3D Localization in Large-Scale Wireless Sensor Networks: A Micro-Differential Evolution Approach

Hojjat Salehinejad, Student Member, IEEE, Robert Zadeh, Student Member, IEEE, Ramiro Liscano, Senior Member, IEEE, and Shahryar Rahnamayan, Senior Member, IEEE
Department of Electrical, Computer, and Software Engineering, University of Ontario Institute of Technology,
2000 Simcoe Street North, Oshawa, ON L1H 7K4, Canada
Email: {hojjat.salehinejad, robert.zadeh, ramiro.liscano, shahryar.rahnamayan}@uoit.ca

Abstract—Most of the recent proposed approaches for sensor(mote) localization are focused on 2-D environments with limited functionalities. This is mostly due to the nature of problem which is non-linear, large-scale, and has limited hardware resources. The micro-evolutionary algorithms (MEAs) utilize a small-size population to solve optimization problems. Therefore, such algorithms require much less processing time and memory than standard evolutionary algorithms (EAs), suitable for implementation on embedded systems. In this paper, a novel protocol for localization of motes in 3-D environments is proposed, simulated, and discussed. The localization problem is modeled as an optimization problem. The proposed model is based on a realistic approach to the localization problem, where possible errors and noises in the localization procedure such as signal strength detection are addressed. To present a suitable approach to solve the proposed optimization model, a comparative study on performance of the micro-differential evolution (MDE) algorithms is performed and the results are discussed.

Index Terms—Localization, Micro-Differential Evolution (MDE), Wireless Sensor Networks (WSNs).

I. INTRODUCTION

Complexity of problems dealing with wireless sensor networks (WSNs) are increasing tremendously. Some of the challenges facing WSNs are spectrum assignment [1] and power management [2]. One of the main challenges is finding the exact location of the mote (sensor), called localization problem. Even though a simple and fast idea to overcome this problem is equipping each mote with geographical positioning system (GPS) modules; however such power hungry systems not only require line-of-sight connection with satellites but also are expensive in comparison with the mote cost itself, particularly for large-scales deployments [3].

Evolutionary algorithms (EAs), such as differential evolution (DE) algorithm, are state-of-the-art methods for solving real-world problems [4]. The term micro-algorithm refers to population-based algorithms with a small population size [5]. The micro-algorithms generally suffer from lack of diversity in the population, which limits the exploration of algorithm for feasible solution. To overcome this problem for micro-differential evolution (MDE) algorithms, methods such as micro-differential evolution with scalar random mutation factor (MDESM) and micro-differential evolution with vectorized random mutation factor (MDEVM) are proposed [5]. The sensor localization problem is formulated as a NP-hard optimization problem [6]. In order to solve such problems, traditional analytical optimization techniques such as quadratic programming and interior-point methods require enormous computational efforts [6], [7]. Therefore, high performance optimization methods with moderate usage of computational resources are of interest. Soft computing algorithms such as genetic algorithm (GA) [9], particle swarm optimization (PSO) [6], and artificial neural networks (ANNs) [10] have been employed for localization in WSNs. In this paper, the localization problem is modeled as an optimization problem and a comparative study on performance of the MDE algorithms family is conducted.

The next section reviews the DE algorithm. In Section III, the proposed approach is discussed and then evaluated in Section IV. Finally, the paper is concluded in Section V.

II. DIFFERENTIAL EVOLUTION

The DE algorithm works based on the scaled difference between two individuals of a population’s set, where the scaling factor is called the mutation factor [12]. In DE algorithm, the population \( P = \{ X_1, ..., X_{NP} \} \) consists of \( NP \) vectors in generation \( g \), where \( X_i \) is a \( D \)-dimensional vector defined as \( X_i = (x_{i1}, ..., x_{iD}) \). A simple DE algorithm consists of the following steps:

Mutation: This step selects three vectors randomly from the population such that \( i_1 \neq i_2 \neq i_3 \neq i \), where \( i \in \{1, ..., NP\} \) and \( NP \geq 4 \) for each vector \( X_i \). The mutant vector is calculated as

\[
V_i = X_{i1} + F(X_{i2} - X_{i3}),
\]

where the factor \( F \in (0,2] \) is a real constant number, which controls the amplification of the added differential variation of \( (X_{i2} - X_{i3}) \). The exploration of DE increases by selecting higher values for \( F \).

