposes a lifetime maximization of WSNs using WOA.

IV. System Model and Assumptions

The following assumptions are used to form the system model:

1) A two-dimensional plane network following a square shape. S one side of the square area of the network in m.
2) The network area is assumed to have a set of obstacles. The number of obstacles is denoted as O. The dimensions of each obstacle is given as O_{size} in m. For simplicity, the obstacles are assumed to be rectangles.
3) A set of unknown nodes (UNs) are distributed following an arbitrary form. The number of these nodes is introduced as N.
4) At first, all UNs inside the network have no prior knowledge about their current locations.
5) All deployed nodes are static, which means no node is able to change its own location once the distribution process is done.
6) Each node has a fixed communication range R_{Tx} in m.
7) A mobile node (MA) is able to move freely in the network in straight directions except in locations where obstacles exist. It is also assumed to have the ability to locate itself in any point in the network. The number of MAs is denoted as M.
8) The MA is able to detect any obstacle in its direction using any detection method. Examples of those detection methods include infrared (IR) sensors or passive infrared (PIR) sensors [43]. Unlike the active IR, the PIR depends on the received IR that is emitted by the objects.
9) The MA movement is constrained by the value of maximum distance (d_{max}), where the MA movement cannot go beyond this value.
10) While the MA is moving, it frequently stops to provide nodes within its communication range with its current position. Each of these positions called a localization point.
11) The MA and UNs cannot communicate with each other except if their locations are within the communication range of each other.
12) Once any three different location information of MA are received by a UN, it estimates its own location by the used localization model.
13) Once the UN succeeds in estimating its location, it turns into a reference node (RN). The RN can share its own location information with other UNs located within its range, which will help in estimating their locations.

V. Proposed Models

In this section, we discuss the constraints and objectives of this model and then introduce the two movement techniques. Then, we describe the localization process from both side, on the MA's side and on the UN's side.

A. Constraints and Objectives Analysis

As in many path planning models in WSNs, a number of constraints is assumed. In this model, we assume four different constraints as follow:

1) In the network area, every visited localization point must be unique. This means that the MA cannot visit a localization point more than once and cannot return to the same point at any time.
2) To avoid the collinearity problem that affects the localization results, the model forces the MA movement to be not collinear by assuming that every three consecutive localization points are not on the same line.
3) The MA cannot exceed the limited movement distance (d_{max}). Once this distance is reached, the MA stops.
4) The network area includes a set of obstacles distributed randomly around the network. The MA has no prior knowledge about the obstacles' locations and has to detect them during its movement, thus, avoid them.

Although it is a rare situation to have the MA trapped in a small area of network, the MA can any of the first three constraint rules once it happens in order to keep moving. The last constraint cannot be broken since the MA is unable to move over the obstacle.

The main objective function of this model is to minimize the average localization error of the deployed nodes. It is represented as

$$Minimize \quad Error_{avg} \tag{22}$$

B. Movement Decision

Before starting the MA movement, a few rules regarding the movement pattern will be assigned. The network area will be virtually divided into a set of lines, each line includes a set of guide points. The distance between each two lines is fixed. Also, the distance between any two guide points in the same line will be fixed, denoted as d_p. Similar to [8], this distance is given as $R_{Tx}/\sqrt{3}\ m$. Therefore, to maintain the condition of fixed distance between each two neighbor points in any direction, the distance between each two lines are given as $d_p/2\ m$. However, the starting point in each consecutive lines will be incompatible. In other words, if the starting point of line a is (x, y), the next line b will be starting a half of d_p as $(0.5x, 2y)$. This is intended to overcome the collinearity problem forming a triangle-shape of virtual points. In addition, since the MA has to consider the movement constraints, a few rules are considered as shown in Figure 3.

Initially, as in Figure 3.**a**, the MA is surrounded with six different points, any of which can be chosen as a next point to visit based on the optimization model decision. Let us say that the MA has the following points in its range $\{a, b, c, d, e, f\}$. Based on the optimization decision, the MA selected the point f as next point to move to. Now, a new set of points will be formed. The last point that MA has just left will be a new point in the new form, called c in Figure 3.**b**. However, this point will be excluded from the potential visiting point since it has already been visited. Thus, the MA will never visit it again. Two more points in Figure 3.**b** will be ignored as well, namely $\{e, f\}$. These points will be excluded for different reasons. The point e is located in an obstacle direction, which MA has to avoid, by considering other directions. On the other hand, the point f will not be considered because of the collinearity problem, which imposes that three consecutive points cannot be collinear. Thus, only the other three points, namely $\{a, b, d\}$, will be considered. The decision of moving to one of them will be made based on the applied optimization model. In this example, the MA selects the point a as the next point, and the same procedure concept will be repeated.

1) *Grey Wolf Optimizer based Obstacle-Avoidance Path Planning (GWPP)*

Once the MA receives the information of the network area and forms its movement virtual points, it starts its journey by making three random movements. This is meant to let the MA to get more information about its starting area [8], [20]. With every movement, the MA will be providing its current location to the nearby UNs. Once three random movements are done, it is time for the optimized movement. In the next movement steps, the MA will select the path to satisfy the collinearity condition and avoid the obstacles. Based on such considerations, the MA will use the GWO to define a candidate direction point by calculating the fitness value of each direction point. The fitness function used here is the objective function, minimizing the localization error rate. It is represented as

$$error_{total} = \left(\sum_{i=1}^{N} error_{(i)}\right) \tag{23}$$

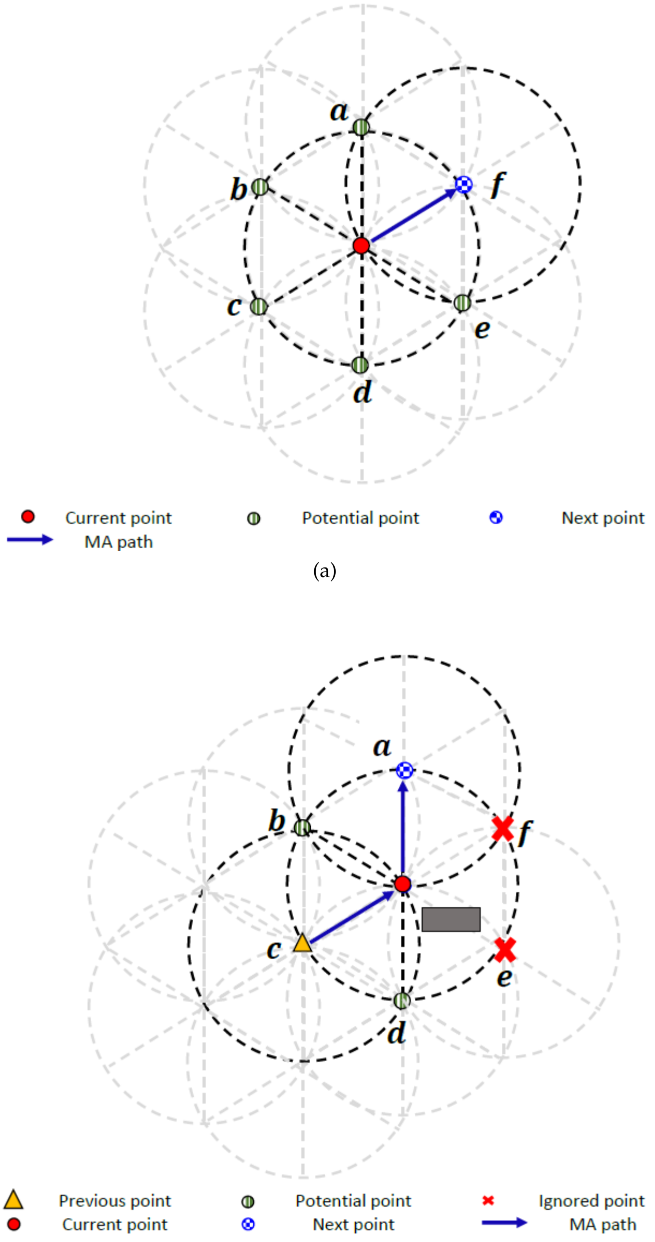


Figure 3: The base of selecting the next movement point in (**a**) first movement, and (**b**) next movements.

