Applied Intelligence
https://doi.org/10.1007/s10489-022-03872-y

A new hybrid localization approach in wireless sensor networks based on particle swarm optimization and tabu search

Elie Tagne Fute [1,2] · Doris-Khöler Nyabeye Pangop [1] · Emmanuel Tonye [3]

Accepted: 7 June 2022

Abstract
In recent years, many works have proposed solutions for indoor localization in Wireless Sensor Networks (WSN). The challenge in these different works is above all to improve localization accuracy. New trends in the field are the use of optimization techniques to improve the accuracy in determining the location of a sensor. Thus, this study aims to propose a new contribution to the indoor localization problem in WSN based on optimization techniques. The designed approach improves the performance of particle swarm optimization (PSO). In this improved version of PSO, on the one hand, a form of tabu search is used by each particle to determine its best local neighbor in order to accelerate its possibilities of convergence towards a better solution. On the other hand, limit and performance checks are introduced into the PSO algorithm to evolve only with better particles belonging to the search space constructed by constraint analysis, around an initial solution obtained by trilateration. This proposed approach called FPSOTS uses the received signal strength indicator (RSSI) method to evaluate inter-sensor distances. Localization accuracy and convergence performances of the FPSOTS approach were evaluated in simulation and compared with other recent localization approaches based on optimization techniques. Results show that FPSOTS succeeds in locating unknown nodes of a WSN with fast convergence and better accuracy than recent state-of-the-art approaches such as HPSOVNS, NS-IPSO, ECS-NL and GTOA. Indeed, in comparison with these four approaches, the accuracy of FPSOTS approach was better by 40%, 35%, 44% and 22% respectively.

Keywords Wireless sensor networks · Indoor localization · Optimization · Particle swarm · Tabu search · Accuracy

1 Introduction

The current generation of WSN allows the construction of applications requiring real-time localization of the active agents of the system, such as applications for monitoring and detecting suspicious or critical events like enemy intrusions, earthquakes, bushfires, etc. The most commonly used positioning system is GPS (Global Positioning System). However, equipping all the sensors with a GPS considerably increases the cost of network deployment, without forgetting the increase in the energy consumption of sensors due to the use of GPS [1, 2]. In addition, GPS signals are greatly affected by security issues and are sometimes poorly received inside buildings and dense environments such as forests [3, 4]. Therefore, it might be necessary to equip only some sensors with GPS (called anchors) and estimate the locations of all other sensors without GPS (called unknown or blind sensors) through an indoor localization technique.

In general, indoor localization consists in accurately estimating the positions of unknown sensors using certain data received from anchors. The estimated positions must be as close as possible to the real positions of unknown sensors. In indoor localization, anchors can be fixed or mobile, and can know their position by a GPS or by a manual configuration during their deployment [5]. However, the use of fixed anchors can lead to high-cost problems due to the need for

✉ Doris-Khöler Nyabeye Pangop
nyabeyedoriskholer@gmail.com

Elie Tagne Fute
eliefute@yahoo.fr

Emmanuel Tonye
tonye2018@hotmail.com

1 Department of Computer Science, University of Dschang, Dschang, Cameroon

2 Department of Computer Engineering, University of Buea, Buea, Cameroon

3 Department of Electrical and Electronics Engineering, University of Yaounde I, Yaounde, Cameroon

Published online: 25 July 2022

numerous fixed anchors for the best environmental coverage. On the other hand, one or a few mobile anchors can be deployed in the network, to move around and broadcast information on their position that the unknown sensors will collect to locate themselves. Usually, in a double dimension (2D), sensors wishing to calculate their location need at least three non-collinear anchor position points.

Indoor localization methods are generally classified into two groups: range-based [6–9] and range-free [10–13]. Range-based localization methods are generally more precise than range-free [14]. Range-based localization methods estimate distances and/or angles between sensors and anchors, using anchor positions. These distances or angles can be obtained using a measurement technique such as Angle of Arrival (AoA), Time of Arrival (ToA), Two Way Ranging (TWR), Time Difference of Arrival (TDoA), Received Signal Strength Indicator (RSSI) [2, 4, 15, 16].

From distances and/or angles, we can estimate a sensor position using a geometric technique such as trilateration, triangulation and multilateration [1, 2, 15]. In range-based localization, the RSSI technique is most often used to estimate distances between nodes. This is because several wireless devices integrate a unit allowing them to measure received signal intensity, and therefore there is no need for additional equipment. However, RSSI measurement may have errors in the estimated distances due to noise in the environment, which adversely affects received signals. Methods such as trilateration, triangulation or multilateration are very sensitive to these errors and can lead to very erroneous results. This causes position estimations with poor precision, increasing error between the estimated position and the actual position of an unknown sensor.

In recent years, several works have focused on using optimization methods to increase the accuracy of estimating the location of an unknown sensor [10, 12, 17–25]. These works use for example: particle swarm optimization (PSO) in traditional or improved version [17, 21, 23, 26]; forms of PSO hybridization with variable neighborhood search [18] or simulated annealing [20]; cuckoo search [25]; group teaching optimization [22]; etc. Thus, the localization problem can be formulated as an optimization problem and be solved using heuristics or meta-heuristics.

In the literature, we observe that PSO is widely used in the field of indoor localization because of its rapid convergence, its ease of implementation and its good performance in terms of localization accuracy [17]. Indeed, PSO can accurately estimate the location of unknown sensors in a short time [21].

In this study, we are also interested in using PSO to estimate the location of unknown sensors in a WSN using mobile anchor data. An initial solution is determined by trilateration. We improve the PSO method to estimate positions with good accuracy and as close as possible to real positions, despite the noise in the environment. More precisely:

- We use a simple method based on constraints analysis to define an area that can contain the position of unknown sensor, also serving as a search space for the optimization process;
- In the iterative process of PSO, we give each particle the ability to search for the best local neighbor around its current position by a form of tabu search, to increase and accelerate its possibilities of convergence towards a better solution;
- Next, we introduce in the iterative process of PSO a limit check to verify that the new particle positions belong to the elaborated search space. A new particle position, outside the search space and less efficient than the current position is adjusted to be brought back into the search space. But, the one outside the search space and better than the current position is maintained and validated.
- Finally, we introduce in the iterative process of PSO a performance check to evolve only with the new solutions better than the current ones.

This improved version of PSO, we called it FPSOTS. After evaluating the performance of FPSOTS against those of other optimization-based indoor localization techniques, we show that FPSOTS allows to quickly obtain, with good convergence and precision, an optimal solution to the location of unknown sensors. To our knowledge, this study is the first to use any hybridization of PSO with tabu search to improve indoor localization performance in WSN.

The rest of this paper is organized into 4 sections. Section 2 discusses our motivations and the PSO-based localization technique. Section 3 presents previous works on localization methods based on RSSI distances and optimization techniques. In Section 4, we describe the functioning of our localization model named FPSOTS i.e. calculation of the initial solution, elaboration of the search space and improvement of PSO. Simulations were performed, and the results are presented in Section 5. This paper ends with a conclusion in Section 6.

2 PSO localization and motivations

PSO is a computational algorithm that finds its source in observations made during computer simulations of grouped flights of birds and shoals of fish [27]. It uses a population of particles to represent candidate solutions in a search space, and iteratively optimizes the problem to move these particles to the best solutions with respect to a given quality measure called an objective function.

In PSO algorithm, the behavior of the swarm is described in terms of a particle. At any time, each particle p in position $x(p)$, is considered as a solution of the problem. The particle must move to explore more solutions. Its displacement depends on:

- Its current speed $v(p)$: influenced by an inertia component ci, which represents the confidence of the particle in its current trajectory. Particle instinctively tries to take its course of movement.
- Its best-visited solution $P_{best}(p)$: influenced by a cognitive component cc, which allows the particle to move towards its best-encountered position until the current instant.
- The best particle of the whole swarm S_{best}: influenced by a social component cs, allowing the particle to be inspired by the experience of other particles, to move towards the best position encountered by its neighbors.

The best position P_{best} corresponds each time to the best position already obtained by the particle among all the positions it has visited. The next particle position (at the iteration $t+1$) is calculated using Eqs. 1, 2 and 3.

$$v_{t+1}(p) = ci^{*}\mathrm{v_t(p)} + \mathrm{cc^{*}(P_{bestt}(p) - x_t(p))} + \mathrm{cs^{*}(S_{bestt} - x_t(p))} \quad (1)$$

$$x_{t+1}(p) = v_{t+1}(p) + x_t(p) \quad (2)$$

$$S_{bestt} = argmin(f(x_t(q)), q \in V(p)) \quad (3)$$

Where t denotes the current iteration; $V(p)$ denotes all other particles of the swarm; $f(x_t(q))$ is the fitness function used to evaluate a particle position.

