Journal of Information Hiding and Multimedia Signal Processing ©2017 ISSN 2073-4212
Ubiquitous International Volume **8**, Number **2**, March **2017**

An Improvement of Flower Pollination Algorithm for Node Localization Optimization in WSN

Jeng-Shyang Pan

Fuzhou University of International Studies and Trade, China
jengshyangpan@gmail.com

Thi-Kien Dao, Tien-Szu Pan

Department of Electronics Engineering,
National Kaohsiung University of Applied Sciences, Taiwan
jvnkien@gmail.com, tpan@cc.kuas.edu.tw

Trong-The Nguyen*

Department of Information Technology, Haiphong Private University, Vietnam
*Corresponding Author:vnthe@hpu.edu.vn

Shu-Chuan Chu, John F. Roddick

School of Computer Science, Engineering and Mathematics,
Flinders University of South Australia
jan.chu@flinder.edu.au, john.roddick@flinders.edu.au

Received December, 2016; revised February, 2017

Abstract. *This paper presents an improvement of the flower pollination algorithm (FPA) for optimization localization issues in wireless sensor networks (WSN). A novel probabilistic is used to generate a new candidate of competition for simulation optimization operations. The actual population of tentative solutions does not employ, but a unique representative probabilistic of them accumulate over generations. Evaluating this proposed method, we firstly used six selected benchmark functions to experiment and then we applied the proposal to solve the optimization problem of localization in WSN to confirm its performance further. The testing results compared with the original version of FPA show that the proposed method produces considerable improvements of reducing variable storing memory and running time consumption. Compared with the other approaches in the literature, the localization obtained from the proposed method is more accuracy and convergence rate indicate that the proposed method provides the effective way of using a limited memory.*
Keywords: Compact Flower Pollination Algorithm, Optimization localization problems, Probabilistic model, Wireless sensor network.

1. **Introduction.** Knowledge of the sensor node location is criteria key for some applications of the wireless sensor network (WSN), especially in applications include environmental monitoring, precision agriculture, vehicle tracking, and logistics [1]. In these applications, information about current locations is used for geographic-based routing, getting data aggregation, and various network services. Hence, self-organization and localization capabilities are one of the most important requirements in sensor networks. Theoretically, location awareness is possible obtained in principle by using a global positioning system (GPS). This solution, however, is not always viable in practice, because a sensor network consists of thousands of nodes and GPS will be very costly. In another

hand, GPS is not well suited to indoor and underground deployments, and the presence of obstacles like dense foliage or high buildings. Such barriers may cause GPS impair communication with satellites [2]. The estimate the location of sensor nodes by optimization localization error is one of the promise ways to deal with these mentioned issues.

Moreover, several applications require a solution to a complex optimization problem whether in limited hardware conditions [3], [4]. An available computational device is in use of the limited hardware state due to cost, memory and space limitations, e.g. WSN [5], [6]. Sensor nodes in WSN are small size, battery-powered, memory-constraint devices. The wireless communication capability of these devices is also over a restricted area [7]. Because the limited memory and the power constraints, WSNs fully functional network must be maintained and stable by the sound design system employment [6],[8]. The problems arise from the insufficient memory of these computational devices for storing the population of candidate solution in the optimization problems.

The compact algorithm is a promising answer to these problems. Compact algorithm simulates the behavior of population-based algorithms by employing the replacement a population of solutions with its probabilistic representation. An efficient way is to use one adaptive solution to present all solutions in the search space for the advantages of population-based algorithms without requirements of storing actual populations of solutions [9]. The representation of candidate solutions is considered based on learning and sampling probabilistic models. It means a built probabilistic model could generate new candidates or selected solutions by sampling. The replacement strategy is used to incorporate new solutions into the virtual population. In the compact algorithms, the number of parameters stored in the memory is smaller than their corresponding algorithms of the population-based structures and the requirement of the memory device is less for every run.

Additionally, Flower Pollination Algorithm (FPA) [10] is a new population-based intelligent optimization algorithm. FPA is a useful optimization algorithm because of balancing between exploring and exploiting like adjusting parameters for these pollination processes. Two different ways of pollination as the self-pollination and cross-pollination considered are local pollination and the global pollination process for simulating FPA.

Above mentioned motivation for this paper. In this article, we employ a probabilistic of a representative population within a few variable memories, rather store the population such the original version. This way is called compact flower pollination algorithm (cFPA). The localization in WSN and six benchmark functions are used to validate the performance of the proposed method. Related works, methodology details, and experimental results will present as following sections.

2. **Localization model.** WSN assumes with n nodes that deployed in two-dimensional space of Z^2, and m anchor nodes. There are $n - m$ unknown nodes in which $m < n$. Distance of each node to its nearby neighbors within its ranging distance is measured after a network deployed. All of the successful distance measurements along with the node specifications are transferred to a base station using multi-hop routing. We construct a graph after all of the measured distances are received at the base station. This graph for WSN can be modeled as G with (V, E) where V and E are the nite set of vertices and edges respectively. A set of sensor nodes is represented the vertices V with $\{v_1, v_2, ..v_n\}$. The connection of these vertices is represented network links as the set of edges E with $\{e_{1,2}, e_{1,3}, ..e_{i,j}, ..e_{n-1,n}\}$. If a connected component of G, $G_1 = (V_1, E_1)$ does have not three or more anchor nodes, then all the sensor nodes in the subgraph G_1 are not localizable. We can assume each connected component of graph G has at least three anchors.

The objective localization in a WSN is to estimate the coordinates of $n-m$ unknown nodes utilize the previous information about the sites of m anchor nodes. The objective function is established for the node localization included two-phase process. First is as ranging process which nodes estimate their distances from anchor nodes using the signal propagation time or the received signal strength indicator (RSSI), and second is position estimation of the nodes, i.e. using the ranging information [11]. The localization error is minimized by applying the optimization algorithms. In the first phase, each anchor nodes in the deployment estimates its distance from each of its neighboring target nodes. RSSI ranging technology can obtain the internode reaching distance.

