

Article

# An Elite Hybrid Particle Swarm Optimization for Solving Minimal Exposure Path Problem in Mobile Wireless Sensor Networks

Nguyen Thi My Binh <sup>1,2</sup>, Abdelhamid Mellouk <sup>3,\*</sup> <sup>(D)</sup>, Huynh Thi Thanh Binh <sup>2</sup> <sup>(D)</sup>, Le Vu Loi <sup>2</sup>, Dang Lam San <sup>2</sup> and Tran Hai Anh <sup>2</sup> <sup>(D)</sup>

- <sup>1</sup> The faculty of Information and Technology, Hanoi University of Industry, Hanoi 100000, Vietnam; binhntm@haui.edu.vn
- <sup>2</sup> School of Information Communication Technology, Hanoi University of Science and Technology, Hanoi 100000, Vietnam; binhht@soict.hust.edu.vn (H.T.T.B.); loi9a1999.nqb@gmail.com (L.V.L.); san1201.bkhn@gmail.com (D.L.S.); anhth@soict.hust.edu.vn (T.H.A.)
- <sup>3</sup> Tinc-NET/LiSSi Laboratory, Networks & Telecommunications (N&T) Department, University Paris-Est Creteil (UPEC), F-94400 Vitry, France
- \* Correspondence: mellouk@u-pec.fr

Received: 28 March 2020; Accepted: 27 April 2020; Published: 1 May 2020



Abstract: Mobile wireless sensor networks (MWSNs), a sub-class of wireless sensor networks (WSNs), have recently been a growing concern among the academic community. MWSNs can improve network coverage quality which reflects how well a region of interest is monitored or tracked by sensors. To evaluate the coverage quality of WSNs, we frequently use the minimal exposure path (MEP) in the sensing field as an effective measurement. MEP refers to the worst covered path along which an intruder can go through the sensor network with the lowest possibility of being detected. It is greatly valuable for network designers to recognize the vulnerabilities of WSNs and to make necessary improvements. Most prior studies focused on this problem under a static sensor network, which may suffer from several drawbacks; i.e., failure in sensor position causes coverage holes in the network. This paper investigates the problem of finding the minimal exposure paths in MWSNs (hereinafter MMEP). First, we formulate the MMEP problem. Then the MMEP problem is converted into a numerical functional extreme problem with high dimensionality, non-differentiation and non-linearity. To efficiently cope with these characteristics, we propose HPSO-MMEP algorithm, which is an integration of genetic algorithm into particle swarm optimization. Besides, we also create a variety of custom-made topologies of MWSNs for experimental simulations. The experimental results indicate that HPSO-MMEP is suitable for the converted MMEP problem and performs much better than existing algorithms.

**Keywords:** minimal exposure path; wireless sensor networks; mobile wireless sensor networks; hybrid particle swarm optimization

# 1. Introduction

Wireless sensor networks (WSNs) are nowadays ubiquitous. They are used in various different domains: military, medical, environment, etc. Research on WSNs has tried to optimize sensor network design [1–3]; i.e., coverage quality, network lifetime, sensor placement, communication and computation effort, etc. Coverage is one of the most widely studied problems in WSNs. Coverage of a WSN reflects the quality of service (QoS) provided by that WSN; e.g., a high-quality WSN used in a security system may detect unauthorized penetration with probability higher than 99% [4–6]. Among the studies on the coverage problem, barrier coverage has drawn tremendous attention from



the academic community due to its huge potential in various security applications [7–10]. To evaluate coverage quality of WSN, especially in barrier coverage problems, a well-known method is using exposure as a metric [11–13]. Great effort in studies related to exposure has been made to investigate the minimal exposure path problem. The goal of the minimal exposure path (MEP) problem is to find a penetration path having minimal exposure value from a source point to a destination point in the sensing field. With the knowledge of MEP, the sensor network designers can appraise weaknesses or worst-case coverage paths of a sensor network, since objects moving across the sensing field along this path have the least capacity to be detected. As a result, information of the MEP can be used in optimizing, managing and maintaining WSNs. Measuring exposure is not only useful in the WSN but also in several other fields, such as evaluating the quality of radio signal propagation or manufacturing path-finding robots.

Recently, mobile wireless sensor networks (MWSNs) have been received increasing interest because of a wide variety of potential applications. A MWSN consists of mobile sensor nodes which are equipped with a locomotive unit and can move around after deployment [14–17]. Sensors can be attached to larger machines like mobile robots or can be a self-contained miniature system with the ability to move to desired areas. Such MWSNs are extremely valuable in situations where traditional static wireless sensor network (SWSN) deployment mechanisms fail or are not suitable; for instance, a hostile environment where sensors cannot be manually deployed, and must be air-dropped. MWSNs also play an important role in homeland security. Sensors can be mounted on vehicles (subway trains, taxis, police cars, fire trucks, boats, etc.) or carried by people (policemen, fire fighters, etc.). These sensors accompany with their carriers in every motion, and can monitor and dynamically patrol the environment (for chemicals, biological matter, wildfires or radiological agents). For example, MWSN can be used in wildfire monitoring applications. The mobile sensors are able to maintain a safe distance from the fire perimeter, and provide updating information to fire fighters that indicate where the perimeter currently is. In other application scenarios, such as atmosphere and under-water environment monitoring, airborne or under-water sensors may move with the surrounding air or water currents [18,19]. The coverage of a MWSN now depends not only on the initial network configurations, but also on the mobility behavior of the sensors. Thanks to the mobility of sensor nodes, MWSNs can improve coverage quality, prolong the network lifetime, optimize use of resources and be relocated very efficiently [20–22].

The MEP problem in SWSNs has been intensively studied by the academic community. However, this problem has not yet been efficiently explored and exploited in MWSNs. Motivated by the advantages of MWSNs and the vital role of the MEP problem, this paper investigates the problem of finding the minimal exposure path in a MWSN (hereinafter MMEP). Specifically, given a set of mobile sensor nodes which are randomly deployed in the region of interest (ROI), the goal of the MMEP problem is to find a crossing path from a source point to a destination point such that an intruder moving along this path has the lowest possibility of being detected, meaning that this path has the minimal exposure value. The MEP problem is an optimization problem. Usually, it is converted into numerical functional extreme problem [23]. Due to high dimensionality and non-linearity of the objective function, and the mobility feature of sensors, the MMEP problem is much more challenging than the traditional MEP problem in SWSNs. Hence, we propose to integrate genetic algorithm into particle swarm optimization to form an efficient algorithm named HPSO-MMEP algorithm.

The main contributions of this paper are as follows:

- Formulating the MMEP problem in MWSNs with several different sensing coverage models.
- Devising an elite hybrid particle swarm optimization algorithm, called HPSO-MMEP, which integrates genetic operators into PSO algorithm to significantly improve the performance of original PSO.
- Designing a new individual representation and proposing a strategy for generating individuals called controlled-point (CP) initialization which ensures the diversity of the generated population.

• Conducting numerous experiments in various scenarios to evaluate the performance of HPSO-MMEP. The experimental results are thoroughly analyzed to give insights into the affects of different factors.

The rest of this paper is organized as follows. Related works regarding the MEP problem are discussed in Section 2. Section 3 provides some preliminaries, the formulation of the MMEP problem and the proposed algorithm HPSO-MMEP. Experimental results are considered in Section 4. Finally, Section 5 presents conclusions and future works of the paper.

#### 2. Related Works

This section provides the big picture of literature regarding the MEP problem. Different MEP models dependent on various factors, such as types of sensors, deployment strategies, deployment environments and solution approaches, are briefly reviewed.

Regarding SWSNs, a number of studies have focused on the MEP problem with several different approaches: computational geography, grid-based and heuristic/metaheuristic. Using Voronoi diagram has been the most widely used method for the computational geometry approach. Voronoi diagram-based methods and grid-based methods were the earliest methods to address the MEP problem. In [24], Meguerdichian et al. was the first to devise the concept "exposure", and contended that finding the MEP in WSNs under arbitrary sensor and intensity models is very meaningful for network designers but is an extremely difficult for optimization tasks. The authors found the closed-form solution in the case of polygonal region with a single sensor under some constraints. In the generic case, where the sensing field consists of multiple sensors, the grid-based method was applied. The MEP problem was then focused on by the academic community [12,24–28]. In [27], Djidjev et al. derived the closed-form solution for the MEP problem with a single-sensor field in both unbounded and polygonal regions. This solution completed the open problems left from [24]; i.e., it worked with both bounded and polygonal regions without any constraints. Based on this result, the authors provided an approximation algorithm for calculating MEP in a multiple-sensor field. The main idea was triangulating the sensing field using a Voronoi diagram, and in the scope of each Voronoi cell, the MEP was calculated by the optimal solution in the single-sensor case. To obtain the solution for the whole sensing field, Dijkstra shortest path algorithm was performed in the final one. While this method successfully dealt with closest-sensor field intensity model, it could not be applied for all-sensor field intensity model. In [12], the authors introduced a very similar concept to MEP which was the maximal breach path—a path running from a single source to a destination point across the sensing field, in which the Euclidean distance from any point on the path to the closest sensor is maximized. The authors proved that the maximal breach path must lie on the edges of Voronoi diagram corresponding to the set of sensors. The maximal breach path was then easily calculated using the Dijkstra shortest path algorithm. Both the minimal exposure path and the maximal breach path provide information about the worst-case coverage of the sensor network; however, the minimal exposure path is more generic since it can be used with a variety of network models. In [29], the authors studied the maximal breach path problem; they then proposed the Voronoi diagram-based algorithm for solving the problem. In [13], the authors extended the previous concept of the worst-case path-based coverage to evaluate the coverage of a given network from a global point of view, taking arbitrary paths into account by considering the arbitrary source and destination pairs. They then presented centralized and distributed algorithms that used knowledge from computational geometry; i.e., Voronoi diagram and Delauney triangulation. The works in [26,30–32] addressed different MEP models using the common idea of breaking the region of interest into small grids. In [26], the region of interest was broken into Voronoi cells and the calculation of the MEP was limited to the scope of each cell. The authors then proposed a localized algorithm that can effectively reduce the communication and computation performed by the sensor network. Song et al. [31] and Liu et al. [32] used uniform square grids to partition the ROI. While almost grid-based methods used the Dijkstra shortest path algorithm to search for the MEP, the authors in [31] proposed a metaheuristic algorithm called Physarum optimization algorithm to achieve the