Algorithm 1 : PSO algorithm

```
Output : S_best
Initialization of N particles ;
Particles evaluation ;
S_best ← position of the best particle according to (3) ;
while (stop criterion not reached) do
    foreach Particle P do
        Update P_best ;
        Update velocity and position of P
        according to (1) and (2) ;
    end
    Particles evaluation ;
    S_best ← position of the best particle according
    to (3) ;
end
return S_best ;
```

In Algorithm 1, we note the following points in PSO method:

1. The local component of a particle (P_{best}) is the best position visited by the particle: in the PSO method, a particle evolves according to its best local performance. This best local performance is simply its best-visited position at the current time. We wondered about the result that the algorithm would have produced if the local component were obtained by searching around the current position of the particle, using for example a trajectory meta-heuristic. Indeed, the works proposing a hybridization of PSO [18, 20] do this hybridization on the global component (S_{best}). They search by another optimization technique for a result to the problem and compare it with the current result provided by PSO to retain the best of the two as the current solution. We think that performance of PSO could be improved if the hybridization was done on the local component (P_{best}).
2. PSO has no limit check: all new particle positions are accepted, even those that are not part of the search space. Performance can be improved by rejecting positions outside the search space that do not improve the overall performance of the algorithm, or by bringing these positions back into the search space.
3. PSO accepts less efficient solutions: all new particle positions, even those less efficient than the current positions are accepted. This behavior can cause a particle to be less good in the next iterations than in past iterations. This can impact the performance of the algorithm. It is possible to control the new positions calculated by a particle to retain only those which make it possible to improve the performance of said particle.

In our FPSOTS proposal, we improve the PSO algorithm to integrate behaviors described by these three points. We use as a trajectory meta-heuristic a form of tabu search, to obtain the component P_{best} which we now baptize as *best local neighbor of a particle* instead of "best-visited position".

Tabu search is a meta-heuristic developed by Glover in 1986 [26]. Our choice of tabu search comes from the fact that it overcomes the problem of local optima during a search process by using a tabu list. Tabu list is a short-term memory that stores previously visited solutions to avoid rollbacks and moves that do not improve the current solution. But it is possible to violate a ban when a prohibited movement achieves the best solution recorded so far. Another point justifying the choice of tabu search is that to the best of our knowledge, no form of hybridization of PSO with tabu search has yet been proposed in the literature.

3 Previous works on optimization-based localization

In the category of localization approaches using population meta-heuristics, one of the most used optimization techniques is PSO proposed in [17]. They are the first to use PSO for localization in WSN. In their study, they compared the performance of PSO algorithm with that of the simulated annealing algorithm for localization. Results showed that PSO is more efficient than simulated annealing for locating a sensor.

Based on swarm intelligence applications to locate a moving sensor, [23] proposed a concept of projecting virtual anchor nodes using RSSI as a measurement technique. They separately implemented H-Best Particle Swarm Optimization (HPSO), Biogeography Based Optimization (BBO) and Firefly (FA) algorithms using a single mobile anchor in the network. Their study showed that HPSO algorithm provided better performance than BBO and FA algorithms.

A localization approach named HPSOVNS is implemented in [18] based on PSO and variable neighborhood search (VNS). At each iteration of PSO algorithm, a solution is sought by VNS technique. If the solution obtained by VNS is better than that provided by PSO then this solution is considered as the best overall solution(S_{best}) of PSO for the next iteration.

Authors of [28] in their study to determine node location with high accuracy by swarm intelligence algorithms, presented a localization algorithm named LMQPDV-hop. In LMQPDV-hop, an improved DV-Hop was used as an underground mechanism to collect estimated distances between nodes. Subsequently, a Quantum Particle Swarm Optimization (QPSO) algorithm, named LMQPSO, was developed to find the best coordinates of unknown nodes. The results of conducted simulations show that the algorithm can effectively improve position accuracy. However, sensors are not self-locating. They just collect information necessary for their location and send them to the base station, which is responsible for calculating the position of each sensor.

To address the challenges of low sampling efficiency and particle depletion in Monte Carlo positioning algorithms, the authors of [10] introduce a chronological Monte Carlo localization algorithm based on PSO (TSMCL-BPSO). First, the sampling region is constructed based on the overlap of the initial sampling region and the Monte Carlo sampling region. Then, a PSO strategy is adopted to search for the optimal position of the target node. But this algorithm uses very complex and expensive calculation methods for sensors with low resources. In addition, the algorithm is designed to work with fixed anchors, which limits its use in a hostile or difficult-to-access area where fixing anchors can be very risky.

In [21], a localization technique based on node segmentation with PSO improvement (NS-IPSO) is proposed. This method divides sensors into segments to improve the accuracy of the estimated distances between pairs of anchors and unknown sensors. They determined candidate sensors that could potentially be used to segment anchors in an area, based on certain specified conditions. To further improve the accuracy of location, the fitness function is improved to take into account the number of jumps between each anchor and unknown sensors. Their improved version of PSO only considered particles that do not change position in order to possibly reduce the risk of trapping in a local optimal.

Moreover, recent works use other forms of optimization to locate sensors in a WSN. For example, the works in [25] provide a cuckoo search enhancement for node localization called Enhanced Cuckoo Search (ECS-NL), to minimize the average localization error and time required to locate an unknown sensor. In this algorithm, the authors introduced an early stop mechanism to improve the search process by breaking out of the search loop whenever the optimal solution is reached.

Authors of [22] designed a group teaching optimization algorithm for node localization named GTOA-NL based on a meta-heuristic for WSN. The goal of GTOA-NL is to determine the position of unknown sensors using anchors with minimum localization error and maximum localization accuracy. A set of simulations was carried out, and the results obtained ensured the performances of GTOA-NL model compared to other methods under a varying number of anchors, telemetry error and transmission ranges.

In our study, we propose a contribution to indoor localization problem in WSN using meta-heuristics. We propose a new localization approach based on an improved version of PSO named FPSOTS according to our motivations presented in Section 2. This improved version introduces in PSO method a limit and performance check in the evolution of particles, while allowing them to find their best local solution using a form of tabu search. During this tabu search, each particle will have the possibility to explore a better solution in order to modify its performance and quickly improve the overall performance of PSO. FPSOTS only considers new particle positions that improve the performance of the algorithm. The principle of the FPSOTS method is detailed in the rest of this work. Performances of FPSOTS are compared to HPSOVNS, NS-IPSO, ECS-NL and GTOA-NL using the parameters of anchor density, standard deviation of signals due to noise and transmission range.

4 FPSOTS-based localization

4.1 FPSOTS Methodology

In our WSN, the process of FPSOTS localization method is described in the following steps.

1. Diffusion: anchors equipped with GPS move and periodically broadcast their coordinates in the environment.
2. Collect: the other sensors, initially fixed, receive these coordinates when they are in the transmission range of an anchor. Sensors must collect three non-collinear

coordinates to be able to start their localization process, since we are in 2D.

3. Calculation of distances: each time a sensor receives a signal from an anchor, it estimates by RSSI the distance d_i separating it from the anchor i. This distance can be expressed by Eq. 4.

$$d_i = \sqrt{(x_c - x_i)^2 + (y_c - y_i)^2} \tag{4}$$

(x_c, y_c) is the position of sensor C and (x_i, y_i) is the position of anchor i.

In real cases, the measured distance may be corrupted by noise [23]. A sensor can define the distance between an anchor i and itself using formula 5.

$$D_i = d_i + n_i \tag{5}$$

d_i is the estimated distance between anchor i and sensor. n_i is the Gaussian variable with zero mean and noise variance σ, which affects the evaluated distance d_i. σ represents the standard deviation caused by noise over the distances evaluated.

4. Find three non-collinear positions: each time the sensor receives an anchor signal and evaluates the distance between the anchor and it, it checks to see if it has already at least three non-collinear positions. If that is the case, he goes to the next step. Otherwise, he remains open to new positions.
5. Initial solution by trilateration: when a sensor collects three non-collinear positions, it calculates S_0, the initial solution to its location by trilateration according to Eq. 7.
6. Search space: then it delimits a search space from its final localization using Eq. 12.
7. Auto-localization: it searches by the FPSOTS method of algorithm 3 for its final location.
8. Displacement: finally, the located sensors move and periodically broadcast their current positions to accelerate the localization process of other sensors.

The overall architecture of FPSOTS is summarized in Fig. 1.