The distance between the unknown node $o(x,y)$ is denoted $d_1, d_2, .., d_n$ and the anchor node is obtained by the hop count and the average hop distance between nodes. The ranging error is $\epsilon_1, \epsilon_2, .., \epsilon_n$ the estimated coordinates (x,y) satisfies the following inequalities:

$$\begin{cases} {d_1}^2 - {\varepsilon_1}^2 < (x-x_1)^2 + (y-y_2)^2 \leq {d_1}^2 + {\varepsilon_1}^2 \\ {d_2}^2 - {\varepsilon_2}^2 < (x-x_2)^2 + (y-y_2)^2 \leq {d_2}^2 + {\varepsilon_2}^2 \\ \qquad\qquad .... \\ \qquad\qquad .... \\ {d_n}^2 - {\varepsilon_n}^2 < (x-x_n)^2 + (y-y_n)^2 \leq {d_n}^2 + {\varepsilon_n}^2 \end{cases} \tag{1}$$

where d is the actual distance between two nodes, and ϵ is a ranging error. Localization problem is transformed into finding coordinates (x,y) which minimize objective function $f(x,y)$ of formula (2). This optimization $f(x,y)$ guarantees minimum total error.

$$f_{(x,y)} = \sum_{j=1}^{m} \sum_{i=m+1}^{n} \left| \sqrt{((x_i - x_j)^2 + (y_i - y_j)^2)} - d_j \right| \tag{2}$$

where (x_i, y_i) and (x_j, y_j) are coordination of node i and j location. d_j is the distance between unknown node to the anchor node j.

3. **Improvement of Flower Pollination Algorithm Optimization.**

3.1. **Meta-heuristic Flower Pollination Algorithm.**

A recently population-based algorithm drawn inspiration from two pollination processes of the flowering plant including self-pollination and cross-pollination is known as Flower Pollination Algorithm (FPA) [10]. In the flowering plant, pollens are transported by pollinators according to the rules of Levy flights, and they can self-pollinate randomly. Moreover, two universal concepts of guides the optimal process in the population-based algorithm are for exploring and exploiting the search space. A self-pollination of the flowering plant viewed as local pollination that expressed for exploitation in the search area. However, a cross-pollination considered as global pollination that represented for exploration a promising area search.

How to switch between the exploring and exploiting phrases in pollination for FPA, a switching probability $p \in [0,1]$ can be used to control their characteristics of local and global pollination. Let's x_j^t and x_k^t be solution vector of the pollen, i.e. pollen in the same plant or the flowers. We could model for the local pollination as following.

$$x_i^{t+1} = x_i^t + u \times \left(x_j^t - x_k^t\right) \tag{3}$$

where u is drawn from a uniform distribution in [0, 1]. The update solution vector can convert into updating equations for the cross-pollination. In the global pollination, flower pollen gametes are carried by pollinators such as insects. Insects can often fly and move in a much longer range. A Levy flight can express flying insects over a long distance with

various length steps, and be used to mimic this characteristic efficiently. Let's L be a Levy distribution with active as drawn formula.

$$L = \frac{\lambda\Gamma(\lambda) \times \sin\left(\frac{\pi\lambda}{2}\right)}{\pi \times s^{i+\lambda}} \tag{4}$$

where $\Gamma(\lambda)$ is the gamma function, and this distribution is valid for large steps $s > 0$. $\Gamma(\lambda)$ is called the step size like the parameter that corresponds to the strength of the pollination. Updating locations of pollen are simulated as given.

$$x_i^{t+1} = x_i^t + \gamma \times L(\lambda) \times \left(x_i^t - g^*\right) \tag{5}$$

where g^* is the current best solution found so far, γ is a scaling factor to control the step size and t is the current generation or iteration.

In the evaluation for updating pollination, according to the fitness function, if x_j^t and x_k^t come from the same species or selected from the same population, u becomes a local random walk. Because of the flower pollination processes can occur at both local and global, to imitate this feature, the proximity probability p is to switch between natural global pollination to intensive local pollination. The switching probability can be used likely $p \in [0, 1]$ to control between the local and worldwide pollination.

3.2. **Compact Flower Pollination Algorithm.** The estimated distribution algorithm (EDA)[12] can process a representation of probabilistic to get fewer variable memory, rather store the population solution. The compact methods use the principle of EDA to simulate the operations of the population-based methods. A probabilistic model is used to represent these operations in the population-based algorithm. The real population of the algorithms considered as a virtual population in compact algorithms. The virtual population can configure by considering probability density functions (PDFs) [13] based on EDA. Not all of population of a solution stored in memory, but it generates a few new candidate of solution based on probability distribution store in memory. A generated new candidate solution is by being iteratively biased toward a promising area of an optimal solution. The probabilistic model of a population of solutions represents the probability vector of each component learned from previous generations. The structure of this vector was called Perturbation Vector (PV) [9]. These principles apply to the improvement of FPA [10] such compact FPA is presented as follows.

For compact the actual population-based of FPA, a virtual population will express by encoding within a data structure of probabilistic vector. A real-valued prototype vector represents the probability of each component represented in a candidate solution. The specified probability for each element in new candidate solutions maintained in the optimum process. The information-processing objective of the compact algorithm is to simulate the behavior of pollination of FPA, but it used with a much smaller memory. PV generates a candidate solution probabilistically from the vector and the competing components toward to the better solutions that change the probability vector. The created trial solutions stayed to allocate in boundary constraints. PV is a matrix for specifying the two parameters of mean and standard deviation values in the PDF of each design variable. It can be defined as.

$$PV^t = \left[\mu^t, \sigma^t\right] \tag{6}$$

where t is time steps. A truncated Gaussian (PDF) for μ and σ values are within the interval of (-1, +1). The PDF normalizes the amplitude of area equal to 1. A generated candidate solution x_i is produced from $PV(\mu_i, \sigma_i)$. The associated Gaussian to mean μ and standard deviation σ in PV is given as.