minimal exposure road-network among multiple points of interest. This algorithm then can be applied to solve the Steiner tree problem. In [32], the authors used percolation theory to derive the critical conditions for the existence exposure path, where an exposure path is a penetration path through the sensor network on which every point is not covered by any sensor.

The advantage of the Voronoi diagram-based method is that it extracts the geometric characteristics of Voronoi diagram to reduce the significant computational effort. However, this method only works under a particular type of network model. Unless the network is homogeneous and the sensors are isotropic, the Voronoi diagram-based method cannot obtain the solution because if not for that case, the intrinsic geometric characteristics of Voronoi diagram will no longer reflect information about the MEP. For the grid-based method, its main idea can be summarized as follows: (1) transform the continuous search domain of the MEP problem into a discrete one by discretizing the region of interest into small grids; (2) construct a graph based on these grids and perform a graph search algorithm to calculate MEP. The grid-based method can deal with almost all types of network model, and different constraints of the particular problem. However, the results of this method are limited; i.e., the path that is found always follows fixed directions, which does not follow realistic scenarios. In addition, there must be a trade-off between the solution accuracy and the computation effort; i.e., to get higher accuracy, the grid size must be smaller, which increases the computational time.

Since the Voronoi diagram-based and the grid-based methods posed several disadvantages, as mentioned above, recently, heuristic/metaheuristic methods which were inspired by the process in nature, such as particle swarm optimization (PSO) and the genetic algorithm, were applied to solving the MEP problem [29,33–37]. These studies converted the MEP problem into the numerical functional extreme (NFE) problem [23] by breaking the penetration path into small enough path intervals and then performed a metaheuristic algorithm to search for the optimal result. Although the general idea was shared among these studies, each work proposed different techniques that are appropriate to the particularities of the models considered. Miao et al. [33] modified the PSO algorithm by adding a Gauss mutation operator and a projection operator to avoid the saw-tooth shape of the obtained path. To eliminate this saw-tooth effect of the metaheuristic algorithm and improve the efficiency, the authors in [34] proposed a genetic algorithm with a special crossover operator, a local search scheme and an upside-down operator. Feng et al. [37] considered the MEP problem with path constraints. A genetic algorithm was then proposed with a local search operator that could effectively remove the saw-tooth shape in the path solution and increase the convergence speed. Binh et al. [35] studied the MEP problem under a practical probabilistic coverage model, which took environmental factors into consideration and regarded them as noise. The problem was then addressed using a genetic algorithm with two efficient crossover operators and one mutation operator. The authors also investigated the MEP problem under a directional heterogeneous sensor network in [36]. This work introduced a novel PSO algorithm based on gravity force theory. All of those studies on the MEP problem were done to achieve a solution as good as possible in the least computational time. Compared to previous Voronoi diagram-based and grid-based approaches, this approach obtained better results in less computational time.

We have delved into the related works of the MEP problem in WSNs. It is certain that almost the authors did not regard moving ability of sensor nodes. Previously, there was only one research paper related to MMEP, written by Zhang et al. [38]. Nevertheless, their study solved the MEP problem under a hybrid sensor network with both static and mobile sensor, in which the minimal exposure path under static sensors that could avoid mobile sensors was introduced. The approach of this work was a Voronoi diagram method which could not be generalized to deal with other network models, such as an entirely mobile sensor network or a heterogeneous sensor network. Our previous publication in 2017 [39] also studied the MEP problem in MWSNs. However, the research result was not optimal, which served as the motivation for us to do further research and come up with a more efficient solution for the MMEP problem. Therefore, in this paper, we introduce a heuristic strategy called control-point initialization and propose the HPSO-MMEP algorithm to effective solve the MMEP problem.

We summarize all the related works presented in this section in Table 1.

Dublingtions	Ν	Network Model			
rubilitations	Voronoi Diagram Grid-Based Metaheuris		Metaheuristic	Static	Mobile
[12,13,24,27,29,30]	$\checkmark$			$\checkmark$	
[25,26,28]	$\checkmark$	$\checkmark$		$\checkmark$	
[32]		$\checkmark$		$\checkmark$	
[31,35]		$\checkmark$	$\checkmark$	$\checkmark$	
[33,34,36,37]			$\checkmark$	$\checkmark$	
[38]	$\checkmark$				$\checkmark$
[39]			$\checkmark$		$\checkmark$

Table 1. Comparative table of related works on the MEP problem.

#### 3. Proposed Algorithm: HPSO-MMEP

# 3.1. Preliminaries

#### 3.1.1. Sensing Model

A sensing model describes the ability of a sensor node to sense its surrounding environment. In reality, almost all types of sensors share a feature in common: the sensing quality, or sensing intensity, of a sensor decreases as the distance away from the sensor increases. In this paper, we adopt a widely used sensing model called *attenuated disk model* and another variant called *truncated attenuated disk model*. Let f(l, p) be the sensing intensity function of the sensor located at the position l on the target point p; the expression of f(l, p) for the attenuated disk model is:

$$f(l,p) = \frac{C}{d^{\lambda}(l,p)}$$
(1)

where d(l, p) is the Euclidean distance between the sensor and the the target point p; C is a constant depending on the essence of sensor;  $\lambda$  is the attenuation exponent, which depends both on the sensor and the environment. For the truncated attenuated disk model, the sensing intensity function is:

$$f(l,p) = \begin{cases} 1 & , \text{if } d(l,p) \le R_1 \\ e^{-\alpha [d(l,p)-R_1]^{\beta}} & , \text{if } R_1 < d(l,p) \le R_2 \\ 0 & , \text{otherwise} \end{cases}$$
(2)

where  $\alpha$  and  $\beta$  are constants,  $R_1$  is a certain range,  $R_2$  is a sensing range and  $R_2 - R_1$  is an uncertain range. Figure 1 illustrates such two attenuated sensing models.



Figure 1. Illustration of (a) an attenuated disk model; (b) a truncated attenuated disk model.

#### 3.1.2. Sensor Field Intensity Model

The sensor field intensity model specifies the collaboration of sensors in the sensing field. In this paper, we use the all-sensor field intensity model, which is described as follows:

$$I(p) = \sum_{i=1}^{N} f(l_i, p)$$
(3)

where *N* is the number of sensor nodes in the sensor network; i.e., the sensor network consists of *N* sensors  $\{s_1, s_2, ..., s_N\}$ .  $l_i$  is the location of sensor  $s_i$ ; *p* is the target point.

#### 3.1.3. Exposure

The concept of exposure was introduced by Meguerdichian et al. [24] as a metric to measure how well an object moving on an arbitrary path can be observed by the sensor network. According to that, the exposure of an object O in the sensing field during time interval  $[t_1, t_2]$  along the path p(t) is defined as:

$$E(p(t), t_1, t_2) = \int_{t_1}^{t_2} I(p(t)) \left| \frac{\mathrm{d}p(t)}{\mathrm{d}t} \right| \mathrm{d}t$$
(4)

In SWSN, the sensing intensity at a specific position does not depend on time, which reduces I(p(t)) to I(p) only. The expression  $\left|\frac{dp(t)}{dt}\right| dt$  can be replaced by dp, which represents a path element. To avoid ambiguity, we denote the object trajectory as  $\mathcal{P}$  instead of p(t). Then, Equation (4) can be rewritten as follows:

$$E(\mathcal{P}) = \int_{\mathcal{P}} I(p) \mathrm{d}p \tag{5}$$

Equation (5) means that exposure of an object O along path  $\mathcal{P}$  does not depend on time but only the geometric characteristic of  $\mathcal{P}$ . Equation (5) can be interpreted as the accumulation of sensing intensity over all points on the path  $\mathcal{P}$ .

