IEEE Access
Multidisciplinary : Rapid Review : Open Access Journal

Received May 7, 2019, accepted May 28, 2019, date of publication June 7, 2019, date of current version June 25, 2019.

Digital Object Identifier 10.1109/ACCESS.2019.2921721

An Improved Flower Pollination Algorithm for Optimizing Layouts of Nodes in Wireless Sensor Network

TRONG-THE NGUYEN[1,2], JENG-SHYANG PAN[1,3,4], AND THI-KIEN DAO[1,2]
[1]Fujian Provincial Key Laboratory of Big Data Mining and Applications, Fujian University of Technology, Fuzhou 350014, China
[2]Department of Information Technology, University of Manage and Technology, Haiphong 180000, Vietnam
[3]College of Computer Science and Engineering, Shandong University of Science and Technology, Qingdao 266510, China
[4]College of Informatics, Chaoyang University of Science and Technology, Taichung 413, Taiwan

Corresponding authors: Jeng-Shyang Pan (jengshyangpan@gmail.com) and Thi-Kien Dao (jvnkien@gmail.com)

This work was supported by the Natural Science Foundation of Fujian Province, under Grant 2018J01638.

ABSTRACT The arrangement of nodes impacts the quality of connectivity and energy consumption in wireless sensor network (WSN) for prolonging the lifetime. This paper presents an improved flower pollination algorithm based on a hybrid of the parallel and compact techniques for global optimizations and a layout of nodes in WSN. The parallel enhances diversity pollinations for exploring in space search and sharing computation load. The compact can save storing variables for computation in the optimization process. In the experimental section, the selected test functions and the network topology issue WSN are used to test the performance of the proposed approach. Compared results with the other methods in the literature show that the proposed algorithm achieves the practical way of reducing the number of its stored memory variables and running times.

INDEX TERMS Improved flower pollination algorithm, layout optimization problems, probabilistic model, wireless sensor network.

I. INTRODUCTION

The metaheuristics have been widely applied to solve optimization problems successfully in many fields of life, e.g., engineering, biology, and finance [1], [2]. The optimization process of the metaheuristic algorithms usually begins with generating a set of randomly initialized agents. The agents are combined, immigrated, or evolved over a predefined number of generations [3], [4]. Flower pollination algorithm (FPA) is a new metaheuristic algorithm [5], which models the rules of pollination of the flower. It utilizes a population of pollens to represent candidate solutions in a search space and optimizes the problem by iteration to move these pollens to the best answers about a given measure of quality. Compared with the other algorithms, the advantages of FPA are as follows: It is easy to implement on hardware using programming language, and the program consumes, and it can generate high quality of the solution [6], [7]. FPA has passed through several changing versions from its original algorithm, e.g. discrete [8], binary [9], multiobjective [10], modified [11], etc., due to customizing with conduction of complex problem [7]. FPA also has applied to solve some problems successfully, e.g., the traveling salesman problem [12], the scheduling problem [13], the graph coloring problem [14], the node localization problem [15]. However, PFA has not any improved versions to consider dealing with a class of the memory saving variables in the optimization problems. This paper attempts improving FPA by hybridizing parallel with communication strategy, and compact for optimal network topology in wireless sensor networks (WSN).

As a metaheuristic algorithm could obtain excellent results on a kind of challenges, but the same metaheuristic could give out the inferior score on a different set of instances that proven in no free lunch theorem [16]. The motivation for this paper is a considered attempt of the strong points of the algorithm for the types and properties of the problem and no free lunch theorem. The processing parallel is very significant for computations in optimization techniques. Pollens in the parallel structure with communication strategy can exchange information with another group, share the computation load, and enhance the diversity of pollens. The accuracy and global search capacity in the structure parallel increased more than that in the original one [11].

In the other hand, the compact technique offered the practical way of using a saving variable memory. For hav-

The associate editor coordinating the review of this manuscript and approving it for publication was Emanuele Lattanzi.

ing alternative pollination in search space, a valid agreement is used instead for the advantages of population-based algorithms without requirements of storing actual populations of pollinations. Compact algorithm simulates the behavior of population-based algorithms by replacing a community of solutions with its probabilistic representation. The representation of candidate pollinations is considered based on learning and probabilistic sampling models. Selected answers figure out based on a probabilistic model by sampling agent that generates new solutions in process optimization. The replacement strategy is used to incorporate fresh solutions into the virtual population. For the compact construction, the number of stored memory parameters is fewer than its population-based one and the requirement devices of the amount of memory less.

Furthermore, WSN, an emerging and promising technology consists of hundreds of distributed sensor nodes to sense relevant data in the environment. The detected value of the situation is processed and forwarded to Base Station (BS) through single hop or via intermediate nodes [17]. Applications of WSN has become widely accessible in real-time monitoring and tracking applications such as military surveillance, agriculture, disaster management, healthcare monitoring, industry automation, inventory control, traffic control, home automation [18]. WSN deployed in regions where human is difficult or not possible to intervene. As the sensors of WSN used in the harsh environment, it is tight or not likely to replace or recharge batteries [19]. However, the sensor nodes are limited in computation capability and storage capacity of the computing unit, in communication range and radio quality of communication unit, and the available energy of power units [20]. Because of these constraints, WSNs fully functional network must be maintained and stable by the sound design network employment. As mentioned saving memory variable, the compact method is a likely answer to optimization problems in WSN.

This paper, we extend our previous conference papers [11], [21] by hybridizing the parallel with the compact schemes for the global optimization problems and layout nodes problem in WSN. The logic behind extending works includes hybrid parallel and compact, an added weight to control the probability of sampling for flexible application of perturbation vector. The sick sampling individuals in the subgroups replaced with the better sampling competition agents according to the fitness evaluation. The parallel process considers diversity pollination for exploring aspect, and the compact considers decrease memory variables by the sampling of the probability model. The contributions to extending the previous works express as follows.

- An innovation proposal of hybridizing the parallel with the compact schemes for the global optimization problems and layout nodes problem in WSN (Section 3 in the document).
- A parameter τ is used as a weight to control the probability of sampling of generating a new solution that is sampling points toward the left or right side (refers to (8) in Section 3 in the document). The previous work is the version of FPA for numerical optimization, but its application is not flexible.
- A new form of the proposed pcFPA for the layout nodes problem in WSN (Section 5 in the document).
- Extensive testing is not only for the monomial issues but also for multimodal global optimization problems through series selected benchmark functions (Section 4 in the document).
- Comprehensive comparison with the other version of FPA, the compact algorithms in the literature, and the topology approach in WSN (Section 4 and 5 in the document).

The rest of the paper is organized as follows. Section 2 provides a brief review of FPA and a statement problem of the quality of the node's layout in WSNs. Section 3 presents the analysis and design for the parallel compact FPA. The simulation test and results demonstrate in Section 4. A solution for Network topology WSNs discusses in Section 5. Section 6 summarizes conclusion.

II. RELATED WORK

This section presents a brief review of FPA and a statement problem of the quality of the node's layout in WSNs.

A. METAHEURISTICS ALGORITHM OF FLOWER

Flower Pollination Algorithm (FPA) [22]-a recently population-based algorithm drawn inspiration from two pollination processes of the flowering plant that included self-pollination and cross-pollination. Pollinators transport the pollens of the flowering plant according to the rules of Lévy flights, and they can self-pollinate randomly. There are two universal concepts of exploring and exploiting guide the optimal process in the population-based algorithm: self-pollination of the flowering plant viewed as local pollination that expressed for exploitation in the search area, and a cross-pollination considered as global pollination that represented for exploration a promising area search. For control the characteristics of local and global pollination, a switching probability is used to switch between the exploring and exploiting phrases in pollination for FPA, (p $\in$ [0,1]). Let x_j^t, x_k^t be solution vectors of the pollens, i.e. pollens in the same plant or the flowers. The local pollination is modeled as follows.

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

where u is a random that is drawn from a uniform distribution in [0, 1]. The cross-pollination can convert into equations for updating solution vectors. If x_j^t and x_k^t come from the same plans or the same selected population, u would become a local random walk. Pollinators, e.g., insects carry pollen gametes of flower that often fly and move more extended range for global pollination of optimal processing. Flying insects over long distances with various length steps expressed a Lévy flight. Let's L be a Lévy distribution with powerful drawn formula.

$$L = \frac{\lambda\Gamma(\lambda) \times sin(\frac{\pi\lambda}{2})}{\pi \times s^{i+\lambda}}, \quad (2)$$

where $\Gamma(\lambda)$ is the gamma function, and valid distribution is the step with $s > 0$. The step size like a parameter $L(\lambda)$ corresponds to the strength of the pollination. Updating the location of global pollination is simulated as follows.

$$x_i^{t+1} = x_i^t + \gamma \times L(\lambda) \times (x_i^t - g^*) \quad (3)$$

where t is the current iteration; g^* is the current best solution found so far; γ is a scaling factor to control the step size. A parameter p is used as the proximity probability to switch between natural global pollination to intensive local pollination in the flower solution process.