Input: The vectors μ, σ, and parameter ω
Output: A new candidate x
1: **for** i= *1* to d **do**
2: Generate a random number $r \in [0,1]$
3: ***if*** $r < \omega$ ***then***
Generate $x_i \in [0,1]$ according to $L_i(\mathrm{x})$ in Eq. (8)
5: ***else***
Generate $x_i \in [0,1]$ according to $R_i(\mathrm{x})$ in Eq. (9)
7: ***end if***
8: ***end for***

FIGURE 1. A sampling scheme for generating new solution in the compact algorithm

$$P_i(\mathrm{x}) = \frac{\sqrt{\frac{2}{\pi}} \times \exp\left(-\frac{(x-\mu_i)^2}{2\sigma_i^2}\right)}{\sigma_i\left(\mathrm{erf}\left(\frac{\mu_i+1}{\sqrt{2}\sigma_i}\right) \text{-erf}\left(\frac{\mu_i-1}{\sqrt{2}\sigma_i}\right)\right)} \tag{7}$$

where $P_i(x)$ is the corresponding value of the PDF to variable x_i. The error function indicated as erf is fined in [14]. PDF could have corresponded to Cumulative Distribution Function (CDF) by constructing Chebyshev polynomials [15]. The CDF describes the probability that a real-valued random variable $X = \{x_1, x_2, ...x\_k\}$, k is constant. The newly calculated value of x_i is inversed to from CDF. A parameter is suggested as a weight to control the probability of sampling of μ_i in PDF (7) between left $[-1, \mu_i]$ and right $[\mu_i, 1]$ as given in Eqs (8) and (9).

$$L_i(\mathrm{x}) = \frac{-\sqrt{\frac{2}{\pi}}}{\sigma_i(\mathrm{erf}\left(\frac{\mu_i+1}{\sqrt{2}\sigma_i}\right)} \times \exp\left(-\frac{(x-\mu_i)^2}{2\sigma_i^2}\right) \quad for -1 \le x \le \mu_i \tag{8}$$

$$R_i(\mathrm{x}) = \frac{-\sqrt{\frac{2}{\pi}}}{\sigma_i(\mathrm{erf}\left(\frac{\mu_i-1}{\sqrt{2}\sigma_i}\right)} \times \exp\left(-\frac{(x-\mu_i)^2}{2\sigma_i^2}\right) \quad for \mu_i \le x \le 1 \tag{9}$$

The generating new candidates of the flowers are employed by sampling from PV. Fig. 1 shows a sampling scheme for generating new solution toward to promising area in the search space.

Two designed variables compete to find out who is win or lose. The comparison between two design variables for individuals of the pollen is performed the win solution biases the PV. Fig 2 shows the competing scheme for winner and loser. PV samples these pollen. The win or lose vector is judged according to the fitness function, i.e., its value is better or worse. The new solution is then evaluated and compared against x_k^t and g^* in (5) and (3) to determine whether the winning and losing individual is.

The elements μ_i^{t+1} and σ^{t+1} of the updated PV for the new solution for winner and loser are expressed over the differential iterative as follows.

$$\mu_i^{t+1} = \mu_i^t + \frac{1}{N_p}(winner_i - loser_i) \tag{10}$$

Input: The objective function f and solutions x_a, x_b
Output: *winner or loser*

1: if $f(x_a) < f(x_b)$ then
2: Let $winner = x_a$
3: Let $loser = x_b$
4: **else**
5: Let $winner = x_b$
6: Let $loser = x_a$
7: **end if**

FIGURE 2. The competing scheme for judging winner or loser

1: **for** i= *1* to d **do**
2: $\mu_{backup} = \mu_i^t$
3: $\mu_i^{t+1} = \mu_i^t + \frac{1}{N_p}(winner_i - loser_i)$
4: $\sigma_i^{t+1} = \sqrt{\max(0, (\sigma_i^t)^2 + (\mu_{backup}^t)^2 - (\mu_i^{t+1})^2 + \frac{1}{N_p}(winner_i^2 - loser_i^2)}$
5: Improving for μ_i^{t+1} and σ^{t+1} according to Eqs. (15), (16) with τ set to 0.01
6: **end for**

FIGURE 3. The updated PV scheme of PDF for new candidates

where N_p is virtual population size. This parameter of N_p is only typical in compact algorithms because of it does not strictly correspond to the population size as in a population-based algorithm, for further reported issue in [16]. Regarding σ values, the update rule of each element is given by:

$$\sigma_i^{t+1} = \sqrt{(\sigma_i^t)^2 + (\mu_i^t)^2 - (\mu_i^{t+1})^2 + \frac{1}{N_p}(winner_i^2 - loser_i^2)} \tag{11}$$

For further perturbing for mean and standard deviation values by adding a weight parameter is to enhance diversity distribution. The values of μ_i^{t+1} and σ^{t+1} of PV are modified as given in Eqs. (12), (13), for $i = 1, 2, .., d$.

$$\mu_i = \mu_i + \beta_i \tau \tag{12}$$

$$\sigma_i = \sqrt{\sigma_i^2 + \alpha_i \tau} \tag{13}$$

where τ is a weight parameter representing the maximum value of the perturbation, β_i is a uniform random number in [-1, 1] and α_i is a random number in [0, 1]. The tness value of the position x^{t+1} is calculated and compared with g^* to determine a winner and a loser. The probability vector PV is updated by applying Eqs (12) to (13). Current values for x^{t+1} and v^{t+1} are retained for subsequent algorithm steps. The update PV scheme is carried out in Fig.3.