In MWSN, Equation (5) cannot be applied to calculate the exposure value due to several problems: (1) The sensing intensity at a specific position changes over time. (2) One would expect the exposure value of an object *O* through the sensor network to get higher if the time for which *O* stays in the sensing field is longer. Suppose *O* freezes for a time interval  $\phi$ ; then Equation (5) causes exposure of *O* in  $\phi$  equal to zero, which is unreasonable in reality. Exposure value should be formulated as the accumulation of sensing intensity over every state of the object. While the position, or the coordinates, determines the state of an object in SWSN, time is the representation for the state of an object in MWSN. Therefore, in this paper, we use the exposure formula defined by Binh et al. [39], which is:

$$E(\mathcal{P},T) = \int_{0}^{T} I(p(t)) dt$$
(6)

Here, we assume that the object always enters the sensing field at t = 0 and exits at t = T.

#### 3.2. Problem Formulation

Finding an analytical solution for the MEP problem is a functional extreme problem in mathematics, which is commonly solved by using Euler–Lagrange differential equation. However, this method works only for several simple cases, where the solution of Euler–Lagrange differential equation can be obtained easily. The authors in [27] derived the exact solution for the MEP in the presence of a single sensor, but none of existing works can provide a closed-form expression for the MEP in multiple sensor scenario. Therefore, this paper investigates to find an approximate solution for the MMEP problem by converting the continuous domain of the problem to a discrete one. Specifically,

time interval [0, T] for which an object *O* stays inside the sensing field is partitioned into a set of finite sub-intervals with the same length  $\Delta t$ .  $\Delta t$  should be small enough such that sensing intensity does not vary too much between two consecutive moments. Then, the right side of (6) can be approximated as follows:

$$\int_{0}^{1} I(p(t)) dt \approx \sum_{i=1}^{m} I(p(i\Delta t)) \Delta t, \text{ with } m = \left\lfloor \frac{T}{\Delta t} \right\rfloor$$
(7)

The MMEP problem can be briefly stated as follows: given a mobile sensor network with N sensor nodes  $\{s_1, s_2, ..., s_N\}$  randomly distributed in the ROI  $\Omega$ . The sensor  $s_i$  has the initial location  $l_i$ , and the location at time t is denoted as  $l_i(t)$ . Each sensor moves in a preset trajectory at constant speed. Any object penetrating the monitoring region always starts from the source point S and goes to the destination point D. The time interval for which the object stays inside the sensing field is [0, T]. Speed of the object is limited at a maximum value, and we assume that the object always moves at its maximum speed, since exposure will get higher if the object stays longer in the sensing field. The goal of the MMEP problem is to find out a penetration path  $\mathcal{P}$  from the beginning point S to the ending point D such that the exposure value of the path  $\mathcal{P}$  is minimal. More precisely, the MMEP problem is formulated as follows:

# Input:

- W, H: the length and the width of the ROI  $\Omega$ .
- $(0, y_S)$ : coordinates of the source point *S*.
- (*W*, *y*<sub>*D*</sub>): coordinates of the destination point *D*.
- *N*: number of mobile sensor nodes.
- $\{s_1, s_2, ..., s_N\}$ : set of sensor nodes deployed in  $\Omega$ .
- $l_i$ : the initial location of sensor  $s_i$ ,  $i = \overline{1, n}$ .
- $\mathcal{R}_i$ : trajectory of sensor  $s_i$ .
- *v<sub>s</sub>*: speed of sensor nodes, all sensor nodes have the same speed.
- *v*<sub>*I*</sub>: maximum speed of the intruder.

## **Output:**

A path  $\mathcal{P}$  connects the source point *S* to the destination point *D*.

## **Objective:**

Exposure of  $\mathcal{P}$  is minimum, which means:

$$\sum_{i=1}^{m} I(p(i\Delta t))\Delta t \to \min, \quad m = \left\lfloor \frac{T}{\Delta t} \right\rfloor$$
(8)

## **Constraint:**

- An intruder always moves towards the right side of Ω; i.e., the distance from the intruder to the right border of Ω gets smaller over time.
- The intruder always moves at its maximum speed *v*<sub>*I*</sub>

## 3.3. Proposed Algorithm

The objective function (OF) of the formulated MMEP problem strongly depends on sensing intensity function I(.), which is specified by sensing model and sensor field intensity model. Commonly, OF is very complicated; i.e., it is non-linear, high-dimensional and non-differential. A deterministic optimization approach like steepest descent or the interpolation method is infeasible since such methods require one to efficiently compute the derivative value of OF. A potential approach to deal with such complicated OF is to use metaheuristic algorithms. Metaheuristics are general algorithmic ideas that can be applied to solve a board range of optimization problems without depending on the particularity of the problem. There are many different existing metaheuristic algorithms and they can be used in combination to achieve better performance. In this paper, we

integrate the genetic algorithm (GA) into particle swarm optimization (PSO) to form the HPSO-MMEP algorithm. While PSO is benefited from the collaboration of biological populations and can rapidly converge to optimal solution, GA keeps the population from getting trapped in local optima by using genetic operators. The following subsections, HPSO-MMEP will be discussed in detail.

## 3.3.1. Individual Representation

An individual corresponds to a solution of the problem. In this problem, an individual is illustrated as a list of angles:  $L = \{\varphi_1, \varphi_2, ..., \varphi_h\}$  with fixed size h, where an angle of a line is the angle between that line and the horizontal axis Ox. A penetration path of solution is formed by the combination of multiple line segments with the predefined length  $\Delta s$  and angles taken from L. The predefined  $\Delta s$  is also called the interval length of the path and the value  $\Delta s$  should be small enough to assure the accuracy of HPSO-MMEP. Figure 2 illustrates the individual representation in HPSO-MMEP. Note that the operator of PSO requires the individual to be at the same size or the same number of angles to be performed. Therefore, the size h of L is fixed at a value that guarantees the formed path  $\mathcal{P}$  shall at least reach the right border of the monitoring region  $\Omega$ . The intersection point between  $\mathcal{P}$  and the right border might not be the destination point D and  $\mathcal{P}$  might go over the right border (dash line in Figure 2). However, the path is restricted to the scope  $0 \le y \le H$ .

To calculate exposure of a penetration path represented by *L*, we ignore proportion of *L* which corresponds to the dash line in Figure 2—i.e., the proportion of the penetration path that goes over the ROI  $\Omega$ . After that, points with equal distance  $\Delta s$  are added to the line segment connecting *I* and *D*, where *I* is the intersection point between the penetration path and the right border of  $\Omega$ . The distance between *D* and its consecutive point is equal to or less than  $\Delta s$ . Finally, the path connecting *S*, *I*, *D* is used to calculate the exposure of an individual represented by *L*. An intuitive description about this is shown in Figure 2.



Figure 2. Individual representation for HPSO-MMEP.

# 3.3.2. Control-Point Initialization

The random initialization method often fail to generate diverse individuals; thus, a heuristic is added to surpass this drawback. We design a new initialization strategy called *control-point* initialization to improve the diversity of the population and the search space of the algorithm. Details of the control-point initialization method are as follows:

1. Partition the ROI  $\Omega$  along *Ox* axis into multiple sub-regions with random size. On each boundary between these sub-regions, randomly choose a point, called *control point*. Note that the number of control point is random.

- 2. Sequentially generate angles according to a probability distribution  $\mathcal{K}$ . The probability distribution is expected to drive the next angle toward to current control point (the left-most control point).
- 3. If the current sub-path, which consists of line segments corresponding to generated angles, reaches the right border, sample the remaining angles, i.e., until the number of angles equal to *h*, according to uniform distribution  $\mathcal{U}\left(\frac{-\pi}{2}, \frac{\pi}{2}\right)$  (totally random). Note that the destination point *D* is considered as the last control point.

Figure 3 depicts our proposed control-point initialization method.



Figure 3. Red stars are the control points that drive the path.

There might be many different probability distributions that can be used for the generation of angles. We briefly describe the probability distribution used in this paper as follows.



Figure 4. Illustration of sample an angle

Supposed that at some steps of the initialization process, we are sampling the next angle  $\varphi$ , the current sub-path reaches point *A* and the current control point is *C* (Figure 4). Let  $\Delta Pr = Pr(\varphi \ge 0) - Pr(\varphi < 0)$ ,  $\Delta y = y_C - y_A$ . In a uniform distribution, we always have  $Pr(\varphi \ge 0) = Pr(\varphi < 0) = 0.5$ . However, to make the control points able to drive the path, we establish a dependency of  $\Delta Pr$  on  $\Delta y$ . We should all agree to the following intuition:

$$\Delta \Pr = \begin{cases} 0, & \text{if } \Delta y = 0\\ 1, & \text{if } \Delta y = H \end{cases}$$
(9)

There is no need to consider negative value of  $\Delta Pr$  here because it is completely symmetric to the positive one. We observe that linear dependency between  $\Delta Pr$  and  $\Delta y$  makes the control points less important because the probability that the path orients toward the control point is not high enough. Therefore, we use a non-linear mapping as in Figure 5.