Where $error_{(i)}$ is the localization error of node i and can be given as

$$error_{(i)} = \sqrt{(x_i - u_i)^2 + (y_i - v_i)^2} \tag{24}$$

Where (x_i, y_i) are the real coordinates of the node i, and (u_i, v_i) are the estimated ones of the same node i. The MA will evaluate all nodes within its range, run the GWO optimization and select the point that satisfies most the fitness function. The modified pseudo code of GWO is given as follows

Algorithm 1 Pseudo code of the GWO algorithm in GWPP

```
Initialize the number of movement steps $X_i (i = 1, 2, ..., n)$
Initialize $a$, $A$, and $C$
Calculate the fitness of each candidate point
$X_\alpha$ = the best candidate point
$X_\beta$ = the second best candidate point
$X_\delta$ = the third best candidate point
while $t < T_{max}$
    for each candidate point
        Update the position of the current point by the eq. 12
    end for
    Update $a$, $A$, and $C$
    Calculate the fitness of all candidate points
    Update $X_\alpha$, $X_\beta$, and $X_\delta$
    $t = t + 1$
end while
return $X_\alpha$
```

Where the X_i here indicates the number of MA movement steps (the grey wolf population), given as the maximum distance of MA divided by the distance between each two points as

$$X_i = \frac{d_{max}}{d_p} + 1 \tag{25}$$

Each potential candidate represents a point search agent , and X_α is the best candidate point (the best search agent) to move to. t is the current iteration and T_{max} is the maximum number of iterations.

2) Whale Optimization Algorithm based Obstacle-Avoidance Path Planning (WOPP)

Similar to GWO, the movement decision here will be made based on the optimization model of WOA. However, it starts first with three random movements. The modified pseudo code of WOA is given as in algorithm 2, where the whale population X_i here denotes the number of MA movement steps, each search agent indicates the candidate next points, and X^* is the best candidate point.

WOA includes more details about the movement pattern, as shown in algorithm 2; there might be a chance to select a random point among the available points. However, this randomness should not conflict with other related constraints, specifically the collinearity and obstacle points. This random selection is based on the 50% chance of choosing either shrinking encircling mechanism or the spiral model shown above in equation 18, in Section III-B (WOA model).

Algorithm 2 Pseudo code of the WOA algorithm in WOPP

```
Initialize the number of movement steps $X_i (i = 1, 2, ..., n)$
Calculate the fitness of each candidate point
$X^*$ = the best candidate point
while $t < T_{max}$
    for each candidate point
    Update $a$, $A$, $C$, $l$ and $p$
        if $p < 0.5$ then
            if $|A| < 1$ then
                Update the position of the current candidate point by the eq. 14
            else
                Select a random candidate point $X_{rand}$
                Update the position of the current candidate point by the eq. 21
            end if
        else if $p \geq 0.5$ then
            Update the position of the current candidate point by the eq. 17
        end if
    end for
    Check if any candidate point goes beyond the search space and amend it
    Calculate the fitness of each candidate point
    Update $X^*$ if there is a better solution
    $t = t + a$
end while
return $X^*$
```

C. Mobility Movement and Localization Process

The movement of the MA and localization procedure are simultaneously preformed in two aspects, the MA side and UN side as follow:

1) Procedure in Unknown Node's Side

The following steps are performed in the UNs side:

1) All UNs are distributed randomly.
2) Each node will initiate a neighboring table, which includes all neighbor nodes located within the communication range of that UN. The neighboring table will include the node id, the node type, the number of neighbors, the neighbor ids, and neighbor status of localization. The node type here indicates the localization status and can be either a UN or an RN.
3) Each UN will wait for MA arrival. Upon the MA arrival to each node, the node will exchange its table with the MA.
4) Once three varied points are received, the UN estimates its own location.

5) When a UN get its location estimation, it turns into an RN, and updates its table.
6) Each RN will share its updates table with all neighboring nodes.

The entire process is shown in the following algorithm 3.

Algorithm 3 Pseudo code of the node localization process in the UN

```
do Initialize the localization process
do UN communicates with its neighbors
do UN updates its neighbors table
if three different RNs in the table equals No
  set three different locations received to No
  while three different locations received equals No
    set MA arrives to UN's range to No
    while MA arrives to UN's range equals No
      do wait
      if MA arrives to UN's range equals Yes:
    exit while
    do UN exchanges its table with MA
    if three different locations received equals Yes:
  exit while
do UN estimate its location, UN turns to RN
```

2) Procedure on MA Side

The following steps are performed in the MA side:

1) MA initiate its movement from a starting point, which can be either randomly chosen or set in advance.
2) MA will be given the maximum distance of movement.
3) The first three movements of MA will be chosen randomly, in any direction. However, it must to be only on the guide points.
4) With each movement, MA stops and contacts all nodes located within communication range to provide its current coordinates.
5) Every movement after that will be made only based on the applied optimization model
6) After each movement, MA will update its localization table that is similar to the neighboring table of each UN.
7) MA will terminate its movement once the maximum movement distance is reached.

More details about the procedure are shown in the following algorithm.

Figure 4 shows an example of the MA movement when both (GWPP) and (WOPP) models are applied. The initial setting are intended to be the same, which include the random starting point, the three first movements, the same obstacles locations, and the same nodes distribution. Figure 4.**a** presents the MA path planning when GWO is used, while Figure 4.**b** shows the movement of MA when WOA is applied. Note how each optimization model works differently and makes its distinctive path although all network settings are the same.

