



PAPER • **OPEN ACCESS**

## Proposal of a wireless sensor network for footstep localization and optimization of its location using bio-inspired metaheuristics

To cite this article: Luis Sánchez-Márquez *et al* 2024 *Meas. Sci. Technol.* **35** 086309

View the [article online](#) for updates and enhancements.

### You may also like

- [Metaheuristic optimization approaches to predict shear-wave velocity from conventional well logs in sandstone and carbonate case studies](#)  
Mohammad Emami Niri, Rasool Amiri Kolajoobi, Mohammad Khodaiy Arbat et al.
- [Power generation from footsteps using piezo electronic sensors](#)  
Christeena Joseph, G Sai Krishna, M Bhaskar Sai et al.
- [Nonlinear model and optimization method for a single-axis linear-motion energy harvester for footstep excitation](#)  
Michael N Struwig, Riaan Wolhuter and Thomas Niesler

# Breath Biopsy Conference



Join the conference to explore the **latest challenges** and advances in **breath research**, you could even **present your latest work!**



5th & 6th November  
Online



Main talks





Early career  
sessions



Posters

**Register now for free!**

# Proposal of a wireless sensor network for footstep localization and optimization of its location using bio-inspired metaheuristics

Luis Sánchez-Márquez<sup>1</sup> , Mario Alfredo Reyes-Barranca<sup>1,\*</sup>,  
Griselda Stephany Abarca-Jiménez<sup>2</sup>, Andrea López-Tapia<sup>1</sup>   
and Luis Martín Flores-Nava<sup>1</sup>

<sup>1</sup> Departamento de Ingeniería Eléctrica, Cinvestav, Ciudad de México 07360, Mexico

<sup>2</sup> Unidad Profesional Interdisciplinaria en Ingeniería Campus Hidalgo Instituto Politécnico Nacional, Hidalgo 42162, Mexico

E-mail: [mreyes@cinvestav.mx](mailto:mreyes@cinvestav.mx), [luis.sanchez.m@cinvestav.mx](mailto:luis.sanchez.m@cinvestav.mx), [gabarcaj@ipn.mx](mailto:gabarcaj@ipn.mx), [andrea.lopez@cinvestav.mx](mailto:andrea.lopez@cinvestav.mx) and [lmflores@cinvestav.mx](mailto:lmflores@cinvestav.mx)

Received 10 March 2024, revised 28 April 2024

Accepted for publication 7 May 2024

Published 15 May 2024



CrossMark

## Abstract

This paper introduces a proposal for a wireless network of smart sensors designed to capture the vibrations produced by the impact of a person's footsteps on a dispersive and damped floor. These vibrations are utilized to pinpoint the location of impacts using a heuristic algorithm, enabling operation at a low data transmission/reception rate. Traditionally, the placement of sensors within a room has not been given significant consideration in the context of localization; however, our findings indicate that a detailed analysis of both the number and location of sensors can substantially enhance the accuracy of footstep localization. In this study, we have optimized the arrangement and quantity of sensors through the application of bio-inspired metaheuristics, aiming to minimize localization errors across the room. Upon evaluating various bio-inspired metaheuristic optimization algorithms, we identified the one that yielded the lowest estimation errors for the room as a whole. Our experimental tests demonstrate that such optimization significantly enhances the efficacy of the localization algorithm, resulting in a reduction of localization error ranging from 18.24% to 46.78% across different trajectories.

Keywords: bio-inspired metaheuristics, footstep localization, occupant tracking, optimization, sensor network

## 1. Introduction

Tracking a person inside a building has multiple applications. For example, by following a person's path, an intruder can be tracked. In an emergency rescue/evacuation scenario, first

responders can locate occupants in distress. In smart homes, if the location of a person is known, energy consumption of lamps, air conditioning, etc, can be reduced [1]. In the health area, monitoring a person's daily walking activity can be used to estimate the energy expenditure during the day and thus prevent, delay, or control some chronic diseases (diabetes, cardiovascular and respiratory diseases, obesity) [2].

To locate and track the trajectory followed by a person (also called a source or objective), a network of sensors (also called nodes) can be used. These sensors can measure some excitation of the medium produced by the objective. Traditional methods for locating objectives include those

\* Author to whom any correspondence should be addressed.



Original content from this work may be used under the terms of the [Creative Commons Attribution 4.0 licence](https://creativecommons.org/licenses/by/4.0/). Any further distribution of this work must maintain attribution to the author(s) and the title of the work, journal citation and DOI.

based on the time of arrival of the signal (TOA), time differences of arrival (TDOA), angle of arrival (AOA), and received signal strength (RSS) [3]. These localization methods were designed mainly for radio frequency (RF) or Wi-Fi systems, where the signals to be studied propagate through the air and the propagation speed is considered as a constant.

As it was already reported, it is possible to locate a person inside a building by measuring the vibrations produced by the person's footsteps while walking. These vibrations are usually measured using accelerometers placed at specific locations on the floor of the room [1, 4–8].

Regarding this approach, it is important to remark that this is a non-invasive technique, i.e. the person does not wear the sensors nor is their activity recorded with cameras or microphones. This respects their privacy and prevents the person under study from modifying their behavior when they know they are being observed, masking, in consequence, the analysis and leading to unreal results.

Concrete is a dispersive and damped medium, i.e. mechanical vibrations are distorted as they travel through the floor, and then the propagation velocity is not a constant [7]. Therefore, the traditional methods (TOA, TDOA, AOA, RSS) are not good options to be used in this medium [1]. For this reason, new algorithms have been proposed based on traditional algorithms that overcome the disadvantage of working in dispersive and damped mediums such as concrete.

With this in mind, indoor footstep localization algorithms consider different levels of complexity and infrastructure requirements. For instance, some of these methods need to store and process several samples of the signal. Alajlouni, *et al* [4], Alajlouni and Tarazaga [5], and Alajlouni and Tarazaga [6] used RSS-based methods, so it was necessary to calculate the average power of the accelerometer signal within a given time window. Poston *et al* [1] stored the acceleration signal and studied the vibrations within a time window; then localization was done using a method based on TDOA. Li *et al* [8] stored and processed the complete signals coming from triaxial accelerometers, then applied a hybrid localization method combining the TDOA and AOA methods.

These methods reported sub-metric errors in footstep localization; however, the heuristic SO-TDOA (sign of the time differences of arrival) algorithm proposed in Bahroun *et al* [7], can deliver sub-metric errors without storing all the footstep information, since it only requires knowing the TOA of the signal. This TOA can be estimated by a threshold method (when the signal amplitude crosses a specific value).

As a sample of the complexity of these kinds of systems, localization algorithms are usually tested in rooms with highly instrumented floors such as Goodwin Hall [9]. This is a building designed for a variety of applications, including structural health monitoring, building dynamics, and human motion. This building is equipped with high-sensitivity piezoelectric sensors installed under the floor and attached to the structural supports of the building.

When such highly equipped facilities are not available, sensors are usually placed in locations such that they are uniformly distributed throughout the room. For example, if four

sensors are used, one sensor is usually placed in each corner of the room. However, we will show that if a study of the location of the sensors within the room is performed, the localization of the footsteps can be improved. The complexity of the study will depend on the localization algorithm used. In this study, we use the SO-TDOA algorithm, since it only requires the value of the times of arrival, thereby simplifying the system and reducing data storage requirements.

Regarding the behavior of the mechanical wave produced by a footstep, it is worthy to mention the properties of the range of frequencies present in the traveling wave along a floor. For instance, lower frequencies contribute most of the energy of the wave-packet and are the most affected by dispersion, as well.

On the other hand, frequency-dependent damping is present mainly at higher frequencies. A complete theoretical analysis about dispersion and damping is presented by [4, 7] considering classical plate theory. Therefore, dispersion and damping will be present in every floor considered, and particular experimental tests should be carried out for each kind of floor. Experimental results using an accelerometer as a sensor are reported for concrete, concrete covered by carpet, and concrete partly dusted with sand in [10]. Finally, it should be mentioned that concrete was selected for the present study since it is the most used material for construction in Mexico, and because it is the intention to adapt the system proposed in a regular living environment.

From the aforementioned device and material characteristics, a clear division of tasks can be identified for the proposal reported here. First, considering the dispersive and damping characteristics of the material used as the test floor, it is imperative and highly important to measure how the footstep signal arrives to sensors located at different positions used in the designed system, while accounting for the requirements of the SO-TDOA algorithm. Once tests and characterization conducted under different conditions deliver reliable and reproducible results, implementation of the measurement system can proceed. Second, and equally important, is ensuring a clean and high quality of the footstep signal. This was achieved after conducting a thorough characterization of the capacitive accelerometers, following the programming options outlined in datasheet [11], which can provide better operating conditions for our goal. Thus, the selection of a reliable and low-cost accelerometer should yield high-quality, clear, and easily measurable footstep signals when integrated with the proposed system.

With this in mind, this paper contributes to the field of measurements by proposing a novel wireless network of smart sensors to capture the vibrations produced by the impact of a person's footsteps on a concrete floor. In this work, a smart sensor is defined as a device that measures floor vibrations, performs threshold crossing detection, and is part of a sensor network.

In contrast to literature [1, 4–8], our network is designed to allow nodes to transmit a single datum per footstep, facilitating low data transmission/reception rates. Additionally, the network conducts threshold crossing detection directly at the

nodes, so it is not necessary to perform this calculation in the computer, thereby simplifying the system. Furthermore, it is engineered to determine the TOA of signals without needing to store comprehensive data on each footstep.

In addition, in [1, 4–8] the precise location of sensors was not a primary focus. Therefore, in this study, both the number of sensors and their location in the room are optimized using bio-inspired metaheuristics to minimize the footstep localization error, demonstrating that strategic application of the SO-TDOA algorithm can substantially mitigate localization errors.