4.2 Initial solution

Suppose a sensor $C(x_c, y_c)$ at a given position and obtaining the different points $A_1(x_{a1}, y_{a1}), A_2(x_{a2}, y_{a2})$ and $A_3(x_{a3}, y_{a3})$ broadcast by anchors. Upon receipt of each signal, the sensor calculates by RSSI the distance separating it from the transmitting anchor. By trilateration, the sought position of sensor C is the intersection point of the three circles, each having a reference point as its center and the estimated distance between this point and the sensor as its radius (Fig. 2).

The coordinates (x_c, y_c) of sensor C are obtained by solving system 6.

$$\begin{cases} (x_c - x_{a1})^2 + (y_c - y_{a1})^2 & = & d_1^2 \\ (x_c - x_{a2})^2 + (y_c - y_{a2})^2 & = & d_2^2 \\ (x_c - x_{a3})^2 + (y_c - y_{a3})^2 & = & d_3^2 \end{cases} \tag{6}$$

Initial sensor localization named $S_0(x_0, y_0)$ is defined by Eq. 7.

$$\begin{cases} x_0 = x_c = \frac{-x_{a1}^2 - y_{a1}^2 + x_{a2}^2 + y_{a2}^2 - 2y_0(y_{a2} - y_{a1}) + d_1^2 - d_2^2}{2(x_{a2} - x_{a1})} \\ y_0 = y_c = \frac{(x_{a2} - x_{a1})(-x_{a1}^2 - y_{a1}^2 + x_{a3}^2 + y_{a3}^2 + d_1^2 - d_3^2) - (x_{a3} - x_{a1})(-x_{a1}^2 - y_{a1}^2 + x_{a2}^2 + y_{a2}^2 + d_1^2 - d_2^2)}{2((x_{a2} - x_{a1})(y_{a3} - y_{a1}) - (x_{a3} - x_{a1})(y_{a2} - y_{a1}))} \end{cases} \tag{7}$$

Once this initial solution S_0 has been found, the sensor can develop a search space around it, which will be used by optimization to find a better solution to the positioning of the sensor.

4.3 Search space

It is true that in the literature, several methods exist for the construction of a box containing the position of an unknown sensor [21]. But we have chosen to use a simple and not computationally expensive process for the development of this zone representing the search space. Constraints analysis is the name we have given to our process of delimiting an area that may contain the position of an unknown sensor. It is inspired by interval analysis, which is a technique that has proven itself in the processing of state estimation problems (localization) [29]. Delimitation consists in analyzing the graphic zone of the sensor and defining constraints to represent it.

An unknown sensor is fixed and receives all the positions broadcast by anchors being in the same place. This assumes that each time an anchor position was received, it was within the transmission range of the transmitting anchor. Then, the intersection zone of the transmission fields of anchors whose diffused positions have been received by this unknown sensor, represents the constraint zone of the sensor as illustrated in Fig. 3.

A_1, A_2 and A_3 are the three non-collinear anchor positions that the sensor has received and selected. R_{com} designates the communication radius of anchors. The points $I_n, 1 \leq n \leq 6$, are intersection points of the supposed different communication fields of anchors having emitted positions A_1, A_2 and A_3. Let:

- I_1 and I_2 intersection points of circles with centers A_1 and A_2;
- I_3 and I_4 intersection points of circles with centers A_1 and A_3;
- I_5 and I_6 intersection points of circles with centers A_2 and A_3.

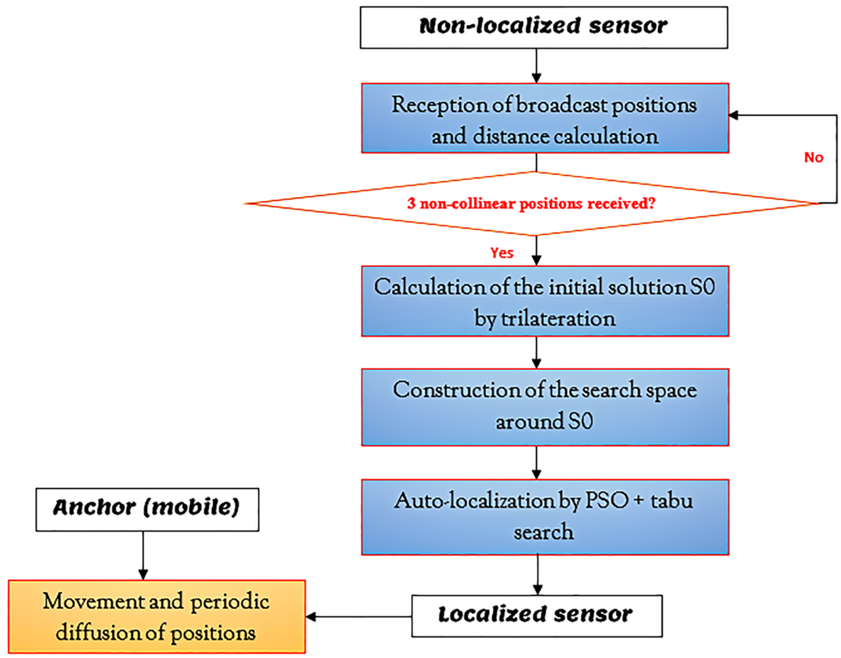


Fig. 1 Architecture of FPSOTS localization method

The constraint zone $Z(C)$ of sensor C represents the intersection zone of the three circles formed by each reference point. This area is delimited by three intersection points, as we can see in Fig. 3. To know which points to retain among the six intersection points, we calculate the distance separating intersection points of two circles from the center of the third circle and we retain the closest point. According to the example in Fig. 3, we calculate and compare the distances:

- $[A_3I_1]$ and $[A_3I_2]$ and we retain I_1
- $[A_2I_3]$ and $[A_2I_4]$ and we retain I_3
- $[A_1I_4]$ and $[A_1I_6]$ and we retain I_5

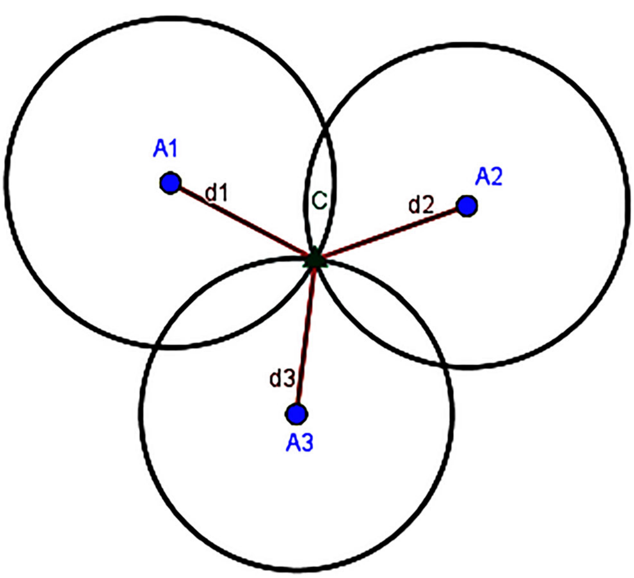


Fig. 2 Initial solution by trilateration

Note that this zone is made up of three arcs of a circle. To represent it, we start from system 8.

$$\begin{cases} (x-x_{a1})^2+(y-y_{a1})^2 & \leq & R_{com}^2 \\ (x-x_{a2})^2+(y-y_{a2})^2 & \leq & R_{com}^2 \\ (x-x_{a3})^2+(y-y_{a3})^2 & \leq & R_{com}^2 \end{cases} \tag{8}$$

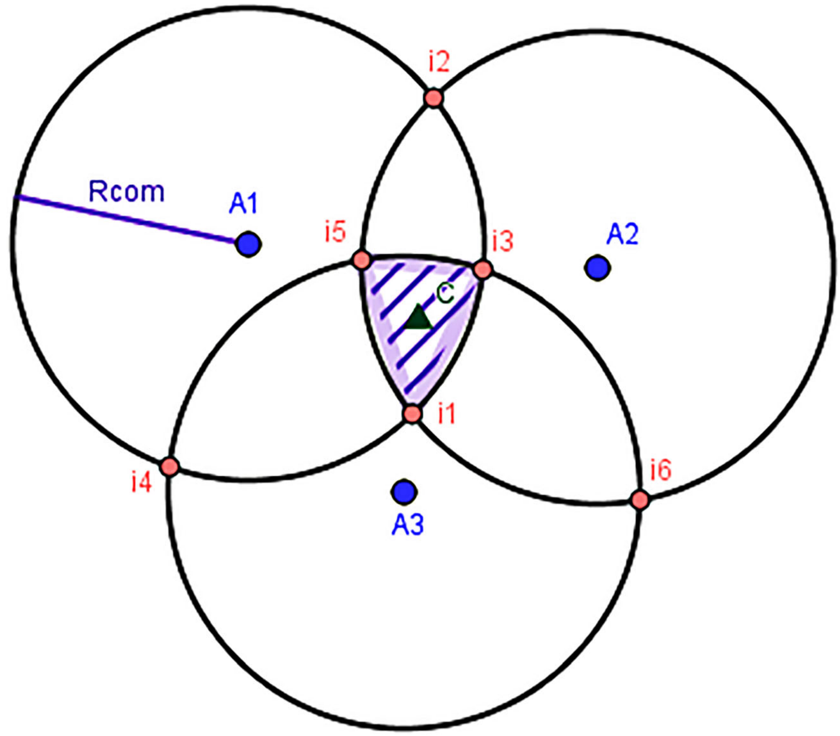
Where (x_{a1}, y_{a1}) are the coordinates of Anchor A_1; (x_{a2}, y_{a2}) are the coordinates of Anchor A_2; (x_{a3}, y_{a3}) are the coordinates of Anchor A_3; R_{com} designates the communication radius of anchors.