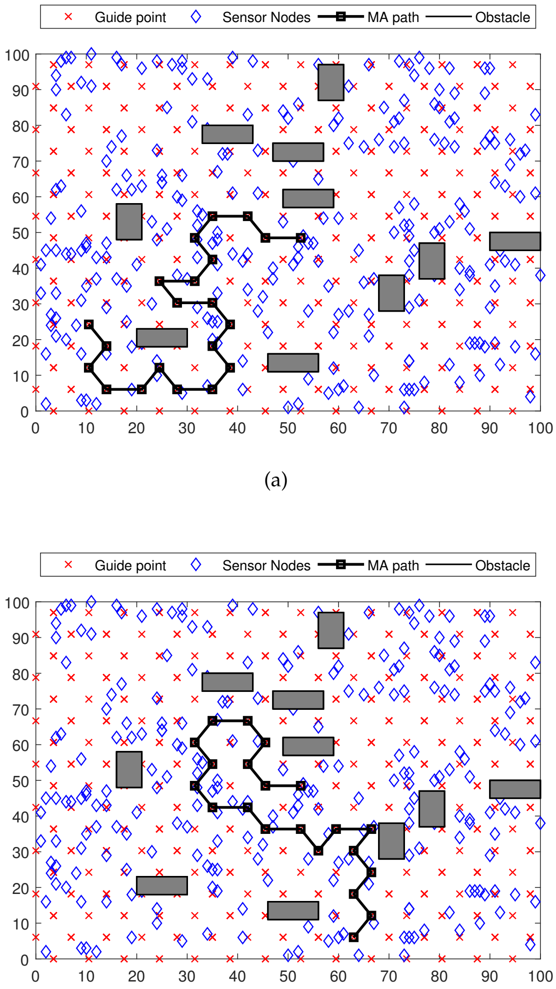
Algorithm 4 Pseudo code of the movement and node localization process in the MA in GWPP or WOPP

```
do Initialize the movement process
set three random movements reached to No
set maximum distance reached to No
while three random movements reached equals No, do
  do MA moves randomly
  do MA stops, provides nodes within its range with its current location
  set three random movements reached to Yes
  while maximum distance reached equals No, do
    do Update routing table
    if three random movements reached equals No
        exit while
    end if
    if three random movements reached equals Yes
      do Evaluate all candidate notes within range
      do Identify collinear node
      do Identify previously visited nodes
      do Run GWO algorithm as outlined in Algorithm 1, or WOA algorithm as outlined in Algorithm 2
      do MA moves to selected point
      if maximum distance reached equals Yes
        exit while
      end if
    end if
  end while
end while
```

VI. Performance Settings

To evaluate the performance of our proposed models, we implemented them along with two obstacle-avoidance models, Z-Curves and Snake-Like. For a better assessment, we used two localization algorithms, namely Weighted Centroid Localization (WCL) [44] and Weight-Compensated Weighted Centroid Localization (WCWCL) [45]. applying two localization models is aimed to provide more comprehensive and objective evaluation. WCL is a simple localization algorithm. Because of its relatively low communication consumption, WCL is known for its energy efficient localization estimation. The key idea in estimating localization information is based on the weights between the anchors and their estimated distance. WCL has been used as the main localization method in multiple path planning models. WCWCL is an improved version of WCL that extends the estimation procedure by giving more impact for the close anchors on the weights, which therefore improves the localization accuracy. Figure 5 shows a magnified example of both localization models of WCL in 5.**a**, and WCWCL in 5.**b** with the same setting in one of the optimizations based movement. A small number of nodes has been used for a better representation.

Matlab was used to model the proposed framework

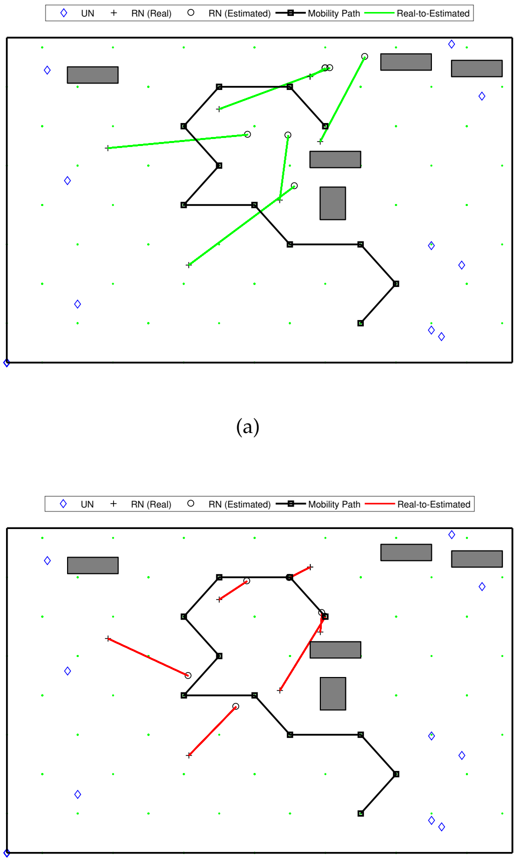


(a)

(b)

Figure 4: The obstacle-avoidance path planning of the two models in (**a**) GWPP, and (**b**) WOPP.

and for performance evaluation of GWPP, WOPP and the other two models. The evaluation has been done in terms of five performance metrics: localization accuracy, localization precision, localization ratio and coverage, and the computation time of the two proposed models. The simulation environment and parameters used were selected for consistency with some other similar works. The network area is assumed to be square with a size, S, of 100 × 100 m. A set of 250 randomly distributed nodes, N, is used with a single mobile anchor, M. Various maximum movements, d_{max}, are applied. The value of d_{max} indicates the maximum distance that MA can take before its ends its journey. In this work, the starting point in all runs is chosen randomly. Chipcon CC1100 radio module [46] specifications are employed as realistic simulation parameters. Such characteristics were already employed in similar proposals including [8], [14], [45]. For simplicity, we round the value of d_p to the nearest integer number. The rest of the parameters are presented in Table I.



(a)

(b)

Figure 5: Example of the localization estimation when the same nodes deployment and the same MA movement are used in (**a**) WCL, and (**b**) WCWCL

VII. Evaluation and Results

To assess the performance of the proposed models, we tested the following metrics, accuracy, precision, and localization ratio from two aspects: the impact of maximum movement distance and the impact of resolution. We also evaluated the computation time for the two algorithms of GWPP and WOPP.

A. Localization Accuracy

To analyze the behavior of the planned models in comparison to the others when different maximum movement distances (d_{max}) are applied, we first performed a test of 50 simulation runs with 250 UNs, 12.5 m of R_{Tx} that is equivalent to R = 1 , and a standard deviation of noise (σ) of 3. The rest of parameters are fixed as shown in Table I.

In path planning for mobility-assisted localization in WSNs, localization accuracy is one of the most important performance metrics. Higher accuracy gives

Table I: Simulation values and parameters.

Parameters	Symbol	Value
Network size (m)	S	100 × 100
Number of mobile anchors	M	1
Number of unknown nodes	N	250
Maximum movement distance (m)	d_{max}	35, 70, 105, 140, 175
Communication range (m)	R_{Tx}	12.5
Resolution	R	0.5, 0.75, 1, 1.5, 1.75, 2
Path loss exponent	β	3.5
Power loss (dB) at d_0	$PL(d_0)$	-60
Reference point (m)	d_0	1
Standard deviation of noise	σ	3
Number of obstacles	O	10
Obstacles dimensions (m)	O_{size}	5 × 10
Maximum number of iterations for GWO and WOA	T_{max}	50-300
Simulation run		50

more confidence about the localization estimation of one model over another. Hence, it is considered as the main factor in many works in this research field. Accuracy is computed using the accomplished localization error. The lower the estimation error, the higher the localization accuracy is. In this work, we assess the accuracy of localization in terms of the average localization error with standard deviation of the all nodes in all implemented models.