**Figure 5.** Dependency of  $\Delta Pr$  on  $\Delta y$ .

Let  $Pr_u$  and  $Pr_d$  stand for  $Pr(\phi \ge 0)$  and  $Pr(\phi < 0)$ . Then we have:

$$\begin{cases} \Pr_{u} + \Pr_{d} = 1\\ \Pr_{u} - \Pr_{d} = \Delta \Pr = \sqrt{1 - \frac{(\Delta y - H)^{2}}{H^{2}}} \end{cases}$$
(10)

From this set of equations,  $Pr_u$  and  $Pr_d$  are calculated and the probability distribution is determined. In case  $\Delta y < 0$ , simply an inverse sign of  $\Delta Pr$  can give the result.

From this initialization method, it can be expected that the generated individuals will be more diverse, as the paths are able to reach the far top or bottom of the region. Therefore, the population can exploit a larger region and the search space will be enlarged as well.

#### 3.3.3. Integration of GA into PSO Algorithm to Form HPSO-MMEP

The original PSO algorithm can be mapped to the process of a bird swarm seeking food. Individuals in the swarm collaborate and use their historical experiences to get better positions. After a while, the swarm might enter a local optima trap and almost be unable to escape, since the convergence degree at the time is quite high, or, stated differently, the searching space is remarkably shrinking. This is the right time for genetic operators to perform. The crossover and mutation operators of the genetic algorithm can deal with a multiple local optima objective function since they can efficiently expand the searching space and maintain the diversity of the population. HPSO-MMEP is formed by periodically performing genetic operators on the population during PSO search. The update operator of HPSO-MMEP is basically the same as in the original PSO. We recall the update operator in Algorithm 1.

 $r_1$ ,  $r_2$  are random numbers in range (0, 1);  $\omega$ ,  $C_1$ ,  $C_2$  are acceleration coefficients of PSO that depend on particular problem. These coefficients are usually obtained from experiments.

For steps at which genetic operators are used, such operators are performed after the PSO update operation. The period of using genetic operators should be configured at an appropriate value such that the population can benefit from the PSO search process. The period is the time for the population to reach a certain convergent level by performing a PSO update operation. If we use genetic operators at every step, the effects of genetic operators may dominate the effects of PSO update operation because genetic operators make the whole population change remarkably.

Algorithm 1: PSO update operator

<b>Input:</b> The individual at step <i>t</i> : $L^t = \{\varphi_1^t, \varphi_2^t,, \varphi_h^t\}$	
<b>Output:</b> The updated individual at step $t + 1$ : $L^{t+1} = \{\varphi_1^{t+1}, \varphi_2^{t+1},, \varphi_h^{t+1}\}$	
1 Veclocity update 2 $v^{t+1} = \omega * v^t + r_1 * C_1 * (pBest_L^t - L^t) + r_2 * C_2 * (gBest^t - L^t)$	
3 Position update 4 $L^{t+1} = L^t + v^{t+1}$	
<pre>5 pBest update /* pBest<sub>L</sub> is the best position found by individual L /* or the best individual among <math>\{L^1, L^2,, L^t\}</math> 6 if L is better than pBest<sub>L</sub> then 7   pBest<sub>L</sub> <math>\leftarrow</math> L 8 end</pre>	*/ */
<ul> <li>9 gBest update <ul> <li>/* gBest is the best position found by the population</li> <li>/* or the best individual among all pBest</li> </ul> </li> <li>10 if L is better than gBest then <ul> <li>11   gBest ← L</li> </ul> </li> <li>12 end</li> </ul>	*/ */

# 3.3.4. Crossover Operator

The crossover operator is a binary operator. It takes two operands corresponding to two individuals in the population. Figure 6 demonstrates the crossover operator of HPSO-MMEP algorithm. It can be described in the two steps as follows:

- 1. Choosing a position to perform the operator by sampling a random integer *k* in {1,2,...,*h*}. Recall that *h* is the number of genes of each individual in the population.
- 2. Concatenating the beginning part of a father individual from 1 to k 1 and the ending part of mother individual from k to h to form the child individual.



Figure 6. Illustration of the crossover operator of HPSO-MMEP.

If the child is better than both of its father and mother, the father will be replaced by the child. If the child is better than one of its parents, that parent will be replaced by the child. Otherwise, nothing changes. Note that when an individual changes, we need to update its pBest and gBest of the population.

#### 3.3.5. Mutation Operator

We use two mutation operators, which are *inverse mutation* and *symmetric mutation*. The *inverse mutation* is obtained by randomly choosing a continuous set of genes and inverse the order of elements in that set. The symmetric mutation shares the same part of choosing genes in common, but instead of inverting the order of those genes, we invert the sign of each gene in the set. The results are shown in Figure 7.



Figure 7. Mutation operators: (a) Inverse mutation. (b) Symmetric mutation.

While the *inverse mutation* results in a relatively similar individual to the original one, *symmetric mutation* creates a significantly different individual. *Inverse mutation* can also be seen as local searching that tries to search for a better solution in neighborhood region. Symmetric mutation greatly expands and explores the searching space, and thus, it has potential to find the global optimum.

#### 3.3.6. Complexity Analysis

In this section, we analyze time complexity of our HPSO-MMEP algorithm by evaluating the time complexity of calculating the objective function. The value of the objective function of an individual is also called the *fitness* of that individual. Each individual is composed of h genes, corresponding to hpath intervals with equal length  $\Delta s$ . The fitness of an individual L is calculated by summing up the fitness of each of its genes. For calculating the fitness of a gene, or a path interval  $\Delta s$ , the algorithm must traverse through all sensors in the sensing field; this requires time O(N), where N is the number of sensors. Therefore, time complexity for calculating the fitness of an individual is O(h \* N).

As aforementioned in Section 3.3.1, the fitness of an individual is calculated by an alternative individual, which corresponds to the path connecting S, I, D in Figure 2. Assume  $\Delta s$  is constant, we prove that length *h* of the alternative individual is O(W + H).

**Proof.** Consider a path with two components: the sub-path from the source point *S* to the right border of the ROI  $\Omega$  and the vertical sub-path towards the destination point D (the path connecting S, I and the path connecting I, D in Figure 2 respectively). For the latter sub-path, denote the number of genes as t. Then, t is equivalent to the number of additional points described in Section 3.3.2, and it is obviously true that  $t = O\left(\frac{H}{\Delta s}\right) = O(H)(*)$ . For the former sub-path, we estimate the probability distribution of its length as follows.

Denote  $\mathcal{L}$  as length of this sub-path and g as the number of its genes; then we have:

$$\mathcal{L} = \Delta s * g$$
  

$$W = \Delta s \left[ \cos(\varphi_1) + \cos(\varphi_2) + ... + \cos(\varphi_g) \right]$$
(11)

The probability distribution of a gene  $\varphi$  has already been presented in Section 3.3.2. According to that,  $\varphi$  is equal to the multiplication of two independent random variables A and  $\alpha$ , where A determines the sign of  $\varphi$  and  $\alpha$  determines the magnitude of  $\varphi$ . Specifically,  $\alpha \sim \mathcal{U}(0, \frac{\pi}{2})$  and A follows a discrete probability distribution:  $Pr(A = 1) = Pr(\varphi \ge 0)$ ,  $Pr(A = -1) = Pr(\varphi < 0)$ . Based on this, we can easily determine the probability distribution of  $|\varphi|$  as follows:

$$|\varphi| = |A * \alpha| = |A| * |\alpha| = A \sim \mathcal{U}(0, \frac{\pi}{2})$$
 (12)

Let  $Z = \cos(|\varphi|)$ ; we can compute the expectation  $\mu(Z)$  and standard deviation  $\sigma(Z)$  of *Z*. We only present here the final results:

$$\mu(Z) = \frac{2}{\pi}, \ \sigma(Z) = \frac{1}{2} - \frac{4}{\pi^2}$$
(13)

We perform some transformations on the second line of Equation (11):

$$W = \Delta s \left[ \cos(\varphi_1) + \cos(\varphi_2) + ... + \cos(\varphi_g) \right]$$
  
=  $\Delta s \left[ \cos |\varphi_1| + \cos |\varphi_2| + ... + \cos |\varphi_g| \right]$   
=  $\Delta s (Z_1 + Z_2 + ... + Z_g)$   
=  $\Delta s * g * \overline{Z}$   
=  $\mathcal{L} * \overline{Z}$  (14)

where  $Z_1, Z_2, ..., Z_g$  are g independent random variables with the same probability distribution as Z,  $\overline{Z} = \frac{Z_1 + Z_2 + ... + Z_g}{g}$ . Assume that  $W \gg \Delta s$ ; then  $g \gg 1$ . According to the central limit theorem, probability distribution of  $\overline{Z}$  can be approximated by the normal distribution  $\mathcal{N}\left(\mu(Z), \frac{\sigma^2(Z)}{g}\right)$ . For a normal distribution  $\mathcal{N}(\mu, \sigma^2)$ , 99.7% of the observations are within  $[\mu - 3\sigma, \mu + 3\sigma]$ . Hence, the probability that  $\overline{Z}$  is within  $\left[\mu(Z) - \frac{3\sigma(Z)}{\sqrt{g}}, \mu(Z) + \frac{3\sigma(Z)}{\sqrt{g}}\right]$  is 99.7%. Since  $g \gg 1$  and  $\sigma$  is small  $\left(\frac{1}{2} - \frac{4}{\pi^2}\right)$ , in practice, we can assume that value of  $\overline{Z}$  is bounded within a specific interval, and hence, from Equation (14), we have  $\mathcal{L} = O(W)$ , which deduces to  $g = \frac{L}{\Delta s} = O(W)(**)$ .