From system 8, we obtain relations of system 9.

$$\begin{cases} f_1(x) & = & y_{a1} - \sqrt{R_{com}^2 - (x-x_{a1})^2} \\ f_2(x)v = & & y_{a1} + \sqrt{R_{com}^2 - (x-x_{a1})^2} \\ g_1(x) & = & y_{a2} - \sqrt{R_{com}^2 - (x-x_{a2})^2} \\ g_2(x) & = & y_{a2} + \sqrt{R_{com}^2 - (x-x_{a2})^2} \\ h_1(x) & = & y_{a3} - \sqrt{R_{com}^2 - (x-x_{a3})^2} \\ h_2(x) & = & ;y_{a3} + \sqrt{R_{com}^2 - (x-x_{a3})^2} \\ X_{MIN} & = & min(x_{I_1}, x_{I_3}, x_{I_5}) \\ X_{MAX} & = & max(x_{I_1}, x_{I_3}, x_{I_5}) \end{cases} \tag{9}$$

The set $Z(C)$ representing the constraint zone of the sensor C is defined by Eq. 10.

Fig. 3 Constraint zone of a sensor



$$Z(C) = \left\{ p\begin{pmatrix} x \\ y \end{pmatrix} | X_{MIN} \leq x \leq X_{MAX} and \left\{ \begin{array}{l} f_1(x) \leq y \leq f_2(x) \\ g_1(x) \leq y \leq g_2(x) \\ h_1(x) \leq y \leq h_2(x) \end{array} \right\} \right\} \tag{10}$$

Let us pose the following system 11:

$$\begin{cases} Y_{MIN} & = & \max(\mathrm{f}_1(\mathrm{x}), \mathrm{g}_1(\mathrm{x}), \mathrm{h}_1(\mathrm{x})) \\ Y_{MAX} & = & min(f_2(x), g_2(x), h_2(x)) \end{cases} \tag{11}$$

The final search space for a sensor localization is defined by system 12.

$$Z(C) = \left\{ p\begin{pmatrix} x \\ y \end{pmatrix} | X_{MIN} \leq x \leq X_{MAX} \ and \ Y_{MIN} \leq y \leq Y_{MAX} \right\} \tag{12}$$

A generated candidate solution is then valid only if it belongs to $Z(C)$. This process will eliminate inconsistent solutions during the optimization process.

4.4 FPSOTS optimization: PSO + tabu search

We denote by *NBParticles* the number of particles generated in the search space $Z(C)$. All the particles are part of the same swarm, and each particle represents a potential solution to the problem. FPSOTS method being an optimization method, the fitness function used to evaluate the quality of candidate solutions is defined by Eq. 13.

$$f(x,y) = \frac{1}{M} \times \sum_{i=0}^{M} \left(\sqrt{(x - x_i^2 + (y - y_i)^2} - D_i \right) \tag{13}$$

(x, y) are the estimated coordinates of the non-localized sensor; M the number of anchor signals received within the transmission range of the non-localized sensor; (x_i, y_i) the coordinates of anchor i. D_i represents the estimated distance between anchor i and sensor.

At the start of PSO, each particle has zero speed and considers its best local neighbor to be its current position. At each iteration of optimization, the particle searches for its new best local neighbor P_{best} and updates its speed using Eqs. 1, 2 and 3 from Section 2. To get this best local neighbor P_{best}, particle performs a form of tabu search in its neighborhood.

4.4.1 Tabu search for the best local neighbor of a particle

Algorithm 2 presents how the best neighbor of a particle is obtained by tabu search. Tabu list is common to all particles in the swarm and has a defined capacity. It stores old visited positions of particles. We denote by the size of a particle neighborhood i.e. its number of possible neighbors or more precisely number of search iterations. Algorithm uses as input a particle, the size of its neighborhood, tabu list and the best global solution. Our form of tabu algorithm allows acceptance of a tabu solution if it improves the current solution.

During the search for its best local neighbor, a particle may explore a better solution than the current global solution; i.e. a better quality solution than the quality of the component S_{best} (the component that designates the best overall solution). In this case, the algorithm updates particle position, its component P_{best} and the global component S_{best} by this position found before continuing the search. This process will help algorithm to converge faster towards an optimal solution.

Algorithm 2 : Tabu search for the best neighbor of a particle (P_{best})

```
Input :  P, NBV, LTabou, S_best;
Output :  P_best^(P);
Initialization:  S ← P.coordinates; i ← 0; S* ←
                 S_best; better ← false;
while (i < NBV) do
    S' ← generate a neighbor of P
    if (f(S') < f(S)) then
        S ← S' ;
    end
    if ( f(S') < f(S*)) then
        S* ← S' ;
        better ← true ;
    end
    i ← i + 1;
end
if (LTabou full) then
    Remove the oldest entries from LTabou;
end
if (better = true and f(S*) < f(S)) then
    LTabou ← LTabou + P.coordinates;
    LTabou ← LTabou + P_best^(P) ;
    P.coordinates ← S* ;
    P_best^(P) ← S* ;
    S_best ← S*
else if ( f(S) < f(P_best^(P))) then
    LTabou ← LTabou + P.coordinates;
    LTabou ← LTabou + P_best^(P) ;
    P_best^(P) ← S ;
  end
end
return P_best^(P)
```

Our form of tabu search algorithm gives each particle chance to improve the overall performance of PSO as well as its performance, if it explores a better solution than the current best overall solution. Therefore, the optimization process will quickly find an optimal solution to sensor localization.

4.4.2 Limit check in particle evolution

As mentioned in Section 2 among our motivations, the PSO algorithm in its traditional form accepts any new particle position, even out of search space. But in the PSO version proposed in this work, the algorithm checks that the new positions belong to the search space.

If a new position is outside the search space and is also of better quality than the current position, then it is accepted as a new position, because it can allow the discovery of a better position. But in the case where the new position is outside the search space and is of lower quality than the best position of the particle, this new position is brought back into the search space according to Eq. 14. $Pmin(X_{MIN}, Y_{MIN})$ and $Pmax \times (X_{MAX}, Y_{MAX})$ are two points representing the bounds of search space. $X_{MIN}, Y_{MIN}, X_{MAX}, Y_{MAX}$ are defined in systems 9 and 11. Limit check is done on each dimension of position coordinates.

$$x_{t+1}(p) = \begin{cases} (x_t(p) + Pmax)/2 & If\, x_{t+1}(p) < Pmin \\ (x_t(p) + Pmin)/2 & If\, x_{t+1}(p) > Pmax \end{cases} \tag{14}$$

Where t denotes the current iteration.

4.4.3 Performance check in particle evolution

PSO algorithm in its traditional form accepts any new position, even one that does not improve the quality of a particle. In the PSO version proposed in this work, the algorithm accepts only the best positions after having checked and adjusted their membership in the search space using system 15.

$$x_{t+1}(p) = \begin{cases} x_{t+1}(p) & Iff(x_{t+1}(p)) < f(x_t) \\ x_t(p) & Iff(x_{t+1}(p)) > f(x_t) \end{cases} \tag{15}$$

Where f is the fitness function allowing to evaluate the performance of each particle defined by Eq. 13; t denotes the current iteration.

4.4.4 Final algorithm of FPSOTS

The final process of FPSOTS localization is described by algorithm 3. The algorithm outputs the best position extracted as the sensor localization. The steps of this algorithm are as follows.