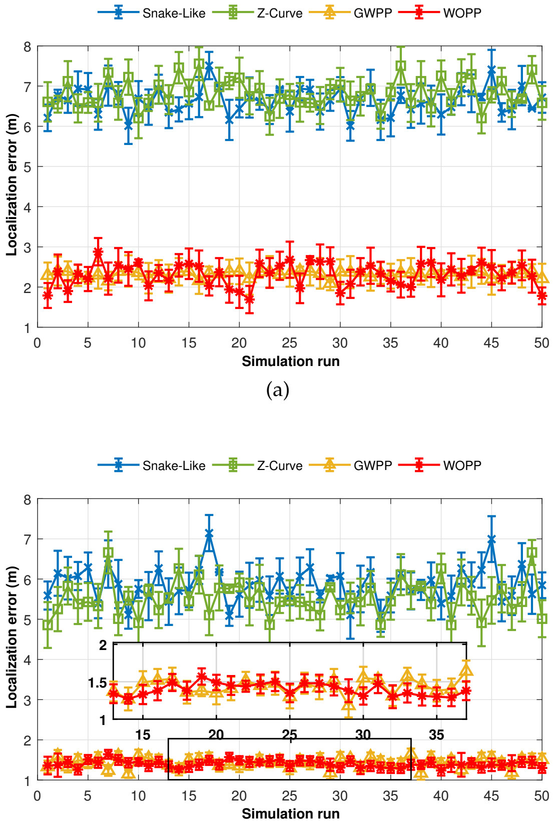
As discussed in Eq. 24, the localization error, $error_i$, indicates the distance between the successfully estimated node and its real location. Here, the average localization error, $error_{avg}$, considers the entire set of unknown nodes, N, and is calculated as:

$$error_{avg} = \Big(\sum_{i=1}^{N} error_{(i)}\Big)/RN \tag{26}$$

Where $error_i$ is the localization error for a localized node i calculated using Eq. 24, and RN is the total number of localized nodes, represented as reference nodes. In addition to evaluating the localization error, we also evaluate the standard deviation of the localization error. A low standard deviation in localization error is desired, since it means a high percentage of error values are close to the mean of all errors. The standard deviation of the localization error for the entire population is calculated as

$$error_{std} = \sqrt{\frac{\sum_{i=1}^{N}(error_{(i)} - error_{avg})^2}{RN}} \tag{27}$$

Where N is the total number of localized nodes, $error_i$ is the node i's localization error, and $error_{avg}$ is the average localization error.



(a)

(b)

Figure 6: Localization errors of all mobility models in (**a**) WCL, and (**b**) WCWCL, ($d_{max} = 140, R = 1$).

However, to analyze the behavior of the proposed models in comparison to the others, we first performed a test of 50 simulation runs with 250 UNs, 140 m of d_{max}, and R of 1. The results in Figure 6 show the performance of all models according to their localization error when the two localization algorithms of WCL and WCWCL are applied.

When WCL is used, as shown in Figure 6.**a**, both of our proposed models offer superior accuracy with the lowest error rate in all of the results shown. In most of the 50 run cases, both models provide a high accuracy with less than 3 m of error. Based on the deployment of nodes and location of obstacles, the results from both models vary over the simulation runs. Both GWPP and WOPP show similar performance during the different simulation runs. Most run results show values less than 1.5 m of standard deviations. On the other hand, Z-Curves and Snake-Like models provide unstable performance. One reason behind this behavior is the inability of the static models to dynamically change their planned path. In case of facing an obstacle in their way, Z-Curves and Snake-Like models will only avoid the obstacle and then continue on the statically planned path, which will affect the performance of the models and may create a collinearity problem.

Indeed, the Snake-Like model does not present a solution to deal with the collinearity problem. This is clear in their performance of accuracy where most runs achieve results around 6.5-7.5 m of error rate. In fact, some results are close to 8 m of error rate depending on the network topology. On the other hand, when WCWCL is used as in Figure 6.**b**, improved outcomes for all models are achieved. However, this improvement is more noticed in both of GWPP and WOPP as they accomplished high accuracy. The localization accuracy is improved in both Z-Curves and Snake-Like models. We start to notice some results around 5 m of localization error in Z-Curves. In general, both models performs better in WCWCL than in WCL.

1) The Impact of Maximum Movement Distance

Then, the average localization error for all movement models when different maximum distances (d_{max}) are applied, is calculated. For a better evaluation, we performed different experiments with changeable values of the maximum distance of movement (d_{max}). The values of d_{max} range from 35 m to 175 m. The area consists of 250 UNs, 10 obstacles of 5 × 10 m each. The resolution value (R) of 1 is assumed. Figure 7 presents the performance of the path models when WCL and WCWCL are used.

Figure 7.**a** shows the average localization error and the corresponding d_{max} for all models when WCL is used, while Figure 7.**b** shows the average localization error when WCWCL is used. Five different d_{max} values of 35 m, 70 m, 105 m, 140 m, 175 m are applied. With both localization algorithms, our proposed optimized path models of GWPP and WOPP offer higher accuracy. Indeed, it is noticed that when d_{max} increases, the localization error decreases. In general, and with both WCL and WCWCL, WOPP offers better performance than GWPP. However, the difference between their performances is small. The main reason that WOPP provides better results than GWPP is that WOPP calculates the next visiting point based on satisfying a fitness function, whereas the next visiting point in GWPP is based on the mean of the three best candidate points. Also, WOPP makes a random movement when an optimum next visiting point is not found. On the other hand, Z-Curves and Snake-Like provide a lower performance of accuracy in comparison to GWPP and WOPP in both WCL and WCWCL localization. However, all path planning models performs better with WCWCL. The localization error for Z-Curves and Snake-Like using both WCL and WCWCL decreases with increasing maximum movement, d_{max}, as shown in Figue 7.**a**. However, their localization error using WCL is more than 6.5 m when d_{max} is 175 m, more than twice that of our proposed models GWPP and WOPP. In WCWCL, as presented in 7.**b**, Z-Curves and Snake-Like become better and the error rate decreases to close to 5.5 m in some cases. In general, Z-Curves performs better when WCWCL is used. Z-Curves takes into account the collinearity problem in its design while Snake-Like does not. GWPP and WOPP have high accuracy with WCWCL with only approximately 2.32 to 1.4 m of error when d_{max} increases from 35 m to other distances.