### 1.1. Optimization

Optimization is the process in which the minimum or maximum of a function is determined. This function is known as the objective function. All possible solutions to the optimization problem are found within a bounded space called the search space. A local minimum of a function refers to a point where the value of the function is less than or equal to nearby points within a certain neighborhood. On the other hand, a global minimum is a point where the value of the function is less than or equal to all other points in its domain.

Optimization methods can be divided into derivative and non-derivative methods [12]. Derivative methods require knowledge of the derivative of the objective function, while non-derivative methods require information (evaluation) of the objective function and not of its derivative. Traditional optimization methods are calculus-based methods or based on random searches [13]. The main disadvantage of these methods is that they are local optimizers, i.e. they can get trapped in a local minimum of the objective function.

However, these methods can be employed as global optimizers by utilizing multi-start initialization techniques [14, 15]. In other words, multiple local searches are conducted from different starting points, allowing for exhaustive exploration of the search space, avoiding local optimality. The local minimum with the best objective function value is considered the global optimum [14].

The search process in optimization algorithms ends when some convergence criterion is met. These criteria are employed to determine when an optimization algorithm has achieved a satisfactory result or has sufficiently approached the optimum. Some common convergence criteria in optimization algorithms include terminating the algorithm when the change in the function value between two consecutive iterations is small, when the partial derivatives (gradient components) of the function are small, when the change in the design variables between two consecutive iterations is small, or when a certain number of iterations is reached. The choice of convergence criteria depends on the specific problem and the optimization method used [12].

**1.1.1. Modern optimization methods.** Approximate algorithms present an alternative methodology to conventional optimization techniques. These algorithms can be divided into

heuristics and metaheuristics. ‘Meta’ comes from Greek and means upper level, while ‘heuristic’ refers to the art of discovering new strategies [13, 16].

Heuristic methods offer practical problem-solving techniques that produce satisfactory solutions, although they may not necessarily be optimal [17]. Metaheuristics are higher-level heuristics that seek, generate, or select lower-level heuristics (partial search algorithms) capable of providing a sufficiently good solution to an optimization problem [13].

Metaheuristic algorithms possess the capability to extensively explore the search space, thereby enabling global optimization without being confined to local minima. The ability to overcome local minima in the objective function enhances the likelihood of finding the global minimum and is referred to as the robustness of the method [18].

While metaheuristics can provide a sufficiently good solution to an optimization problem, they may not necessarily be the optimal [13, 18].

The different modern optimization methods include bio-inspired population-based metaheuristic algorithms; these are stochastic global optimizers inspired by nature. In these algorithms, a function is optimized by evaluating individuals, where each individual represents a potential solution [19].

These algorithms are iterative; they start with random solutions, and with each iteration, the individuals move within the search space trying to reach the minimum of the objective function. Individuals share and store information of the search space, help each other to avoid local minima, and use memory to store the best solution obtained so far. The direction and magnitude of the displacements of the individuals depend on the algorithm used.

Since there are many problems (objective functions) with different characteristics (conditions and constraints), there is no single algorithm that universally optimizes better than the others [20]. This is why a wide variety of bio-inspired metaheuristic algorithms have been developed.

Some of the most popular are particle swarm optimization (PSO) [21], artificial bee colony (ABC) [22], grey wolf optimizer (GWO) [23] and combinations of them, to take advantage of the strengths of each method. For example, the Hybrid PSO-GWO (HPSGWO) algorithm, proposed by Şenel *et al* [24], takes advantage of GWO exploration (global search) and PSO exploitation (local search) [25].

**1.1.2. PSO.** The PSO algorithm [21] is inspired by the social behavior and movement dynamics of flocks of birds and schools of fish. In this algorithm, each particle represents a potential solution.

This algorithm finds the best global solution by adjusting the displacement of each particle according to its personal best position and the best global position of the particles of the whole swarm. There are different variants of the original PSO algorithm, with the objective of improving it [26, 27].

Each particle  $i$  has its own trajectory, i.e. it has a position  $x_i$  and a velocity  $v_i$ , and moves in the search space updating its trajectory according to the following equation:



$$v_i(t+1) = wv_i(t) + c_1r_1[x_i^*(t) - x_i(t)] + c_2r_2[x^g(t) - x_i(t)]$$

$$x_i(t+1) = x_i(t) + v_i(t+1), i = 1, 2, \dots, M$$

where  $t$  is the current iteration;  $M$  is the number of particles in the swarm;  $c_1$  and  $c_2$  are acceleration constants, which are named cognitive and social parameter, respectively;  $r_1$  and  $r_2$  are uniform random numbers within  $[0, 1]$ ;  $x_i^*$  is the particle best solution;  $x^g$  is the global best solution;  $w = \frac{MaxIt-t}{MaxIt}$  is called the inertia weight, which modifies the particle velocity [26].

The algorithm keeps searching for solutions until a stopping criterion is met, which can be a maximum number of iterations,  $t = MaxIt$ .

**1.1.3. ABC.** The ABC algorithm [22] is inspired by the foraging behavior of bees. In the artificial colony, bees are divided into two main classes: employed bees and unemployed bees, with the latter further divided into onlooker bees and scout bees.

In the ABC algorithm, food sources represent potential solutions to the problem; the amount of nectar in a food source corresponds to the quality of the solution (fitness). All employed bees are associated with specific food sources (solutions).

The algorithm consists of three phases: employed bees phase, onlooker bees phase, and scout bees phase.

First, the employed bees are sent to random food sources  $\mathbf{x}_i$  and the employed bees phase begins. In this phase, employed bees search within their neighborhood for a new food source with more nectar, conducting a local search and staying at the source with the best fitness (a greedy selection is applied between the two sources). Then, the employed bees share their information with onlooker bees.

The onlooker bees phase begins when all employed bees have shared their food source information. In this phase, the onlookers select a food source  $i$  with a probability  $P_i$  given by:

$$P_i = \frac{f_i}{\sum_{j=1}^M f_j}$$

where  $f_i$  is the fitness of the food source  $i$  and  $M$  is the total number of food sources. This fitness is given by:

$$f_i = \begin{cases} 1/(1+f(\mathbf{x}_i)), & \text{if } f(\mathbf{x}_i) \geq 0 \\ 1+|f(\mathbf{x}_i)|, & \text{if } f(\mathbf{x}_i) < 0 \end{cases}$$

where  $f(\mathbf{x}_i)$  is the objective function evaluated at food source location  $\mathbf{x}_i$ .

Once the food source has been selected, a local search is performed as in the employed bees phase, i.e. the onlooker bees search for another food source in their neighborhood and stay at the source with the best fitness. If the nectar of the food sources decreases or is exhausted (if a local minimum was found) after a certain number of attempts (local search

abandoning limit), then the employed bees of those sources become scout bees.

In the scout bee phase, scout bees randomly search for a new food source  $\mathbf{x}_i$  and afterward, they again become employed bees. Finally, the best food source (solution) found so far is memorized.

These three phases are repeated until a stop criterion is met, which can be a maximum number of iterations.

**1.1.4. GWO.** The GWO algorithm [23] is inspired by the hunting behavior of gray wolf packs, where each wolf has a role and hierarchy within the group. The alpha wolves ( $\alpha$ ) are the leaders, the betas ( $\beta$ ) are the second in hierarchical level, followed by the deltas ( $\delta$ ) and finally the omegas ( $\omega$ ).

In the GWO algorithm, each wolf represents a potential solution to the optimization problem, and the prey symbolizes the minimum (or maximum) to be found within the search space. It is posited that there is only one alpha ( $\alpha$ ), one beta ( $\beta$ ), and one delta ( $\delta$ ), with all other potential solutions being considered omega ( $\omega$ ) wolves.

The alpha, beta, and delta wolves are assumed to be the closest to the prey, and thus, to determine the hierarchy among the wolves, they are evaluated against the function to be optimized. Accordingly, the alpha is the best solution, the beta is the second-best solution, and the delta is the third-best solution. The remaining solutions are considered omegas.

Then, the positions of the three solutions ( $\mathbf{x}_\alpha, \mathbf{x}_\beta, \mathbf{x}_\delta$ ) are stored and the rest of the wolves (solutions  $\mathbf{x}_i$ ) update their position relative to the top three as follows:

$$\mathbf{x}_i(t+1) = \frac{\mathbf{x}_1 + \mathbf{x}_2 + \mathbf{x}_3}{3}$$

where  $t$  is the current iteration,

$$\mathbf{x}_1 = \mathbf{x}_\alpha - \mathbf{A}_1 \cdot \mathbf{D}_\alpha; \mathbf{x}_2 = \mathbf{x}_\beta - \mathbf{A}_2 \cdot \mathbf{D}_\beta; \mathbf{x}_3 = \mathbf{x}_\delta - \mathbf{A}_3 \cdot \mathbf{D}_\delta$$

$$\mathbf{D}_\alpha = |\mathbf{C}_1 \cdot \mathbf{x}_\alpha - \mathbf{x}_i|; \mathbf{D}_\beta = |\mathbf{C}_2 \cdot \mathbf{x}_\beta - \mathbf{x}_i|; \mathbf{D}_\delta = |\mathbf{C}_3 \cdot \mathbf{x}_\delta - \mathbf{x}_i|.$$

Vectors  $\mathbf{A}$  and  $\mathbf{C}$  are coefficient vectors that determine the movement of each omega wolf and are calculated as follows:

$$\mathbf{A} = 2\mathbf{a} \cdot \mathbf{r}_1 - \mathbf{a}; \mathbf{C} = 2 \cdot \mathbf{r}_2$$

where the components of  $\mathbf{a}$  decrease linearly in each iteration from 2 to 0. The vectors  $\mathbf{r}_1$  and  $\mathbf{r}_2$  are random vectors in the range  $[0, 1]$ .