1. Generate particles around S_0 in $Z(C)$
2. Evaluate each particle

3. Determine S_{best}: position of the best particle
4. By tabu search of algorithm 2, find the best local neighbor P_{best} of each particle P
5. Calculate the new velocity of the particle
6. Calculate the new position named X
7. Adjust X if $X \notin Z(C)$ and $f(X) > f(P)$ according to system 14
8. Update position of P with X according to system 15
9. Start over at step 2 until the number of iterations is reached
10. Return S_{best}

Algorithm 3 : FPSOTS localization

```
Input : ParticlesList, S_0, ParticlesNumber, ci, cc,
        cs, cg, MaxIteration;
Output : S_best (Global best position obtained);
ParticlesList ← ParticlesNumber particles
  generate around S_0 and ∈ Z(C) ;
Evaluate each particle according to (13);
S_best ← best particle position;
for i = 1 to MaxIteration do
    foreach Particle P ∈ ParticlesList do
        search for P_best^(P) according to algorithm
        2;
        Calculate the new speed of P according
        to (1);
        X ← New position of P according to
        (2);
        if ( X ∉ Z(C) et f(X) > f(P.
        coordinates)) then
            Adjust X according to (14);
        end
        Update P with X according to (15);
        Evaluate P according to (13);
    end
    for k = 1 to ParticlesNumber do
        if (f(P_best^(k)) < f(S_best)) then
            S_best ← P_best^(k) ;
        end
    end
end
return S_best;
```

5 Simulations and interpretation of results

We have chosen as simulation environment, the CupCarbon environment. CupCarbon is a recent WSN simulator, developed by Mehdi and al. in 2014 [30]. Its purpose is to design, visualize and validate the algorithms proposed for WSNs. Its simulation environment allows the design of mobility scenarios. The CupCarbon platform is developed with the following main objectives [31]:

- study the deployment of WSNs by considering the mobility and availability of the radio spectrum;
- simulate and analyze the performance of a proposed WSN in a 2D/3D environment;
- study the feasibility and reliability of communication in the network;
- detect areas of high radio interference in the network;
- accurately simulate radio propagation in an urban environment in real-time;
- better visualize simulation results to debug and validate a developed algorithm.

All these elements motivated our choice of this environment for our simulations. But even more, our choice was favored by the programming language of this environment. Indeed, CupCarbon is developed in Java and can be easily integrated and used in a Java application; the Java language being the one chosen for our implementations.

The performance in terms of localization accuracy (localization error) of FPSOTS was evaluated in comparison with approaches HPSOVNS [18], NS-IPSO [21], ECS-NL [25] and GTOA-NL [22] using the parameters of anchor density, standard deviation of signals due to noise, transmission range as well as the evaluation of the fitness function.

The localization error of a sensor c represents the difference between its estimated position (x_e, y_e) and its real position (x_r, y_r). The average network localization error for N sensors is evaluated by Eq. 16.

$$\xi = \frac{1}{N} \sum_{i=0}^{N} \sqrt{(x_{e_i} - x_{r_i})^2 + (y_{e_i} - y_{r_i})^2} \tag{16}$$

Each result presented is the sum of the results of 5 simulation rounds. To carry out these simulations, we used the following general parameters of Table 1.

Specifically, Table 2 presents the parameters used to simulate each method.

Table 1 General simulation parameters

Parameters	Values
Environment size	700×700
Number of anchors	1 to 10
Number of sensors	20
Standard deviation (σ)	0.1 to 1
Transmission range (R_{Com})	10 to 40
Population size	50
Maximum iteration count	100

Table 2 Specific simulation parameters

Algorithm	Parameters
HPSOVNS	NS=1, ci=0.729, cc=1.494, cs=1.494,NI_{VNS} = 50,K_{VNS} = 5
NS-IPSO	NS=5, ci=0.729, cc=1.494, cs=1.494, cg=1.494
ECS-NL	α = [0.9-1.0], Pa= [0.05–0.25]
GTOA-NL	TF=1
FPSOTS	NS=1, ci=0.729, cc=1.494, cs=1.494,NI_{TS} = 50

NS: number of swarms; NI: number of iterations; TF: teaching factor; Pa: probability of mutation

For the choice of parameters in Tables 1 and 2, we relied heavily on data from references [18, 21, 22, 25, 32].

5.1 Localization accuracy according to anchors number

Table 3; Fig. 4 present, according to anchors number, a comparison of 5 samples of different algorithms. We can observe on data of these comparisons, the capacity of the FPSOTS method to provide a good localization whatever the number of anchors used. In fact, the position estimation error obtained by FPSOTS is much lower than those obtained by others algorithms. We believe this is due to our improved version of PSO and our form of tabu search algorithm, which allows a particle to improve quickly and easily while also improving the overall performance of the optimization process. The different simulation runs were carried out with $\sigma = 1$ and $R_{Com} = 25$.

5.2 Localization accuracy according to environmental noise

As RSSI measurements are often affected by environmental noise, we assess in our way the robustness of algorithms studied according to various standard deviation values of sensor signals. Results obtained, presented in Table 4; Fig. 5, testify to the robustness of FPSOTS with respect to environmental noise compared to other algorithms. The simulations of this category were made with 6 anchors and $R_{Com} =$ 25.

It can be seen that the FPSOTS method considerably reduces position estimation errors obtained by other methods. This allows us to validate the operating mode efficiency of FPSOTS in minimizing the position estimation error in presence of noise.

5.3 Localization accuracy according to the transmission range

In principle, when the communications range is great, localization is more precise, because sensors are well covered by anchor signals. They thereby receive a lot of information for the adjustment of their localization estimation function. We then evaluate the impact of transmission range on localization error in FPSOTS. Results of the evaluation carried out with 6 anchors and a standard deviation $\sigma = 1$ are presented in

Table 3 Average localization error values vs. number of anchors

Anchor number	Round	HPSOVNS	NS-IPSO	ECS-NL	GTOA	FPSOTS
1	1	2,823	2,837	2,483	2,28	1,815
	2	2,3	1,991	2,308	1,82	1,601
	3	2,674	1,897	1,918	1,728	1,691
	4	1,901	2,439	1,981	1,908	1,764
	5	2,815	2,855	2,801	2,123	1,897
	Average	*2,5026*	*2,4038*	*2,2982*	*1,9718*	*1,7536*
3	1	1,955	2,262	2,955	1,852	1,611
	2	2,141	1,878	1,888	1,855	1,603
	3	1,877	1,828	1,903	1,911	1,728
	4	1,812	1,912	1,937	1,872	1,718
	5	1,898	1,894	1,814	1,798	1,649
	Average	*1,9366*	*1,9548*	*2,0994*	*1,8576*	*1,6618*
5	1	1,811	1,711	1,831	1,726	1,581
	2	1,645	1,645	1,648	1,605	1,345
	3	1,629	1,695	1,806	1,555	1,416
	4	1,639	1,635	1,744	1,395	1,515
	5	1,733	1,603	1,768	1,533	1,398
	Average	*1,6914*	*1,6578*	*1,7594*	*1,5628*	*1,451*
7	1	1,329	1,297	1,538	1,297	1,191
	2	1,596	1,296	1,347	1,226	1,111
	3	1,357	1,426	1,517	1,327	1,08
	4	1,392	1,286	1,506	1,266	0,973
	5	1,565	1,357	1,385	1,257	1,107
	Average	*1,4478*	*1,3324*	*1,4586*	*1,2746*	*1,0924*
10	1	1,383	0,992	1,034	0,983	0,796
	2	0,929	0,986	0,983	0,908	0,625
	3	1,098	0,985	1,982	1,098	0,898
	4	0,938	0,977	0,952	0,777	0,687
	5	0,987	0,951	0,957	0,878	0,638
	Average	*1,067*	*0,9782*	*1,1816*	*0,9288*	*0,7288*

It is the average of the 5 localization errors obtained by each approach, for each number of anchors used.

Fig. 4 Localization error vs. number of anchors

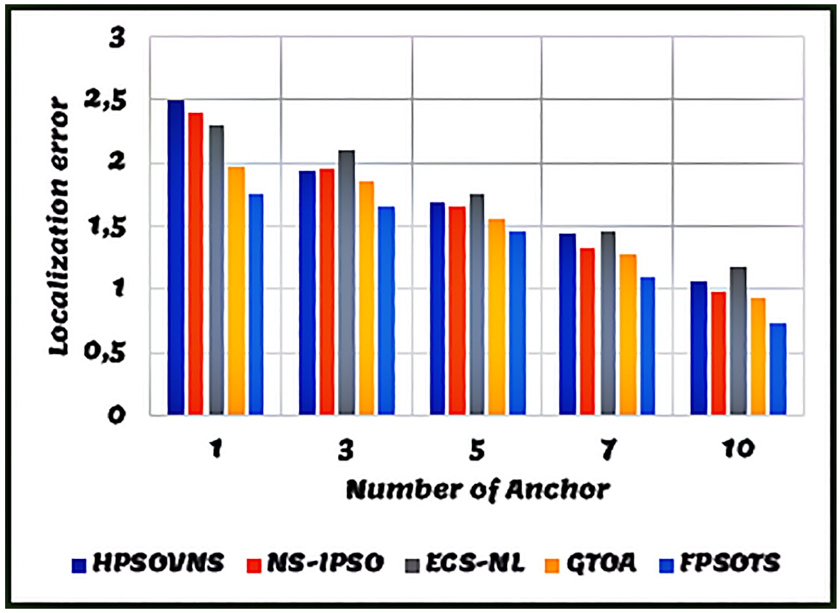


Table 5; Fig. 6. Note that the proposed FPSOTS approach is more precise than the other algorithms regardless of transmission range.