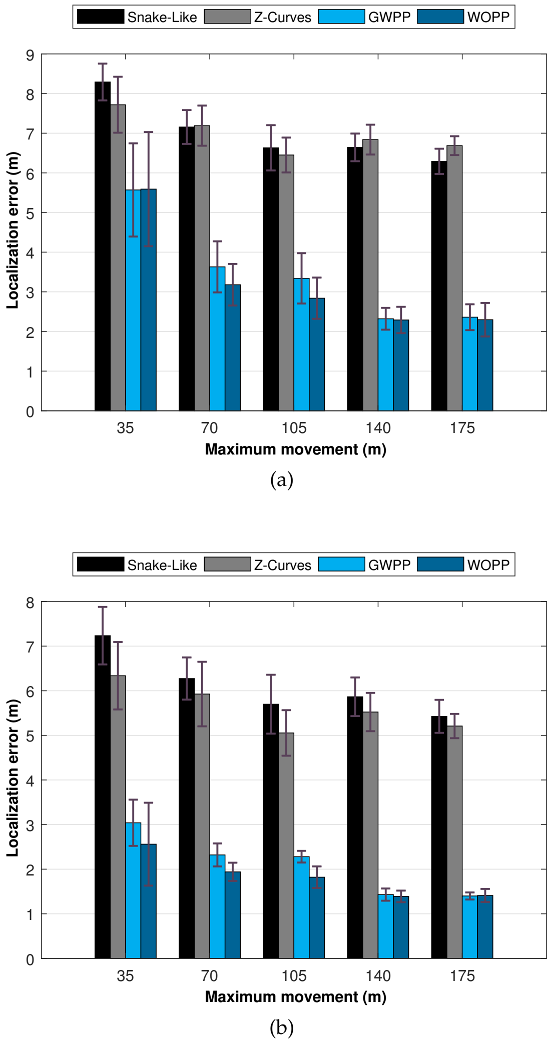


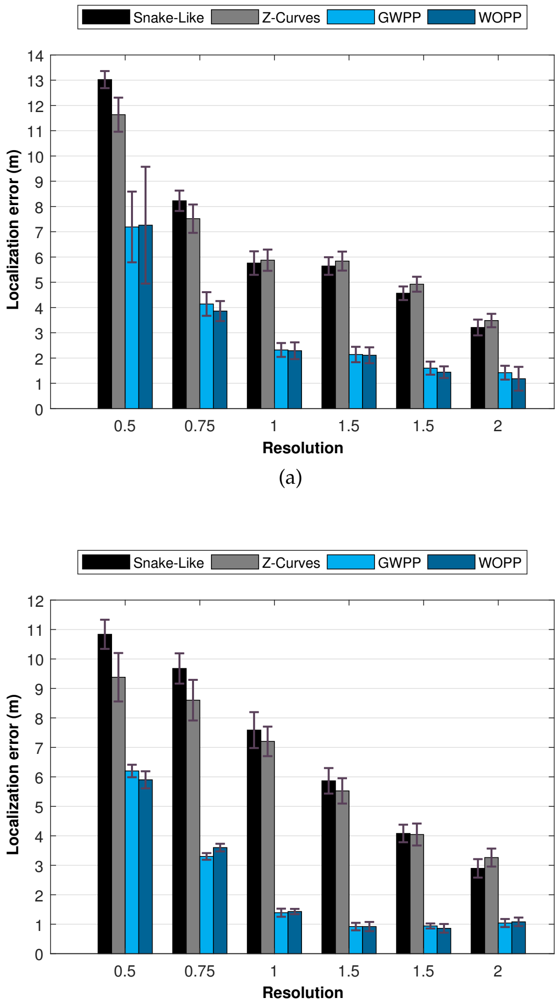
Figure 7: Average localization errors versus maximum movement in (**a**) WCL, and (**b**) WCWCL.

2) The Impact of Resolution

In addition to evaluating the localization error when different d_{max} values are applied, we also evaluate the impact of resolution R on the localization error. The resolution values R refer to the relationship between the communication range, R_{Tx}, and the distance between every two points in each static path planning model. It has been used in some other similar works including [14], [17], [21]. It is formulated as:

$$R = \frac{d_p}{R_{Tx}} \tag{28}$$

For fair comparisons, we applied the same communication range and resolution values to all models including GWPP and WOPP. Reference R values of 0.5, 0.75, 1, 1.5, 1.75, and 2 are used. For example, if the communication range of the MA and UNs is a when R = 0.5, the distance between every two points d_p will be 0.5 × a and so



(a)

(b)

Figure 8: Average localization errors and resolution in (**a**) WCL, and (**b**) WCWCL.

on. Figure 8 shows the average localization error with different R when R_{Tx}=12.5, and d_{max}=140 m.

In the first test where WCL is used, as in Figure 8 .**a**, the accuracy of all model is substantially decreased when small resolution values of 0.5 and 0.75 are used. This is applicable to all models. In fact, error values of 13 m and 11 m are shown in Snake-Like and Z-Curves respectively. Even in our models, localization error of 7 m are presented when the same resolution of 0.5 is used. However, in all models, increasing R decreases the error. Very high accuracy is achieved when R of 2 is used, with less than 2 m of error in GWPP and WOPP. The same concept is shown also when WCWCL is used as presented in Figure 8 .**b**. However, the results in general are better than these in WCL. Indeed, we can attain an acceptable accuracy around 1 m when R is 1.

B. Precision

The location of each node in a network has an accuracy determined by the localization error, which is the distance between the actual location of the node and its location as calculated by the localization algorithm. The proportion of localization errors smaller than a certain threshold error value is known as localization precision. For example, if 80% of the nodes have a localization error of less than 3 m, the precision is 0.8 at < 3 m, which we could write P_3 =0.8.

Precision can be formulated as

$$P_k = \frac{\sum_{i=1}^{N}(b_i)}{RN} \quad \begin{cases} b_i = 1, & \text{if } LE(i) <= k \\ 0, & \text{otherwise} \end{cases} \tag{29}$$

Where P_k is the precision values achived under the k threshold of distance in m, LE is localization error, and RN is the set of all localized nodes in the network.

We considered five precision values as follows: less than 1 m, less than 3 m, less than 5 m, less than 7 m, and less than 9 m. Each localization precision is the mean of 50 different simulation runs, when d_{max} = 140 m, and R = 1. Figure 9 shows the precision evaluation usign WCL and WCWCL.

C. Localization Ratio

The localization ratio, or coverage, indicates the number of localized nodes (reference nodes) divided by the total number of nodes. High localization ratio gives an impression of how successful the path planning is. The localization ratio is represented as

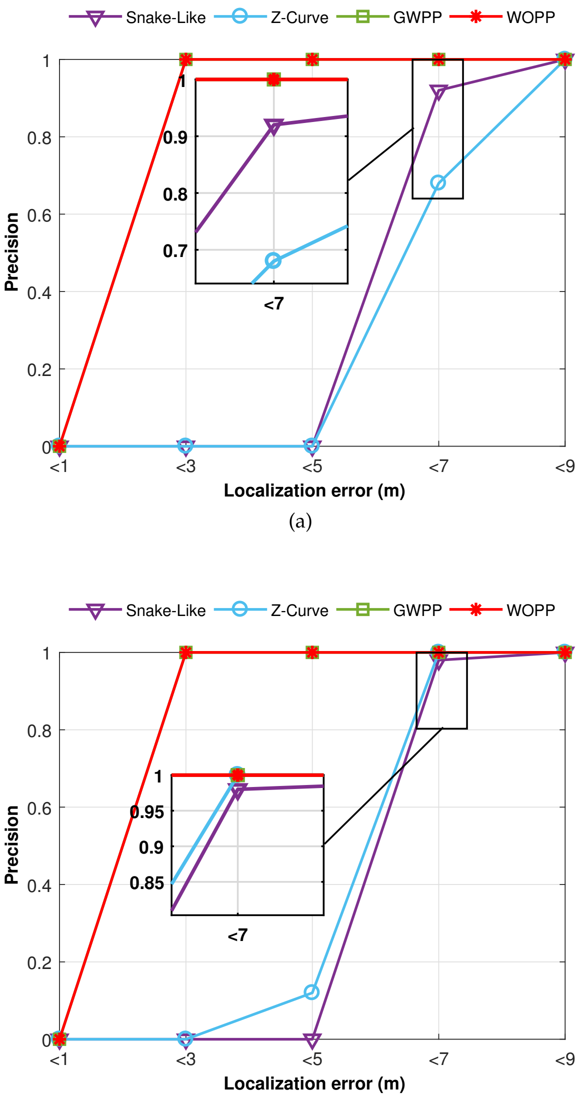
$$L_{avg} = \frac{RN}{N} \tag{30}$$

Where RN is the total number of reference nodes, and N is the total number of deployed nodes. Here, we evaluate the localization ratio from two prespectives, the impact of maximum movement distance d_{max}, and the impact of resolution value R.