This algorithm balances exploration and exploitation. When the magnitude of vector  $|\mathbf{A}| > 1$  the wolf moves away from the prey (exploration), but when  $|\mathbf{A}| < 1$ , it approaches the prey (exploitation). This dynamic helps the algorithm avoid local minima, enabling global searches. However, although  $|\mathbf{A}|$  takes random values at each iteration, it tends to decrease due to the vector  $\mathbf{a}$ , which, decreases linearly with each iteration. On the other hand, the vector  $\mathbf{C}$  always takes random values, favoring exploration, avoiding local minima even in the last iterations.

**1.1.5. HPSGWO algorithm.** The HPSGWO algorithm [24] combines the PSO and GWO methods. This combination is beneficial because when the PSO approaches a good solution, the particles trend towards it, even if it is a local minimum, meaning the PSO's exploitation ability is high relative to its exploration ability [24].

The HPSGWO algorithm introduces some GWO-enhanced particles into the PSO to reduce the likelihood of the PSO algorithm being trapped in a local minimum. Particle substitution occurs with a low probability (*prob*), allowing the main loop of the PSO to run while some particles enter a nested loop of the GWO.

The nested loops mean the HPSGWO algorithm runs longer than the original PSO and GWO, so the number of iterations (*MaxItW*), probability (*prob*) and wolf population ( $N_w$ ) of the nested GWO loop are kept small. For example,  $N_w = 10$ , *MaxItW* = 10, *prob* = 0.01 [24].

When new optimization algorithms are proposed, they are typically compared with other algorithms to demonstrate their effectiveness by optimizing specific functions. In this work, we do not aim to develop a new optimization algorithm; however, a comparison of the previously mentioned bio-inspired metaheuristics applied to the SO-TDOA algorithm is made to select the one that best optimizes the location of the sensors within a room.

## 2. Materials and methods

In order to propose a smart sensor network and to optimize the positioning of the sensors inside the room, it is necessary to define some concepts and give a brief review of the heuristic SO-TDOA algorithm proposed in Bahroun *et al* [7].

The algorithm assumes that the order of arrival of the signals at different sensors is preserved (even in a dispersive and damped medium such as concrete); that is, the signal arrives first at the sensors closest to the source of vibration. This can be expressed by equation (1),

$$\text{sign}(t_{si} - t_{sj}) = \text{sign}(d_{si} - d_{sj}) \quad (1)$$

where  $\text{sign}(\cdot)$  is the sign operator;  $(i, j)$  are a pair of sensors;  $d_{si}$  is the distance between the position of the source ( $\mathbf{p}_s$ ) and the sensor  $i$ ;  $t_{si}$  is the TOA from the source to the sensor  $i$ .

The TOA is defined as the time when the wave, produced by a hit over the floor, arrives from the vibration source to the sensor's location and can be estimated by a threshold method (see figure 1).

The algorithm is divided into two stages: The floor segmentation stage and the measurement and localization stage.

### 2.1. Floor segmentation stage

First, the test surface is selected, and the sensors are placed in known locations. Then, the test room is divided into  $Q$  regions, where each region ( $R_k$ ) is formed by the perpendicular bisectors of the line segment joining each pair of sensors (see figure 2(a)). Each region will have a characteristic

vector  $\mathbf{z}_k$  whose elements take values  $+1$  or  $-1$ , as can be seen in equation (2), where each element of the vector is given by the sign of the differences of distance between a pair of sensors  $(i, j)$  and a region  $R_k$ ,

$$z_k(l) = \text{sign}(d_{ki} - d_{kj}); l = \frac{(j-2)(j-1)}{2} + i \quad (2)$$

$$\forall (i, j) \in \{(1, 2), (1, 3), (2, 3), (1, 4), \dots, (N-1, N)\}$$

where  $N$  is the number of sensors,  $d_{ki}$  is the distance between region  $R_k$  and sensor  $i$ ,  $d_{kj}$  is the distance between region  $R_k$  and sensor  $j$ .

Finally, the centroid  $\mathbf{p}_k^c$  of each region  $R_k$  is calculated (see figure 2(b)). This segmentation stage is done only once, just after the sensors are fixed on the floor.

### 2.2. Measurement and localization stage

Afterwards, a foot impact is made on the floor and the TOA from the source to each sensor  $i$  is estimated ( $\hat{t}_{si}$ ). TOA is estimated with a threshold method, starting from an arbitrary time  $t_0$ . After that, the characteristic vector of the source ( $\mathbf{z}_s$ ) is calculated, which is given by the sign of the measured TDOA. The elements of  $\mathbf{z}_s$  take values  $+1$  or  $-1$ .

$$z_s(l) = \text{sign}(\hat{t}_{si} - \hat{t}_{sj}) \quad l = \frac{(j-2)(j-1)}{2} + i \quad (3)$$

Following, the set of regions  $M_r$  that minimize the Hamming distance to the measured characteristic vector  $\mathbf{z}_s$  is calculated.

$$M_r = \arg \min_{k \in [1 \dots Q]} \sum_{i=1}^{N(N-1)/2} (z_s(a) \oplus z_k(a)) \quad (4)$$

$$M_r \subset [1 \dots Q]$$

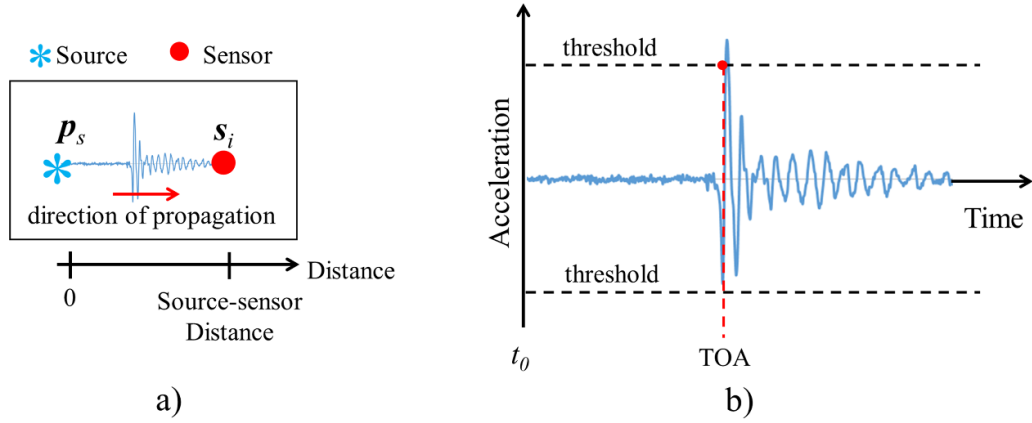
where  $\oplus$  is the exclusive or operator and the elements  $z_s(a)$  and  $z_k(a)$  represent the  $a$ th elements of the vectors  $\mathbf{z}_s$  and  $\mathbf{z}_k$ , respectively.

Finally, the source position  $\hat{\mathbf{p}}_s$  is estimated by averaging the centroids of the regions that minimize the Hamming distance to the measured vector.

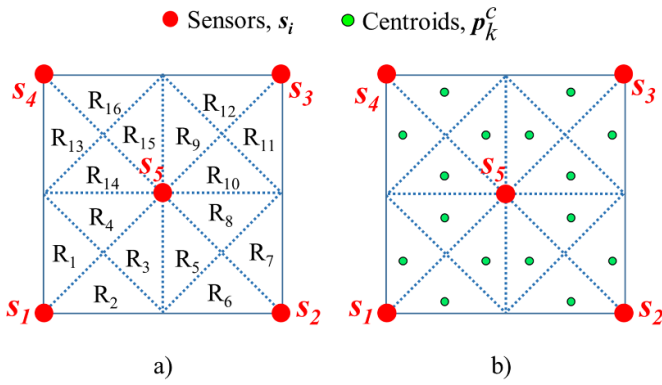
$$\hat{\mathbf{p}}_s = \frac{1}{|M_r|} \sum_{r \in M_r} \mathbf{p}_r^c \quad (5)$$

where  $|M_r|$  are the cardinal numbers of the set  $M_r$ .

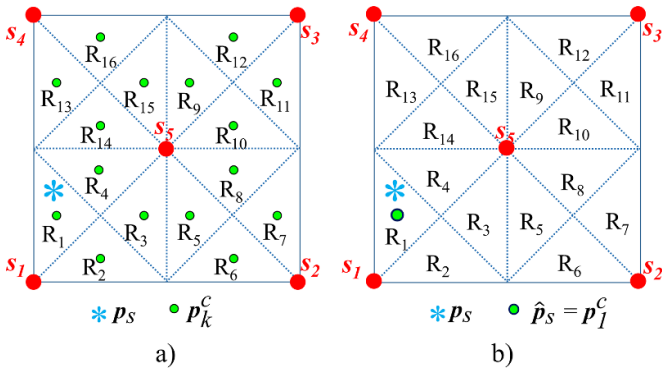
In the absence of measurement errors, the estimation of the location of the footstep  $\hat{\mathbf{p}}_s$  is given by the centroid of the region  $\mathbf{p}_k^c$  whose characteristic vector  $\mathbf{z}_k$  minimizes the Hamming distance to the vector  $\mathbf{z}_s$ . For example, in figure 3(a), the footstep  $\mathbf{p}_s$  is located within the region  $R_1$ , so the estimated position  $\hat{\mathbf{p}}_s$  will be the centroid  $\mathbf{p}_1^c$  as shown in figure 3(b). That is, the footstep localization error depends on the number and shape of the regions, and then the estimation error is directly related to



**Figure 1.** (a) The vibration propagates from the source located at  $p_s$  to sensor  $i$  located at  $s_i$ . (b) The time of arrival is estimated using a threshold method and measured with respect to an arbitrary time origin  $t_0$ .