B. STATEMENT PROBLEM IN WSN

The autonomous sensor nodes of the WSN are distributed spatially to monitor environmental conditions in the specific space. We can model a WSN as a connected graph G with *(V, E),* where V and E are the finite 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 $\{e_{1,2}, e_{1,3}, ..e_{i,j}, ..e_{n-1,n}\}$. The number of network nodes n is set to $|V|$ and the number of network links l is set *to* $|E|$. A *link* can be defined as follows.

$$e_{i,j} = \begin{cases} 1, & if\ v_i, v_j\ have\ edge \\ 0, & otherwise \end{cases}$$
$$(j = i + 1, 2, n; i = 1, 2, ..n - 1), \quad (4)$$

where e denoted a existed link of v_i, andv_j edge. It means that a link e equals $(v_i,)$ from $v_i \epsilon V$ to $v_j \epsilon$ *V,* and vice versa a existed ink e' equals $(,v_i)$ from v_j to v_i.

A positive real number is associated l as a link value of the attributed fitness function if the edge of $e_{i,j}$ is existed in the modeled graph. Some attributes of WSNs were defined as referencing on [23]. For example the represented weight of an attribute could be denoted $w_{i,j}^k = \{w_{ij}^1, w_{ij}^2, ..w_{ij}^l\}$, where $k = 1, 2, \ldots, l$. Each *link e* could have an associated value which corresponding to the obtained value from objective function.

Objective functions could be the cost, delay, data fusion, loss rate, or power consumption that related with the attributes in WSN. Let l be a positive real value of the objective function that is associated with *link e, (eϵ E).* The cost, coverage, delay, data fusion, loss rate, or power consumption of objective function could be represented the symbols letters of C, B, D, F, L and P respectively. It is written descriptive symbols for cost $C(e)$: $E \rightarrow R^+$, coverage $B(e) : E \rightarrow R^+$, delay $D(e) : E \rightarrow R^+$, data fusion $F(e) : E \rightarrow R^+$, loss rate $L(e) : E \rightarrow R^+$, power consumption $P(e) : E \rightarrow R^+$. Let $P_T(Point_S, Point_T)$ be a path from a point of source node s to the point of target node $T \epsilon M$. Let α be the coverage constraint, β be the delay constraint, δ be data aggregation constraint, ζ be the loss rate constraint, η be the dissipated energy constraint. The number of multi-criteria destination nodes M is the target group and *{{s}∪M}* is the multi-criteria group.

The minimum cost of the multi-criteria problem is defined as follows.

$$\begin{aligned} &Minimize\ Cost\ -function\ C(Tr(s, M)) \\ &\text{subjects to: } B_{x_iy_i}(X) \leq \alpha \forall T \in M, \\ &\qquad D(P_T(Point_S, Point_T)) \leq \beta \forall T \in M, \\ &\qquad F(P_T(Point_S, Point_T)) \leq \delta \forall T \in M, \\ &\qquad L(P_T(Point_S, Point_T)) \leq \zeta \forall T \in M, \\ &\qquad \vdots \\ &\qquad P(P_T(Point_S, Point_T)) \leq \eta \forall T \in M \end{aligned} \quad (5)$$

where *Tr(s, M)*is a multi-criteria tree as a sub-graph of G, the source node $s \epsilon V$ and the set of destination nodes $M \subseteq$ V - {s}. The represented solutions on the searching space of swarms could be the vector x. Let x be equal to$(x_{1,2}, x_{1,3}, \ldots, x_{i,j}, \ldots x_{n-1,n})$. A graph G can be expressed by the vector x,where $x_{i,j} = [1,\ \text{or}\ 0]$ if $e_{i,j}$ is selected or not. The set of all such vectors are set to X in corresponding to spanning trees in graph G and the multi-criteria degree constrained minimum problem can be formulated the multiobjective functions as follows.

$$\begin{aligned} &\text{minimize}\ f_k(x) = \sum w_{i,j}^k x_{i,j} \\ &\qquad \vdots \\ &\qquad (k = 1, 2, l; i = 1, 2, ..n - 1; j = i + 1, .., n) \\ &\qquad 1 \leq \sum w_{i,j}^k x_{i,j} \leq T \\ &\qquad (i = 1, 2, .., n; x \in X; j = 1, 2, \ldots n) \end{aligned} \quad (6)$$

where $f_k(x)$ is the k objective function and T denotes the degree constraint. A multi-objective function is considered to implement in Section 5 based on the selection two objective functions of these mentioned functions, e.g., the remaining energy nodes and contentions in WSNs.

III. HYBRID PARRALLEL AND COMPACT FPA

This section, we implement an improvement of FPA by hybridizing the parallel and the compact. The improvement of the FPA is an extension of our previous works [11], [21] that considered two techniques of the parallel and compact. The parallel process with communication strategy has significant for computations that exchanges information with other groups, shares the computation load, and enhances the diversity of individuals. The compact scheme simulates the behavior of population-based algorithms by employing the replacement of a community of solutions with its probabilistic representation. The compact offers an effective way of using a saving variable memory.

A. PARRALLEL SCHEME

Computing parallel is a carried computational form out simultaneously [11], [24]. The proven parallel method with communication strategies is the faster convergence and more accuracy than the original algorithm [25], [26]. In parallel with communication strategy of the metaheuristic algorithm, several created subpopulations by dividing the population into could evolve themselves separately over iterations, and the best solutions are selected to continue next generation according to the measured fitness. The communication scheme in the parallel processing is to exchange their properties among swarms, e.g. moving, copying, immigrating, or replacing randomly. A promising region would swap with weak areas within the solution space, and exploration of a promising area is carried out.

The communication strategies are suggested as follows: the best to all, neighboring groups, a pair swapping, etc. The strategy with the best to all is the finest pollens among all swarms migrate to every group, mutate them by replacing the worst pollens in each of these groups and update them after the period exchanging time of running. The strategy with the neighboring groups is to move the best pollen of one group to its surrounding groups then replace some poorer pollens after the period of running. The strategy with a pair swapping is a pair of two subgroups that the finest pollen of this subgroup replaces the worst pollen in the other subgroup and reverse. The drawback of communication in those algorithms was fixed the picking out one of the communication strategies. The effectiveness of such these strategies did not take full advantages. In this section, a new communication scheme is proposed to overcome this drawback. In the draft scheme, the exchanging strategies are combined rationally based on a switching parameter as a weight control. In the experimental section, a weight control ω is set to 0.7 to 0.5. This scheme enhances dynamically the communication strategies. A pseudo code of communication scheme is shown in Fig.1, namely *Communication (pollens)*.

Input: The subgroups $G_i, .. G_m, i = 1, 2.., m < N_p$ the population size
Output: Promising regions in subgroups G after communicating.

if $m>2$ **then**
  if $random \le \omega$ **then** // Stategy1- eighboring groups
  for i = 1 to m **do**
   replace the *worst*(G_i) with *best* random from ($G_i, .. G_m$)
  endfor
  else // Stategy2- the best to all
   for i = 1 to m **do**
     replace the *worst*(G_i) with *best* ($G_i, .. G_m$)
   endfor
  endif
else // Stategy3- a pair swapping
   replace *worsts*(G_1) with *best*(G_2)
   replace *worsts*(G_2) with *best*(G_1)
endif

FIGURE 1. Pseudo code of communication for parallel scheme.

B. COMPACT SCHEME

The estimated distribution algorithm (EDA) can process a representation of probabilistic to get fewer variable memory, rather store the population solution in the metaheuristics algorithms [27]. The compact method uses the principle of EDA to simulate the operations of metaheuristic algorithm. 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) [28] 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 stored in the memory. An attracted new candidate solution is by being iteratively biased toward a promising area of an optimal solution. The likelihood of a population of individuals in algorithm represents the probability vector of each component learned from previous generations. The structure of this vector was called Perturbation Vector (PV) [29]. These principles apply to the improvement of memory saving variable for compact FPA.

Different with the population-based algorithm such FPA, the compact technique considered population as "virtual community" by expressing the encoded data structure of probabilistic vector. A real-valued prototype vector represents the probability of each component described in a candidate solution. The specified probability for each element in new candidate solutions maintained in the optimum process. The optimization processing objective of the compact algorithm is to simulate the behavior of pollination of FPA, but it used with a much smaller stored variable memory. PV generates a candidate solution probabilistically from the vector. Competing for components toward to the better solutions reflect in the updated probability vector. The created trial solutions stayed to allocate in boundary constraints. PV is a matrix for specifying the two parameters of mean μ and standard deviation σ values in the PDF of each design variable. It can be defined as: $PV^t = [\mu^t, \sigma^t]$, 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. We use $PV(\mu_i, \sigma_i)$ to generate candidate solution x_i in the compact method. The solution is corresponding to the virtual pollens based on the associated Gaussian of μ and σ as following expressed PV.