The behavior of FPA simulated in compact algorithm by sampling probabilistic models for a population of solutions. The probabilistic expression simulated the operations in

Step 1. Initialization: pollen x is generated from PV.
Step 2. Calculate for the best solution g^* with initial pollen, assign F_{min} to the fitness at g^*.
Step 3. For each pollen
if rand $< p$, // A switch variable $p \in [0\ 1]$
Global agent is updated via Eq.(5) // compute L according to Lévy Eq.(4)
else
Local agent is processed via Eq.(3) // Draw u from [0,1]
$x_k^t = PV$
end if
Step 4. Evaluate new solutions
[winner, loser] = compete(x^{t+1}, g^*); *Fnew=f*(x_k^t);
// μ^{t+1}, σ^{t+1}according to Update PV scheme 3, Fig.3, the global best update
[winner, loser] = complete(g^*, x^{t+1});
if Fnew<Fmin
$g^* = winner$; *Fmin=Fnew;*
end if
t= t+1;
Step 5. Check termination condition, if not safety, go back Step 3.

FIGURE 4. A pseudo code for improving FPA scheme

TABLE 1. Six selected test functions as a benchmark

Series	Test Functions	Dim	Boundaries	Iteration
1	$F_1(x) = \sum_{i=1}^{n} x_i^2$	30	[-100,100]	2000
2	$F_2(x) = \sum_{i=1}^{n} \lvert x_i \rvert + \prod_{i=1}^{n} \lvert x_i \rvert$	30	[-10,10]	2000
3	$F_3(x) = \sum_{i=1}^{n} (\sum_{j=1}^{i} x_j)^2$	30	[-100,100]	2000
4	$F_4(x) = \max_i \{\lvert x_i \rvert, 1 \le i \le n\}$	30	[-100,100]	2000
5	$F_5(x) = \sum_{i=1}^{n-1} [100(x_{i+1} - x_i^2)^2 + (x_i - 1)^2$	30	[-100,100]	2000
6	$F_6 = \sum_{i=1}^{n} ([x_i + 0.5])^2$	30	[-100,100]	2000

optimal algorithms. The working principle of compact method provides a pseudo code for improving FPA as shown in Fig. 4.

3.3. **Experiment with numerical problems.** Six optimal numerical problems as benchmark tests [17] are utilized for testing the accuracy and the speed of the compact flower pollination algorithm (cFPA). The solution quality of the cFPA method compared with the original flower pollination algorithm (FPA). The experiments employed for all the test functions averaged values over 25 runs. Each function contains the full iterations of 2000 and is repeated by different random seeds over twenty-five runs. The optimization goal is to minimize the output for all benchmarks. The initial range, the dimension, and total iteration number for all test functions listed in Table 1.

The parameters setting for FPA and cFPA are as follows. Population size n for FPA is set to 30, the dimension of the solution space M 30. The final result obtained by taking the average of the outcomes from all runs. The results compared with the FPA.

TABLE 2. The comparison between FPA and cFPA regarding quality of the performance evaluation and speed

Test Functions	Performance evaluation		Rate %	Consumed Time (minutes)		Rate %
	FPA	cFPA	RD	FPA	cFPA	Comparison
$F_1(x)$	1.84E+03	1.81E+03	2%	1.5659	1.0141	54%
$F_2(x)$	4.82E+00	4.56E+00	6%	2.1764	1.2936	68%
$F_3(x)$	8.67E+03	8.38E+03	3%	1.498	1.2598	19%
$F_4(x)$	2.16E+01	2.09E+01	3%	1.1847	0.6921	71%
$F_5(x)$	1.19E+06	1.09E+06	9%	1.3823	0.8845	56%
$F_6(x)$	4.07E+03	3.82E+03	7%	1.1616	1.0768	8%
Average	**2.00E+05**	**1.84E+05**	**5%**	**1.494817**	**1.036817**	46%