**Figure 2.** Room segmentation. (a) Room divided into regions  $R_k$ , (b) Centroids of the regions  $p_k^c$ . The dotted lines are the bisectors that form the regions.



**Figure 3.** (a) A footprint  $p_s$  in a room with 16 regions (b) The estimated location of the footprint is at the centroid of the region where the footprint occurred.

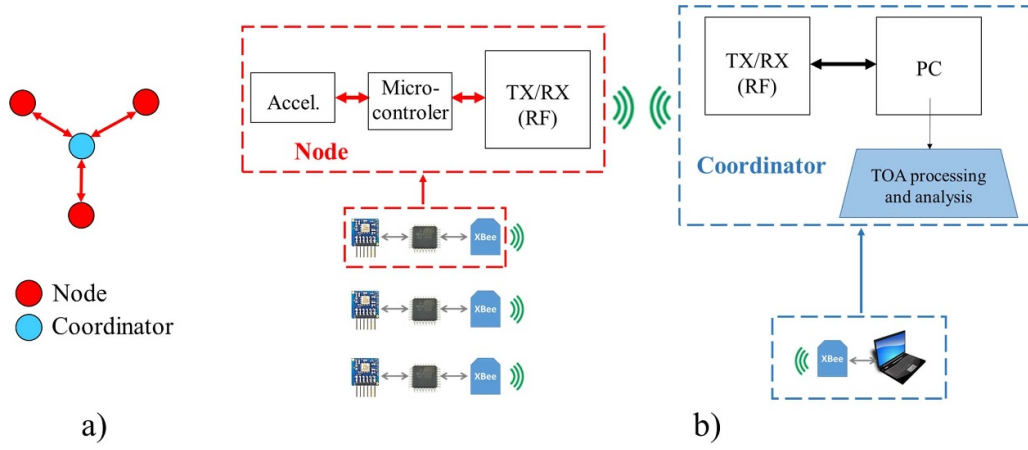
the number of sensors and their location. Finding the location of the sensors that minimizes the localization error is a prohibitive task if done manually, since changing the location of a single sensor completely modifies all the regions. Therefore, in this paper, the number of sensors and their location are computationally optimized using bio-inspired metaheuristics.

### 2.3. Proposed smart sensor network

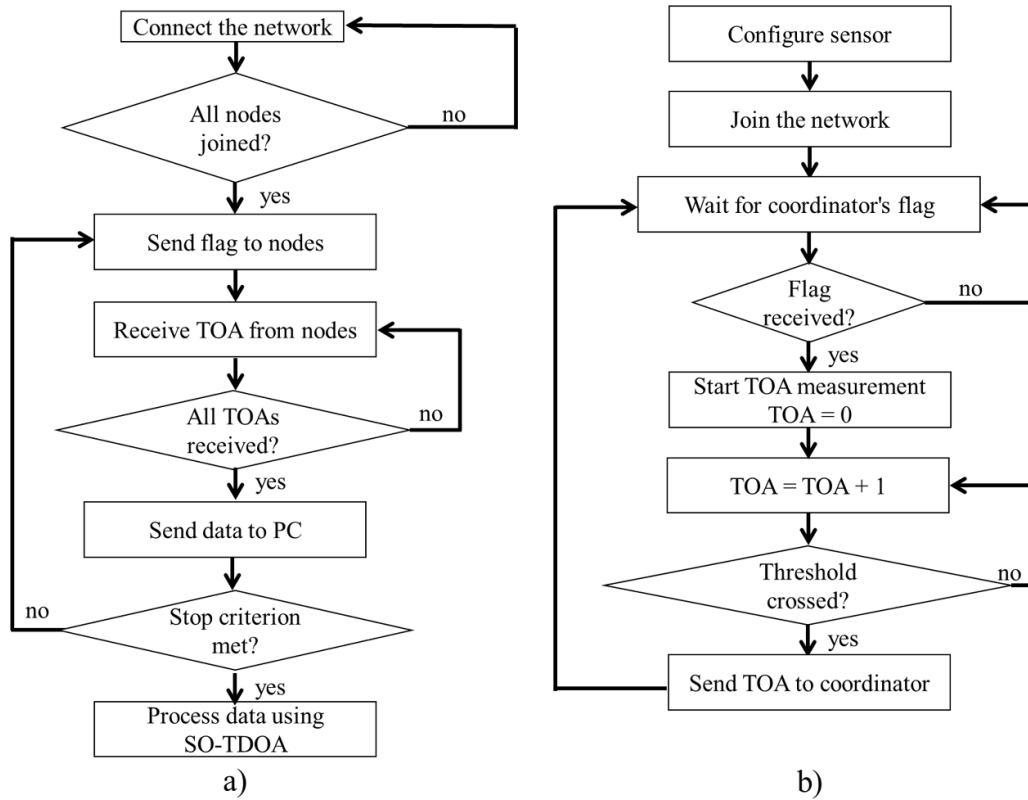
As seen, to perform the localization using the SO-TDOA algorithm, it is only necessary to know the TOA from the source to the sensors. This allows using smart sensors to detect threshold crossing without the need to store and analyze all the footprint information in a computer, which simplifies the system.

In this work, we present a novel wireless smart sensor network for footprint localization, where all nodes (smart sensors) communicate bidirectionally with a central node (Coordinator) using a point-to-multipoint topology, as illustrated in figure 4(a). Figure 4(b) shows a schematic of the proposed network. The nodes consist of a capacitive accelerometer to measure vibrations and detect threshold crossings, a microcontroller to configure and read the sensor, and a wireless transmitter/receiver (TX/RX) module to communicate with the coordinator. The coordinator consists of a TX/RX wireless module and a computer where data processing is performed.

Figure 5 depicts the flow diagrams for both the coordinator and the nodes. The first task of the Coordinator is to create the network. It then sends a flag to all nodes to instruct them to start the TOA measurement. Once the flag is received, the nodes begin measuring the TOA using a digital counter and continuously check if the vibration amplitude crosses the threshold. When a sensor detects that the signal crosses the threshold, it sends the TOA to the coordinator (see figure 6). This technique allows the nodes to send only one datum per footprint, enabling the wireless modules to operate at a low data transmission/reception rate. When the coordinator receives the TOAs from all nodes, it sends them to a computer to store them and subsequently estimate the location of the footprint using the second stage of the SO-TDOA algorithm. Once this is done, the flag is sent again to the nodes, and the cycle repeats continuously. It is important to mention that this first proposal of the system does not work in real time. The system stores all the measurements in the computer, and when a stopping criterion is met, then it estimates the trajectory followed by the person. The stopping criterion can be a certain time window of measurements or a certain number of footsteps.



**Figure 4.** (a) Point-to-multipoint topology, where the central node communicates bidirectionally with the other nodes in the network. (b) Block diagram of the proposed sensor network.



**Figure 5.** Flow diagrams of the sensor network. (a) Coordinator (b) Nodes.

In this work, we use the number of footsteps as a stopping criterion.

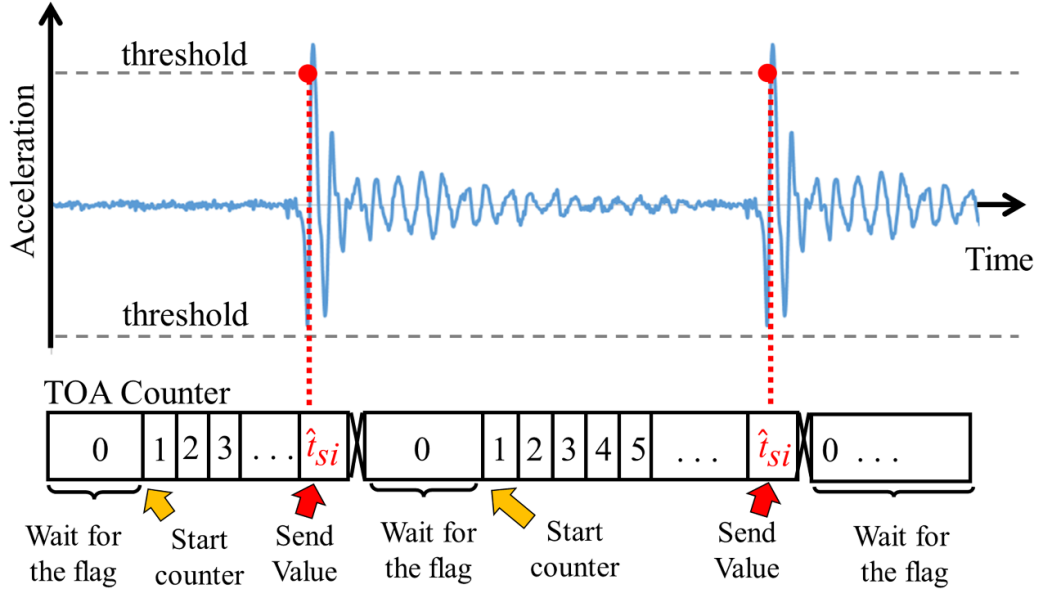
The proposed network requires a sensor capable of detecting small changes in acceleration caused by footsteps, an adequate bandwidth to detect floor vibrations (the dominant floor vibrations produced by a footstep are below 500 Hz [6]), and a threshold crossing detection circuit as well as a circuit to adjust the signal offset. EVAL-ADXL355-PMDZ [11] capacitive sensors were used together with an ATmega328P micro-controller of which 21% of its resources are used (the code uses 6492 bytes of 32 256 bytes available in the flash memory,

while the global variables use 440 bytes of 2048 bytes available in the SRAM). XBee 3.0 modules were used to form the sensor network [28].