Crossover: The crossover operation increases diversity of the population by shuffling the mutant and parent vector as follows:

\[
U_{i,d} = \begin{cases} V_{i,d}, & \text{rand}_d(0,1) \leq Cr \text{ or } d_{rand} = d, \\ x_{i,d}, & \text{otherwise} \end{cases}
\]

where \( d = 1, ..., D, Cr \in [0,1] \) is the crossover rate.
In this section, the proposed protocol for mote localization in WSNs is presented.

A. Localization Protocol

In real-world applications of WSNs, such as temperature detection in jungles, the motes are deployed randomly on the landscape using air-crafts such as UAVs. In most cases, the motes are not time-synchronized with each other or with their outer environment. In order to localize motes after deployment, we have proposed a protocol as presented in Figures 2 to 3. In the proposed topology, after deployments of sensors on the landscape, the UAV flights on the landscape and broadcasts a signal every $\Delta t_i$ seconds, after flying a distance $\Delta X_i$ as denoted in Figure 2. The message package sent from the UAV is consisted of the packet identification number $P_i$, time sent $t_i$, UAV current position $DC_i = (X_{DC_i}, Y_{DC_i}, Z_{DC_i})$, UAV flew distance from last position $DC_{i-1}$ to current position $DC_i$, i.e. $\Delta X_{i-1}$, and UAV average speed from last sent package at time $t_{i-1}$ to $t_i$, i.e. $\Delta V_{i-1}$. Each sensor on the landscape $S$ receive a number of packages $N_S$. Since we are considering the asynchronous mode, which is a more challenging mode than

algorithm is typically a constant mutation factor (CMF), generally set to $F = 0.5$ [5]. In order to deliver diversity to the population, two ideas are proposed recently [5]. The first one is utilizing a random mutation factor for all individuals in each generation in a scalar manner, i.e. the MDESM method. In this case, the mutation factor is considered as

$$F_i = \text{rand}(0.1, 1.5), \forall i \in \{1, ..., N_P\}. \quad (4)$$

The second idea is utilizing a random vector (not scalar) $F$ for each individual in the population is proposed in [5], the MDEVM algorithm. In this algorithm, for each individual $i$ in the population vector, $F_i = \text{rand}(0.1, 1.5)$ selects a value randomly from the interval $[0.1, 1.5]$. Therefore, the mutation factor can be defined for each individual $i$ as

$$F_i = \{F_{i,1}, ..., F_{i,D}\}, \forall i \in \{1, ..., N_P\}, \quad (5)$$

where $F_{i,j} = \text{rand}(0.1, 1.5), \forall j \in \{1, ..., D\}$, [5].

III. PROPOSEDMOTE LOCALIZATION SCHEME

A. Micro-Differential Evolution

The reduced population size in the MDE algorithm, i.e. $N_P \leq 5$, decreases the computational time dramatically. However, the population size reduction increases the risk of stagnation as well as premature convergence in finding the global optimum solution. The stagnation is not the same as premature convergence. During stagnation, the population remains non-converged but divert and the optimization process does not progress. A large population size offers a more diversified pool of individuals whose recombination offers higher likelihood to locate the global solution [5]. Therefore, reducing the population size while raising the diversity of the population is a key point to achieve a faster convergence speed while maintaining a low risk of premature convergence or stagnation.

B. Micro-Differential Evolution with Vectorized Random Mutation Factor

The mutation factor $F$ plays a major role to deliver diversity into population. The mutation factor $F$ in the DE

$$\mu_{i} = (U_{i,1}, ..., U_{i,D}). \quad (3)$$

Selection: The $U_i$ and $X_i$ vectors are evaluated and compared with respect to their fitness values; the one with better fitness is selected for the next generation.