1) The Impact of Maximum Movement Distance

A collection of 250 sensor nodes is used with various d_{max}, and R of 1. In this metric, only one of the localization algorithms is shown, since their localization ratios are similar. Figure 10 show the localization ratio for the four implemented movement models.

Unlike the static path planning models, dynamic models do not guarantee that all nodes inside the network will be able to receive the localization information. In addition, since there is a limited distance of MA movement, it is difficult to cover the entire area with such assumption. However, both of our dynamic models, GWPP and WOPP, offer higher localization ratio in most cases in comparison to other two models. All four models provide weak localization ratios when d_{max} is short as 35 m. This ratio get improves with the increase of d_{max}. When the d_{max} is 175 m, the proposed models can achieve about 35 percentage of localized nodes. Snake-Like and Z-Curves can acquire less than 28 percentage in its best case.



(a)

(b)

Figure 9: Precision of all mobility models versus the localization error in (**a**) WCL, and (**b**) WCWCL, ($d_{max} = 140, R = 1$)

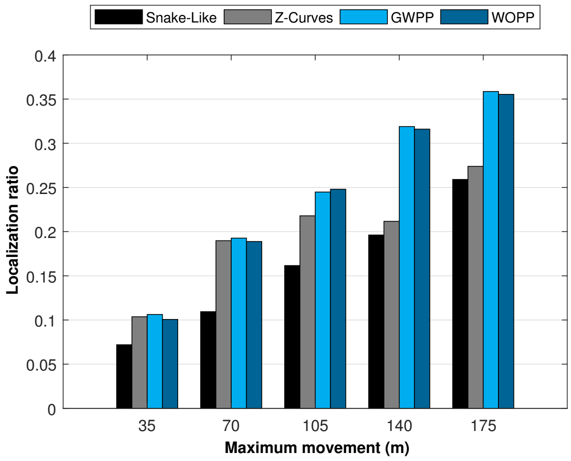


Figure 10: Localization ratio versus the maximum movement distance of all mobility models in both WCL and WCWCL

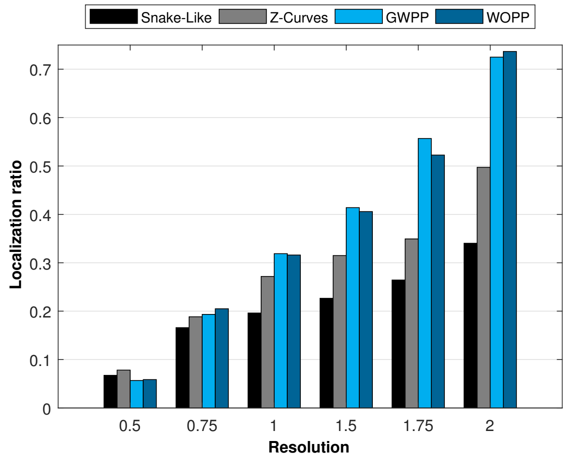


Figure 11: Localization ratio versus the resolution of all mobility models in both WCL and WCWCL

2) The Impact of Resolution

In the other experiment, we assessed the localization ratio in terms of different resolution values. We used similar assumptions of those in the former experiment presented in VII-C1, except with a fixed d_{max} of 140 m, and changable R. The results are plotted in Figure 11.

When a resolution value R of 0.5 is used, the localization ratio of all models is poor. More nodes are covered when R increases, thus, more UNs will be able to estimate their locations. When R is 1.5, more than 40 percentage of nodes are localized in GWPP and WOPP comparison to less than 35 percentage when R = 1. The ratio increases gradually when R = 1.75. Both GWPP and WOPP can achieve high results of more than 72 percentage of localized nodes when R = 2, while Z-Curves can cover about 50 percentage of its UNs when the same assumption is applied. Snake-Like still performs poorly with all R values since it is affected by the collinearity problem.

D. Computation Time

We also extended our performance metrics evaluation to compare the computation time for the proposed models. Computation time here refers to the time spent from the first execution of the code to the end of it. The computation time is measured in seconds (s). A comparison of 50 runs with different maximum iteration times (T_{max}) for both models are conducted. The average of every 50 runs of T_{max} is taken. The following parameters were used for the performance, 250 UNs, d_{max} = 140, R = 1, and the rest are fixed as shown in Table I. The executed computation time is shown in Figure 12.

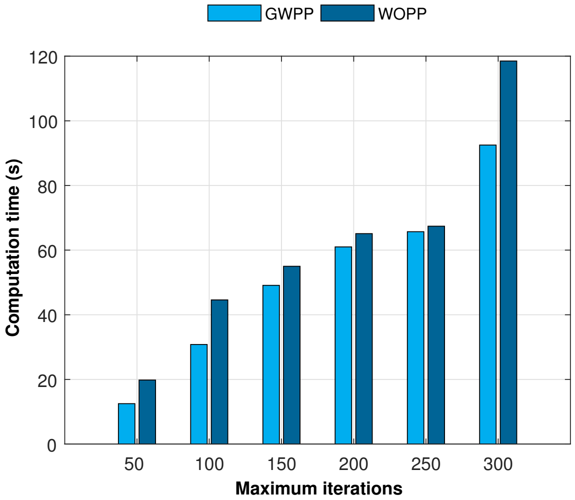


Figure 12: Average computation time for both techniques in seconds

The result of computation time shows that GWPP takes less time to run in comparison to WOPP regardless of the value of T_{max}. When T_{max} increase the computation time increases. GWPP spends only 12.5 seconds when a T_{max} of 50 is used, while WOPP takes 19.8 seconds for the same T_{max}. The time spent on execution of both models increases gradually. However, even in its longest period, GWPP and WOPP need 92.5 seconds and 118.5 seconds respectively when T_{max} is 300.

VIII. Discussion

As seen above, the dynamic proposed models and their optimized base help to improve their performance in many metrics. Both models provide better accuracy compared to the other two models. The flexibility of the MA moving based on the network parameters and node locations improves such metric. Unlike the other models, GWPP and WOPP can avoid obstacles, avoid collinearity, avoid visited points, and make their own path in real-time based on information from the network. In addition, all movements will depend on the fitness function. Moreover, GWPP and WOPP have high precision. Our proposed dynamic models GWPP and WOPP provide better localization ratio than the static models Snake-Like and Z-Curves, even in the presence of obstacles and with limited movement distance. Indeed, when high resolution values are considered, both models get competitive accuracy. In general, WOPP has a better performance than GWPP. However, GWPP has a shorter run computational time. Thus, choosing which model to apply can be based on these metrics. If time is an important constraint, GWPP may be preferable. Alternatively, WOPP may be the better candidate if high performance is required.