5.4 Evolution of fitness function

Graphs in Fig. 7 present samples of the evolution of fitness function of different algorithms studied. In general, we can note that FPSOTS converges quickly towards an optimal solution compared to other algorithms. This rapid convergence is due not only to the limit and performance checks introduced in PSO, but also to the form of tabu search used, which gives particles the chance to visit a better solution to the problem very quickly.

5.5 Discussion

Table 6 globally presents, a summary of the localization errors obtained by existing approaches and the proposed FPSOTS approach. In Table 6, we note that whatever the anchor's number in the network, the error in distances measurement and the node's transmission range in the network, the FPSOTS algorithm always provides a more precise localization than HPSOVNS, NS-IPSO, ECS-NL, and GTOA algorithms which are fairly good and recent localization algorithms in the literature.

From these results, we can therefore affirm that, allowing each particle of a PSO process to determine its best local neighbor by a tabu search method, allows the algorithm to achieve an efficient result. Moreover, equipping the particles with limit and performance checks allows them to evolve efficiently towards a promising result.

Thus, the FPSOTS algorithm is an efficient search algorithm for solving the indoor localization problem in a WSN. But we believe that FPSOTS can also be used to solve other problems in WSN such as the coverage problem or finding optimal paths between nodes of a network.

As an example of using the FPSOTS method, we can think of the location of mines on a battlefield. We can also think of the location of fire risk areas in forests or the location of survivors of a natural disaster. In these different examples, a group of sensors can be sent on a mission in the environment in question. In this group of sensors, very few can be equipped with GPS to limit the negative impact of GPS on the network. Other sensors without GPS will be able to use our indoor location method to determine their location. Indeed, for a sensor to locate a mine, a fire risk area, or a survivor of a disaster, it must know its location itself. In such application examples, the indoor location method used must be very precise on the different calculated positions. And we have shown that FPSOTS is a method that provides much more accurate localization than the existing methods studied.

6 Conclusions

In this paper, a new method to localize sensors in a WSN is proposed and evaluated. The proposed method named FPSOTS is a range-based localization approach using optimization techniques for estimating the localization of sensors. FPSOTS improves the PSO algorithm by introducing a form of tabu search for the determination of the best local neighbor of a particle, as well as limit and performance

Table 4 Average localization error vs. standard deviation

Standard deviation	Round	HPSOVNS	NS-IPSO	ECS-NL	GTOA	FPSOTS
0,1	1	0,316	0,301	0,481	0,28	0,215
	2	0,33	0,311	0,38	0,32	0,301
	3	0,41	0,395	0,418	0,28	0,261
	4	0,401	0,39	0,433	0,38	0,24
	5	0,315	0,325	0,41	0,33	0,307
	Average	*0,3544*	*0,3444*	*0,4244*	*0,318*	*0,2648*
0,3	1	0,555	0,52	0,655	0,52	0,5
	2	0,541	0,55	0,615	0,455	0,403
	3	0,6	0,48	0,603	0,411	0,38
	4	0,62	0,69	0,67	0,572	0,418
	5	0,67	0,54	0,691	0,498	0,38
	Average	*0,5972*	*0,556*	*0,6468*	*0,4912*	*0,4162*
0,5	1	0,78	0,699	0,801	0,7	0,468
	2	0,765	0,689	0,864	0,654	0,505
	3	0,777	0,748	0,806	0,755	0,417
	4	0,699	0,752	0,791	0,712	0,494
	5	0,763	0,702	0,808	0,622	0,499
	Average	*0,7568*	*0,718*	*0,814*	*0,6886*	*0,4766*
0,7	1	0,793	0,713	0,792	0,633	0,632
	2	0,841	0,827	0,91	0,79	0,579
	3	0,769	0,746	0,873	0,686	0,679
	4	0,892	0,879	0,889	0,734	0,614
	5	0,799	0,801	0,878	0,792	0,658
	Average	*0,8188*	*0,7932*	*0,8684*	*0,727*	*0,6324*
1	1	0,952	0,959	1,334	0,886	0,779
	2	0,979	0,911	0,999	0,951	0,788
	3	0,963	0,911	1,162	0,847	0,725
	4	1,222	0,927	0,997	0,839	0,803
	5	0,979	1,001	0,977	0,847	0,798
	Average	*1,019*	*0,9418*	*1,0938*	*0,874*	*0,7786*

It is the average of the 5 localization errors obtained by each approach, for each value of standard deviation.

checks in the evolution of particles. The implemented tabu search allowed particles to quickly discover a better solution during the optimization process. In FPSOTS, the initial solution is determined by trilateration and search space is constructed by a simple process of constraints analysis. The final solution to sensor localization is subsequently determined by our optimization process which merges PSO and tabu search. This tabu search, coupled with limit and performance checks introduced in PSO, made it possible to efficiently determine the location of a sensor. The accuracy of FPSOTS was evaluated in simulation based on anchor density, noise effect and anchor transmission range. Overall, evaluation of localization error vs. anchors density of network demonstrates that FPSOTS improves HPSOVNS, NS-IPSO, ECS-NL and GTOA by approximately 39%, 33%, 42% and 19% respectively. Evaluation of the impact of signal standard deviation on localization accuracy showed that FPSOTS improves HPSOVNS, NS-IPSO, ECS-NL and GTOA approaches by approximately 24%, 23%, 27% and 18%. According to the analysis of the impact of nodes transmission range on the determination of sensors location, the proposed FPSOTS approach improved HPSOVNS, NS-IPSO, ECS-NL and GTOA methods by about 30%, 28%, 25% and 21%. Analyzes of results show the efficiency of FPSOTS as an optimization algorithm for solving indoor localization problems in WSN with a fast convergence towards an optimal solution. In future directions for this work, we can study the efficiency of the algorithm if the tabu search was replaced by a population metaheuristic. In this case, it will be necessary to evolve not only localization accuracy, but also localization time and energy consumption of the sensors. In addition, we want to verify the efficiency of FPSOTS in real