In contrast to the literature where the footstep signal is sampled at 20 kHz [7] and transmitted complete for analysis, in this work, we propose to send to the computer only one datum per footstep. Assuming one footstep per second and five nodes, then, only five data per footstep would be stored instead of 100 000 data.

On the other hand, wireless modules consume 40 mA while transmitting data [28]. If only one datum is transmitted instead





**Figure 6.** TOA estimation. Sensor  $i$  starts measuring the time after receiving the flag from the coordinator. TOA is obtained when the signal crosses the threshold.

of 20 000 data, then there will be a saving in power consumption. However, a more in-depth study such as the one conducted in [29] is required to give an exact numerical value of the power consumption savings in our system.

Regarding data processing, in the literature, the threshold crossing detection is performed after transmitting the data. In this work, the threshold crossing detection is performed directly at the nodes with the help of the internal circuits of the sensor, so it is not necessary to perform this calculation in the computer, which simplifies the data processing.

#### 2.4. Sensor network optimization

As previously mentioned, the number of sensors and their location within the room can be obtained through optimization. Hereafter, we define the objective function.

**2.4.1. Objective function.** If footsteps are distributed throughout the room, it might seem intuitive that placing sensors uniformly throughout the room (see figure 7(a)) would yield the smallest average localization errors. However, changing the location of the sensors can lead to a larger number of centroids (see figure 7(b)) and thus decrease the localization error for the entire room.

Optimization is performed with footsteps uniformly distributed (see figure 7(c)). Hence, the optimization aims to find the sensor locations that minimize the mean squared error (MSE) of the location estimation of uniformly distributed footsteps.

Then the objective function (the function to be optimized) is given by the MSE, calculates as follows:

$$\text{MSE} = \frac{1}{N_p} \sum_{m=1}^{N_p} (error_m)^2 \quad (6)$$

where  $N_p$  is the number of footsteps and  $error_m$  is the localization error of the footstep  $m$  and is computed as

$$error_m = |\mathbf{p}_{sm} - \hat{\mathbf{p}}_{sm}|$$

$$error_m = \sqrt{(p_{sm_x} - \hat{p}_{sm_x})^2 + (p_{sm_y} - \hat{p}_{sm_y})^2} \quad (7)$$

where:

$\mathbf{p}_{sm} = (p_{sm_x}, p_{sm_y})$  is the actual location of the footstep  $m$ ,  $\hat{\mathbf{p}}_{sm} = (\hat{p}_{sm_x}, \hat{p}_{sm_y})$  is the estimated location of the footstep  $m$  using the SO-TDOA algorithm.

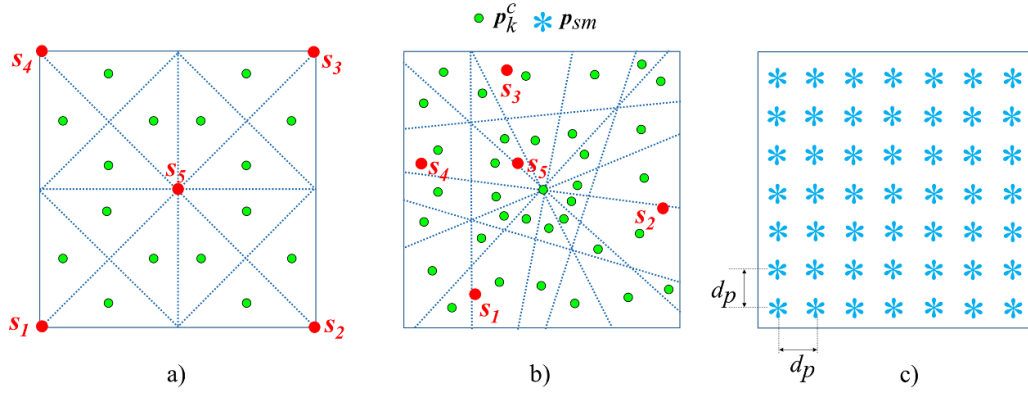
As previously discussed, in the SO-TDOA algorithm, the test room is divided into regions. The number and shape of these regions depends on both the quantity and the placement of the sensors. Consequently, the footstep localization error is substantially affected by these parameters.

By substituting equations (2)–(5) and (7) into equation (6), the objective function is obtained as follows:

MSE

$$= \frac{1}{N_p} \sum_{m=1}^{N_p} \left| \mathbf{p}_{sm} - \frac{1}{\arg \min_{k \in [1 \dots Q]} \sum_{i=1}^{N(N-1)/2} (\mathbf{z}_{sm}(a) \oplus \mathbf{z}_k(a))} \right| \times \sum_{r \in M_{rm}} \mathbf{p}_{rm}^c \Big|^2$$

where  $M_{rm}$  is the set of regions  $r$  that minimize the Hamming distance between the measured characteristic vector  $\mathbf{z}_{sm}$  and the characteristic vector  $\mathbf{z}_k$ .  $\oplus$  is the exclusive or operator,  $\mathbf{z}_{sm}(a)$  represents the  $a$ th element of the vectors  $\mathbf{z}_s$  of the  $m$ th footstep, and  $\mathbf{p}_{rm}^c$  is the centroid of the region  $r$  that minimize the Hamming distance.



**Figure 7.** Centroids of the regions of a room for (a) uniformly distributed sensors, (b) non-uniformly distributed sensors. (c) Footsteps uniformly distributed throughout the room.

Such that, the objective function depends on both the number of sensors  $N$  and their location  $s_i = (s_{ix}, s_{iy})$  due to the characteristic vector  $z_k$  (equation (2)), which can be rewritten as follows:

$$z_k(l) = \text{sign} \left( \sqrt{(s_{ix} - p_{kx}^c)^2 + (s_{iy} - p_{ky}^c)^2} - \sqrt{(s_{jx} - p_{kx}^c)^2 + (s_{jy} - p_{ky}^c)^2} \right); l = \frac{(j-2)(j-1)}{2} + i.$$

If the vector  $\mathbf{S} = [s_{1x} s_{1y} s_{2x} s_{2y} \dots s_{Nx} s_{Ny}]$  contains all the coordinates  $s_i = (s_{ix}, s_{iy})$  of the sensors, where  $i = 1, 2, \dots, N$  and  $N$  is the number of sensors, then the goal is to search for the vector  $\mathbf{S}$  that minimizes the MSE within a  $2N$ -dimensional search space. The boundaries of the objective function are given by the spatial limits of the room. Then, the optimization problem can be stated as follows:

$$\text{Minimize MSE}(\mathbf{S}) \text{ s.t. } \text{MSE}(\mathbf{S})_{s_{ix}} \in (0, U_x); s_{iy} \in (0, U_y)$$

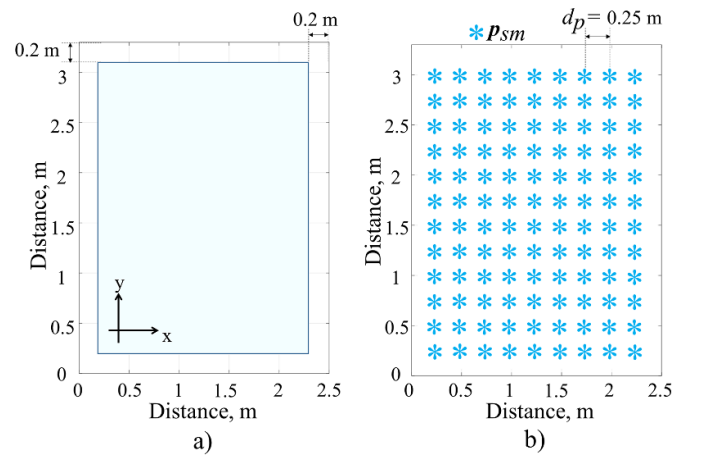
where  $U_x, U_y$  are the maximum values for the horizontal and vertical dimensions of the room, respectively.

**2.4.2. Characteristics of the optimization problem.** A room measuring  $2.5 \text{ m} \times 3.3 \text{ m}$  was selected for the study. This room featured an 11 cm thick concrete floor covered with ceramic tile and was situated on the second floor of a residential house.

The boundaries of the objective function were set to  $0.2 \text{ m} \leq s_{ix} \leq 2.3 \text{ m}$  and  $0.2 \text{ m} \leq s_{iy} \leq 3.1 \text{ m}$  as depicted in figure 8(a). The footsteps for optimization were evenly distributed and spaced  $0.25 \text{ m}$  apart, as illustrated in figure 8(b).

To determine the optimal number of sensors, the MSE was optimized using the HPSGWO algorithm for  $N = 3, 4 \dots 10$ . The population size was set to 50 individuals; the algorithm was executed for 200 iterations and repeated ten times. The parameters of the HPSGWO were  $c_1 = c_2 = 1.49445$ ,  $w = \frac{\text{MaxIt} - \text{It}}{\text{MaxIt}}$  [26], where  $\text{It}$  is the current iteration and  $\text{MaxIt} = 200$  is total number of iterations.

Once the number of sensors was selected, their locations were optimized using all the metaheuristics to identify the best



**Figure 8.** (a) Room and limits for optimization. (b) Footsteps evenly distributed throughout the room.

one. For these optimizations, the population size was maintained at 50 individuals, and each algorithm was executed for 200 iterations and repeated 20 times. The parameters of the PSO (also utilized in the HPSGWO) were  $c_1 = c_2 = 1.49445$ ,  $w = \frac{\text{MaxIt} - \text{It}}{\text{MaxIt}}$  and the local search abandoning limit for the ABC algorithm was set to Limit = 10 [30].