IX. Conclusion and Future Works

In this work, we introduced two dynamic obstacle-avoidance path planning models, called GWPP and WOPP, for mobile anchor-assisted localization in WSNs. The proposed models work on optimizing the path design based on the real-time information from the network. The optimization models help not only to avoid the obstacles located in the MA's way but also to design an outstanding optimized path when the MA has a limited movement. To examine the efficiency of the proposed models, we compare them to other two models that consider obstacle-avoidance methods in their design. The final results demonstrate that our proposed models, GWPP and WOPP, maximize the localization efficiency in the aspects of accuracy, localization precision, and localization ratio (coverage) when they are tested from two aspects, the maximum movement distance and the resolution values. We also compared the proposed models in terms of computation time in order to study them from different aspects. We were able to summarize the outcomes of this paper based on the four metrics analyzed as follow:

1) **Localization accuracy :** indicated as localization error, the results show that our proposed models offer superior outcomes in both experiments as shown in Figures 7, 8.
2) **Localization precision :** The dynamic models of GWPP and WOPP present the best outcomes of precision in WCL and WCWCL as presented in Figure 9.
3) **Localization ratio**: Typically, static path planning models provide better performance than the other kinds of mobility in terms of coverage. However, when the MA has a limited and constrained movement distance, and there are some obstacles in the area, the static models performance is affected. GWPP and WOPP consider the network and node distribution, and take them into account when they design their path. The results show that they still provide competitive performance as shown in Figure 10. Higher localization ratio can be obtained when the resolution values increase as shown in Figure 11.
4) **Computation time :** To make the final decision of choosing either of GWPP or WOPP, we also analyzed the computation time. The results show that GWPP has shorter computation time than WOPP as shown in Figure 12, which may be beneficial when a faster optimized model required.

To sum up, we have shown that employing an optimization model for forming a movement path leads to optimal outcomes in a number of metrics. As future works, we may consider a multi-objective optimization model that forms the optimal path based on the localization error and localization ratio as well. Also, we may assume irregular obstacle shapes and let the MA to control its movement based on this assumption. Three-dimensional movement pattern may also be considered.

Acknowledgment

Abdullah Alomari is thankful to Albaha University and the Saudi Arabian Cultural Bureau in Canada for the generous financial support during his studies.

References

[1] Z. Fei, B. Li, S. Yang, C. Xing, H. Chen, and L. Hanzo, "A survey of multi-objective optimization in wireless sensor networks: Metrics, algorithms, and open problems," *IEEE Communications Surveys Tutorials*, vol. 19, no. 1, pp. 550–586, Firstquarter 2017.

[2] L. M. Borges, F. J. Velez, and A. S. Lebres, "Survey on the characterization and classification of wireless sensor network applications," *IEEE Communications Surveys Tutorials*, vol. 16, no. 4, pp. 1860–1890, Fourthquarter 2014.

[3] H. Karl and A. Willig, *Protocols and architectures for wireless sensor networks*. Hoboken, NJ: Wiley, May 2005.

[4] A. B. Noel, A. Abdaoui, T. Elfouly, M. H. Ahmed, A. Badawy, and M. S. Shehata, "Structural health monitoring using wireless sensor networks: A comprehensive survey," *IEEE Communications Surveys Tutorials*, vol. 19, no. 3, pp. 1403–1423, third quarter 2017.

[5] D. N. Sandeep and V. Kumar, "Review on clustering, coverage and connectivity in underwater wireless sensor networks: A communication techniques perspective," *IEEE Access*, vol. 5, pp. 11 176–11 199, 2017.

[6] A. M. Abu-Mahfouz and G. P. Hancke, "Alwadha localization algorithm: Yet more energy efficient," *IEEE Access*, vol. 5, pp. 6661–6667, 2017.

[7] Y. S. Chen, D. J. Deng, and C. C. Teng, "Range-based localization algorithm for next generation wireless networks using radical centers," *IEEE Access*, vol. 4, pp. 2139–2153, 2016.

[8] A. Alomari, W. Phillips, N. Aslam, and F. Comeau, "Dynamic fuzzy-logic based path planning for mobility-assisted localization in wireless sensor networks," *Sensors*, vol. 17, no. 8, 2017.

[9] S. Halder and A. Ghosal, "A survey on mobile anchor assisted localization techniques in wireless sensor networks," *Wireless Networks*, vol. 22, no. 7, pp. 2317–2336, 2016. [Online]. Available: http://dx.doi.org/10.1007/s11276-015-1101-2

[10] N. A. Alrajeh, M. Bashir, and B. Shams, "Localization techniques in wireless sensor networks," *International Journal of Distributed Sensor Networks*, vol. 9, no. 6, p. 304628, 2013. [Online]. Available: https://doi.org/10.1155/2013/304628

[11] G. Han, J. Jiang, C. Zhang, T. Q. Duong, M. Guizani, and G. K. Karagiannidis, "A survey on mobile anchor node assisted localization in wireless sensor networks," *IEEE Communications Surveys Tutorials*, vol. 18, no. 3, pp. 2220–2243, thirdquarter 2016.

[12] S. Mirjalili, S. M. Mirjalili, and A. Lewis, "Grey wolf optimizer," *Advances in Engineering Software*, vol. 69, pp. 46 – 61, 2014. [Online]. Available: http://www.sciencedirect.com/science/article/pii/S0965997813001853

[13] S. Mirjalili and A. Lewis, "The whale optimization algorithm," *Advances in Engineering Software*, vol. 95, pp. 51 – 67, 2016. [Online]. Available: http://www.sciencedirect.com/science/article/pii/S0965997816300163

[14] A. Alomari, F. Comeau, W. Phillips, and N. Aslam, "New path planning model for mobile anchor-assisted localization in wireless sensor networks," *Wireless Networks*, pp. 1–19, 2017. [Online]. Available: http://dx.doi.org/10.1007/s11276-017-1493-2

[15] D. Koutsonikolas, S. M. Das, and Y. C. Hu, "Path planning of mobile landmarks for localization in wireless sensor networks," *Comput. Commun.*, vol. 30, no. 13, pp. 2577–2592, Sep. 2007. [Online]. Available: http://dx.doi.org/10.1016/j.comcom.2007.05.048

[16] R. Huang and G. V. Zaruba, "Static path planning for mobile beacons to localize sensor networks," in *Pervasive Computing and Communications Workshops, 2007. PerCom Workshops '07. Fifth Annual IEEE International Conference on*, March 2007, pp. 323–330.

[17] G. Han, C. Zhang, J. Lloret, L. Shu, and J. Rodrigues, "A mobile anchor assisted localization algorithm based on regular hexagon in wireless sensor networks," *The Scientific World Journal*, pp. 1–13, 2014.

[18] G. Han, H. Xu, J. Jiang, L. Shu, T. Hara, and S. Nishio, "Path planning using a mobile anchor node based on trilateration in wireless sensor networks," *Wireless Communications and Mobile Computing*, vol. 13, no. 14, pp. 1324–1336, 2013. [Online]. Available: http://dx.doi.org/10.1002/wcm.1192

[19] A. Alomari, N. Aslam, W. Phillips, and F. Comeau, "Three-dimensional path planning model for mobile anchor-assisted localization in wireless sensor networks," in *2017 IEEE 30th Canadian Conference on Electrical and Computer Engineering (CCECE)*, April 2017, pp. 1–5.