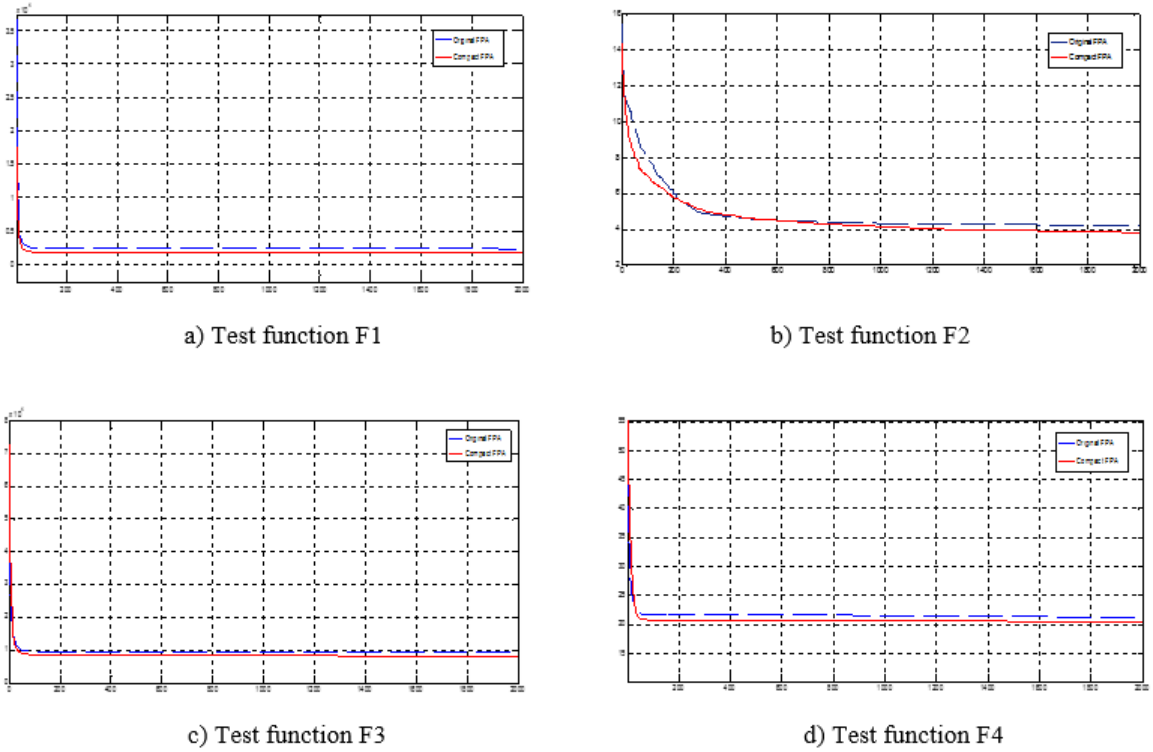
Table 2 compares the quality of performance and running time for numerical problem optimization between cFPA and FPA. In Table 2, columns of the FPA and the cFPA are the averages of the outcomes of 25 runs for six test functions respectively. RD is a rated deviation. It means the percentage of the deviation on the previous version of algorithm outcomes for the cFPA respectively. Clearly, the most cases of benchmark functions for optimizing in both cFPA and oBC have the small percentage of the deviation. It says that the accuracy of the cFPA is as good as that of the FPA. The average proportion of the deviated rate of eight test functions evaluation is only 5%. But the average time consuming for cFPA is 46% faster than that for FPA.

Fig. 5 shows the averaged function minimum values for four test functions in 25 runs output with the same iteration of 2000. Clearly, all cases of testing benchmark functions for the compact FPA (red lines) are equal to or faster than original FPA in convergence.

Fig. 5 compares the performance quality of running with different population sizes for test functions between cFPA and FPA. The most cases of testing functions for cFPA is smaller than that for FPA in convergence. Most cases of test functions for cFPA are not affected as variety population size as in FPA. The mean of obtained value functions for cFPA is more stable than the FPA due to the virtual population in cFPA.

Obverse Table 3, the computations of saving memory in two algorithms cFPA and FPA use. Clearly, the number of memory variables of cFPA use is smaller than those of FPA use in the same condition of computation such as iterations. The real number of population or population size for FPA is N, but that size for cFPA is only one. The used equations for optimizing computation in cFPA is six such as Eqs. (03), (04), (05), (07), (12) and (13). However, the used equations for optimizing computation in FPA is three of them such as Equations (03), (04), and (05). But the computing complexity of cFPA is the only $6 \times T \times iteration$ whereas those of FPA is $3 \times T \times N \times iteration$ which is higher $N/2$ times. Thus, the rate of saving memory of the computing complexity of cFPA per the computing complexity of FPA equals to $2/N$.

Figure 4 illustrates the executing time comparison of cFPA and FPA in 25 runs with iteration 2000 for eight benchmark functions. It is clear that the most cases of test functions for executing time in the proposed cFPA (red colored bar) are smaller than those executing in FPA (blue colored bar). The average consuming time of eight tests for cFPA is 85% faster than those for FPA. The reason for a fast processing speed is that some the memory stored parameters for cFPA are smaller than those for FPA.



a) Test function F1 b) Test function F2

c) Test function F3 d) Test function F4

FIGURE 5. Convergence comparison of cFPA and FPA algorithms for four first benchmark functions

TABLE 3. The comparison of memory variables between cFPA and FPA

Algorithms	Population size	Occupied memory size	Equations Numbers	Computing parameters
oFPA	*N*	*3×N*	*(3),(4),(5)*	*3×T×N ×iteration*
cFPA	*1*	*6*	*(3),(4),(5),(7), (12),(13)*	*6×T ×iteration*

The results regarding the comparison for performance of the proposed cFPA outcomes with the other Compact algorithms in the literature for 6 test functions is available in Table 4, where it shown that cFPA outperforms its competitors. The best results in among them for each function have highlighted in rows. The performance of compared ratio r is set for each pair of comparisons of FPA with other methods of the cPSO, cDE, and rcGA respectively. The noticed indication $'+, -$ and $\sim$' mean the 'better', 'worse', and 'approximation' of the deviation on their outcomes respectively. Visibly, almost the highlighted cases of testing benchmark functions belong to the proposed method of the cFPA. It means that the proposed method is an alternative method. According to Figures of 4 compact methods, the cFPA method shows a comparatively better convergence behavior on the selected tests than the other algorithms.

4. **Localization optimization in WSN based on cFPA.** In this section, we can formulate the position estimation of the given unknown nodes as an optimization problem that involving in minimization of a represented objective function about the localization precision. We investigate the estimation of hidden nodes for the optimal localization in a

TABLE 4. Comparison the outcomes of the proposed method of cFPA with the state of the art of compact algorithms such as rcGA, cDE, cPSO for 6 test functions

Functions	**cFPA**	**cPSO**	**r**	**cDE**	**r**	**rcGA**	**r**
$F_1(x)$	1.81E+03	1.84E+03	~	1.91E+03	+	1.81E+03	~
$F_2(x)$	**4.56E+00**	4.82E+00	+	4.76E+00	+	4.56E+00	+
$F_3(x)$	8.38E+03	8.67E+03	-	8.48E+03	+	8.38E+03	+
$F_4(x)$	2.09E+01	2.06E+01	-	2.09E+01	~	2.09E+01	+
$F_5(x)$	1.09E+06	1.19E+06	+	1.09E+06	+	1.09E+06	+
$F_6(x)$	**3.82E+03**	4.07E+03	+	3.82E+03	-	3.82E+03	+
AVG	**1.84E+05**	1.90E+05	+	1.94E+05	+	2.04E+05	+
Summary		4 + 1~ 2-		5+ 1~ 1-		6+ 1~ 0-	