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

Input: parameter μ, σ of probability vector, dimension d, and τ
Output: A new candidate x
for $i = 1$ to d **do**
    Generated $r \in [0,1]$ randomly in uniform distribution
    if $r < \tau$ then
        Generating $x_i \in [1,0]$ via $L_i(\mathrm{x})$ of Eq. (8)
    else
        Generating $x_i \in [1,0]$ via $R_i(\mathrm{x})$ of Eq. (8)
    end if
end for

FIGURE 2. Sampling scheme for Perturbation Vector *(PV)* for generating new agent.

1: Initialization of $PV(\mu, \sigma)$
    for i=1:n **do**
    $\mu_i^t = 0$;
    $\sigma_i^t = k$;
    end for
2: Initializing pollen location x via PV
3: Initializing g^* with the best location value: $g^* = \arg\min f[x]$.

FIGURE 3. Initialization phase of the cFPA.

where $P_i(\mathrm{x})$ is the probability distribution of PV that associated to the μ and δ in a truncated Gaussian PDF. This is the corresponding value of the PDF to variable x_i. The error function indicated as *erf* is fined in [30]. PDF could have corresponding to Cumulative Distribution Function (CDF) by constructing Chebyshev polynomials [31]. The arranged codomain of CDF is from 0 to 1. The described*CDF* is a real-valued random variable X in a value given distribution at $\leq x_i$. The value of newly calculated candidate x_i is a value of its inversed CDF.

C. HIBRID PARRALLEL AND COMPACT SCHEME

This subsection presents an implementation of a hybrid of the parallel and compact methods for FPA. Construct parallel, the whole population split into several subpopulations, and the communication scheme triggers among the subpopulations. The subpopulations run in parallel and evolve independently based on FPA optimization. The communicating subgroup is carried out, e.g., the most excellent pollination among the subgroups immigrated to another subset, replaced with the weakest bats according to a measured fitness, and updated the subgroups over the period.

Then we compact the subgroups based on probability vectors through the competition. The described compact process is through the illustrations in Figures 2 to 7. A couple of improvements for the perturbation vector are proposed through the process of sampling and updating PV. New generated candidate by learning and sampling from explicit probabilistic models is the promising solutions in search space.

We extend a couple of improvements for the perturbation vector through the process of sampling and updating PV. New candidates generated by learning and sampling from explicit probabilistic models that forward to promising solutions in search space. To control the probability of sampling of μ_i in PDF, a parameter τ as a weight is suggested from between left $[-1, \mu_i]$ and right $[\mu_i, 1]$. Thus, PV can be computed into two sides of the left and right as follows:

$$\begin{cases} L_i(x) = \dfrac{-\sqrt{\frac{2}{\pi}}}{\sigma_i(erf\left(\frac{\mu_i+1}{\sqrt{2}\sigma_i}\right)} \\ \quad \times \exp\left(-\dfrac{(x-\mu_i)^2}{2\sigma_i^2}\right) \; for -1 \leq x \leq \mu_i \\ R_i(x) = \dfrac{-\sqrt{\frac{2}{\pi}}}{\sigma_i(erf\left(\frac{\mu_i-1}{\sqrt{2}\sigma_i}\right)} \\ \quad \times \exp\left(-\dfrac{(x-\mu_i)^2}{2\sigma_i^2}\right) \; for \; \mu_i \leq x \leq 1 \end{cases} \tag{8}$$

The new sampling extended approach for PDF is for generating new candidates of the swarm that depicted in Figure 2. It means the PV scheme is to generate the agents of x randomly.

Figure 2 shows the described initialization of compact FPA (cFPA). The initializing PV scheme is as pseudo code. The best pollens g^* can be computed based on learning scheme of a sampling trial pollen. If the temporary new solutions are the better according to the evaluated fitness, x is then updated. A variable k is a large constant, e.g., k is set 10.

Two designed variables of winner or loser compete together to find out who is the better one. According to the evaluated fitness value, the winner or loser vector is the better or the worst. These sampled solutions are from PV. The winner is toward to a promising area in searching space based on the comparison between two design variables for pollens of the subgroup. A newly selected candidate is assigned to evaluate the given objective function. Determining winning solution is employed based on the comparison the chosen candidate agents. Figure 4 displays the competing scheme for winner/loser.

Moreover, 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. A typical parameter called virtual population Np is not strict variable as the correspond to the population size variable as in a population-based algorithm.

$$\mu_i^{t+1} = \mu_i^t + \frac{1}{N_p}(winner_i - loser_i) \tag{9}$$

where t is current iteration, and $i = 1, 2, ..N_p$. Regarding σ values, the update rule of each element is given by:

$$\sigma_i^{t+1} = \sqrt{(\sigma_i^t)^2 + (\mu_i^t)^2 - \left(\mu_i^{t+1}\right)^2 + \frac{1}{N_p}\left(winner_i^2 - loser_i^2\right)} \tag{10}$$

Input: The objective function f and solutions x_a, x_b
Output: *winner or loser*
if $f(x_a) < f(x_b)$ then
  Let $winner = x_a$
  Let $loser = x_b$
else
  Let $winner = x_b$
  Let $loser = x_a$
end if

FIGURE 4. Compete scheme for ***winner*** and ***loser.***

for *i= 1* to *d* **do**
  $\mu_{backup} = \mu_i^t$
  $\mu_i^{t+1} = \mu_i^t + \frac{1}{N_p}(winner_i - loser_i)$
  $\sigma_i^{t+1} = \sqrt{\max(0, (\sigma_i^t)^2 + \left(\mu_{backup}^t\right)^2 - (\mu_i^{t+1})^2 + \frac{1}{N_p}(winner_i^2 - loser_i^2)}$
  Improving for μ_i^{t+1} and σ^{t+1} via Eqs. (11), (12) with τ set to 0.01
end for

FIGURE 5. Updated PV scheme of PDF for new candidates.

Another improvement for updating PV, the values of μ_i^{t+1} and σ_i^{t+1} are modified with control parameter for expressing the maximum value of perturbation. An added parameter ϑ is a weight control of the expressed the perturbations maximum value.

$$\mu_i = \mu_i + \beta_i \times \vartheta \tag{11}$$

$$\sigma_i = \sqrt{\sigma_i^2 + \alpha_i \times \vartheta} \tag{12}$$

where β_i and α_i are random number distributed in $[-1, 1]$, and distributed in [0, 1] respectively. Evaluated the fitness function with selected location x is compared with g^* to obtain winner for next generation. Current location x is maintained in following steps of the scheme. Figure 5 shows the carried updating PV scheme out.

Figure 6 indicates the pseudo-code of the steps for the compact FPA. It simulated as the behavior of population-based algorithms by sampling probabilistic model. A virtual population is encoded its probabilistic representation.

The described the steps of the parallel compact FPA algorithm are as follows. The first, initialized population divided into G subgroups, objective function f and period of R for executing the communication strategy are assigned the pollens. The second, the compacted subsets evaluate, the communication scheme activates, and assessed results are compared to find the current best solution. The third is the termination that checks terminating condition, goes to the Second step if the stop condition is not met, otherwise records the best pollens and obtains value of the function $f(x)$.

Input: The objective function f, t = 0 and the swarm
Output: The best solution x_{gbest} , F_{min}
Initialization phase according to Initial scheme via Fig.3
while *stop criteria are not met* **do**
  Generating x by PV, via Fig.2
  Update pollens via Eqs.(1) and (3)
  Select best by Compete scheme via Fig.4.
  [*winner, loser*] = *compete(x, newx);*
  F_{new}=*f (newx);*
  Update PV scheme μ^{t+1}, σ^{t+1}, via Fig.5
  Update global best
  [*winner, loser*] = *complete(newx, g^*);*
  if *(F_{new}<F_{min})*
    $g^* = winner$; F_{min}=F_{new};
  end if
  t= t+1;
end while

FIGURE 6. Compact FPA scheme, namely ***cFPA***(subgroups).

Step 1. Initialization
(a) generate $G_{1..m}$ $(m \leq N_p)$ swarms, each G has n pollens
(b) assign period exchanging time *R*, counter *t =1*
(c) solutions x_{ij}^t in the *j-th* subgroup with n_j pollens, i =1,2,..,n; j=1,2,..m.
Step 2. **while** *termination is not satisfied* **do**
(a) **for** *j = 1* to *m* **do**
  cFPA(G_i) according to Compact scheme, via Fig.6
  end do
(b) **if***(mod(t,R)==0)* **then**
  Communication ($G_{1..m}$); according to Communication scheme, via Fig.1
  (c) Find the current best solution g^*
  t = t +1
endwhile
Step 3. Output the best solutions found

FIGURE 7. Pseudo code for Parallel and Compact FPA algorithm-pcFPA.