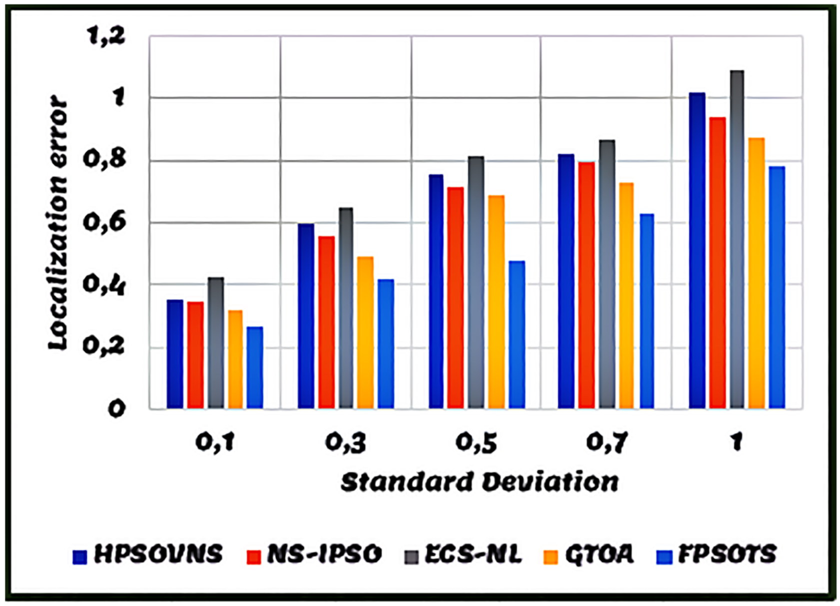


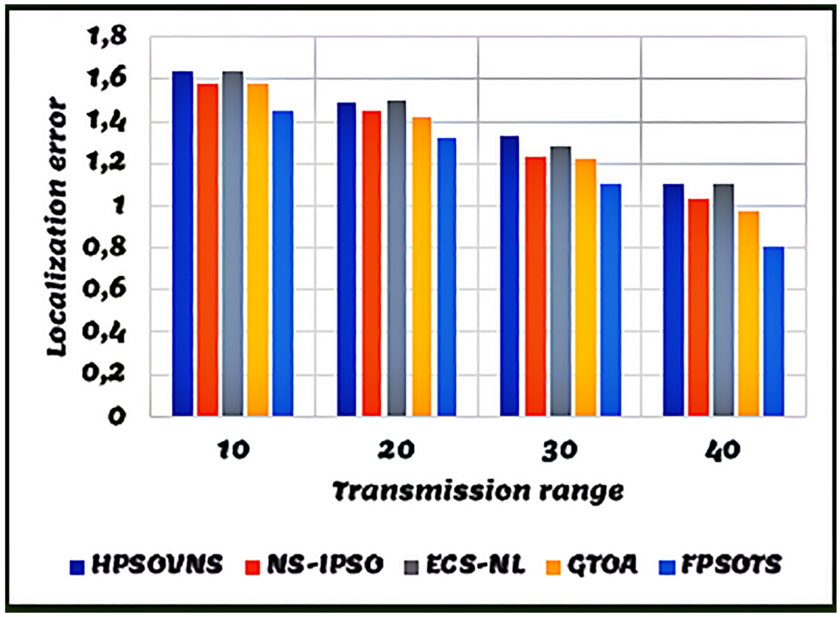
Fig. 5 Localization error vs. standard deviation

Table 5 Localization error values vs. transmission range

Transmission range	Round	HPSOVNS	NS-IPSO	ECS-NL	GTOA	FPSOTS
10	1	1,576	1,326	1,576	1,467	1,315
	2	1,607	1,633	1,617	1,611	1,497
	3	1,654	1,645	1,65	1,599	1,491
	4	1,672	1,661	1,675	1,642	1,464
	5	1,666	1,632	1,671	1,599	1,497
	Average	*1,635*	*1,5794*	*1,6378*	*1,5836*	*1,4528*
20	1	1,581	1,549	1,582	1,546	1,321
	2	1,458	1,346	1,448	1,337	1,323
	3	1,574	1,539	1,57	1,512	1,258
	4	1,477	1,396	1,427	1,326	1,321
	5	1,365	1,436	1,455	1,377	1,363
	Average	*1,491*	*1,4532*	*1,4964*	*1,4196*	*1,3172*
30	1	1,348	1,266	1,268	1,249	1,151
	2	1,281	1,206	1,258	1,203	1,057
	3	1,271	1,177	1,271	1,154	1,129
	4	1,362	1,242	1,302	1,238	1,12
	5	1,386	1,251	1,325	1,247	1,047
	Average	*1,3296*	*1,2284*	*1,2848*	*1,2182*	*1,1008*
40	1	1,198	1,024	1,051	1,025	0,851
	2	1,215	0,96	1,147	1,026	0,855
	3	0,998	1,026	1,107	0,927	0,719
	4	1,117	1,206	1,206	0,946	0,811
	5	0,996	0,957	0,999	0,937	0,799
	Average	*1,1048*	*1,0346*	*1,102*	*0,9722*	*0,807*

It is the average of the 5 localization errors obtained by each approach, for each value of node transmission range.

Fig. 6 Localization error vs. transmission range



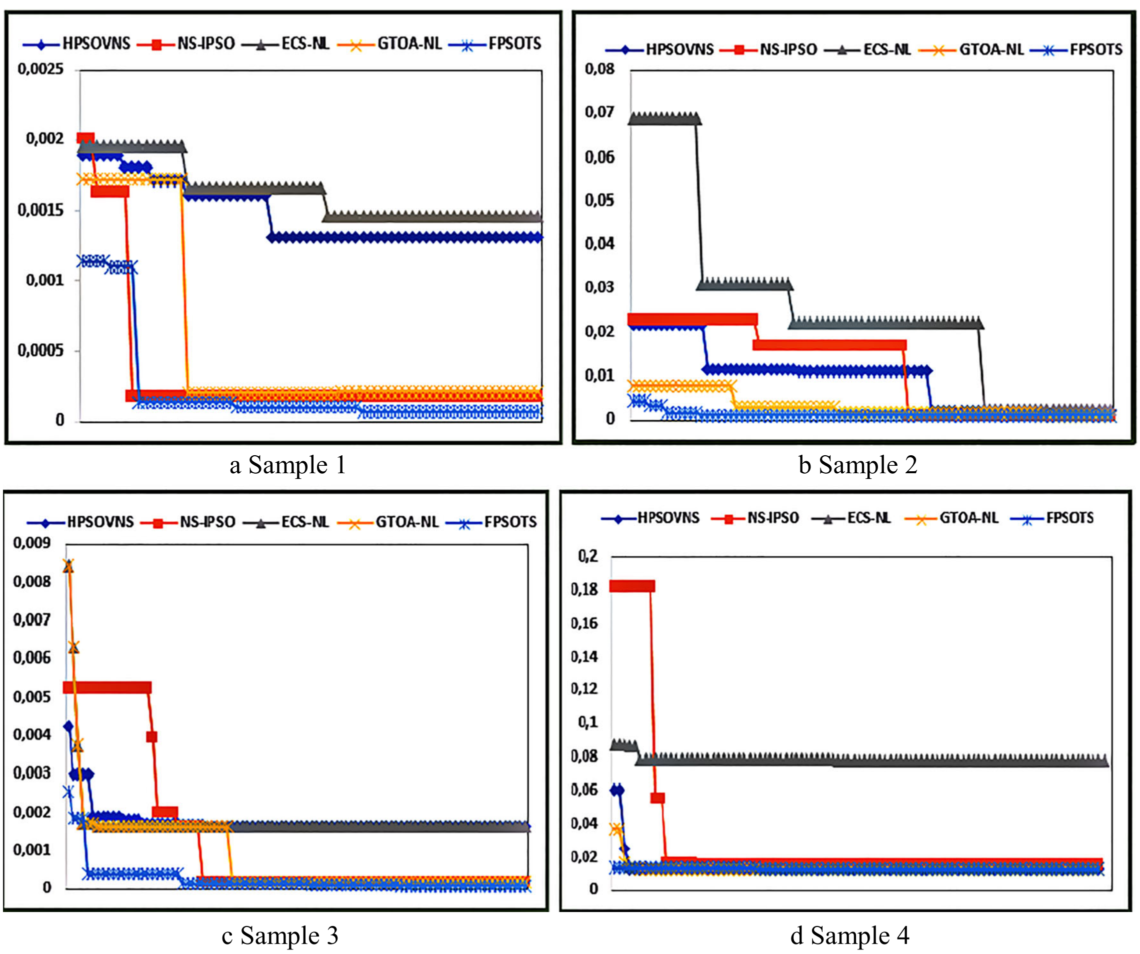


Fig. 7 Evolution of fitness function. **a** Sample 1. **b** Sample 2. **c** Sample 3. **d** Sample 4

Table 6 Results of localization accuracy vs. anchor densities, localization accuracy vs. standard deviation and localization accuracy vs. transmission range

Anchor number	HPSOVNS	NS-IPSO	ECS-NL	GTOA	FPSOTS
1	2,50	2,40	2,30	1,97	**1,75**
3	1,94	1,95	2,10	1,86	**1,66**
5	1,69	1,66	1,76	1,56	**1,45**
7	1,45	1,33	1,46	1,27	**1,09**
10	1,07	0,98	1,18	0,93	**0,73**
Standard deviation	HPSOVNS	NS-IPSO	ECS-NL	GTOA	FPSOTS
0,1	0,35	0,34	0,42	0,32	**0,26**
0,3	0,60	0,56	0,65	0,49	**0,42**
0,5	0,76	0,72	0,81	0,69	**0,48**
0,7	0,82	0,79	0,87	0,73	**0,63**
1	1,02	0,94	1,09	0,87	**0,78**
Transmission range	HPSOVNS	NS-IPSO	ECS-NL	GTOA	FPSOTS
10	1,63	1,58	1,64	1,58	**1,45**
20	1,49	1,45	1,50	1,42	**1,32**
30	1,33	1,23	1,28	1,22	**1,10**
40	1,10	1,03	1,10	0,97	**0,81**

Bold values express the smallest localization error obtained by each simulation. A simulation is described by a line of the table. This also in order to show that the FPSOTS method therefore has the smallest localization errors.

experimentation on a WSN and study the introduction of an early stop of the optimization process, when no change has been observed for a certain time.

Data availability Not applicable.

Code availability Not applicable.