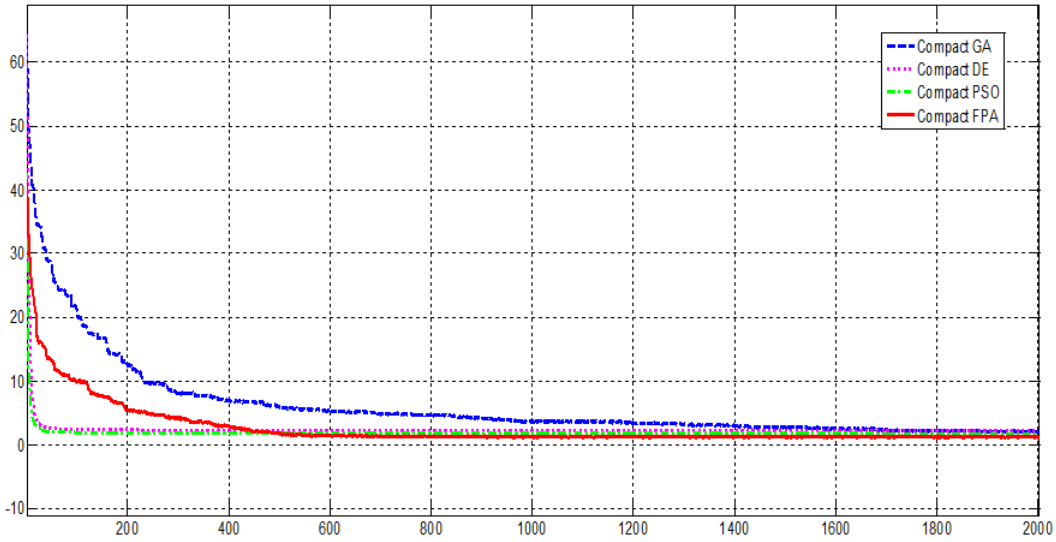


FIGURE 6. Comparison of six compact algorithms of rcGA, cDE, cPSO and cFPA, for the testing function F1

sparse network based on the proposed cFPA. The simulations have been run cFPA for the objective function as in Eq.(2). Each localizable unknown node can be estimated cFPA algorithm independently to localize itself by finding its coordinates (x, y). The different situations in the localization issue have been implemented varying scenarios such as nodes densities, varying anchor nodes, and diversity of the solutions for optimization localizations. We conducted sensor localization for the whole sensor network in the following manner. The network consists of total n nodes are randomly deployed in a specified area with m anchor nodes being randomly generated from these nodes. Each node can communicate in range of RSSI as assuming that ranging error e_{ij} follows a Gaussian distribution. An unknown node can be estimated its location if it has at least 3 non-coplanar anchor nodes in neighbor. That node is said to be localizable node. Each localizable node measures its distance from each of its neighboring anchors. $n - m$ is actual distance between

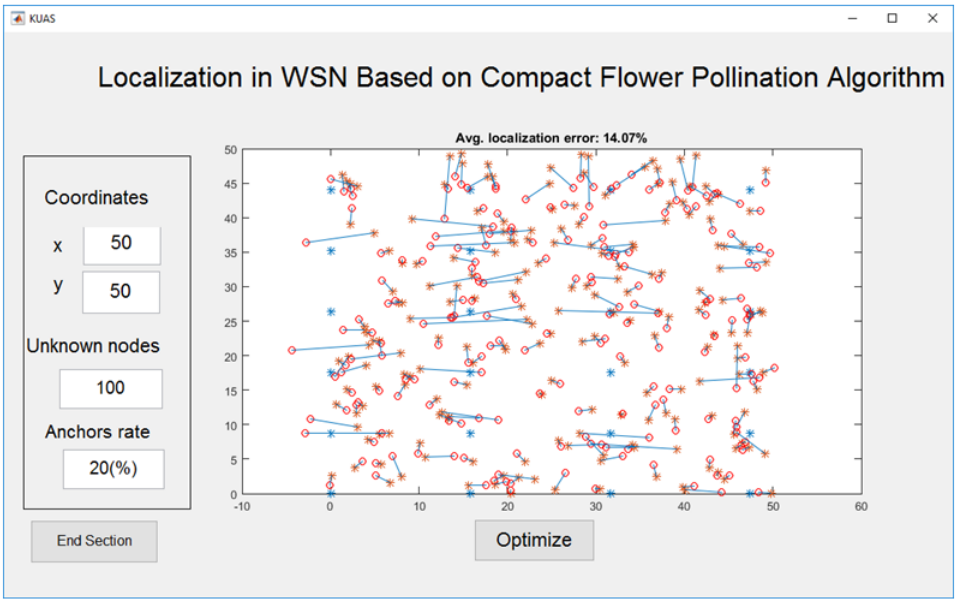


FIGURE 7. The optimal location of unknown nodes in WSN based on compact flower pollination algorithm

the localized nodes and anchor nodes. The objective function for localization problem is defined as in Eq.(2). The cFPA method evolves the optimal location of unknown nodes, i.e. (x, y) by minimizing the error function.