Figure 7 shows main processing steps as summarized pseudo code of pcFPA. In which G is subpopulations; n is the number of pollens in each swarm; m is the number of swarms; R is the exchanging period, and cFPA is a compact scheme.

IV. EXPERIMENT WITH TESTING PROBLEMS

In section, a series of selected optimal numerical problems [32] are utilized as benchmark functions to evaluate the performance of the proposed pcFPA. Table 2 displays the obtained results of the proposed pcFPA for testing functions in comparison with the variety FPA versions, e.g., the original FPA (FPA) [5], parallel FPA (pFPA) [11], and compact FPA (cFPA) [21]. The results also compared with popular meta-heuristic algorithms in the literature, e.g., Partcle swarm optimization (PSO) [33], Differential evolution (DE) [34], Bat

TABLE 1. Fifteen selected benchmark functions.

Name	Test functions	Range	Dimension	Iteration
Rosenbrock	$f_1(x) = \sum_{i=1}^{n-1}(100 \times (x_{i-1} - x_i^2)^2 + (1 - x_i)^2$	±100	30	2000
Quadric	$f_2(x) = \sum_{i=1}^{n}\sum_{k=1}^{i} x_i$	±100	30	2000
Ackley	$f_3(x) = 20 + e - 20e^{-0.2\sqrt{\frac{\sum_{i=1}^{n} x_i^2}{n}}} - e^{\frac{\sum_{j=1}^{n}\cos(2\pi x_i)}{n}}$	±32	30	2000
Rastrigin	$f_4(x) = \sum_{i=1}^{N}[10 + x_i^2 - 10cos2\pi x_i]$	±5.12	30	2000
Griewangk	$f_5(x) = 1 + \sum_{i=1}^{N}\frac{{x_i}^2}{4000} + \prod_{i=1}^{N} cos\frac{x_i}{\sqrt{i}}$	±100	30	2000
Spherical	$f_6(x) = \sum_{i=1}^{N} x_i^2$	±100	30	2000
Quartic Noisy	$f_7(x) = random[0,1) + \sum_{i=1}^{N} i \times x_i^4$	±1.28	30	2000
Schwefel	$f_8(x) = 418.983n - \sum_{i=1}^{N} x_i \times \sin\left(\sqrt{\lvert x_i \rvert}\right)$	±100	30	2000
Langermann	$f_9(x) = \sum_{i=1}^{n}[x_i^2 - 10\cos(2\pi x_i) + 10]$	±5.12	30	2000
Shubert	$f_{10}(x) = -20\exp(-0.2\sqrt{\frac{1}{n}\sum_{i=1}^{n} x_i^2}) - \exp(\frac{1}{n}\sum_{i=1}^{n}\cos(2\pi x_i)) + 20 + e$	±32	30	2000
Dixon &Price	$f_{11} = (x_1 - 1)^2 + \sum_{i=2}^{d} i(2 \times x_i^2 - x_{i-1})^2$	±32	30	2000
Michalewicz	$f_{12} = -\sum_{i=1}^{d}\sin(x_i) \times \sin^{20}(\frac{\mathrm{i} \times \mathrm{x}_i^2}{\pi})$	±5.12	30	2000
Schaffer N.2	$f_{13} = \frac{1}{2} + \frac{sin^2(x_1^2 - x_2^2) - 0.5}{[1 + 0.001 \times (x_1^2 - x_2^2)]^2}$	±100	30	2000
Matyas	$f_{14} = 0.26(x_1^2 + x_2^2) - 0.48x_1x_2 - 10$	± 10	30	2000
Drop-Wave	$f_{15} = \frac{1 + \cos(12\sqrt{x_1^2 + x_2^2}}{0.5(x_1^2 + x_2^2) + 2}$	±5.12	30	2000

algorithm (BA) [35], and parallel genetic algorithm (GA) [26] are shown in Table 3.

Table 4 shows the proposed algorithm compared with four methods of compact algorithms in the literature included the real compact Genetic algorithm (rcGA) [36], the compact Differential evolution (cDE) [37], the compact Artificial bee colony (cABC) [38], and the compact Bats algorithm (cBA) [39] regarding solution quality and time running. The experiments employed for the test functions are averaged outcomes values with all runs. Each function contains 2000 of full iterations and is repeated by different random seeds over twenty-five runs. The selected testing functions are the minimizing outcomes of global problems in the sequences. The reason why we selected fifteen benchmark functions for testing the proposed method is that the benchmark functions have been created more and more by adding a new one or compounding the existing ones. These selected benchmark problems consist of a broad range of test problems, e.g., uni-modal, multi-modal, and fixed-dimension functions. The initialized range, the dimension, and max iteration for all test

TABLE 2. Comparison of the proposed pcFPA with the FPA cFPA, and pFPA algorithms respectively for 15 test functions.

Functions	pcFPA	FPA	r	cFPA	r	pFPA	r
$f_1(x)$	**9.20E-01**	1.56E+00	+	1.07E+00	+	9.56E-01	~
$f_2(x)$	**3.38E+00**	4.34E+00	+	3.68E+00	~	3.49E+00	+
$f_3(x)$	**1.53E+00**	4.11E+00	+	2.10E+00	+	1.96E+00	+
$f_4(x)$	4.29E-01	5.62E-01	+	4.55E-01	+	4.29E-01	~
$f_5(x)$	**1.17E+01**	2.28E+01	+	2.44E+01	+	1.19E+01	~
$f_6(x)$	**2.15E+00**	6.79E+00	+	**2.15E+00**	~	2.38E+00	+
$f_7(x)$	**2.64E+00**	4.13E+00	+	3.61E+00	+	**2.46E+00**	-
$f_8(x)$	**-4.37E+02**	-3.67E+02	~	-4.09E+02	~	-4.17E+02	~
$f_9(x)$	**8.57E+01**	1.38E+02	+	1.15E+02	+	9.57E+01	+
$f_{10}(x)$	1.93E+00	1.93E+00	-	1.96E+00	~	**1.70E+00**	-
$f_{11}(x)$	**4.70E-02**	1.34E-01	+	6.06E-02	~	4.16E-01	+
$f_{12}(x)$	**1.91E-01**	5.48E-01	+	2.83E-01	+	6.38E-01	+
$f_{13}(x)$	2.08E+00	3.17E+00	+	2.29E+00	+	2.63E+00	+
$f_{14}(x)$	9.79E+00	1.14E+01	+	**8.86E+00**	-	**8.14E+00**	-
$f_{15}(x)$	**9.82E-03**	2.78E-02	~	1.57E-02	+	3.36E-02	~
AVG	-2.10E+01	-1.12E+01	+	-1.03E+01	+	-1.90E+01	+
Summary		13+ 2~ 1-		10+ 5~ 1-		8+ 5~ 3-	

functions listed display in Table 1. The parameters setting for pcFPA, and other algorithm are as follows. Virtual and real Population size N of the mentioned algorithms set to 80. The dimension of the solutions space-D isset based on the problem dimension requirements listed in Table 1.

Inertia weights used for PSO starts with 0.9 down to 0.4, which is decreased based on the increasing rate of iteration; factors of the speed control c_1 and c_2 set to 1.46, for the further setting PSO [33] and of compact methods [37]. The final results are taking the average of the outcomes from all runs.

Table 2 displays the comparison of the performance of the proposed pcFPA with the FPA, cFPA, and pFPA algorithms for fifteen benchmark optimization problems. In the columns of Table 2, the pcFPA, FPA, cFPA, and pFPA are the mean of the outcomes of 25 runs for the functions respectively. The highlighted numbers are the best results in among them in each function (each row of the table). The compared ratio r is a paired comparison between pcFPA and other algorithms respectively, i.e., a paired pcFPA and FPA, a paired pcFPA and cFPA, and a paired pcFPA and pFPA. The ratio r is set to the noticed indication '+, - and ∼.' The symbols of '+,' '−,' and '∼' mean the 'better,' 'worse,' and 'approximated' of the deviation with respecting to their outcomes respectively. If the optimized results of pcFPA are the better (smaller the value for minimized, or bigger for maximized problems) than others (the obtained from methods) of FPA, cFPA, and pFPA, then r is set symbol '+.' Do the same with symbols "−." and "∼." for the 'worse' and 'approximated' cases. Visibly, almost the highlighted cases of testing benchmark functions belong to pcFPA. It means that the proposed method outperforms the others approaches in observing the Table.

Table 3 compares the performance for fifteen numerical optimization problems of the proposed pcFPA with the other popular metaheuristic algorithms such as DE, PSO, GA, and BA algorithms. The highlighted numbers are the best results of the obtained average outputs in among them in each function.