Parameter, and $\text{rand}(a, b)$ generates a random number in the interval $[a, b]$ with a uniform distribution. Therefore, the trial vector $U_i \forall i \in \{1, ..., N_P\}$ can be generated:

$$U_i = (U_{i,1}, ..., U_{i,D}). \quad (3)$$

In this case, the mutation factor is considered as

$$F_i = \text{rand}(0.1, 1.5), \forall i \in \{1, ..., N_P\}. \quad (4)$$

The second idea is utilizing a random vector (not scalar) $F$ for each individual in the population is proposed in [5], the MDEVM algorithm. In this algorithm, for each individual $i$ in the population vector, $F_i = \text{rand}(0.1, 1.5)$ selects a value randomly from the interval $[0.1, 1.5]$. Therefore, the mutation factor can be defined for each individual $i$ as

$$F_i = \{F_{i,1}, ..., F_{i,D}\}, \forall i \in \{1, ..., N_P\}, \quad (5)$$

where $F_{i,j} = \text{rand}(0.1, 1.5), \forall j \in \{1, ..., D\}$, [5].

Fig. 1. Motes deployment on a sample landscape and corresponding localization with assistance of a UAV, flying on a random path and broadcasting a detection signal on the landscape.

Fig. 2. Sample signal coverage model for four detected coordinates (DCs) to the mote.

Fig. 3. Timing-diagram of packets arrival from four detection coordinates (DCs) to the mote.
the synchronized mode, the first received packet is used to set the detection clock of sensor to \( T_1 \) as denoted in the timing diagram illustrated in Figure 3. As the time passes and arrives to \( T_2 \) on the sensor detection clock, that is the time second packet is received, the sensor calculates the time \( t_{d2} \) spent on packet signal arrival from UAV at position \( DC_2 \) to sensor by using the timing relation as

\[
T_2 = T_1 + \Delta t_1 + t_{d2},
\]

where \( \Delta t_1 \) is the time UAV spends to fly from \( DC_1 \) to \( DC_2 \). By having \( t_{d2} \) in mind, the distance from detected position of UAV \( DC_2 \) to the sensor location \((X_s, Y_s, Z_s)\) is estimated as \( d_2 = V_{EM} \times t_{d2} \) where \( V_{EM} \) is the electromagnetic signal speed set to \( 2.99 \times 10^8 m/sec \) [11]. The same procedure is repeated for the other \( N_S \) received packets in the general form

\[
t_{di} = T_i - (T_{i-1} + \Delta t_{i-1}),
\]

where the distance between each \( DC_i \) and the sensor location \((X_s, Y_s, Z_s)\) is computed as \( d_i = V_{EM} \times t_{d2} \). The above procedure is conducted for each sensor separated from the rest of sensors on the landscape in a standalone manner. By estimating the distances \( d_i \) \( \forall i \in \{1, ..., N_S\} \) from all received detection coordinates \( DC_i \)'s to the sensor location \((X_s, Y_s, Z_s)\), the sensor localization problem can be modeled.

### B. Localization modeling

By considering the \( DC_i \) \( \forall i \in \{1, ..., N_S\} \) coordinates as well as the estimated distances \( d_i \) \( \forall i \in \{1, ..., N_S\} \) to the UAV by each mote, the location of mote, i.e. \( (X_s, Y_s, Z_s) \), can be estimated by intersection of \( N_S \geq 3 \) spheres for a 3-D environment, where the interconnection point is the estimated mote location. However, as it is demonstrated in Figure 4 for a typical 2-D localization problem, in real-world scenarios many errors may occur during the localization procedure such as signal coverage model which is not ideally a sphere, communication channel estimation and modeling, signal path loss and fading, computational errors for instance in speed of UAV or speed of electromagnetic waves, and etc. Therefore, solving the ideal model achieved by interconnecting of spheres is not a realistic approach due to the existence of an additive estimation error between the computed interconnection of spheres and the exact location of mote. To overcome this problem, the additive estimation error is considered in our modeling. In this approach, on the average an equal error for each estimated distance \( d_i \) between detection coordinated and the sensor location is assumed.