The setting simulation is in specified sensor network area (e.g.a 50m *times* 50m) with n unknown nodes e.g.100 unknown nodes and m like the percentage of anchor nodes e.g. 20 as shown in Fig.7. The coordinates of each node (x, x) are randomly dispersed in this area. Each node has a transmission range of 25m. In the simulation, the size of the pollen population is fixed at 30, and the number of iterations is 1000. Each result is the average of 30 replications. Simulation results show the effectiveness of the proposed objective function in tackling the fine-grained localization problem in WSNs. The results are compared with those obtained from the firefly algorithm (FA), parallel firefly algorithm (pFPA) [18] and the original method of flower pollination algorithm (FPA) [19]. The average localization error is defined as the distance between the real and estimated locations of an unknown node which is computed as the mean of square of distance of computed node coordinates.

$$Age = \frac{\sum_{i=m+j}^{n} \sqrt{((x_i - x_j)^2 + (y_i - y_j)^2)}}{N_L} \tag{14}$$

4.1. **Effect of Anchor node density.** The obtained results by the cFPA for real position of unknown nodes and the coordinates of the estimated nodes in a trial run presented in Fig.8. We experimented with the anchor node density and unknown node density to verify the factors influencing to the localization error and number of localized nodes.

Table 5 indicates the localization results taken at the given unknown node number of 100 and transmission range of 25m with the percentage of anchor nodes is varying from 10% to 30%. Observed, if having more number of anchors is in network space, the most favorable references for unknown nodes are. The number of nodes that get localized relies on the number of anchors. Less number of localized nodes are due to insufficient anchor nodes in the surrounding. The number of localized nodes has a significant increase, and

TABLE 5. The effect of the proportion for localization errors with different anchor nodes rate

Methods	Anchor node proportion					
	5%	10%	15%	20%	25%	30%
FA[21]	29%	25%	23%	21%	20%	20%
FPA[22]	29%	22%	20%	20%	19%	18%
pFPA[21]	27%	21%	19%	19%	18%	17%
cFPA	29%	22%	19%	19%	19%	17%

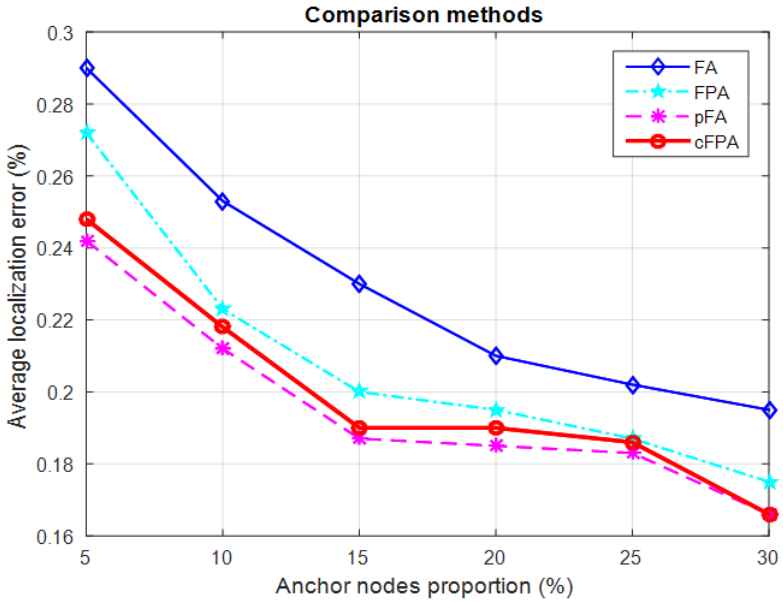


FIGURE 8. Comparison of four methods in the localization errors with the different anchor node proportion

the average localization error decreases with the increase of anchor node density as shown in Fig. 8.

4.2. **Effect of Anchor node density.** We also analyzed the effect of unknown node density for the performance of localization. The analyzed results presented in Table 6. The varying density of unknown node has resulted on the success rate of localized node. Clearly, if the higher density of nodes is, the estimated error is the less, but the cost is high.

In generally, average localization error decreases when unknown node density increases shown in Fig. 9. The comparison curves of the approaches illustrated in Fig. 9 shows that the proposed method archived more accuracy and convergence rate than others.

5. **Conclusion.** In this paper, we proposed an improvement of flower pollination algorithm (FPA) based on the compact technique for optimization problems of the class of devices limited memory. In the proposed method, we replaced the actual design variable of solutions of FPA with a probabilistic representation of the population. A probabilistic model was used to generate new candidate solutions with searching promising solutions so far. The probability density functions (PDF) and cumulative distribution functions

TABLE 6. The effect of the density for localization errors with different node number

Methods	Total number of nodes					
	50	100	120	140	160	180
FA[21]	34%	26%	21%	20%	19%	19%
FPA[22]	27%	25%	20%	20%	19%	18%
pFPA[21]	27%	25%	19%	19%	19%	17%
cFPA	25%	24%	19%	19%	18%	17%

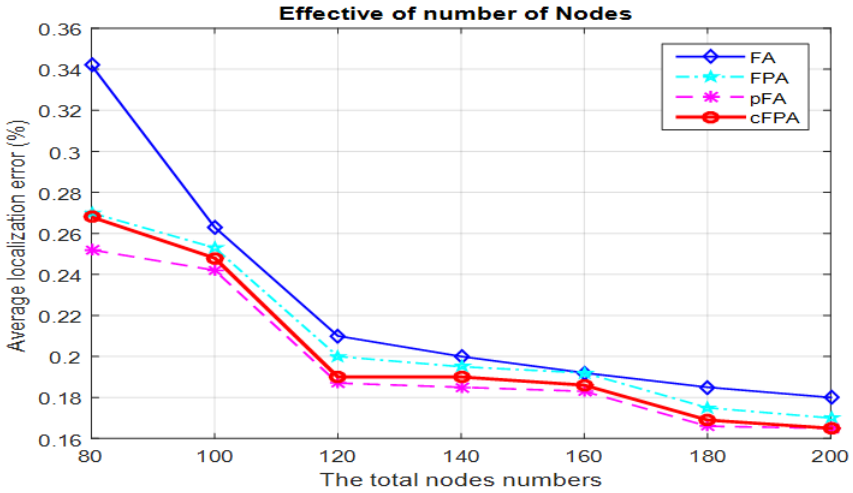


FIGURE 9. Comparison of the four methods for the effective density of unknown nodes to localization errors

(CDF) are explored thoroughly to construct the operations of selecting and optimizing behaviors. This probabilistic model can open a valid alternative optimization method for even on the available devices whose limited sources hardware.