Table 4 shows the compared performance quality optimization between pcFPA and the other compact algorithms such as cABC, cBA, cDE, and rcGA algorithms. The highlighted numbers are the best results of the obtained average outputs in among them in each function. Observed Tables 2, 3 and 4, the most highlight number, the symbol " + " of better points belong to the proposed algorithm. That means the proposed approach offers a competitive algorithm.

Figure 8 illustrates the comparison of executing time of the proposed pcFPA and FPA, PSO, BA, cFPA, and pFPA for the first eight functions. Clearly, all cases of the time consumption for testing functions of the pcFPA are smaller than the other algorithms, but the shortest running time belongs to cFPA. The results of the fast processing speed are that some the memory-stored parameters of cFPA and pcFPA are smaller than the stored solutions in the population-based algorithms.

TABLE 3. Comparison of the proposed pcFPA with the DE, PSO, BA, and GA algorithms respectively for 15 test functions.

Functions	pcFPA	DE	r	PSO	r	BA	r	GA	r
$f_1(x)$	9.20E-01	9.31E-01	~	**7.54E-01**	-	1.09E+00	+	1.25E+00	+
$f_2(x)$	3.38E+00	3.34E+00	+	3.49E+00	+	4.09E+00	+	4.38E+00	+
$f_3(x)$	1.45E+00	**1.24E+00**	+	1.91E+00	+	2.19E+00	+	2.91E+00	+
$f_4(x)$	**4.29E-01**	**5.93E-01**	+	4.37E-01	+	4.93E-01	~	**4.60E-01**	~
$f_5(x)$	**1.17E+01**	1.21E+01	+	**1.04E+01**	-	1.34E+01	+	1.98E+01	+
$f_6(x)$	**2.15E+00**	2.20E+00	~	2.38E+00	+	2.40E+00	~	2.17E+00	~
$f_7(x)$	**2.64E+00**	2.92E+00	+	2.46E+00	~	**3.12E+00**	+	7.40E+00	+
$f_8(x)$	**-4.33E+01**	-3.63E+01	+	-2.70E+01	+	-8.46E+00	+	-1.25E+01	+
$f_9(x)$	**5.76E+00**	5.39E+00	~	6.40E+00	+	6.39E+00	+	6.80E+00	+
$f_{10}(x)$	1.90E+00	**2.40E+00**	-	2.69E+00	+	2.36E+00	+	2.19E+00	+
$f_{11}(x)$	**4.70E-02**	4.16E-01	+	**4.16E-01**	+	4.16E-01	+	1.22E+00	+
$f_{12}(x)$	**1.92E-01**	3.81E-01	+	**2.38E-01**	~	3.90E-01	+	3.75E-01	+
$f_{13}(x)$	2.08E+00	2.63E+00	+	2.63E+00	+	2.63E+00	+	3.95E+00	+
$f_{14}(x)$	9.79E+00	1.03E+01	~	1.11E+01	+	9.30E+00	-	1.27E+01	+
$f_{15}(x)$	**9.82E-03**	8.33E-02	+	3.36E-01	~	8.33E-02	+	6.47E-02	~
AVG	-5.88E-02	5.73E-01	+	1.24E+00	+	2.66E+00	+	3.55E+00	+
Summary		11+ 4~ 1-		11+ 3~ 2-		12+ 3~ 1-		13+ 3~ 0-	

TABLE 4. Comparison of the proposed pcFPA with the cABC, cBA, cDE and rcGA algorithms respectively for 15 test functions.

Functions	pcFPA	cABC	r	cBA	r	cDE	r	rcGA	r
$f_1(x)$	**9.25E-01**	9.54E-01	~	7.39E+00	+	1.51E+00	+	9.44E-01	~
$f_2(x)$	3.38E+00	**2.99E+00**	-	1.15E+01	+	4.61E+00	+	1.02E+01	+
$f_3(x)$	**1.53E+00**	1.51E+00	~	6.10E+00	+	4.92E+00	+	7.45E+00	+
$f_4(x)$	3.67E-01	**1.53E-01**	-	7.94E-01	+	6.98E-01	+	7.99E-01	+
$f_5(x)$	**1.17E+01**	1.34E+01	+	1.11E+01	~	2.23E+01	+	1.76E+01	+
$f_6(x)$	**2.15E+00**	3.91E+00	+	1.01E+01	+	**1.72E+00**	-	5.17E+00	+
$f_7(x)$	**2.64E+00**	8.81E+00	+	2.05E+01	+	5.49E+00	+	7.40E+00	+
$f_8(x)$	**-4.37E+01**	-2.50E+01	+	-2.67E+01	~	-2.38E+01	+	-1.25E+01	+
$f_9(x)$	**8.57E+01**	1.21E+02	+	1.08E+02	+	**7.94E+01**	-	1.28E+02	+
$f_{10}(x)$	1.93E+00	**1.66E+00**	-	3.55E+00	+	1.81E+00	+	3.18E+00	+
$f_{11}(x)$	**4.70E-02**	**5.93E-02**	~	3.46E-01	+	9.39E-02	~	3.22E-01	+
$f_{12}(x)$	**4.91E-01**	9.81E-01	+	1.75E+00	+	8.24E-01	+	1.32E+00	+
$f_{13}(x)$	2.08E+00	6.93E-01	-	4.67E+00	+	**6.07E-01**	-	**1.95E+00**	-
$f_{14}(x)$	9.79E+00	1.22E+01	+	1.01E+01	+	1.27E+01	+	1.23E+01	+
$f_{15}(x)$	**9.82E-03**	2.40E-02	~	1.93E+00	+	6.32E-02	~	2.75E-02	~
AVG	5.27E+00	9.55E+00	+	1.14E+01	+	7.52E+00	+	1.23E+01	+
Summary		8+ 4~ 4-		14+ 2~ 0-		11+ 2~ 3-		13+ 2~ 1-	

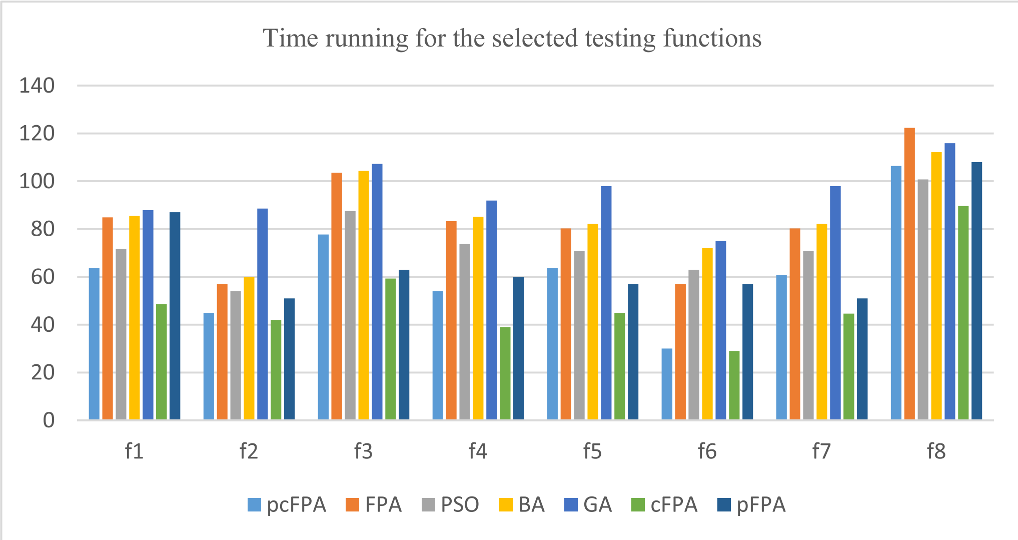

FIGURE 8. Comparison of the pcFPA for running times with FPA, PSO, BA, GA, cFPA and pFPA algorithms for the first eight test functions.

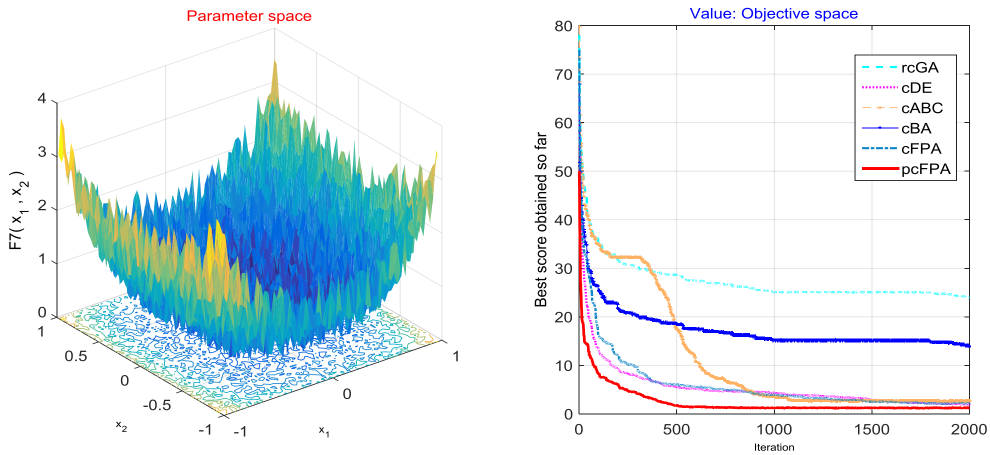

FIGURE 9. Comparison of the evaluated performance of the proposed pcFPA with the rcGA, cDE, cABC, cBA, and pFPA algorithms for the testing function f7 of Quartic Noisy.

