Constrained Relay Node Deployment using an improved multi-objective Artificial Bee Colony in Wireless Sensor Networks

Wenjie Yu, Xunbo Li, Xiang Li and Zhi Zeng
School of Mechatronics Engineering,
University of Electronic Science and Technology of China,
Chengdu, 611731 – China
[e-mail:wenjie.y@outlook.com]
*Corresponding author: Wenjie Yu

Received July 22, 2016; revised November 8, 2016; revised February 24, 2017; accepted March 10, 2017; published June 30, 2017

Abstract

Wireless sensor networks (WSNs) have attracted lots of attention in recent years due to their potential for various applications. In this paper, we seek how to efficiently deploy relay nodes into traditional static WSNs with constrained locations, aiming to satisfy specific requirements of the industry, such as average energy consumption and average network reliability. This constrained relay node deployment problem (CRNDP) is known as NP-hard optimization problem in the literature. We consider addressing this multi-objective (MO) optimization problem with an improved Artificial Bee Colony (ABC) algorithm with a linear local search (MOABCLLS), which is an extension of an improved ABC and applies two strategies of MO optimization. In order to verify the effectiveness of the MOABCLLS, two versions of MO ABC, two additional standard genetic algorithms, NSGA-II and SPEA2, and two different MO trajectory algorithms are included for comparison. We employ these metaheuristics on a test data set obtained from the literature. For an in-depth analysis of the behavior of the MOABCLLS compared to traditional methodologies, a statistical procedure is utilized to analyze the results. After studying the results, it is concluded that constrained relay node deployment using the MOABCLLS outperforms the performance of the other algorithms, based on two MO quality metrics: hypervolume and coverage of two sets.

Keywords: WSNs, constrained relay node deployment, ABC with a linear local search, MO optimization, hypervolume, coverage of two sets

This research is supported by National Natural Science Foundation of China (Grant No.: 51405066) and Science and Technology Planning Project of Guangdong Province (Grant No.: 2016A010102002).

https://doi.org/10.3837/tiis.2017.06.005 **ISSN : 1976-7277**

1. Introduction

Recent years have witnessed significant advances in wireless sensor networks (WSNs), which have evolved in many areas due to their large applicability and development possibilities, for example, for forest fire detection, environmental control, home automation, industrial process monitoring, intensive agriculture, and robotics [1].

Traditional WSNs typically consist of sensor nodes (SNs) and a base station (BS). This kind of network is easy to perform badly due to some intrinsic properties of sensors, such as limited coverage radius, powered by non-replaceable batteries with limited energy, etc. Some studies considered to improve this situation by deploying sensors reasonably [2, 3]. However, it has been unaddressed that the network may break down due to some sensor nodes with heavy burden run out of energy early. Therefore, a new kind of nodes with higher energy capacity, namely relay nodes (RNs), was proposed to be added into the WSN[4]. It was verified that reasonable deployment of RNs is beneficial to improve the network properties, such as network connectivity/lifetime maximization [5] and network load balancing [6].

Most approaches studied the RNs deployment problem (RNDP) without setting any physical limitations, while most deployment regions in reality may contain forbidden regions or impose lower bounds on internode distances. We cannot deploy nodes anywhere we want. Therefore, a more practical situation, the constrained RNDP (CRNDP), is considered. CRNDP is an NP-hard optimization problem [7], which cannot be solved with conventional methods. Instead, some works assume non-conventional methods, such as heuristics [8] or metaheuristics [9].

In this paper, we consider solving CRNDP by metaheuristics, which could solve very general types of problems. To this end, a novel MO metaheuristic, namely MOABCLLS, is developed to solve the problem of RNs deployment with lower bounds on internode distances constraint, aiming to optimize some important factors in the industry. Our contribution can be summarized as follows:

- By introducing a practically lower bounds distances constrained framework for WSNs, we conduct an MO research on CRNDP, optimizing average energy consumption (AEC) of the sensors and average network reliability (ANR), which are two important factors in the industry.
- A novel MO metaheuristic, namely MOABCLLS, is introduced for solving CRNDP. This algorithm is an extension of traditional ABC, adding a linear local search factor and integrating concepts of non-dominated sorting and crowding distance from NSGA-II [10].
- We compare MOABCLLS with two versions of MO ABC, two standard genetic algorithms, NSGA-II [10] and SPEA2 [11], and two different MO trajectory algorithms, MOVNS and MOVNSwP [9], through a widely accepted statistical methodology. In this respect, the results acquired are analyzed via two MO quality metrics: hypervolume and coverage of two sets [12, 13].