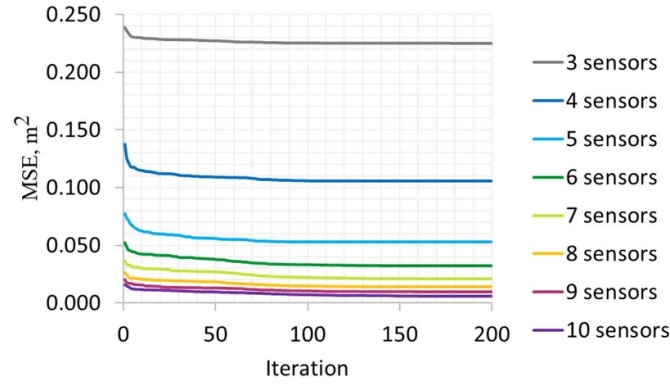
The optimization algorithms were implemented in MATLAB R2022a, on a PC with an Intel Core i7-7500 U processor.

To implement the SO-TDOA algorithm effectively, it was necessary to ensure the condition regarding the order of arrival of the signals, as outlined in equation (1). In the absence of TOA measurements ( $\hat{t}_{si}$ ) in the optimization, equation (1) can be satisfied by assuming a constant propagation velocity, thus  $\hat{t}_{si} = d_{si}/c$  where  $c = 1000 \text{ m/s}$  is the propagation velocity in concrete [7].

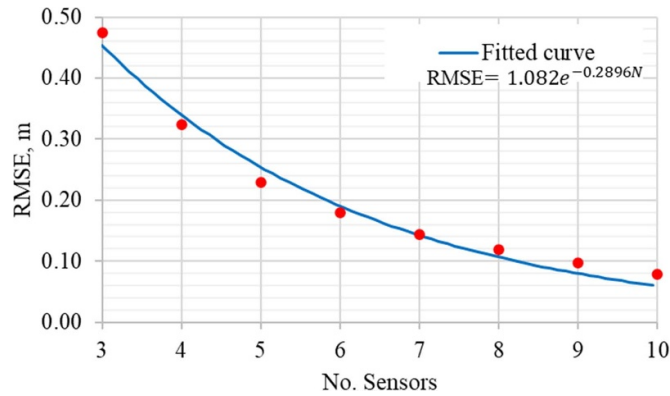
### 3. Results

#### 3.1. Number of sensors optimization

Figure 9 illustrates the evolution of the average convergence from the 10 runs of the HPSGWO algorithm for varying



**Figure 9.** Average convergence of the HPSGWO for varying sensor numbers.



**Figure 10.** RMSE as a function of the number of sensors.

**Table 1.** Comparison of optimization algorithms.

Optimization algorithm	MSE Min. (m <sup>2</sup> )	MSE Max. (m <sup>2</sup> )	SD (m <sup>2</sup> )	Avg. time per run (min)
Non-optimized	0.091	0.091	0	—
Simplex	0.058	0.1094	0.0127	0.076
ABC	0.048	0.055	0.0017	5.322
GWO	0.048	0.059	0.0033	2.552
PSO	0.050	0.063	0.0028	2.901
HPSGWO	0.047	0.058	0.0027	5.545

numbers of sensors. It shows that as the number of sensors increases, the MSE decreases. Additionally, figure 10 presents the average root mean square error (RMSE) for different numbers of sensors, fitting the exponential model  $RMSE = 1.082e^{-0.2896N}$ .

The number of sensors can be chosen based on the desired location accuracy for the entire room and the project's budget constraints. In this study, five sensors were selected, providing a localization accuracy of approximately 25 cm for the entire room.

### 3.2. Location of the sensors optimization

After selecting the number of sensors, their optimal locations were determined using metaheuristics and simplex algorithms to identify the most effective one. The simplex method [31] is a traditional non-derivative optimization approach that

compares function values at the  $n + 1$  vertices of a simplex (where  $n$  is the number of variables) and iteratively replaces the worst vertex with a better point.

To increase the possibility to find a global solution, metaheuristics and simplex algorithms were executed for 200 iterations and repeated 20 times, choosing random initial points for each run.

Table 1 summarizes the minimum and maximum MSE, standard deviation (SD), and average runtime for each algorithm run. The HPSGWO algorithm achieved the lowest MSE (0.047 m<sup>2</sup>) among all runs. Additionally, the metaheuristics found a better optimum than the simplex method (0.058 m<sup>2</sup>). It was also noted that metaheuristics require more time to find the optimum compared to traditional simplex method, due to the frequent function evaluations in each iteration. However, as this step is a one-time process, selecting the method that optimizes the function most effectively is

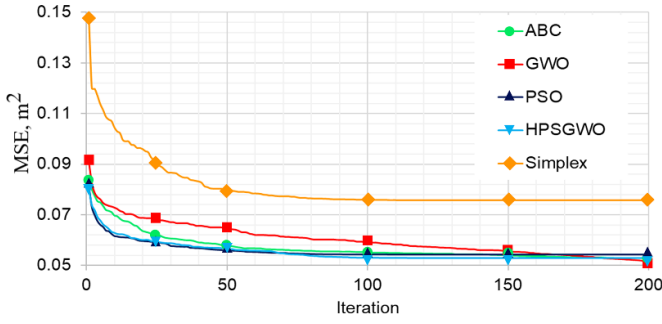


Figure 11. Average convergence of the different algorithms.

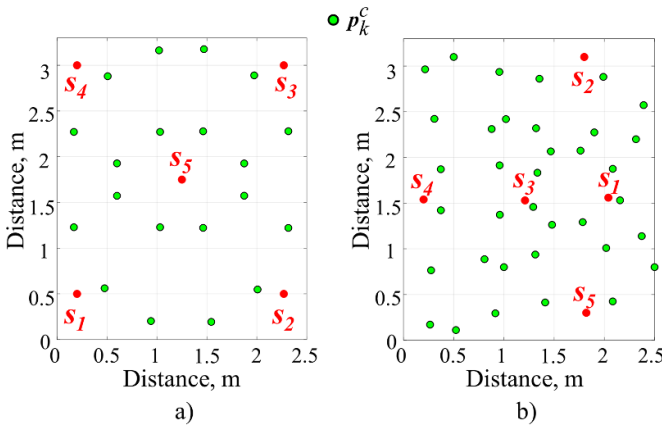


Figure 12. Sensors and centroids of the room regions (a) non-optimized, (b) optimized.

advantageous, and all metaheuristics outperformed the simplex method in this regard.

Moreover, figure 11 depicts the average convergence over 20 runs of both metaheuristic and Simplex algorithms, showing that metaheuristic algorithms reduce the MSE by 28% on average compared to the Simplex method. The GWO algorithm reaches a lower average MSE ( $0.051 \text{ m}^2$ ), but the HPSGWO achieves convergence with fewer iterations.

Due to the inherent nature of metaheuristics, we cannot guarantee that the global minimum was found. Nevertheless, the solution obtained was deemed satisfactory since it led to a significant reduction in RMSE for the entire room by 27.9% compared to the non-optimized locations shown in figure 12(a) (non-optimized RMSE =  $0.301 \text{ m}$ ; optimized RMSE =  $0.217 \text{ m}$ ). Figure 12(b) displays the optimized sensor locations, indicating that the new positions increase the number of centroids.

### 3.3. Measurements in the room

As previously discussed, the SO-TDOA algorithm involves two stages: the floor segmentation stage and the measurement and localization stage.

In this study, the first stage initiates when sensors are positioned on the room's floor. Their locations are recorded and inputted into the computer to calculate the centroids of the

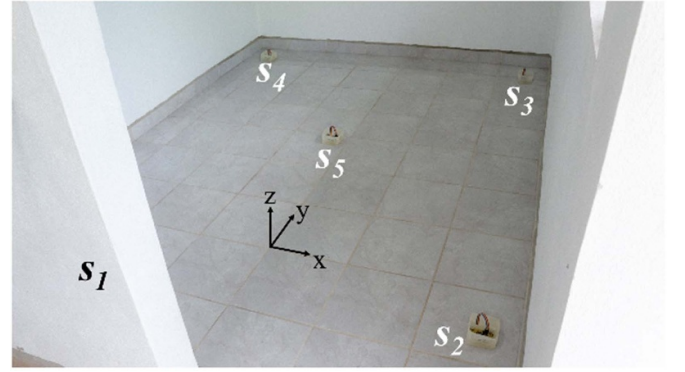


Figure 13. Photograph of the test room. Sensors are placed in the non-optimized locations.

regions. This calculation is performed only once, assuming the sensor locations remain unchanged.

The second stage of the SO-TDOA algorithm commences upon receiving all TOA measurements. During this stage, the locations of the footsteps are estimated utilizing the TOA data. The algorithm was executed in MATLAB R2022a.

Consequently, sensors were positioned at non-optimized locations ( $S_{\text{non-opt}} = [0.20 \ 0.50 \ 2.27 \ 0.50 \ 2.27 \ 3.00 \ 0.20 \ 3.00 \ 1.25 \ 1.75]$ ) with their  $z$ -axis perpendicular to the floor plane as shown in figure 13. An offset was applied to the sensors to achieve a reading of  $+1 \text{ g}$  ( $1 \text{ g} = 9.81 \text{ m s}^{-2}$ ) on the  $z$ -axis. A threshold of  $\pm 0.005 \text{ g}$  was set, meaning the TOA is determined when the signal surpasses  $+1.005 \text{ g}$  or drops below  $+0.995 \text{ g}$ , as depicted in figure 14.