By having \( N_S \) detection coordinates, the sphere equation for spheres \( i \in \{1, ..., N_S\} \) is defined as

\[
\begin{align*}
(X_s - X_{DC_i})^2 + (Y_s - Y_{DC_i})^2 + (Z_s - Z_{DC_i})^2 &= d_i^2, \\
&\vdots \\
(X_s - X_{DC_{N_S}})^2 + (Y_s - Y_{DC_{N_S}})^2 + (Z_s - Z_{DC_{N_S}})^2 &= d_{N_S}^2,
\end{align*}
\]

where by summing up the equations we will have

\[
\sum_{i=1}^{N_S} ((X_s - X_{DC_i})^2 + (Y_s - Y_{DC_i})^2 + (Z_s - Z_{DC_i})^2) = \sum_{i=1}^{N_S} d_i^2.
\]

By considering the estimated distances as \( \tilde{d}_i \) \( \forall i \in \{1, ..., N_S\} \) and \( e_i \) as the distance estimation error, the exact distance is defined as

\[
d_i = \tilde{d}_i + e_i
\]

where its square is

\[
d_i^2 = (\tilde{d}_i + e_i)^2
\]

By adding up the Eq. (11) for \( i \in \{1, ..., N_S\} \) we have

\[
\sum_{i=1}^{N_S} d_i^2 = \sum_{i=1}^{N_S} (\tilde{d}_i + e_i)^2.
\]

Then, by substituting Eq. (12) into Eq. (9) we have

\[
\sum_{i=1}^{N_S} ((X_s - X_{DC_i})^2 + (Y_s - Y_{DC_i})^2 + (Z_s - Z_{DC_i})^2) = \sum_{i=1}^{N_S} \tilde{d}_i^2 + 2 \sum_{i=1}^{N_S} \tilde{d}_i e_i + \sum_{i=1}^{N_S} e_i^2.
\]

By expanding the right side of Eq. (13) and replacing the left side with \( A \), we have

\[
A = \sum_{i=1}^{N_S} \tilde{d}_i^2 + 2 \sum_{i=1}^{N_S} \tilde{d}_i e_i + \sum_{i=1}^{N_S} e_i^2.
\]

where by considering equal errors for all estimations as \( e = e_i \) \( \forall i \in \{1, ..., N_S\} \), we can simplify Eq. (14) and arrive to a quadratic equation as

\[
N_S e^2 + (2 \sum_{i=1}^{N_S} \tilde{d}_i) e + (\sum_{i=1}^{N_S} \tilde{d}_i^2 - A) = 0.
\]

By solving the Eq. (15) for \( e \), the possible solutions are

\[
e = \frac{-2 \sum_{i=1}^{N_S} \tilde{d}_i \pm \sqrt{(2 \sum_{i=1}^{N_S} \tilde{d}_i)^2 - 4 N_S (\sum_{i=1}^{N_S} \tilde{d}_i^2 - A)}}{2 N_S}
\]

where the negative part of the square root is not applicable. Since we are willing to minimize the estimation error, \( e \), the sensor localization problem can be defined as an optimization problem such as

\[
\min(e) = \min \left( - \sum_{i=1}^{N_S} \tilde{d}_i + \sqrt{(\sum_{i=1}^{N_S} \tilde{d}_i)^2 - N_S (\sum_{i=1}^{N_S} \tilde{d}_i^2 - A)} \right)
\]

where the sensor exact location coordinates, i.e. \((X_s, Y_s, Z_s)\), are the variables of the optimization problem.
Mutation (MDESM) for Sensor Localization

Algorithm 1 Micro-Differential Evolution with Scalar Mutation (MDESM) for Sensor Localization