The rest of this paper is organized as follows: In Section 2, we discussed related works concerning CRNDP and metaheuristics methods applied in WSNs. Section 3 is devoted to the description of the WSN model and the problem definition. The MOABCLLS proposed is detailed in Section 4. The experimental configuration and the results analysis are given in Section 5 and 6, respectively. Finally, Section 7 concludes the paper with future research directions.

2. Related Work

In this section, we describe research about how to efficiently deploy RNs into traditional WSNs without and with constraints. By routing structures [14], RNDPs can be classified into either single-tiered or two-tiered. In single-tiered RN deployment, an SN forwards packets received from other SNs or RNs. In two-tiered RN deployment, an SN only forwards its own sensed information to an RN or a BS.

First, we analyze previous works on single-tiered RN deployment, in which both RNs and SNs forward the received packets. Lloyd and Xue [15] optimized the network lifetime in single-tiered network with heuristics, while they assured the network connectivity; in this regard, they conducted two different types of research: first, between each pair of sensors, there was a connecting path composed of RNs and/or sensors, and another one is that the path was solely composed of RNs. Zhao and Chen [16] optimized the energy efficiency of the WSN by using a particle swarm algorithm, with the objective of minimizing the average path length in the single-tiered network. Cheng. et al. [17] guaranteed global connectivity by placing a minimum of RNs in single-tiered WSN with heuristics. Han et al. [18] optimized the fault-tolerance in single-tiered network considering sensors with adjustable transmission radius with heuristics. Lanza-Gutierrez and Gomez-Pulido [9] studied how to use metaheuristics to deploy RNs into single-tiered WSN with the objective of optimizing average energy consumption and average sensitivity area of the network. Ranga et al. [19] proposed a new solution based on a zero gradient point inside the convex hull polygon to restore the lost connectivity by the placement of RNs in WSNs. Truong et al. [20] introduced MO network repairing algorithms for restoring WSN connectivity in a known area.

Next, we give a review of previous works on two-tiered RNs deployment, in which only the RNs forward the packets received. Hao et al. [21] tried to place a minimum number of RNs in two-tiered WSN with heuristics such that every SN can reach at least two RNs and there exist at least two node-disjoint paths between every pair of RNs. Liu et al. [22] explored how to deploy minimum RNs into a two-tiered WSN with heuristics, considering two cases, making the network connected and making the network 2-connected. Hou et al. [4] developed a heuristic algorithm, named SPINDS, to prolong the network lifetime and mitigate the network geometric deficiencies in a two-tiered network. Tang et al. [23] presented two polynomial time approximation algorithms to guarantee connectivity and fault-tolerance in a two-tiered WSN via deploying the minimum number of RNs. Wang et al. [24] applied heuristics to minimize the network device cost in a two-tiered WSN under the constraints of coverage, lifetime and connectivity. They considered two scenarios. The first is RNs with limited energy. The second is all nodes with limited energy. Zhang et al. [25] studied RNDP that ensured 2-connectivity in both single-tiered and two-tiered WSN by using heuristics. Xu et al. [5] discussed the impacts of random node deployment on connectivity and lifetime in a two-tiered WSN. Peiravi et al. [26] proposed a clustering method using a genetic algorithm in a homogeneous two-tiered WSN, optimizing the network lifetime with different delay values. Azharuddin and Jana [27] intended to minimize the number of RNs and maximize network connectivity by using a metaheuristic method-genetic algorithm in a two-tiered WSN. Chen et al. [28] considered converting RNDP into the minimum geometric disk cover problem and they proposed a linear time approximation algorithm for this problem. Hashim et al. [29] proposed an enhanced deployment algorithm based on Artificial Bee Colony to extend the lifetime of the WSN by optimizing the network parameters and constraining the total number of deployed RNs.

All of above approaches concern on RNs deployment with no constraints. It implies that the RNs can be deployed anywhere. However, in practice, there may be some physical limits on the RNs deployment. For single-tiered network, Misra et al. [30] deployed minimum RNs to a WSN under constraint that RNs were limited to be placed at a subset of candidate positions. The network connectivity is ensured at the meantime. By reaching out along this constrained approach, Misra et al. [8] ensure connectivity and survivability by deploying a minimum number of RNs in an energy-harvesting single-tiered WSN. The candidate locations with the energy harvesting potential are pre-specified. Perez et al. [31] employed an MO algorithm to optimize both the energy cost and the number of routers in a single-tiered WSN. Nigam et al. [32] proposed a branch-and-cut algorithm to place the minimum number of RNs at a subset of candidate locations in a single-tiered WSN, ensuring the sensors communicated with the sink node within a pre-specified delay bound. Ozkan and Ermis [33] studied how to ensure the connectivity of the network by deploying RNs in a single-tiered heterogenous WSN. They converted the problem of constrained RN placement into a mixed-integer programming model. Two meta-heuristics Genetic Algorithm and Simulated Annealing were employed to solve this model with the objective of finding the minimum number and reasonable position of the RNs. Yang et al. [34] used heuristics to deploy the minimum number of RNs into a two-tiered WSN under both connectivity and survivability requirements.