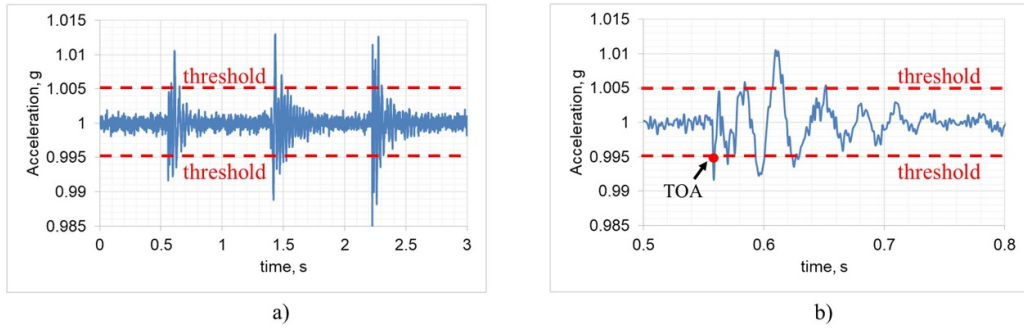
Once the network was connected, ten impacts were made on the floor with the sole at different locations in the room, following the three trajectories illustrated in figure 15. These tests were repeated for sensors placed at locations optimized with the HPSGWO algorithm ( $S_{\text{opt}} = [2.04 \ 1.56 \ 1.80 \ 3.10 \ 1.20 \ 1.53 \ 0.20 \ 1.54 \ 1.82 \ 0.30]$ ).

Figures 16(a)–(c) present the best estimation for the three trajectories for the non-optimized locations, figures 16(d)–(f) depict the best estimations for the optimized locations. Table 2 depicts the comparison of the RMSE for the estimations before and after optimization highlighting that optimization reduced the RMSE by between 18.24% and 46.78% for the three proposed trajectories.

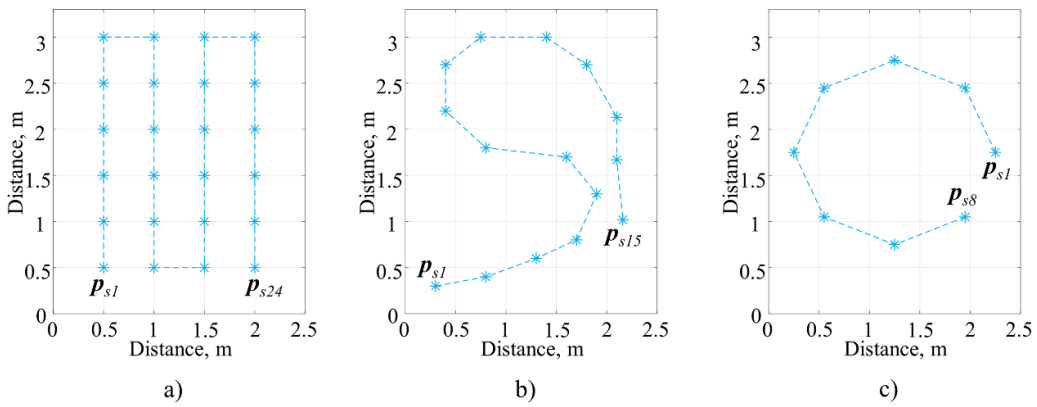
## 4. Discussion

Table 3 summarizes the characteristics and results of various studies, including the current work. It highlights that while almost all reported sub-meter accuracy in their findings, the specific location of sensors was not a focus of their research. In [1, 4–6] experiments were conducted in Goodwin Hall [9], limiting any potential adjustments to sensor placement. Conversely, in [7, 8], sensors were uniformly distributed across the floor.

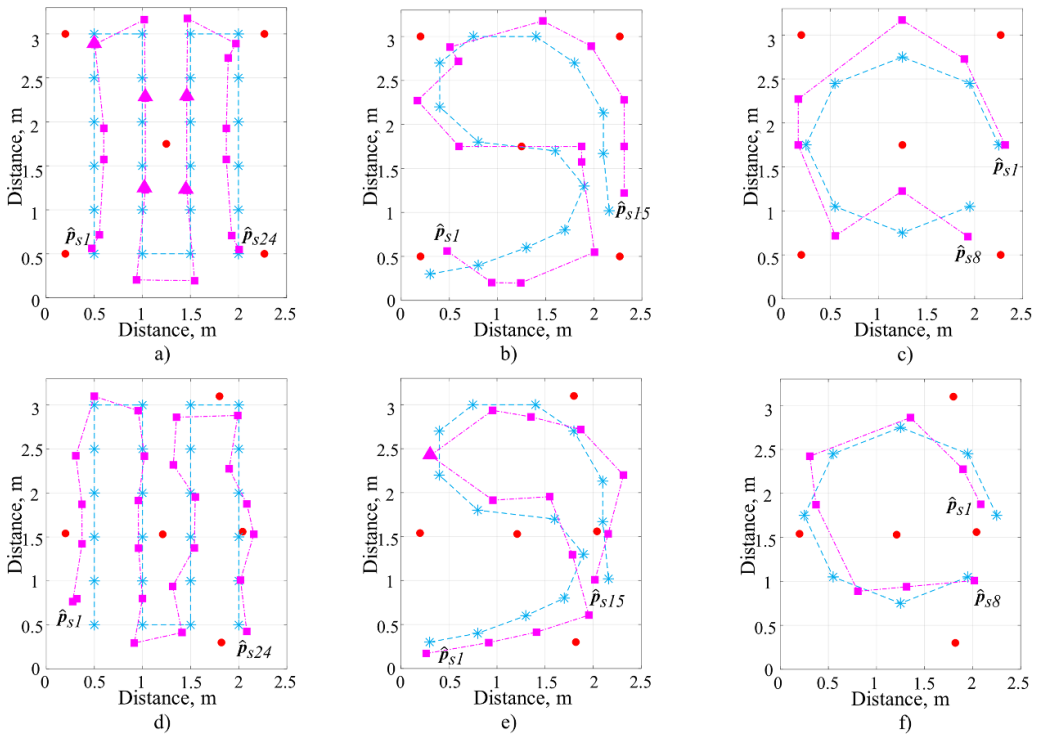
Table 4 underlines the significance of optimizing sensor networks. It presents a comparison of localization errors between a non-optimized six-sensor network and an optimized



**Figure 14.** (a) Footstep signal and threshold. (b) The time of arrival is given when the signal crosses the threshold.



**Figure 15.** Displays the ten impacts made at each point along the proposed trajectories. (a) Trajectory 1. (b) Trajectory 2. (c) Trajectory 3.



**Figure 16.** Presents the best estimation for the three trajectories. (a)–(c) Non-optimized. (d)–(f) optimized. asterisks represent the actual trajectories; squares denote the estimated trajectories, and triangles indicate locations where two estimations from two different footsteps coincide.



**Table 2.** RMSE comparison before and after optimization.

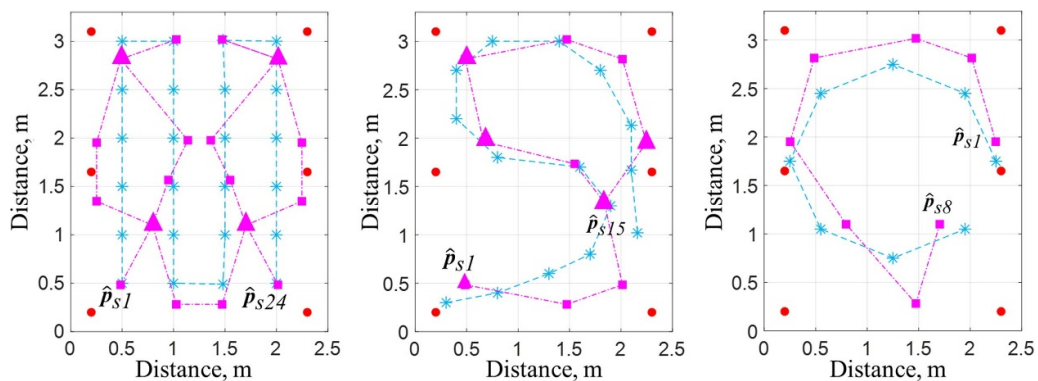
Sensor location	RMSE (m)		
	Trajectory 1	Trajectory 2	Trajectory 3
Non-optimized	0.259	0.285	0.389
Optimized (HPSWGO)	0.211	0.233	0.207
Error reduction percentage	18.53%	18.24%	46.78%

**Table 3.** Summary of the characteristics and results from different studies.

Authors	Method (Based on)	Sensor	No. of sensors	Location of sensors	Room dimensions	No. of footsteps	RMSE (m)
Poston <i>et al</i> [1]	TDOA	Piezo-electric accelerometer	120	Non-Optimized <sup>a</sup>	25.5 m × 9.4 m	30	0.590
Alajlouni <i>et al</i> [4]	RSS	Piezo-electric accelerometer	4	Non-Optimized <sup>a</sup>	3 m × 1 m	13	0.253 <sup>b</sup>
Alajlouni and Tarazaga [5]	RSS	Piezo-electric accelerometer	6	Non-Optimized <sup>a</sup>	13 m × 2 m	66	0.845
Alajlouni and Tarazaga [6]	RSS	Piezo-electric accelerometer	11	Non-Optimized <sup>a</sup>	16 m × 2 m	162	1.020
Bahroun <i>et al</i> [7]	SO-TDOA	Piezo-electric/capacitive accelerometers	9	Non-Optimized	3.6 m × 5.4 m	7	0.475
Li <i>et al</i> [8]	ATDOA	Seismometer	4	Non-Optimized	4 m × 3 m	12	0.270 <sup>b</sup>
Present work	SO-TDOA	Capacitive accelerometer	5	Non-Optimized	2.5 m × 3.3 m	24	0.259
Present work	SO-TDOA	Capacitive accelerometer	5	Optimized	2.5 m × 3.3 m	24	0.211

<sup>a</sup> Performed at Goodwin Hall.<sup>b</sup> Indicates mean error.**Table 4.** Comparison of the RMSE between a non-optimized network utilizing 6 sensors and an optimized network with 5 sensors.