1: Procedure MDESM
2: \( g = 0 \) //Initial Population Generation
3: for \( i = 1 \rightarrow N_p \) do
4: for \( d = 1 \rightarrow D \) do
5: \( X_{i,d} = x_{d}^{\text{min}} + \text{rand}(0, 1) \times (x_{d}^{\text{max}} - x_{d}^{\text{min}}) \)
6: end for
7: \( P_{g} = X_{i} \)
8: end for//End of Initial Population Generation
9: while \((|BFV - VTR| > EVTR & NFC < NFC_{\text{Max}})\) do
10: for \( i = 1 \rightarrow N_p \) do
11: //Mutation
12: Select three random population vectors from \( P_{g} \) where \((i_1 \neq i_2 \neq i_3 \neq i)\)
13: \( F = \text{rand}(0.1, 1.5) \)
14: for \( d = 1 \rightarrow D \) do
15: \( V_{i,d} = X_{i,d} + F(X_{i_2,d} - X_{i_3,d}) \)
16: end for
17: //End of Mutation
18: //Crossover
19: for \( d = 1 \rightarrow D \) do
20: if \( \text{rand}(0, 1) < C_r \) or \( d_{\text{rand}} = d \) then
21: \( U_{i,d} = V_{i,d} \)
22: else
23: \( U_{i,d} = x_{i,d} \)
24: end if
25: end for
26: //End of Crossover
27: //Selection
28: if \( f(U_{i}) \leq f(X_{i}) \) then
29: \( X_{i} = X_{i}^\prime \)
30: else
31: \( X_{i} = X_{i}^\prime \)
32: end if
33: //End of Selection
34: end for
35: \( P_{g+1} = \{X_1, ..., X_{N_p}\} \)
36: \( g = g + 1 \)
37: end while

C. Solving Localization Model

By modeling the sensor localization problems as the optimization problem in Eq. (17), each sensor can utilize the MDESM algorithm (or MDEV algorithm for more complex problems) [5] to find its own \((X_s, Y_s, Z_s)\) location.

The pseudocodes of the MDESM algorithm is presented in Algorithm 1. The termination criterion is met when the difference between best fitness value \((BFV)\) and fitness value to reach \((VTR)\) is less than fitness error-value-to-reach \((EVTR)\), or the searching procedure exceeds the maximum number of function calls \(NFC_{\text{Max}}\).

IV. Simulation Results

In section, performance of the proposed protocol is evaluated by presenting a comparative study on performance of using the MDE family algorithms.

A. Parameters Setting

To simulate the sensor localization problem, 50 sensors are deployed randomly on the landscape illustrated in Figure 1 and the UAV has broadcasted 256 DC messages on the landscape, following a random flight path. The parameter setting of algorithms for all simulations are set as in Table I. The reported values are averaged for \(N_{\text{Run}}\) independent runs per sensor localization per algorithm to minimize the effect of the stochastic nature of the algorithms on the results [5].

B. Performance Evaluation

In order to study the effect of number of received packets by each sensor in sensor localization performance, the simulations are performed for \(N_S \in \{4, 15\} \) received number of packets. The average of performance results for 50 deployed sensors are reported in terms of number of function calls \((NFCs)\), \(N_{\text{Run}}\) number of runs, success rate \((SR)\), and best achieved accuracy in exact location finding \((Best)\) in Table II. As the results demonstrate, performance of the MDESM algorithm is the best among the other algorithms. The MDESM methods has \(SR = 99.77\) with \(NFC = 429.46\) and has achieved the error of \(Best = 3.06E-10\) for \(N_S = 4\). This is while for the \(N_S = 15\) has the same performance for all the approaches. This means that increasing the number of received samples does not help the system for to achieve better performance.

In order to study the convergence of algorithms toward the solution (exact location of sensor), algorithms performance for best value achieved so far versus \(NFC\) for 4 randomly selected sensors are presented in Figure 5.
For the sensor localization problem in WSNs, the simulation results demonstrate that the MDESM method performs much better than the other approaches. This is due to the dimensionality of problem which is three, where the MDESM can provide enough diversity for the search procedure. This is while the MDE diversity is limited for this problem and the MDEV and DE algorithms provide more than enough diversity.

VI. CONCLUSION AND FURTHER WORKS

In this paper a new protocol for sensor localization in WSNs is proposed. In this approach, an unmanned aerial vehicle (UAV) flies on the landscape where the sensors are deployed on, and it consequently broadcasts packets while flying. Each sensor on the landscape individually receives a number of packets and form its localization problem as an optimization model by replacing received data in the model. Then, the micro-differential evolution with scalar random mutation factor (MDESM) algorithm is utilized to solve the problem.

The proposed protocol can be further developed for indoor environments. In addition, since the MDESM algorithm works based on a small population size, it can be developed for parallel processing by assigning each individual of the population to a nearby mote.

REFERENCES