From (\*) and (\*\*), we have h = t + g = O(H) + O(W) = O(W + H), which is what must be proven.  $\Box$ 

To conclude, time complexity of our HPSO-MMEP algorithm is O((W + H) \* N).

#### 4. Experimental Results

This section investigates effects of important factors on the performance of the proposed HPSO-MMEP algorithm. To prove the efficiency of HPSO-MMEP, we compare the algorithm with two existing algorithms, which are GAMEP [39] and HPSO [33], in various experimental scenarios. In the last case, effects of several factors relating to the MMEP problem such as speed of intruder  $v_I$ , number of sensors N and network model (mobile/static) are thoroughly analyzed.

## 4.1. Experimental Settings

## 4.1.1. Data Settings

We establish eight different experimental scenarios. For each scenario, the number of sensors N varies in {25, 50, 75, 100}, which in turn contains 10 topologies. Since HPSO-MMEP is approximate algorithm, we run on each topology 20 times to get average value and measure the standard deviation. A scenario is determined by a sensing model, a sensor trajectory and a sensor distribution, which are as follows:

#### Sensing model

- Attenuated disk model
- Truncated attenuated disk model

## Sensor trajectory

- Random point trajectory.
- Rectangle trajectory.

Sensor trajectory is characterized by a set of fixed positions, and sensor will move from one position to the next in that set. The movement repeats when the sensor reaches the last position in the position set [17]. Rectangle trajectory corresponds to a position set with four points that are four corners of a rectangle. Random point trajectory corresponds to a random-size set of random positions. Figure 8 illustrates these two sensor trajectory.



Figure 8. Sensor trajectory: (a) Rectangle trajectory. (b) Random point trajectory.

## Sensor distribution

- Uniform distribution.
- Gauss distribution.

Differently from SWSNs, sensors in MWSNs change their positions over time. Hence, sensor distribution in MWSNs refers to the distribution of every position in the position set of every mobile sensor.

An experimental instance corresponds to an experimental scenario with a specific number of sensors and is named in the format "*Dis\_Model\_Traject\_Num*" in which *Dis* presents the sensor distribution: "*u*" for uniform and "*g*" for Gauss; *Model* is the sensing model: "*a*" for attenuated and "*t*" for truncated attenuated; *Traject* is the trajectory of sensor: "*ran*" for random point and "*rec*" for rectangle; "*Num*" is the number of sensor. Although 10 topologies are generated per a specific number of sensors per a scenario, we select only one topology for experiments presented in later sections and do not regard the topology in an experimental instance.

## 4.1.2. Parameters and System Settings

All subsequent sections share the following parameters: W = 100 m, H = 40 m, the source point S = (0, 30), the destination point D = (100, 10), the speed of sensors  $v_S = 1$  and the speed of intruder  $v_I = 2$ . The sensor network is homogeneous and entirely mobile. A position of a sensor is pair of independent random variables (*X*, *Y*). For uniform distribution,  $X \sim U(0, W)$  and  $Y \sim U(0, H)$ , where U(a, b) denotes uniform distribution on interval [a, b]. For Gauss distribution,  $X \sim U(0, W)$ and  $Y \sim \mathcal{N}(\mu, \sigma^2)$  where  $\mathcal{N}(\mu, \sigma^2)$  is Gauss distribution with expectation  $\mu$  and standard deviation  $\sigma$ . The following are settings of the Gauss distribution used in all our experiments:

• For rectangle trajectory,  $\mu = y_S = 30$  and  $\sigma^2 = 200$  for all positions in the position set of all sensors.

• For random point trajectory,  $\mu = y_S = 30$  and  $\sigma^2 = 64$  for initial position of all sensors,  $\mu = y_S = 30$  and  $\sigma^2 = 200$  for all other positions in the position set of all sensors.

Configurations of attenuated and truncated attenuated sensing models are given in Table 2 and are applied to all sensors.

Sensing Model	Parameter	Value
Attenuated disk model	Constant C	1
	Path attenuation exponent $\lambda$	2
	Constant <i>α</i>	0.5
Truncated attenuated disk model	Constant $\beta$	1
function attenuated disk model	Certain range $R_1$	1
	Sensing range $R_2$	10

Table 2. Experimental parameters for attenuated disk model and truncated attenuated disk model.

Parameters settings for HPSO-MMEP algorithm are shown in Table 3.

Table 3. Experimental parameters for HPSO-MMEP algorithm.

Parameter	Value
Swarm size	100
Number of GA iterations	100
Number of PSO iterations per GA iteration	10
-	$C_1 = 0.8$
Acceleration coefficients	$C_2 = 0.7$
	$\omega = 1$
Crossover rate	100%
Inverse mutation rate	30%
Symmetric mutation rate	70%

All our experiments were run on a machine with Intel<sup>®</sup>Core<sup>TM</sup>i7-4790 CPU at 3.60 GHz 16 GB RAM using the Java language.

#### 4.2. Computation Results

- 4.2.1. Effects of Important Factors on the Performance of HPSO-MMEP
- (a) Performance of HPSO-MMEP using different  $\Delta s$  ( $\Delta s = v_I \Delta t$ )

 $\Delta s$  does not appear in exposure calculations; however, it still reflects the accuracy of the exposure approximation; i.e., transform the integral form of exposure  $E(\mathcal{P}, T) = \int_{0}^{T} I(p(t)) dt$  to sigma sum form

 $E(\mathcal{P}) = \sum_{i=1}^{m} I(i\Delta t)\Delta t$ . We have  $\Delta t = \frac{\Delta s}{v_I}$ , and  $\Delta t$  should be at an appropriate value such that sensing intensity does not vary too much on that time interval. The expectation of having similar sensing intensity in  $\Delta t$  can be achieved by choosing small enough value of  $\Delta s$  because points with small distances from each other tend to have similar sensing intensity values. On the other hand, decreasing of  $\Delta s$  causes the exposure calculation to be more complex, thereby increasing the computation time of HPSO-MMEP. Figure 9 shows the effects of  $\Delta s$  on the minimal exposure value and the computation time achieved by HPSO-MMEP algorithm. For this experiment, we examine on the experimental instance  $u_a$ -rec.



**Figure 9.** Effect of  $\Delta s$  on (**a**) minimal exposure value; (**b**) computation time.

From Figure 9a, we can see that for a small number of sensors (N = 25 and N = 50), the minimal exposure value continuously slightly rises when the value of  $\Delta s$  increases from 0.1 to 5. This is caused by two reasons: (1) the large value of  $\Delta s$  results in the lower dimension of the searching space, which limits the searching capability of HPSO-MMEP, and thus, gives a worse solution; (2) the large value of  $\Delta s$  degrades the accuracy of exposure calculation, which increases the error between the approximate exposure value and the true exposure value of a penetration path. However, there is no significant difference in the minimal exposure value at  $\Delta s = 0.1$  (the smallest) and  $\Delta s = 5$  (the largest). This can be explained that due to uniform distribution and the sparsity of sensors, there exist wide areas in the ROI  $\Omega$  in which every point has relatively similar and low sensing intensity. Therefore, the large value of  $\Delta s$  still guarantees that sensing intensity does not vary too much on that path interval. In contrast, for a large number of sensors (N = 75 and N = 100), the minimal exposure value changes significantly and unpredictably when  $\Delta s$  increases. In all cases, we obtain quite stable solution at  $\Delta s = 0.1$  and  $\Delta s = 0.2$ . Figure 9b indicates that the computation time of HPSO-MMEP is inversely proportional to  $\Delta s$ . When  $\Delta s = 0.1$ , computation time jumps up an extremely big distance compared to other values of  $\Delta s$ . Note that too small a value of  $\Delta s$  makes the penetration path not realistic since it is not smooth and has a saw-tooth shape. Based on what we have analyzed so far, we choose the optimal value of  $\Delta s$  to be 0.2 and use this value for experiments in all subsequent sections.

# (b) Performance of HPSO-MMEP in the presence of genetic operators

We evaluate the effects of different genetic operators on HPSO-MMEP algorithm by creating different versions of HPSO-MMEP, as shown in Table 4, and perform each version on the experimental scenario  $u_a$ *rec*. The results are shown in Figure 10.

From Figure 10a, we can see that three versions, HPSO-MMEP1, HPSO-MMEP2 and HPSO-MMEP3, give quite similar minimal exposure values when the number of sensors varies. This means that three genetic operators, which are crossover, inverse mutation and symmetric mutation, tend to give the same effects on the performance of HPSO-MMEP. The computational times of these three versions (Figure 10b) are also not too different. The difference becomes clearer when all the three operators are in used or all of them are unused. For the minimal exposure value, HPSO-MMEP performs best with big number of sensors. This can be explained: when the number of sensors is large, there is little chance of finding a good solution. Hence, expanding the search space by using the genetic operators results in much better solution. However, when number of sensors is small, just using original PSO is enough to get the solution.