	RMSE (m)		
	Trajectory 1	Trajectory 2	Trajectory 3
Six sensors non-optimized	0.2691	0.2867	0.3304
Five sensors optimized	0.211	0.233	0.207
Error reduction percentage	21.59%	18.73%	37.34%

**Figure 17.** Best estimation of the three trajectories for a non-optimized 6-sensor network. The asterisks represent the actual trajectories; squares to denote the estimated trajectories, the triangles to indicate locations where estimations from two different footsteps coincide.

five-sensor network across the three proposed trajectories (with figure 17 illustrating the estimated trajectories). Remarkably, the optimized network achieves a reduction in localization error ranging from 18.73% to 37.34% (equating to between 5.3 cm and 12.3 cm) for the three trajectories, despite having one fewer sensor.

This comparison demonstrates the impact of optimization in lowering localization errors. Furthermore, by decreasing the number of sensors, the overall system cost could also be reduced. In this instance, optimization led to a reduction in sensor count from 6 to 5, equating to a cost reduction of 16.66%. Thus, it is advisable to thoroughly evaluate both

the number and placement of sensors within a room before deploying localization algorithms.

## 5. Conclusions

This study introduced a wireless network of smart sensors designed to determine the TOA of signals without needing to store comprehensive data on each footstep. Specifically, the network is engineered to allow nodes to transmit a single datum per footstep, facilitating low data transmission/reception rates. This efficiency is attributed to the utilization of the SO-TDOA algorithm, which relies solely on the sign of the TDOA for its operation.

A critical hurdle in the application of the SO-TDOA algorithm within our network was the synchronization of nodes, necessitating precise measurement initiation at the reference time  $t_0$ . To address this, the system was devised to prompt nodes to commence TOA measurement upon receiving a signal from the coordinator, followed by a reset of counters after each measurement.

Through the evaluation of various bio-inspired metaheuristic optimization algorithms, the hybrid HPSGWO method emerged as superior, yielding the lowest estimation errors across the entire room. The optimization of sensor placement significantly reduced the RMSE for three distinct trajectories by 18.24%–46.78%, demonstrating that strategic application of the SO-TDOA algorithm can substantially mitigate localization errors.

Furthermore, a comparative analysis between a non-optimized 6-sensor network and an optimized five-sensor configuration revealed that optimization could reduce localization errors by 18.73%–37.34% across the three trajectories, despite a reduction in the number of sensors.

Consequently, it is advisable to thoroughly assess both the quantity and arrangement of sensors prior to the deployment of any localization algorithm. Such preliminary considerations can enhance the accuracy of trajectory tracking within a given space and potentially lower the overall costs of the system.

## 6. Future work

The proposed system, while promising, has identified limitations that are currently being addressed for improvement. Notably, the system does not function in real-time; it stores all measurements before estimating the individual's trajectory. Additionally, it lacks the capability to distinguish between the footsteps of multiple people within the same space. Another limitation is that the system's testing was conducted in an empty room. Hence, developing a real-time operational system, enhancing the system to differentiate between multiple individuals, and conducting experiments in environments with obstacles represent significant avenues for future research.

## Data availability statement

All data that support the findings of this study are included within the article (and any supplementary files).

## Acknowledgments

The authors declare that there is no conflict of interest in this manuscript.

This research did not receive any specific grant from funding agencies in the public, commercial, or not-for-profit sectors.

## ORCID iDs

Luis Sánchez-Márquez  <https://orcid.org/0000-0001-5927-8494>

Andrea López-Tapia  <https://orcid.org/0000-0001-8004-0763>

## References

- [1] Poston J D, Buehrer R M and Tarazaga P A 2017 Indoor footstep localization from structural dynamics instrumentation *Mech. Syst. Signal Process.* **88** 224–39
- [2] Lee H, Park J W and Helal A 2009 Estimation of indoor physical activity level based on footstep vibration signal measured by MEMS accelerometer for personal health care under smart home environments *Mobile Entity Localization and Tracking in GPS-less Environments* vol 5801 (Springer) pp 148–62
- [3] Zekavat (Reza) S A and Buehrer R M 2019 *Handbook of Position Location Theory, Practice, and Advances* (Wiley)
- [4] Alajlouni S, Albakri M and Tarazaga P 2018 Impact localization in dispersive waveguides based on energy-attenuation of waves with the traveled distance *Mech. Syst. Signal Process.* **105** 361–76
- [5] Alajlouni S and Tarazaga P 2019 A new fast and calibration-free method for footstep impact localization in an instrumented floor *J. Vib. Control* **25** 1–10
- [6] Alajlouni S and Tarazaga P 2020 A passive energy-based method for footstep impact localization, using an underfloor accelerometer sensor network with Kalman filtering *J. Vib. Control* **26** 941–51
- [7] Bahroun R, Michel O, Frassati F, Carmona M and Lacoume J L 2014 New algorithm for footstep localization using seismic sensors in an indoor environment *J. Sound Vib.* **333** 1046–66
- [8] Li F, Clemente J, Valero M, Tse Z, Li S and Song W Z 2020 Smart home monitoring system via footstep-induced vibrations *IEEE Syst. J.* **14** 3383–9
- [9] Hamilton J M 2015 *Design and Implementation of Vibration Data Acquisition in Goodwin Hall for Structural Health Monitoring, Human Motion, and Energy Harvesting Research* (Blacksburg)
- [10] Ekimov A and Sabatier J M 2006 Vibration and sound signatures of human footsteps in buildings *J. Acoust. Soc. Am.* **120** 762–8
- [11] Analog Devices 2020 Low noise, low drift, low power, 3-Axis MEMS accelerometers, ADXL354/ADXL355 datasheet (available at: [www.analog.com/media/en/technical-documentation/data-sheets/adxl354\\_adxl355.pdf](http://www.analog.com/media/en/technical-documentation/data-sheets/adxl354_adxl355.pdf)) (Accessed 1 January 2023)
- [12] Rao S S 2020 *Engineering Optimization: Theory and Practice* (Wiley)
- [13] Du K-L and Swamy M 2016 *Search and Optimization by Metaheuristics Techniques and Algorithms Inspired by Nature* (Birkhäuser Cham)

- [14] Liberti L and Maculan N 2006 *Global Optimization. From Theory to Implementation* (Springer)
- [15] Theodorakatos N P, Lytras M D, Moschoudis A P and Kantoutsis K T 2023 Implementation of optimization-based algorithms for maximum power system observability using synchronized measurements *AIP Conf. Proc.* **2872** 120006
- [16] Talbi E-G 2009 *Metaheuristics: From Design to Implementation* (Wiley)
- [17] Rothlauf F 2011 *Design of Modern Heuristics: Principles and Application* (Springer)
- [18] Cavazzuti M 2013 *Optimization Methods. From Theory to Design Scientific and Technological Aspects in Mechanics* (Springer)
- [19] El-Hasnony I M, Barakat S I and Mostafa R R 2020 Optimized ANFIS model using hybrid metaheuristic algorithms for Parkinson's disease prediction in IoT environment *IEEE Access* **8** 119252–70
- [20] Wolpert D H and Macready W G 1997 Free lunch theorems for optimization *IEEE Trans. Evol. Comput.* **1** 67–82
- [21] Kennedy J and Eberhart R 1995 Particle swarm optimization *Proc. ICNN'95—Int. Conf. on Neural Networks* vol 4 (IEEE) pp 1942–8
- [22] Karaboga D and Basturk B 2007 A powerful and efficient algorithm for numerical function optimization: artificial bee colony (ABC) algorithm *J. Glob. Optim.* **39** 459–71
- [23] Mirjalili S, Mirjalili S M and Lewis A 2014 Grey wolf optimizer *Adv. Eng. Softw.* **69** 46–61
- [24] Şenel F A, Gökçe F, Yüksel A S and Yiğit T 2019 A novel hybrid PSO–GWO algorithm for optimization problems *Eng. Comput.* **35** 1359–73
- [25] Zhao W, Wang L and Zhang Z 2021 *New Optimization Algorithms and Their Applications* (Elsevier)
- [26] Cekus D and Skrobek D 2018 The influence of inertia weight on the particle swarm optimization algorithm *J. Appl. Math. Comput. Mech.* **17** 5–11
- [27] Eberhart R C and Shi Y 2000 Comparing inertia weights and constriction factors *Particle Swarm Optimization Proc. 2000 Congress on Evolutionary Computation. CEC00 (Cat. No.00TH8512)* vol 1 pp 84–88
- [28] DIGI 2020 DIGI XBEE3 ZIGBEE 3.0, Easy-to-add connectivity in a compact, low-power, low-profile footprint
- [29] Horvat G, Šoštari D and Žagar D 2012 Power consumption and RF propagation analysis on ZigBee XBee modules for ATPC *35th Int. Conf. on Telecommunications and Signal Processing (TSP)* pp 222–6
- [30] Akay B and Karaboga D 2009 Parameter tuning for the artificial bee colony algorithm *Int. Conf. on Computational Collective Intelligence. ICCCI 2009: Computational Collective Intelligence. Semantic Web, Social Networks and Multiagent Systems* pp 608–19
- [31] Nelder J and Mead R 1965 A simplex method for function minimization *Comput. J.* **7** 308–13