[20] Y. Chen, S. Lu, J. Chen, and T. Ren, "Node localization algorithm of wireless sensor networks with mobile beacon node," *Peer-to-Peer Networking and Applications*, vol. 10, no. 3, pp. 795–807, 2017. [Online]. Available: http://dx.doi.org/10.1007/s12083-016-0522-8

[21] J. Rezazadeh, M. Moradi, A. Ismail, and E. Dutkiewicz, "Superior path planning mechanism for mobile beacon-assisted localization in wireless sensor networks," *Sensors Journal, IEEE*, vol. 14, no. 9, pp. 3052–3064, Sept 2014.

[22] S. M. Mazinani and F. Farnia, "Localization in wireless sensor network using a mobile anchor in obstacle environment," *International Journal of Computer and Communication Engineering*, vol. 2, no. 4, p. 438, 2013.

[23] L. Bianchi, M. Dorigo, L. M. Gambardella, and W. J. Gutjahr, "A survey on metaheuristics for stochastic combinatorial optimization," *Natural Computing*, vol. 8, no. 2, pp. 239–287, Jun 2009. [Online]. Available: https://doi.org/10.1007/s11047-008-9098-4

[24] J. H. Holland, "Genetic algorithms," *Scientific American*, vol. 267, no. 1, pp. 66–73, 1992. [Online]. Available: http://www.jstor.org/stable/24939139

[25] S. Kirkpatrick, C. Gelatt Jr., and M. Vecchi, "Optimization by simulated annealing," *Science*, vol. 220, no. 4598, pp. 671–680, 1983, cited By 21902. [Online]. Available: https://www.scopus.com/inward/record.uri?eid=2-s2.0-26444479778&partnerID=40&md5=d90f6f0271cad18bc63145c671398950

[26] J. Kennedy and R. Eberhart, "Particle swarm optimization," in *Neural Networks, 1995. Proceedings., IEEE International Conference on*, vol. 4, Nov 1995, pp. 1942–1948 vol.4.

[27] M. Dorigo and T. Stützle, *Ant Colony Optimization*. Scituate, MA, USA: Bradford Company, 2004.

[28] D. Karaboga and B. Basturk, "A powerful and efficient algorithm for numerical function optimization: artificial bee colony (abc) algorithm," *Journal of Global Optimization*, vol. 39, no. 3, pp. 459–471, Nov 2007. [Online]. Available: https://doi.org/10.1007/s10898-007-9149-x

[29] G. Abbas, J. Gu, U. Farooq, M. U. Asad, and M. El-Hawary, "Solution of an economic dispatch problem through particle swarm optimization: A detailed survey - part i," *IEEE Access*, vol. 5, pp. 15 105–15 141, 2017.

[30] M. Saleem, G. A. D. Caro, and M. Farooq, "Swarm intelligence based routing protocol for wireless sensor networks: Survey and future directions," *Information Sciences*, vol. 181, no. 20, pp. 4597 – 4624, 2011, special Issue on Interpretable Fuzzy Systems. [Online]. Available: http://www.sciencedirect.com/science/article/pii/S0020025510003233

[31] A. M. Zungeru, L.-M. Ang, and K. P. Seng, "Classical and swarm intelligence based routing protocols for wireless sensor networks: A survey and comparison," *Journal of Network and Computer Applications*, vol. 35, no. 5, pp. 1508 – 1536, 2012, service Delivery Management in Broadband Networks. [Online]. Available: http://www.sciencedirect.com/science/article/pii/S1084804512000719

[32] R. Kumar and D. Kumar, "Hybrid swarm intelligence energy efficient clustered routing algorithm for wireless sensor networks," *Journal of Sensors*, vol. 2016, pp. 1 – 19, 2016. [Online]. Available: http://dx.doi.org/10.1155/2016/5836913

[33] P. C. S. Rao, P. K. Jana, and H. Banka, "A particle swarm optimization based energy efficient cluster head selection algorithm for wireless sensor networks," *Wireless Networks*, Apr 2016. [Online]. Available: https://doi.org/10.1007/s11276-016-1270-7

[34] Y. Zhou, N. Wang, and W. Xiang, "Clustering hierarchy protocol in wireless sensor networks using an improved pso algorithm," *IEEE Access*, vol. 5, pp. 2241–2253, 2017.

[35] D. S. Deif and Y. Gadallah, "An ant colony optimization approach for the deployment of reliable wireless sensor networks," *IEEE Access*, vol. 5, pp. 10 744–10 756, 2017.

[36] D. Lavanya and S. K. Udgata, *Swarm Intelligence Based Localization in Wireless Sensor Networks*. Berlin, Heidelberg: Springer Berlin Heidelberg, 2011, pp. 317–328. [Online]. Available: https://doi.org/10.1007/978-3-642-25725-4_28

[37] Z. Sun, L. Tao, X. Wang, and Z. Zhou, "Localization algorithm in wireless sensor networks based on multiobjective particle swarm optimization," *International Journal of Distributed Sensor*

Networks, vol. 11, no. 8, p. 716291, 2015. [Online]. Available: https://doi.org/10.1155/2015/716291

[38] Y. Zhang, J. Liang, S. Jiang, and W. Chen, "A localization method for underwater wireless sensor networks based on mobility prediction and particle swarm optimization algorithms," *Sensors*, vol. 16, no. 2, 2016.

[39] M. M. Fouad, A. I. Hafez, A. E. Hassanien, and V. Snasel, "Grey wolves optimizer-based localization approach in wsns," in *2015 11th International Computer Engineering Conference (ICENCO)*, Dec 2015, pp. 256–260.

[40] N. A. Al-Aboody and H. S. Al-Raweshidy, "Grey wolf optimization-based energy-efficient routing protocol for heterogeneous wireless sensor networks," in *2016 4th International Symposium on Computational and Business Intelligence (ISCBI)*, Sept 2016, pp. 101–107.

[41] M. Sharawi and E. Emary, "Impact of grey wolf optimization on wsn cluster formation and lifetime expansion," in *2017 Ninth International Conference on Advanced Computational Intelligence (ICACI)*, Feb 2017, pp. 157–162.

[42] M. M. Ahmed, E. H. Houssein, A. E. Hassanien, A. Taha, and E. Hassanien, *Maximizing Lifetime of Wireless Sensor Networks Based on Whale Optimization Algorithm.* Cham: Springer International Publishing, 2018, pp. 724–733. [Online]. Available: https://doi.org/10.1007/978-3-319-64861-3_68

[43] R. Ismail, Z. Omar, and S. Suaibun, "Obstacle-avoiding robot with ir and pir motion sensors," *IOP Conference Series: Materials Science and Engineering*, vol. 152, no. 1, p. 012064, 2016. [Online]. Available: http://stacks.iop.org/1757-899X/152/i=1/a=012064

[44] J. Blumenthal, R. Grossmann, F. Golatowski, and D. Timmermann, "Weighted centroid localization in zigbee-based sensor networks," in *Intelligent Signal Processing, 2007. WISP 2007. IEEE International Symposium on*, Oct 2007, pp. 1–6.

[45] Q. Dong and X. Xu, "A novel weighted centroid localization algorithm based on rssi for an outdoor environment, journal of communications," in *Journal of Communications*, Sep 2014, pp. 279–285.

[46] Chipcon, *CC1100, Low-Power Sub- 1 GHz RF Transceiver*, Texas Insturments, 2014.