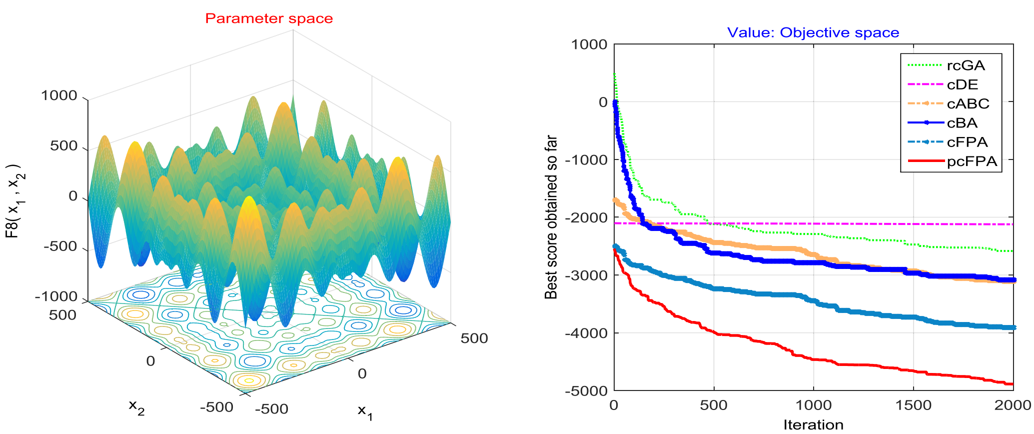

FIGURE 10. Comparison of performance of the proposed pcFPA with the rcGA, cDE, cABC, cBA, and pFPA algorithms for the testing function f8 of Schwefel.

Figures 9 to 11 show the compared the best score results of the proposed pcFPA with the rcGA, cDE, cABC, cBA, and pFPA algorithms for three selected testing functions $f_7(x)$,$f_8(x)$ *and* $f_9(x)$ over 25 runs outputs in the same

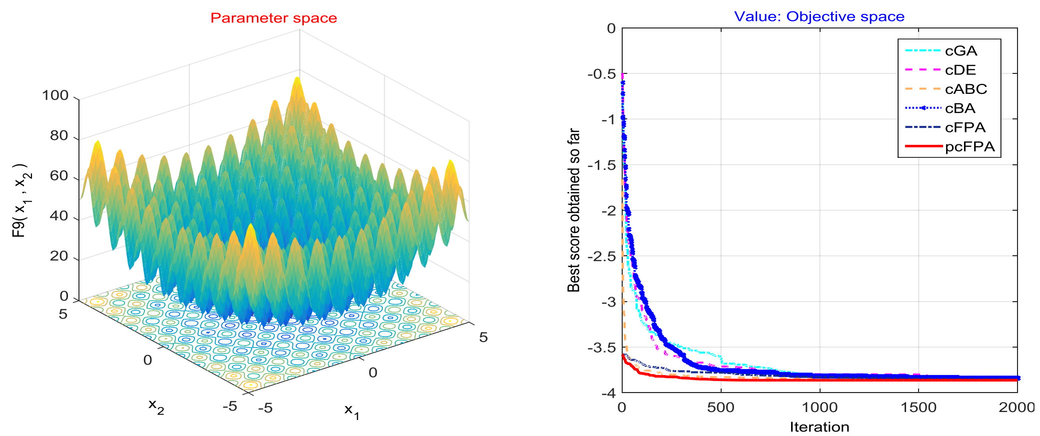

FIGURE 11. Comparison of performance of the proposed pcFPA with the rcGA, cDE, cABC, cBA, and pFPA algorithms for the testing function f9 of Langermann.

iteration of 2000. Clearly, the cases of these testing functions on the pcFPA (indicated red curve) shows a comparatively faster convergence than other algorithms It says the accuracy of the proposed method of pcFPA is improved significantly.

V. APPLIED NODES LAYOUT OPTIMIZATION IN WSNs

In mentioned Session 2.1 of constrained optimization in WSN, the limited resources of sensor nodes are the primary constrained criterion that causes the driving factors of the provided solutions to a performance quality of WSN. Because of non-rechargeable sensor nodes, prolonging lifetime is a crucial requirement of good designed WSN. The quality of connectivity in the network and the residual energy of nodes are two impacting factors to ensure for prolonging the lifetime of the sensor networks. The maintaining network connectivity and the minimum consuming energy of nodes are considered to optimize the WSN. This section applies the above-proposed method of saving memory variable of the pcFPA to deal with an optimum node layout as a constrained issue in WNS. Objective functions for the node layout optimization based on topology scheme in WSN. The node layout could be constructed based on the topology for remaining energy nodes and connections. The experimental results of the proposed approach for the node layout optimization are compared with the PSO [40] and the GA [41] optimizations in WSN.

A. NETWORK MODEL

A simulated graph G (V, E) of WSN with distributed sensor nodes spatially in the desired area is the network topology with the abstracted nodes in the transmitting range. The transmission range r is the communication area of reaching among nodes in a network. It means that a node can communicate with others in transmission range. If node i is in the transmission range r of node j, node i can receive the signal of node j. In the graph G, V and E are the finite set of vertices and edges as the set of nodes in the network and the set of connectable communication of the pair nodes respectively. The set of vertices V is set to $\{v_1, v_2, ..v_n\}$, and the set of edges

TABLE 5. A sample of representing for the locations of the sensor nodes.

Index	*Node$_i$*	**1**	**2**	**3**	**4**	**5**	**6**	**..**	**n**
x	*09*	63	91	112	71	62	55	..	12
y	*01*	4	11	30	15	20	45	..	81

TABLE 6. Attribution of current links if flag = 1,otherwise flag = 0.

Index	*Node$_{ij}$*	**1**	**2**	**3**	**4**	**5**	**6**	**7**	**.**	**.**	**.**	**n**
Edge	**1**	0	0	0	0	1	0	0	.	.	.	0
	2	1	0	1	0	1	0	0				1
	..	..	..	..	..	..	..	..	..	..	.	.
	n	0	1	1	0	0	1	0	.	.	.	0

E is set to $\{e_{1,2}, e_{1,3}, ..e_{i,j}, ..e_{n-1,n}\}$, where n is the number of network nodes. To optimize the topology problem in WSN, we first need to construct a model solution including spatial coordinates of the nodes that could obtain the distances to the base station, and attribute links between nodes. We then optimize the mathematical model of the problem in WSN. Table 5 shows an example of representation the location of sensor nodes in two-dimensional coordinates.

Table 6 indicates a case of representation the attribution of status links related location nodes in a modeled solution. Attributions of the existing link of node i and j in the sensor networks are the edge of a graph. If the connection can form between nodes $\mathbf{v_i}$, and $\mathbf{v_j}$; then $e_{i,j} = e_{,ji}$ is set to 1; otherwise, $e_{i,j} = e_{,ji}$ is set to *0,* where $\forall\ \boldsymbol{v_i}, \boldsymbol{v_j} \in \boldsymbol{G_n}$; $\boldsymbol{i = 1, 2, 3, \ldots n}; \boldsymbol{j = i + 1, \ldots n}$. This modeled graph could map to the associated real numbers, which could obtain the best results from the attributed fitness functions.

The formed objective functions are figured out by analyzing the attributes of the WSNs subject to constraints criteria.

Let *D(i,j)*be the Euclidean distance of $nodes_{i,j}$ in the network, r_{max} be maximum wireless transmission range of the connectable neighbors of node i to communicate together, and CN_i be the set of nearby nodes. Induced subgraph of G is set to $G_i =<V_i, E_i>$ for each node i belong to V.

The sub-graph G_i is the connectable node graph of node i, and E_i is a subset of edges E and V_i is equal to CN_i, $\times E_i$.