3. System models and Problem formulation

3.1 Network model

We assume that the network on a two-dimension sensing field of size $l_x \times l_y$ ($l_x > 0$, $l_y > 0$) is composed of three types of devices: one BS, N_s SNs and N_r RNs, as depicted in **Fig. 1**. Only SNs are powered by batteries and the reminders have unlimited power supply. SNs with sensitivity radius r_s sense the environment, generate data, and immediately transmit the data to the BS simultaneously, starting at time $t = 1 \in \tau$ (set of time periods, $\tau = \{0, 1, 2, \cdots\}$). The BS is the sole connection point of the WSN to the outside. Any two devices can communicate if they are at a Euclidean distance lower than r_c. All SNs are in the same energy charge initially. If an SN runs out of energy, it cannot be linked. We consider S-MAC as the medium access protocol for simplicity [35]. The routing protocol based on the shortest path is provided by SPFA algorithm [36]. To reduce interference in practice, the minimum Euclidean distance d for any pair of devices is more than d_{min}.

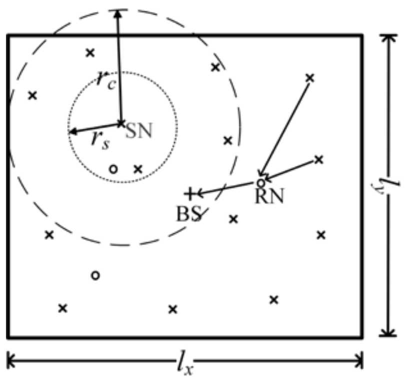


Fig. 1. Wireless sensor network model

3.2 Energy model

The energy model employed in this investigation took the most energy-consuming task of the packets sending into consideration, and neglected receiving, processing and sensing tasks [37].

At time $t > 0$, the energy expenditure suffered by a sensor i is

$$EP_i(t) = P_i(t) \cdot amp \cdot k \cdot \| i - \varsigma_i^s(t) \|^{\alpha} \cdot \beta \quad , \tag{1}$$

where $P_i(t)$ is the number of information packets, *amp* is energy cost per bit of the power amplifier (*amp*>0), k is information packet size in bits, $\|\cdot\|$ is the Euclidean distance between two devices, $\varsigma_i^s(t)$ is the variable which provides the next device in the minimum path, α is path loss exponent ($\alpha \in [2,4]$) and β is the transmission quality parameter ($\beta > 0$). This equation simulates extra cost due to packet loss.

The residual energy of a sensor i is shown as

$$EL_i(t) = \begin{cases} EL_i(t-1) - EP_i(t) & if\ t > 0 \\ ie & if\ t = 0 \end{cases} , \tag{2}$$

where $EL_i(t)$ is the residual energy of sensor i, and *ie* is the initial energy of a sensor.

Base on the above energy expenditure process, we define the lifetime as the number of time units. The network stops working when the energy of any sensor turn to be zero. Thus, the lifetime of the network is

$$lt = \left| \{ t > 0 \in \tau \,/\, EL_i = 0 \} \right| \quad i \in S_s \ , \tag{3}$$

where S_s is the set of initial sensor coordinates and $|.|$ is the cardinal of the set.

3.3 Problem formulation

The average energy consumption (AEC) of the sensors f_{aec}, over the network lifetime, is formulated as

$$f_{aec} = \frac{\sum_{t=1}^{lt} \left(\sum_{i \in S_s(t)} EP_i(t) \right)}{N_s \cdot lt} \ , \tag{4}$$

where N_s is the number of initial sensors. It is the cardinal of S_s. $S_s(t)$ is the set of sensors coordinates with an energy charge more than 0 at time $t > 0 \in \tau$, $S_s(t) \subseteq S_s$.