Figure 10. (a) Minimal exposure value; (b) computation time.

Table 4. Different versions of HPSO-MMEP using different genetic operators.

	HPSO-MMEP	HPSO-MMEP1	HPSO-MMEP2	HPSO-MMEP3	HPSO-MMEP4
Crossover	$\checkmark$	$\checkmark$	$\checkmark$		
Inverse mutation	$\checkmark$	$\checkmark$		$\checkmark$	
Symmetric mutation	$\checkmark$		$\checkmark$	$\checkmark$	

# (c) Performance of HPSO-MMEP using control-point initialization method

We evaluate effect of the control-point initialization by comparing it with random initialization. The experiments are conducted on Gauss and uniform distributions of sensors, corresponding to Figure 11a,b respectively. The trajectory of the sensor is random point; the sensing model is the attenuated disk model.

From Figure 11, we can see that the control-point initialization greatly surpasses the random initialization to obtain smaller minimal exposure value in case of Gauss distribution. On the other hand, there is no significant difference between two methods when sensors are uniformly distributed. Figure 12 illustrates a penetration path created by these two methods. While the path created by the control-point method looks rugged and can reach the far the top or the bottom of the ROI, the other one seems to flow horizontally from the left to the right. When sensors are distributed according to Gauss distribution, an intruder tends to travel along the top or bottom border since the density of sensor is high in the middle of sensing field. In this case, the control-point gives much better results than the random method. Whereas, when sensors are uniformly distributed in the ROI, there might be multiple paths that have low exposure and our HPSO-MMEP still runs effectively with the random initialization.



Figure 11. Random vs. control-point initialization: (a) Gauss distribution. (b) Uniform distribution.





Figure 12. Penetration path created by (a) control-point initialization; (b) random initialization.

4.2.2. Comparison between HPSO-MMEP and GAMEP [39]

We run our HPSO-MMEP algorithm and GAMEP [39] algorithm on various experimental instances to prove the efficiency of HPSO-MMEP over GAMEP. Parameters of GAMEP are set as in Table 5. Tables 6 and 7 show the computation results of our HPSO-MMEP algorithm and the GAMEP algorithm in case of uniform distribution and Gauss distribution of sensors respectively. Tables 6 and 7 indicate that:

- For the minimal exposure value, HPSO-MMEP wins over GAMEP in all experimental instance. The gap between the minimal exposure value achieved by the two algorithms is extremely large. For uniform distribution of sensors, the minimal exposure value obtained by HPSO-MMEP is on average 2.7 times smaller than one obtained by GAMEP. For Gauss distribution of sensors, the ratio is 2.1. The overall far better performance of HPSO-MMEP over GAMEP in various scenarios provides strong evidence of the effectiveness of our proposals for making the combination between GA and PSO, and making heuristics on the initialization of the population (control-point initialization). On the other hand, GAMEP is a pure genetic algorithm with the whole population generated from an initial individual. This causes lack of population diversity, which is one of the essential criteria for population-based meta-heuristics algorithm. As a result, GAMEP usually converges to a local optima, leaving a potential domain to get a better solution of the searching space uncovered.
- For standard deviation, HPSO-MMEP obtains a much smaller exposure than GAMEP in almost cases. Put Tables 6 and 7 together: there are 32 experimental instances in total and we capture several statistical results. For our HPSO-MMEP algorithm, 25 out of 32 instances have standard deviations smaller than 2.0, while this fraction is only 8 out of 32 for the GAMEP algorithm. Among eight experimental instances in which GAMEP has a standard deviation smaller than 2.0, there are seven instances falls at N = 25. This implies that GAMEP is stable only for a small number of sensors, and often fails to obtain stability when sensors are densely distributed. There are a few instances in which standard deviation of HPSO-MMEP is larger than GAMEP. However, except for the instance  $g_t_{rec}_{100}$ , there is only a slightly difference between standard deviation achieved by HPSO-MMEP and GAMEP at those instances; i.e., standard deviation in both algorithms is smaller than 1.0. In summary, HPSO-MMEP is highly stable since it obtains small standard deviations in almost all experimental instances and the standard deviation does not change significantly among those instances; i.e., standard deviation mostly falls on [0.0, 4.0]. Contrastingly, GAMEP has an extremely large standard deviation in some instances, and on average, the standard deviation of GAMEP is very high; i.e., 5.0 on average. The attractive results in standard deviation of HPSO-MMEP compared to GAMEP also suggest that HPSO-MMEP may approach close to global optima of the MMEP problem.
- For the computation time, HPSO-MMEP is longer than GAMEP in all instances. The correlation between the computation al time of the two algorithms is closed to linearity. This makes sense since both algorithms run in a fixed number of iterations. The extreme ratios between the computational time of HPSO-MMEP and GAMEP are 1.1 and 2.0, which is acceptable for the trade-off between the computation time and the solution accuracy and stability. Note that the

complexity of the fitness calculation is the same in HPSO-MMEP and GAMEP. The difference in computation time comes from the speed of convergence of each algorithm. For HPSO-MMEP, it takes a longer time than GAMEP for the algorithm to converge. This again confirms that HPSO-MMEP explores the searching space of MMEP problem more thoroughly to give a better solution.

Parameters	Value
Number of running on each instance	30
Population size	100
Number of iterations	200
Crossover rate	50%
Mutation rate	5%

 Table 5. Parameters setting for GAMEP [39].

**Table 6.** Computation results of HPSO-MMEP in comparison with GAMEP in uniform distribution ofsensors (Mev: minimal exposure value, Sd: standard deviation).

Sancing Model	Trajectory	Num	HPSO-MMEP			GAMEP		
Sensing woder			Mev	Sd	Time (ms)	Mev	Sd	Time (ms)
		25	4.0943	0.2752	33,976	7.3192	0.1889	20,970
	Destanala	50	6.5169	0.6242	60,410	26.898	1.8234	37,516
	Rectangle	75	12.2150	0.23488	83,351	33.8265	3.8030	53,504
		100	26.2017	2.5010	92,361	45.0035	4.3400	75,283
Attenuated		25	5.0750	0.3284	33,603	14.4034	2.0216	19,744
	DanDaint	50	6.8178	0.5703	60,269	21.1367	0.6702	42,236
	KanPoint	75	9.7004	0.18515	86,460	42.4746	2.3046	60,835
		100	29.3876	3.6232	105,081	41.7169	3.240	76,955
		25	3.2934	0.7606	34,828	8.3944	0.7621	18,973
	Destanala	50	4.2928	0.1333	56,075	32.667	12.568	43,700
	Rectangle	75	14.5890	0.5100	78,390	43.26501	8.6802	61,154
Truncated		100	36.7583	1.8389	100,346	65.4624	6.6035	83,299
	RanPoint	25	6.4533	1.5017	36,802	19.2233	5.7220	25,598
		50	4.2184	0.3298	57,800	34.5452	3.1451	48,874
		75	4.4952	0.3681	83,477	48.8881	20.4467	68,922
		100	31.1390	3.7327	103,037	62.1939	4.6001	84,194

Sensing Model	Trajectory	Num	HPSO-MMEP			GAMEP		
		Inum	Mev	Sd	Time (ms)	Mev	Sd	Time (ms)
		25	3.7705	0.5323	38,815.7	6.3678	0.3921	20,252.9
	Postanala	50	4.8387	1.5128	58,371.8	16.1140	3.07555	38,618.4
	Rectangle	75	14.6659	1.5564	76,501.1	21.6330	5.9376	55,233.2
		100	22.9836	2.0796	86,158	46.2440	2.63185	74,937
Attenuated	RanPoint	25	2.5580	0.1083	38,155.9	8.5071	0.5490	20,359
		50	6.5290	0.4990	63,089	17.2894	3.7424	41,227.1
		75	10.7616	1.0230	85,041.9	30.2552	5.5275	61,047.1
		100	17.3493	3.9408	91,420.3	39.6380	6.6926	79,798.6
		25	2.1049	0.9336	37,528.7	8.0387	0.5067	18,565
	Destanala	50	1.8612	2.5485	60,164.4	3.5942	3.7870	38,110.9
	Rectangle	75	20.5613	1.4335	77,512.4	27.3091	19.3064	57,500.7
Truncated		100	18.0101	9.7088	90,404.2	24.5270	3.9843	75,411.9
		25	0.6000	0.0691	37,441.5	11.9273	0.9837	20,707.8
	RanPoint	50	4.2179	0.5602	59,339.2	17.2173	3.0961	40,000.1
		75	7.4190	1.3398	81,291.4	26.2230	15.8393	61,232
		100	11.0246	1.2378	90,070.1	22.4935	3.1480	75,302.6

**Table 7.** Computation results of HPSO-MMEP in comparison with GAMEP in Gauss distribution of sensors (Mev: minimal exposure value, Sd: standard deviation).