Declarations

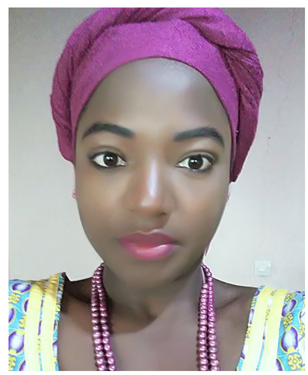
Conflict of interest Authors declare no conflict of interest.

References

1. Singh M, Bhoi SK, Khilar PM (2017) Geometric constraint-based range free localization scheme for wireless sensor networks. IEEE Sens J 17(16):5350–5366
2. Han G, Xu H, Duong TQ, Jiang J, Hara T (2013) Localization algorithms of wireless sensor networks: a survey. Telecommun Syst 52(4): 2419–2436
3. Sakpere W, Adeyeye-Oshin M, Mlitwa NBW (2017) A state-of-the-art survey of indoor positioning and navigation systems and technologies. South Afr Comput J 29:145–197
4. Guyonneau R (2013) Méthodes ensemblistes pour la localisation en robotique mobile. PhD Thesis, Université de Angers
5. Singh M, Khilar PM (2017) Mobile beacon-based range free localization method for wireless sensor networks. Wirel Netw 23(4):1285–1300
6. Sharma R, Prakash S (2021) Hho-lpwsn: Harris hawks optimization algorithm for sensor nodes localization problem in wireless sensor networks. EAI Endorsed Transactions on Scalable Information Systems 8(31). https://doi.org/10.4108/eai.25-2-2021.168807
7. Yang J, Li Y, Cheng W (2018) An improved geometric algorithm for indoor localization. Int J Distrib Sens Netw 14(3). https://doi.org/10.1177/1550147718767376
8. Kumar A, Khosla A, Saini JS, Singh S (2012) Computational intelligence-based algorithm for node localization in wireless sensor networks. In: 2012 6th IEEE International Conference Intelligent Systems, pp 431–438. https://doi.org/10.1109/IS.2012.6335173
9. Lakhbab H (2018) A novel hybrid approach for optimizing the localization of wireless sensor networks. In: MATEC Web of Conferences 200, 00005. https://doi.org/10.1051/matecconf/201820000005
10. Li C, Xie J, Wu W, Tian H, Liang Y (2019) Monte Carlo localization algorithm based on particle swarm optimization. Automatika 60(4): 451–461
11. Kanoosh HM, Houssein EH, Selim MM (2019) Salp swarm algorithm for node localization in wireless sensor networks. J Comput Networks Commun 2019. https://doi.org/10.1155/2019/1028723
12. Gumaida BF, Luo J (2019) A hybrid particle swarm optimization with a variable neighborhood search for the localization enhancement in wireless sensor networks. Appl Intell. Springer. https://doi.org/10.1007/s10489-019-01467-8
13. Aloor G, Jacob L (2009) Performance of some metaheuristic algorithms for localization in wireless sensor networks. Int J Netw Manag 19:355–373. https://doi.org/10.1002/nem.714
14. Paul AK, Li Y, Sato T (2012) A distributed range free sensor localization with friendly anchor selection strategy in anisotropic wireless sensor network. Trans Japan Soc Simul Technol 4(3):96–106
15. Singh P, Khosla A, Kumar A (2018) Computational intelligence-based localization of moving target nodes using single anchor node in wireless sensor networks. Telecommun Syst. https://doi.org/10.1007/s11235-018-0444-2
16. Gharib A, Benhra J (2015) tuning manuel de l'algorithme d'optimisation par essaims de particules appliqué au problème de voyageur. Conception et Production Intégrées, Xme Conférence Internationale
17. Mekelleche F, Haffaf H (2017) Classification and comparison of range-based localization techniques in wireless sensor networks. J Commun 12(4):221–227

18. Yassin A, Nasser Y, Awad M, Al-Dubai A, Liu R, Yuen C, Raulefs R, Aboutanios E (2017) Recent Advances in Indoor Localization: A Survey on Theoretical Approaches and Applications. IEEE Commun Surv Tutorials 19(2):1327–1346. https://doi.org/10.1109/comst.2016.2632427
19. Li L, Wu Y, Ren Y, Yu N (2013) A RSSI localization algorithm based on interval analysis for indoor wireless sensor networks. IEEE International Conference on Green Computing and Communications and IEEE Internet of Things and IEEE Cyber, Physical and Social Computing
20. Wu H, Liu J, Dong Z, Liu Y (2020) A Hybrid Mobile Node Localization Algorithm Based on Adaptive MCB-PSO Approach in Wireless Sensor Networks. Wirel Commun Mob Comput, 1–17. https://doi.org/10.1155/2020/3845407
21. Aloor G, Jacob L (2008) Localization in wireless sensor networks using particle swarm optimization. In: IET (ed.) 2008 IET International Conference on Wireless, Mobile and Multimedia Networks, pp 227–230. IET. https://doi.org/10.1049/cp:20080185
22. Kazem S, Hossien YM, Baradaran KM, Hossein D (2010) Localization in wireless sensor networks using tabu search and simulated annealing. In: 2010 The 2nd International Conference on Computer and Automation Engineering (ICCAE), vol 2, pp 752–757. https://doi.org/10.1109/ICCAE.2010.5451779
23. Sekhar P, Lydia EL, Elhoseny M, Al-Akaidi M, Selim MM, Shankar K (2021) An effective metaheuristic-based node localization technique for wireless sensor networks enabled indoor communication. Phys Commun 48:101411. https://doi.org/10.1016/j.phycom.2021.101411
24. Vaibhav K, Abhilash S, Sandeep S, Jaiprakash N, Cheng-Chi L (2021) Ecs-nl: An enhanced cuckoo search algorithm for node localization in wireless sensor networks. Sensors 21(11). https://doi.org/10.3390/s21113576
25. Phoemphon S, So-In C, Leelathakul N (2020) A hybrid localization model using node segmentation and improved particle swarm optimization with obstacle-awareness for wireless sensor networks. Expert Syst Appl 143:113044. https://doi.org/10.1016/j.eswa.2019.113044
26. Han G, Jiang J, Zhang C, Duong TQ, Guizani M, Karagiannidis GK (2016) A Survey on Mobile Anchor Node Assisted Localization in Wireless Sensor Networks. IEEE Commun Surv Tutorials 18(3):2220–2243. https://doi.org/10.1109/comst.2016.2544751
27. Mesmoudi A, Feham M, Labraoui N (2013) Wireless sensor networks localization algorithms: A comprehensive survey. Int J Comput Netw Commun Secur (IJCNC) 5(6). https://doi.org/10.5121/ijcnc.2013.5603
28. Yang J, Cai Y, Tang D, Liu Z (2019) A Novel Centralized Range-Free Static Node Localization Algorithm with Memetic Algorithm and Lévy Flight. Sensors 19(14):3242. https://doi.org/10.3390/s19143242
29. Halder S, Ghosal A (2016) A survey on mobile anchor assisted localization techniques in wireless sensor networks. Wirel Netw 22(7):2317–2336
30. Kamal M, Lounis M, Bounceur, Tahar Kechadi M (2014) Cupcarbon: a multi-agent and discrete event wireless sensor network design and simulation tool. In Barros F, Perumalla KS, Ewald R (eds) 7th International ICST Conference on Simulation Tools and Techniques, SIMUTools '14, Lisbon, Portugal, March 17–19. ICST/ACM, pp 126–131
31. Alzyoud F, Sharman NAL, Al-Roosan T, Alsalah Y (2019) Smart accident management in jordan using cupcarbon simulation. Eur J Sci Res 152(2):128–135
32. Cui H, Shu M, Song M, Wang Y (2017) Parameter Selection and Performance Comparison of Particle Swarm Optimization in Sensor Networks Localization. Sensors 17(3):487. https://doi.org/10.3390/s17030487

Elie Tagne Fute is a lecturer in the Department of Mathematics and Computer Science, Faculty of Science, University of Dschang, Cameroon. He is currently the Head of Department of Computer Engineering, Faculty of Engineering and Technology, University of Buea, Cameroon. His research works are focused on protocols and security, multi-agent patrolling, sensor networks and multi-sensor patrolling.



Doris-Khöler Nyabeye Pangop is a PH.D. student in fundamental computer science at University of Dschang. She got his Master degree at University of Dschang in 2016. Her research interests focus on localization in wireless sensor network, on multi-agent patrolling, sensor networks and multi-sensor patrolling.

Emmanuel Tonye is a lecturer in the Department of Electrical and Telecommunication Engineering of the National Advanced School of Engineering (University of Yaounde I). His field of study is Telecommunication and his current research interests include communication system, signal and image processing, E-learning.