Six selected numerical optimization problems, and a localization problem in wireless sensor network (WSN) were used to evaluate the proposed method regarding the accuracy, computational time and the saving memory. The simulation results compared with those obtained from the other methods such as the original version of FPA, GA, DE, and PSO show that the proposed method produces considerable improvements of reducing variable storing memory and running time consumption. We also compared the node localization optimization in WSN obtained from the proposed method with the other approaches in the literature. The proposed method archived more accuracy and convergence rate than others, and it can provide an effective way of using a limited memory.

REFERENCES

[1] E. N. Szynkiewicz, Localization in wireless sensor networks: Classification and evaluation of techniques, *Int. J. Appl. Math. Comput.* Sci., vol. 22, no. 2, pp. 281-297, 2012.

[2] Z. Farid, R. Nordin, and M. Ismail, Recent advances in wireless indoor localization techniques and system, *Journal of Computer Networks and Communications*, 2013.

[3] C. F. Garca-hernndez, P. H. Ibargengoytia-gonzlez, J.Garca-hernndez, and J. aPrez-daz, Wireless Sensor Networks and Applications: a Survey, *J. Comput. Sci.*, vol. 7, no. 3, pp. 264-273, 2007.

[4] T. K. Dao, S. C. Chu, T. T. Nguyen, C. S. Shieh, and M. F. Horng, Compact Artificial Bee Colony, *Modern Advances in Applied Intelligence*, Springer, vol. 8481, pp. 96-105, 2014.
[5] T. K. Dao, T. S. Pan, T. T. Nguyen, and S. C. Chu, A Compact Articial Bee Colony Optimization for Topology Control Scheme in Wireless Sensor Networks, *Journal of Information Hiding and Multimedia Signal Processing* , vol. 6, no. 3, pp. 297-310, 2015
[6] T. S. Pan, T. T. Nguyen, T. K. Dao, and S. C. Chu, An Optimal Clustering Formation for Wireless Sensor Network Based on Compact Genetic Algorithm, *Third International Conference on Robot, Vision and Signal Processing (RVSP)*, pp. 294-299, 2015.
[7] T. T. Nguyen, C. S. Shieh, M. F. Horng, and T. K. Dao, A Genetic Algorithm with Self-Configuration Chromosome for the Optimization of Wireless Sensor Networks, *Proc. of the 12th International Conference on Advances in Mobile Computing and Multimedia. ACM*, Kaohsiung, Taiwan, pp. 413-418, 2014.
[8] T. T. Nguyen, M. F. Horng, and C. S. Shieh, T. K. Dao, An Energy-based Cluster Head Selection Algorithm to Support Long-lifetime in Wireless Sensor Networks, *Journal of Network Intelligent*, vol. 1, no. 1, pp. 23–37, 2016.
[9] G. R. Harik, F. G. Lobo, and D. E. Goldberg, The compact genetic algorithm, *IEEE Trans. Evol. Comput.*, vol. 3, no. 4, pp. 287-297, 1999.
[10] X. S. Yang, Flower Pollination Algorithm for Global Optimization, Unconventional Computation and Natural Computation, Springer, pp. 240-249, 2013.
[11] G. Mao, B. Fidan, and B. D. O. Anderson, Wireless sensor network localization techniques, *Comput. Networks*, vol. 51, no. 10, pp. 2529-2553, 2007.
[12] P. Larranaga and J. A. Lozano, Estimation of Distribution Algorithms - A New Tool for Evolutionary Computation, *Springer Science & Business Media*, vol. 2, 2002.
[13] P. Billingsley, Probability and Measure - *Third Edition*, 1995.
[14] I. N. Bronshtein, K. A. Semendyayev, G. Musiol, and H. Mhlig, Handbook of mathematics, *sixth edition*, 2015.
[15] S. K. Ghosh, Handbook of Mathematics, vol. 18, no. 3, 1989.
[16] E. Mininno, F. Neri, F. Cupertino, and D. Naso, Compact differential evolution, *IEEE Trans. Evol. Comput.*, vol. 15, no. 1, pp. 32-54, 2011.
[17] X. Yao, Y. Liu, and G. Lin, Evolutionary programming made faster, *IEEE Trans*, *Evol. Comput.*, vol. 3, no. 2, pp. 82-102, 1999.
[18] V. O. Sai, C. S. Shieh, T. T. Nguyen, Y. C. Lin, M. F. Horng, and Q. D. Le, Parallel Firefly Algorithm for Localization Algorithm in Wireless Sensor Network, *Third Int. Conf. Robot. Vis. Signal Process*, pp. 300-305, 2015.
[19] S. Goyal and M. S. Patterh, Flower pollination algorithm based localization of wireless sensor network, *2nd International Conference on Recent Advances in Engineering & Computational Sciences (RAECS)*, pp. 1-5, 2015.
[20] T. T. Nguyen, J. S. Pan, S. C. Chu, J. F. Roddick, T. K. Dao, Optimization Localization in Wireless Sensor Network Based on Multi-Objective Firefly Algorithm, *Journal of Network Intelligence*, vol. 1, no. 4, pp. 130-138, 2016.