## 4.2.3. Comparison between HPSO-MMEP and HPSO [33]

Settings of HPSO are given in Table 8. We run our HPSO-MMEP algorithm and HPSO algorithm on the experimental instance *u\_a\_ran* and the results are shown in Figure 13. Note that HPSO deals with the MEP problem in a static sensor network. Hence, for the two algorithms to be comparable, we simply set value of  $v_S$  in our MMEP model to zero. Experimental results show that our HPSO-MMEP algorithm outperforms HPSO in both the minimal exposure value and the computational time. Specifically, in Figure 13a, there is a remarkably big gap between HPSO-MMEP and HPSO with respect to the minimal exposure value. The higher value of N is, the bigger the gap in the minimal exposure value between the two algorithms is. Especially, the gap grows up a big distance between N =75 and N = 100. Since HPSO-MMEP performs stably in almost cases, this might be due to the ineffectiveness of HPSO when dealing with complex experimental instances. For the computational time, Figure 13b shows that HPSO-MMEP obtains much better results than HPSO. We observe that the correlation between HPSO-MMEP and HPSO with respect to the minimal exposure value and computational time are quite the same. There is also a big jump in the gap between the computation time of HPSO-MMEP and HPSO from N = 75 to N = 100, which implies that HPSO is not stable in term of the computational time. Hence, when the number of sensors increases, HPSO poses disadvantages in both the solution accuracy and the computational time, which can be effectively addressed using our proposed HPSO-MMEP algorithm. Although both HPSO-MMEP and HPSO are inspired by the idea of combining genetic algorithm and particle swarm optimization, the key that makes HPSO-MMEP distinguished is the ability of vigorously searching, which is missing in HPSO. In this paper, HPSO-MMEP is proposed to solve the particular MMEP problem. However, HPSO-MMEP is not only restricted in the scope of this work but can also be applied to a variety of MEP models or even more general in the optimization problem with promise performance.

Parameters	Value
Number of running on each instance	20
Number of iterations	200
Swarm size	200
Acceleration coefficients	$C_1 = C2 = 2$ $\omega = 1$
Mutation rate	100%

Table 8. Parameters settings for HPSO [33].



**Figure 13.** Comparison between HPSO-MMEP algorithm and HPSO algorithm regarding (**a**) the minimal exposure value; (**b**) the computation time when the number of sensor varies from 25 to 100

#### 4.2.4. Effects of Several Factors of Mmep Problem on the Solution

The minimal exposure value represents the lowest capacity to be detected of an intruder crossing the sensor network. This value also measures the network quality. This section studies the effects of several factors on the minimal exposure value: the speed  $v_I$  of intruder and some network properties.

HPSO-MMEP is an approximate algorithm aiming at finding a solution as close as possible to the true solution. As analyzed so far, HPSO-MMEP is highly stable and is capable of effectively exploring the massive searching space of the MMEP problem. Hence, the minimal exposure value obtained from HPSO-MMEP can reliably reflect the network quality and we use this value to evaluate the network quality in all subsequent sections. Note that the network quality is independent from any algorithm; it depends only on the intrinsic properties of the network, such as the number of sensors, the topology of sensors and type of sensors.

# (a) Effect of intruder's speed $v_I$

Exposure of an intruder is expected to increase when time for which the intruder stays in the sensing field gets longer. Hence, it makes sense that exposure value is inversely proportional to the speed  $v_I$  of the intruder. We run our HPSO-MMEP algorithm with different values of  $v_I$  in several experimental instances to test the effect of this parameter on the minimal exposure value. The results are shown in Figure 14.



Figure 14. Effects of intruder's speed on minimal exposure value.

The experimental results completely matched our expectations. From this, we can conclude that our exposure definition is more appropriate compared to the traditional one.

Note that although the minimal exposure value varies when the speed  $v_I$  of intruder varies,  $v_I$  affects only to the capacity to be detected of the intruder, it does not reflect the network quality.

#### (b) Effect of number of sensors N

Experimental results in Table 6, Table 7 indicates that in all experimental scenarios, the network quality is improved when the number of sensors increases (recall that we used the minimal exposure value obtained from HPSO-MMEP to evaluate the network quality). For both Gauss distribution and uniform distribution of sensors, the minimal exposure value slightly rises between N = 25 and N = 50, but soars between N = 75 and N = 100. In some rare cases, the minimal exposure value is even lower when the number of sensors is larger, e.g., instances  $g_t rec_25$  and  $g_t rec_50$ , instances  $u_t rec_25$  and  $u_t rec_50$ . This implies that the number of sensors is not the only factor to affect to the network quality. The other factor is network topology. For the same the number of sensors, performance of the network might vary significantly when network topology changes. To summarize, we can conclude that the network quality depends largely on the number of sensors in the ROI.

We also observe that there is no limitation for the minimal exposure value; i.e., the exposure value approaches infinity when the number of sensors approaches infinity. In practice, the minimal exposure value of a network higher than a threshold is sufficient to perform monitoring tasks. The selection of the threshold depends on the expectation of the network designer on the network performance.

#### (c) Comparison between mobile sensor network and static sensor network

Network quality when using mobile sensors and static sensors are compared with each other on two sensor distributions, using the attenuated disk model and the random point sensor trajectory. The results are shown in Figure 15.



**Figure 15.** Performances of a mobile sensor network and a static sensor network: (**a**) Gauss distribution; (**b**) uniform distribution.

We see that for Gauss distribution, MWSN always performs better than SWSN, whereas, for uniform distribution, MWSN is often less efficient. The main reason behind is the effects of sensor movements. For Gauss distribution, initial positions of sensors form a network topology with many holes in coverage; i.e., the regions closed to the top or bottom border of the ROI are poorly covered. In this case, the movements of sensors help to fill up the coverage holes, thus improve the network quality. In contrast, for uniform distribution, coverage quality at every position in the ROI is relatively even, and sensor movements might cause new coverage holes and degrade the quality of sensor network. However, when the number of sensors is large (N = 100), MWSN results in better performance for both Gauss and uniform distributions. This is because coverage holes no longer matter when N becomes large; i.e., there hardly exists coverage holes in the ROI both in static and mobile scenarios. In addition, expected coverage quality over time at almost positions in the ROI might be higher with the movements of sensors, thus, MWSNs obtain the expected performance of improving network coverage quality.

# 5. Conclusions

The MEP problem serves as an effective tool to evaluate the coverage quality of a sensor network. In this paper, we investigated the MEP problem in a mobile wireless sensor network (MMEP problem), which is a new and potential model that can offer a lot of additional advantages compared the traditional static MEP model. With the idea of combining the strengths of the genetic algorithm and particle swarm optimization, the HPSO-MMEP algorithm was devised to solve the MMEP problem. Different experimental instances were generated with the variation of the sensor regarding number, sensor distribution and sensor trajectories to examine our HPSO-MMEP algorithm. Effects of different factors on the performance of HPSO-MMEP algorithm and the network quality were thoroughly analyzed and explained after conducting numerous experiments. The results show that our algorithm successfully addresses the MMEP problem and is more efficient than previous algorithms. Furthermore, HPSO-MMEP is not restricted in the scope of this work as it is also applicable to a variety of MEP models or even general optimization problems.

**Author Contributions:** N.T.M.B.: Methodology, Algorithms, Writing manuscript; Abdelhamid Mellouk: Methodology, Writing manuscript, Supervision; H.T.T.B.: Methodology, Writing manuscript, Supervision; L.V.L.: Algorithms, Coding, Experiment, Writing manuscript; D.L.S.: Algorithms, Coding, Experiment, Writing manuscript; T.H.A.: Methodology, Writing manuscript. All authors have read and agreed to the published version of the manuscript.

**Funding:** This research is funded by Hanoi University of Science and Technology (HUST), under grant number "T2018-PC-207".