The centered disk of converging node i with its certain radius r is denoted $C(i,r)$. Node i could affect to all nodes located in area of radius r, so r could be set to $D(i,j)$. Let $L_{i,j}$ be the edge coverage between node i and node j in the established disk of nodes.

$$L_{i,j} = [\{v_i \in V | D(v_i, i) \le R_1\}] \cup [v_j \in V | D(v_j, j) \le R_2)] \quad (13)$$

where R_1, R_2 are the radius of $C(i,r)$, $C(j,r)$ respectively. The strength of an edge between node i and node j is defined as:

$$S_{i,j} = \frac{Ep_i \times Ep_j}{\sqrt{Ep_i^2 + Ep_j^2}} \quad (14)$$

where Ep_i and Ep_j are the remaining energy of node$i$,and node j respectively. The energy consumption between nodes v_i, v_j is considered as given $P_{i,j} = k \times d^{\beta}$, where k is the system constant, d is the communication distance; the value of β is often predefined constant as the path loss exponent. A positive real number is the weight of a communication link, denoted W_{ij} that is assigned to each edge of the node i and node j in the modeled graph. The aspects of base energy consumption, and low communication interference are configured out by weighting the topology network in the graph. The formula is given as.

$$W_{i,j} = \alpha_1 \times \frac{P_{i,j}}{S_{i,j}} + \alpha_2 \times L_{i,j} \quad (15)$$

where α_1 and α_2 are the predefined parameters, and $\alpha_1 + \alpha_2$ set to 1. Moreover, the density of the nodes in the network is also a factor of the considered affects the network performance.

The neighbor density associated with the pollen i^{th} is measured with that denoted $Nei(i)$ following:

$$Nei(i) = \sum_{j=1}^{NP} sh(W_{i,j}) \quad (16)$$

where $sh(W_{i,j})$ is sharing function and NP is population size. $sh(W_{i,j})$ is set to one if $W_{i,j} \le \delta$, in which δ is a sharing parameter and set to zero otherwise. m is dimensions and $W_{i,j}$ is set to $W_{i,j}^1 + W_{i,j}^2 + \ldots + W_{i,j}^m$.

Additional, the multi-criteria optimum problem should put in Pareto optimal solution that can be expected to reach a globally optimal set of the algorithm. A measured dominance is as the state of domination the i^{th} agent on the current population that denoted $Domi(i)$ is as following:

$$Domi(i) = \sum_{j=1}^{p} x_{i,j} \quad (17)$$

where $x_{i,j}$ is set to 1 if pollen j dominates pollen i, and set to 0 otherwise. A vector could be represented by Eq.(17). The graph G_n with spanning tree can be expressed by the vector X.The constrained optimization problem of WSN can be transformed into the unconstrained optimization problem and the fitness function can be formulated as the following equation.

$$ObjectFunction_i = (1 + Domi(i)) \times (1 + Nei(i)) \quad (18)$$

TABLE 7. Comparison of performance and speed of the pcFPA with the GA, the PSO, pFPA, and FPA algorithms for topology optimization in WSN.

Appr-oaches	Pop. size (N)	Min	Max	Mean	Std.	Exe. times
GA	160	2.21E-01	4.45E+00	8.62E-01	1.18E+00	*2.91E+00*
PSO]	160	1.57E-01	4.01E+00	5.46E-01	8.23E-01	*2.41E+00*
pFPA	160	1.74E-01	3.90E+00	5.96E-01	9.77E-01	*2.61E+00*
FPA	160	1.54E-01	4.10E+00	6.77E-01	1.10E+00	2.53E+00
pcFPA	**4**	**9.40E-02**	**3.10E+00**	**3.76E-01**	**2.19E-01**	**2.34E+00**

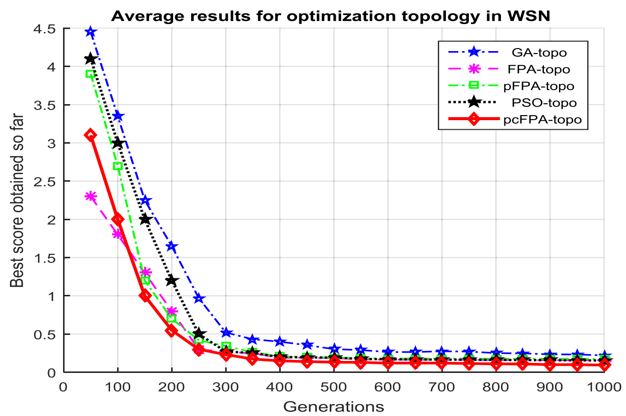

FIGURE 12. Comparisons of the pcFPA performance with the PSO, FPA, pFPA and GA methods for optimization in WSN.

We generate a vetor of the coordinate of G vertices and the weights of each edge randomly. Each edge is supposed to have two real number weights. The node layout is constructed based on the topology for remaining energy nodes and connections in WSN. The equation (18) is the objective function for the layout of the nodes in WSN.

B. EXPERIMENTAL RESULT OF NODES LAYING OUT WSN

Setting environmental network arranges of the deployed area is $M \times M$ (M = 100,200,300m). The dimension of the solution space is equal to the number of agents or sensor nodes set to N (N = 500, 1000, 1500). The number of objective functions set to obj (obj = 2). The fixed iterations for the triggered communication set to Ri (e.g., Ri = 20). The remaining energy begins at initializing nodes are set to 2.0J. The circuit energy set to 50nJ/bit. For each generation need consumption run is about 0.05pJ. The initial coverage of the edge is a randomized $L_{i;j} \epsilon$ [0, 1]. The coefficients k and β are set to 4. α_1 and α_2 equal to 1. The setting parameters in the algorithms of GA, PSO, FPA and pcFPA are as follows. The virtual population size $N = 80$. The objective functions contain 1000 of iterations, and it is repeated by different random seeds with 10 runs.

The parameters for the GA are set as follows; crossover probability $p_c = 0.2$, mutation probability $p_m = 0.05$, maximum generation max-gen equal to 1000 in each run, for further reference in setting [41]. The final result is obtained by taking the average of the outcomes from 15 runs.

The experimental results of the proposed pcFPA are compared with the GA, PSO, and FPA methods for the constrained optimizations.

Figure 12 illustrates the comparison of the proposed pcFPA for node's layout based on topology in WSN, with the GA, PSO, pFPA, and FPA methods. Observed, the performance of the proposed method of pcFPA for the fitness functions evaluation values in 15 runs outperform the other methods.

VI. CONCLUSION

In this paper, an improved Flower pollination algorithm (FPA) based on a hybrid of the parallel and compact schemes was proposed (namely pcFPA) for the optimization problems and the laying out nodes issues in Wireless sensor networks (WSN). The parallel technique enhanced to avoid the optimum local issue in compound constrained optimization problems and allowed fast convergence. Because the compact scheme used a probabilistic model to generate a new candidate solution in space search of the algorithm which forwards to promising areas so far, the compact technique reduced the variables of computing optimization. The implemented hybrid the parallel and the compact scheme shows significant advantages from each of the test schemes and achieves improved collaboration in the optimization process. In the simulation section, a set of the selected optimization problems and the node's layout problem based on topology in WSN are used to evaluate the accuracy, executing time and the saving memory of the proposed algorithm. Compared results with the other algorithm of FPA and the different algorithms in the literature shows that the proposed algorithm outperforms its competitors.

REFERENCES

[1] S. Mahdavi, M. E. Shiri, and S. Rahnamayan, "Metaheuristics in large-scale global continues optimization: A survey," *Inf. Sci.*, vol. 295, pp. 407–428, Feb. 2015.

[2] I. Mandoiu and A. Zelikovsky, *Bioinformatics Algorithms: Techniques and Applications*, vol. 3. Hoboken, NJ, USA: Wiley, 2008.

[3] R. T. Marler and J. S. Arora, "Survey of multi-objective optimization methods for engineering," *Struct. Multidiscipl. Optim.*, vol. 26, no. 6, pp. 369–395, Apr. 2004.

[4] E. Elbeltagi, T. Hegazy, and D. Grierson, "Comparison among five evolutionary-based optimization algorithms," *Adv. Eng. Inform.*, vol. 19, no. 1, pp. 43–53, 2005.

[5] X. S. Yang, "Flower pollination algorithm for global optimization," in *Including Subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics* (Lecture Notes in Computer Science), vol. 7445. Berlin, Germany: Springer, 2012, pp. 240–249.

[6] N. Diab and E. El-Sharkawy, "Recent advances in flower pollination algorithm," *Int. J. Comput. Appl. Technol. Res.*, vol. 5, no. 6, pp. 338–346, 2016.

[7] H. Chiroma, N. L. M. Shuib, S. A. Muaz, A. I. Abubakar, L. B. Ila, and J. Z. Maitama, "A review of the applications of bio-inspired flower pollination algorithm," *Procedia Comput. Sci.*, vol. 62, pp. 435–441, Jan. 2015.

[8] X. He, X.-S. Yang, M. Karamanoglu, and Y. Zhao, "Global convergence analysis of the flower pollination algorithm: A discrete-time Markov chain approach," *Procedia Comput. Sci.*, vol. 108, pp. 1354–1363, Jan. 2017.

[9] D. Rodrigues, X.-S. Yang, A. N. De Souza, and J. P. Papa, "Binary flower pollination algorithm and its application to feature selection," in *Recent Advances in Swarm Intelligence and Evolutionary Computation*. Cham, Switzerland: Springer, 2015, pp. 85–100.

[10] X.-S. Yang, M. Karamanoglu, and X. He, "Multi-objective flower algorithm for optimization," *Procedia Comput. Sci.*, vol. 18, pp. 861–868, Jan. 2013.

[11] T.-T. Nguyen, C.-S. Shieh, M.-F. Horng, T.-K. Dao, and T.-G. Ngo, "Parallelized flower pollination algorithm with a communication strategy," in *Proc. 7th Int. Conf. Knowl. Syst. Eng. (KSE)*, Oct. 2015, pp. 103–107.

[12] Y. Zhou *et al.*, "Discrete greedy flower pollination algorithm for spherical traveling salesman problem," *Neural Comput. Appl.*, 2017. doi: 10.1007/s00521-017-3176-4.

[13] K. Bibiks, J.-P. Li, and Y. F. Hu, "Discrete flower pollination algorithm for resource constrained project scheduling problem," *Int. J. Comput. Sci. Inf. Secur.*, vol. 13, no. 7, pp. 8–19, 2015.

[14] M. Bensouyad and D. Saidouni, "A discrete flower pollination algorithm for graph coloring problem," in *Proc. IEEE 2nd Int. Conf. Cybern. (CYBCONF)*, Jun. 2015, pp. 151–155.

[15] J.-S. Pan, T.-K. Dao, T.-S. Pan, T.-T. Nguyen, S.-C. Chu, and J. F. Roddick, "An improvement of flower pollination algorithm for node localization optimization in WSN," *J. Inf. Hiding Multimedia Signal Process.*, vol. 8, no. 2, pp. 486–499, 2017.

[16] D. H. Wolper and W. G. Macready, "No free lunch theorems for optimization," *IEEE Trans. Evol. Comput.*, vol. 1, no. 1, pp. 67–82, Apr. 1997.

[17] T.-T. Nguyen, T.-K. Dao, M.-F. Horng, and C.-S. Shieh, "An energy-based cluster head selection algorithm to support long-lifetime in wireless sensor networks," *J. Netw. Intell.*, vol. 1, no. 1, pp. 23–37, 2016.

[18] C. F. García-Hernández, P. H. Ibarguengoytia-Gonzalez, J. García-Hernández, and J. A. Pérez-Díaz, "Wireless sensor networks and applications: A survey," *J. Comput. Sci. Netw. Secur.*, vol. 7, no. 3, pp. 264–273, 2007.

[19] J.-S. Pan, T.-T. Nguyen, T.-K. Dao, T.-S. Pan, and S.-C. Chu, "Clustering formation in wireless sensor networks: A survey," *J. Netw. Intell.*, vol. 2, no. 4, pp. 287–309, 2017.

[20] I. F. Akyildiz, W. Su, Y. Sankarasubramaniam, and E. Cayirci, "Wireless sensor networks: A survey," *Comput. Netw.*, vol. 38, no. 4, pp. 393–422, 2002.

[21] T.-K. Dao, T.-S. Pan, T.-T. Nguyen, S.-C. Chu, and J.-S. Pan, "A compact flower pollination algorithm optimization," in *Proc. 3rd Int. Conf. Comput. Meas. Control Sensor Netw. (CMCSN)*, May 2016, pp. 76–79.

[22] X.-S. Yang, "Flower pollination algorithm for global optimization," in *Unconventional Computation and Natural Computation*. Berlin, Germany: Springer, 2013, pp. 240–249.

[23] M. Li and B. Yang, "A survey on topology issues in wireless sensor network," in *Proc. Int. Conf. Wireless Netw.*, 2006, p. 503.

[24] S. W. Mahfoud and D. E. Goldberg, "Parallel recombinative simulated annealing: A genetic algorithm," *Parallel Comput.*, vol. 21, no. 1, pp. 1–28, 1995.

[25] S. C. Chu, J.-F. Chang, J. F. Roddick, and J.-S. Pan, "A parallel particle swarm optimization algorithm with communication strategies," *J. Inf. Sci. Eng.*, vol. 21, no. 4, pp. 809–818, 2005.

[26] D. Abramson and J. Abela, "A parallel genetic algorithm for solving the school timetabling problem," Div. Inf. Technol., Hobart, TAS, Australia, Tech. Rep., 1992, pp. 1–11.

[27] P. Larranaga and J. A. Lozano, *Estimation of Distribution Algorithms—A New Tool for Evolutionary Computation*, vol. 2. Boston, MA, USA: Springer, 2002.

[28] P. Billingsley, *Probability and Measure*, 3rd ed. Hoboken, NJ, USA: Wiley, 2013.

[29] 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, Nov. 1999.

[30] I. N. Bronshtein, K. A. Semendyayev, G. Musiol, and H. Muehlig, *Handbook of Mathematics*, 6th ed. Berlin, Germany: Springer, 2015.

[31] J. C. Mason and C. D. Handscomb, *Chebyshev Polynomials*. London, U.K.: Chapman & Hall, 2002.

[32] X. Yao, Y. Liu, and G. Lin, "Evolutionary programming made faster," *IEEE Trans. Evol. Comput.*, vol. 3, no. 2, pp. 82–102, Jul. 1999.

[33] M. Clerc and J. Kennedy, "The particle swarm-explosion, stability, and convergence in a multidimensional complex space," *IEEE Trans. Evol. Comput.*, vol. 6, no. 1, pp. 58–73, Feb. 2002.

[34] R. Storn and K. Price, "Differential evolution—A simple and efficient heuristic for global optimization over continuous spaces," *J. Global Optim.*, vol. 11, no. 4, pp. 341–359, 1997.

[35] X.-S. Yang, "A new metaheuristic bat-inspired algorithm," *Nature Inspired Cooperat. Strategies Optim.*, vol. 284, pp. 65–74, Apr. 2010.
[36] E. Mininno, F. Cupertino, and D. Naso, "Real-valued compact genetic algorithms for embedded microcontroller optimization," *IEEE Trans. Evol. Comput.*, vol. 12, no. 2, pp. 203–219, Apr. 2008.
[37] E. Mininno, F. Neri, F. Cupertino, and D. Naso, "Compact differential evolution," *IEEE Trans. Evol. Comput.*, vol. 15, no. 1, pp. 32–54, Feb. 2011.
[38] T.-K. Dao, S.-C. Chu, T.-T. Nguyen, C.-S. Shieh, and M.-F. Horng, "Compact artificial bee colony," in *Modern Advances in Applied Intelligence*, vol. 8481. Cham, Switzerland: Springer, 2014, pp. 96–105.
[39] T. K. Dao, J. S. Pan, T. T. Nguyen, S. C. Chu, and C. S. Shieh, "Compact bat algorithm," in *Advances in Intelligent Systems and Computing*, vol. 298, J.-S. Pan, V. Snasel, E. S. Corchado, A. Abraham, and S.-L. Wang, Eds. Springer, 2014, pp. 57–68.
[40] W. Z. Guo, B. Zhang, G. L. Chen, X. F. Wang, and N. X. Xiong, "A PSO-optimized minimum spanning tree-based topology control scheme for wireless sensor networks," *Int. J. Distrib. Sens. Netw.*, vol. 9, no. 4, Apr. 2013, Art. no. 985410.
[41] G. Zhou and M. Gen, "Genetic algorithm approach on multi-criteria minimum spanning tree problem," *Eur. J. Oper. Res.*, vol. 114, no. 1, pp. 141–152, 1999.

JENG-SHYANG PAN received the B.S. degree in electronic engineering from the National Taiwan University of Science and Technology, in 1986, the M.S. degree in communication engineering from National Chiao Tung University, Taiwan, in 1988, and the Ph.D. degree in electrical engineering from the University of Edinburgh, U.K., in 1996. He is currently the Dean of the College of Information Science and Engineering, Fujian University of Technology. He joined the Editorial Board of LNCS Transactions on Data Hiding and Multimedia Security, the Journal of Computers, the Journal of Information Hiding, and Multimedia Signal Processing. His current research interests include soft computing, information security, and signal processing.

TRONG-THE NGUYEN received the Ph.D. degree in communication engineering from the National Kaohsiung University of Science and Technology, Taiwan, in 2016. He is currently a Lecturer with the Department of Information Engineering, UMT Haiphong-Vietnam, and College of Information Science and Engineering, Fujian University of Technology, China. His research interests include computational intelligence and wireless sensor networks.

THI-KIEN DAO received the Ph.D. degree in communication engineering from the National Kaohsiung University of Science and Technology, Taiwan, in 2019. She is currently a Lecturer with the College of Information Science and Engineering, Fujian University of Technology, China. Her research interests include computational intelligence and industry applications.