Conflicts of Interest: The authors declare no conflict of interest

## References

- Aziz, N.A.B.A.; Mohemmed, A.W.; Sagar, B.D. Particle swarm optimization and Voronoi diagram for wireless sensor networks coverage optimization. In Proceedings of the 2007 International Conference on Intelligent and Advanced Systems, Kuala Lumpur, Malaysia, 25–28 November 2007; IEEE: Piscataway, NJ, USA, 2007; pp. 961–965.
- 2. Munir, A.; Gordon-Ross, A. Optimization approaches in wireless sensor networks. In *Sustainable Wireless Sensor Networks*; InTech: Rijeka, Croatia, 2010; pp. 313–338.
- 3. Yuen, K.V.; Kuok, S.C. Efficient Bayesian sensor placement algorithm for structural identification: A general approach for multi-type sensory systems. *Earthq. Eng. Struct. Dyn.* **2015**, *44*, 757–774. [CrossRef]
- 4. Ghosh, A.; Das, S.K. Coverage and connectivity issues in wireless sensor networks: A survey. *Pervasive Mob. Comput.* **2008**, *4*, 303–334. [CrossRef]
- Sangwan, A.; Singh, R.P. Survey on coverage problems in wireless sensor networks. *Wirel. Pers. Commun.* 2015, *80*, 1475–1500. [CrossRef]
- 6. Mulligan, R.; Ammari, H.M. Coverage in wireless sensor networks: A survey. *Netw. Protoc. Algorithms* **2010**, 2, 27–53.
- Arampatzis, T.; Lygeros, J.; Manesis, S. A survey of applications of wireless sensors and wireless sensor networks. In Proceedings of the 2005 IEEE International Symposium on, Mediterrean Conference on Control and Automation Intelligent Control, Limassol, Cyprus, 27–29 June 2005; pp. 719–724.
- 8. Puccinelli, D.; Haenggi, M. Wireless sensor networks: Applications and challenges of ubiquitous sensing. *IEEE Circuits Syst. Mag.* **2005**, *5*, 19–31. [CrossRef]
- Khalil, N.; Abid, M.R.; Benhaddou, D.; Gerndt, M. Wireless sensors networks for Internet of Things. In Proceedings of the 2014 IEEE Ninth International Conference on Intelligent Sensors, Sensor Networks and Information Processing (ISSNIP), Singapore, 21–24 April 2014; pp. 1–6.
- 10. Ishmanov, F.; Bin Zikria, Y. Trust mechanisms to secure routing in wireless sensor networks: Current state of the research and open research issues. *J. Sens.* **2017**, 2017. [CrossRef]
- 11. Zou, Y.; Chakrabarty, K. Sensor deployment and target localization in distributed sensor networks. *ACM Trans. Embed. Comput. Syst. (TECS)* **2004**, *3*, 61–91. [CrossRef]
- 12. Megerian, S.; Koushanfar, F.; Potkonjak, M.; Srivastava, M.B. Worst and best-case coverage in sensor networks. *IEEE Trans. Mob. Comput.* 2005, *4*, 84–92.
- 13. Lee, C.; Shin, D.; Bae, S.W.; Choi, S. Best and worst-case coverage problems for arbitrary paths in wireless sensor networks. *Ad Hoc Netw.* **2013**, *11*, 1699–1714. [CrossRef]
- Poduri, S.; Sukhatme, G.S. Constrained coverage for mobile sensor networks. In Proceedings of the 2004 IEEE International Conference on Robotics and Automation, New Orleans, LA, USA, 26 April–1 May 2004; pp. 165–171.
- 15. Akewar, M.C.; Thakur, N.V. A study of wireless mobile sensor network deployment. *Int. J. Comput. Wirel. Commun.* **2012**, *2*, 533–540.
- Camp, T.; Boleng, J.; Davies, V. A survey of mobility models for ad hoc network research. *Wirel. Commun. Mob. Comput.* 2002, 2, 483–502. [CrossRef]
- 17. Rezazadeh, J. Mobile wireles sensor networks overview. Int. J. Comput. Commun. Netw. (IJCCN) 2012, 2, 1.
- 18. Batalin, M.A.; Sukhatme, G.S. Coverage, exploration and deployment by a mobile robot and communication network. *Telecommun. Syst.* **2004**, *26*, 181–196. [CrossRef]
- Amundson, I.; Koutsoukos, X.D. A survey on localization for mobile wireless sensor networks. In Proceedings of the International Workshop on Mobile Entity Localization and Tracking in GPS-less Environments, Orlando, FL, USA, 30 September 2009; pp. 235–254.
- 20. Cortes, J.; Martinez, S.; Karatas, T.; Bullo, F. Coverage control for mobile sensing networks. *IEEE Trans. Robot. Autom.* **2004**, *20*, 243–255. [CrossRef]
- 21. Senouci, M.R.; Mellouk, A.; Asnoune, K.; Bouhidel, F.Y. Movement-assisted sensor deployment algorithms: A survey and taxonomy. *IEEE Commun. Surv. Tutor.* **2015**, *17*, 2493–2510. [CrossRef]
- 22. Senouci, M.R.; Mellouk, A.; Oukhellou, L.; Aissani, A. WSNs deployment framework based on the theory of belief functions. *Comput. Netw.* 2015, *88*, 12–26. [CrossRef]
- 23. Arfken, G.; Weber, H.J.; Harris, F. *Mathematical Methods for Physicists*, 7th ed.; Acadamic Press: Orlando, FL, USA, 2013.

- Meguerdichian, S.; Koushanfar, F.; Qu, G.; Potkonjak, M. Exposure in wireless ad-hoc sensor networks. In Proceedings of the 7th Annual International Conference on Mobile Computing and Networking, Rome, Italy, 16–21 July 2001; pp. 139–150.
- 25. Meguerdichian, S.; Slijepcevic, S.; Karayan, V.; Potkonjak, M. Localized algorithms in wireless ad-hoc networks: Location discovery and sensor exposure. In Proceedings of the 2nd ACM International Symposium on Mobile Ad Hoc Networking & Computing, Long Beach, CA, USA, October 2001; pp. 106–116.
- Veltri, G.; Huang, Q.; Qu, G.; Potkonjak, M. Minimal and maximal exposure path algorithms for wireless embedded sensor networks. In Proceedings of the 1st International Conference on Embedded Networked Sensor Systems, Los Angeles, CA, USA, 5–7 November 2003; pp. 40–50.
- 27. Djidjev, H.N. Approximation algorithms for computing minimum exposure paths in a sensor field. *ACM Trans. Sens. Netw.* (*TOSN*) **2010**, *7*, 1–25. [CrossRef]
- Liu, L.; Zhang, X.; Ma, H. Minimal exposure path algorithms for directional sensor networks. *Wirel. Commun. Mob. Comput.* 2014, 14, 979–994. [CrossRef]
- Binh, H.T.T.; Binh, N.T.M.; Hoang, N.H.; Tu, P.A. Heuristic algorithm for finding maximal breach path in wireless sensor network with omnidirectional sensors. In Proceedings of the 2016 IEEE Region 10 Humanitarian Technology Conference (R10-HTC), Agra, India, 21–23 December 2016; pp. 1–6.
- 30. Megerian, S.; Koushanfar, F.; Qu, G.; Veltri, G.; Potkonjak, M. Exposure in wireless sensor networks: Theory and practical solutions. *Wirel. Netw.* **2002**, *8*, 443–454. [CrossRef]
- Song, Y.; Liu, L.; Ma, H.; Vasilakos, A.V. A Biology-Based Algorithm to Minimal Exposure Problem of Wireless Sensor Networks. *IEEE Trans. Netw. Serv. Manag.* 2014, 11, 417–430. [CrossRef]
- 32. Liu, L.; Zhang, X.; Ma, H. Percolation theory-based exposure-path prevention for wireless sensor networks coverage in internet of things. *IEEE Sens. J.* **2013**, *13*, 3625–3636. [CrossRef]
- 33. Miao, Y.; Wang, Y.; Wei, J.-X. Hybrid particle swarm algorithm for minimum exposure path problem in heterogeneous wireless sensor network. *Int. J. Wirel. Mob. Comput.* **2015**, *8*, 74–81. [CrossRef]
- Ye, M.; Wang, Y.; Dai, C.; Wang., X. A hybrid genetic algorithm for the minimum exposure path problem of wireless sensor networks based on a numerical functional extreme modele. *IEEE Trans. Veh. Technol.* 2016, 65, 8644–8657. [CrossRef]
- 35. Binh, H.T.T.; Binh, N.T.M.; Ngoc, N.H.; Ly, D.T.H.; Nghia, N.D. Efficient approximation approaches to minimal exposure path problem in probabilistic coverage model for wireless sensor networks. *Appl. Soft Comput.* **2019**, *76*, 726–743. [CrossRef]
- 36. Binh, N.T.M.; Binh, H.T.T.; Van Linh, N.; Yu, S. Efficient meta-heuristic approaches in solving minimal exposure path problem for heterogeneous wireless multimedia sensor networks in internet of things. *Appl. Intell.* **2020**, 1–19. [CrossRef]
- 37. Feng, H.; Luo, L.; Wang, Y.; Ye, M.; Dong, R. A novel minimal exposure path problem in wireless sensor networks and its solution algorithm. *Int. J. Distrib. Sens. Netw.* **2016**, *12*, 1550147716664245. [CrossRef]
- Zhang, L.; Chen, X.; Fan, J.; Wang, D.; Lin, C.K. The minimal exposure path in mobile wireless sensor network. In Proceedings of the 2015 Seventh International Symposium on Parallel Architectures, Algorithms and Programming (PAAP), Nanjing, China, 12–14 December 2015; pp. 73–79.
- Binh, N.T.M.; Thang, C.M.; Nghia, N.D.; Binh, H.T.T. Genetic algorithm for solving minimal exposure path in mobile sensor networks. In Proceedings of the 2017 IEEE Symposium Series on Computational Intelligence (SSCI), Honolulu, HI, USA, 27 November–1 December 2017; pp. 1–8.



© 2020 by the authors. Licensee MDPI, Basel, Switzerland. This article is an open access article distributed under the terms and conditions of the Creative Commons Attribution (CC BY) license (http://creativecommons.org/licenses/by/4.0/).