The average network reliability (ANR) is f_{anr}, which presents the probability of the information transmitting from the sensor node to the sink. That is,

$$f_{anr} = \frac{1}{N_s} \sum_{i \in S_s} r_i \ , \tag{5}$$

where $f_{anr} \in [0,1]$ and r_i is the reliability of the sensor i, defined in [38] as

$$r_i = 1 - \prod_{k=1}^{edp_i^s} \left(1 - (1 - err)^{h_k^{i,s}}\right) , \tag{6}$$

where edp_i^s is the number of disjoint paths between *i* and the sink node, $h_k^{i,s}$ is the number of hops in *k*th disjoint path between both devices, and *err* is the local channel error. The disjoint paths are calculated through max-flow method proposed by Ford and Fulkerson [39].

This way, we define CRNDP as an MO problem, where given a traditional wireless sensor network, i.e. a set of sensors $S_s(t)$ and a BS, the objective is to deploy a set of RNs S_r to

$$\min(f_{aec}), \max(f_{anr}) . \tag{7}$$

subject to

$$d(m,n) \geq d_{\min} \tag{8}$$

and

$$\forall z \in S_r, \quad z = (x,y) / x \in [0,l_x], y \in [0,l_y] , \tag{9}$$

where *m* and *n* are any two in all nodes (including SNs, RNs and the BS), S_r is the set of router coordinates.

4. Methodology

In this section, we introduce a novel MO metaheuristic, MOABCLLS, to solve the CRNDP. MOABCLLS is developed from an improved Artificial Bee Colony algorithm. Before a detailed description of our proposed algorithm, the presentation of encoding in this algorithm is defined as every individual is composed of N_r components. A component is the two-dimensional coordinate of an RN, i.e. $r_i = (x_i, y_i), r_i \in S_r, x_i \in [0,l_x], y_i \in [0,l_y]$.

4.1 ABC with a linear local search (ABCLLS)

The ABC [40] algorithm belonging to Swarm Intelligence is a random optimization algorithm and detailed in reference [41]. Our main concern for this algorithm is how to generate a better candidate solution according to the search equation. The formula for choosing a candidate solution is

$$v_{ij} = x_{ij} + \phi_{ij}(x_{ij} - x_{kj}) \tag{10}$$

where $j \in \{1,2,\cdots,N_r\}$ is a randomly selected index for the candidate solution v_i, the current solution x_i and the neighbor solution x_k ($k \neq i$). ϕ_i is a random number between [-1,1].

Due to ϕ_i is a random number between [-1, 1], the ABC algorithm has a good global search ability but ignored the local search ability, which leads to slow convergence speed. In literature [42], they proposed GABC with a new searching strategy:

$$v_{ij} = x_{ij} + \phi_{ij}(x_{ij} - x_{kj}) + \varphi_{ij}(y_j - x_{ij}) \tag{11}$$

where k, j, ϕ_{ij} are yielded in the same manner as in Eq. (10), φ_{ij} is a uniform random number between [0,C], where C is a nonnegative constant. y_j is the *j*th element of the global best solution. The algorithm improves the local search ability without harming the global search ability in a degree.

Then, we rewrite Eq. (11) as

$$\begin{cases} v_{ij} = x_{ij} + v_{ij}^{(1)} + v_{ij}^{(2)} \\ v_{ij}^{(1)} = \phi_{ij}(x_{ij} - x_{kj}) \\ v_{ij}^{(2)} = \varphi_{ij}(y_j - x_{ij}) \end{cases} \tag{12}$$

In Eq. (12), $v_{ij}^{(1)}$ stands for global search and $v_{ij}^{(2)}$ aims at local search. It is important to note that this algorithm assumes a random local search ability throughout optimization process. However, according to a general optimization process, the global search ability is more important than local search ability in the early stage, which is helpful for the solution space sufficiently searched and avoiding the algorithm trapping into local optimum. While, in the late stage of the optimization process, the local search ability becomes more important than the global search ability, because better local search ability means a higher speed of convergence. Based on the previous analysis, we propose a method which improves GABC by utilizing a linear local search strategy. A linear parameter is defined as follows:

$$l_f = \mathrm{C} * Curitr / Maxitr \tag{13}$$

where *Maxitr* is the maximum number of iterations. *Curitr* is the current number of iterations and C is a positive constant [42]. One can note that l_f increases from 0 to C as the number of iteration increases. This increases the weight of local search linearly.

A new ABC variant ABCLLS is proposed with the search equation given in Eq. (14), aiming to obtain the more accurate results and the higher convergence speed.

$$v_{ij} = x_{ij} + \phi_{ij}(x_{ij} - x_{kj}) + l_f(y_j - x_{ij}) . \tag{14}$$

4.2 MOABCLLS for CRNDP

CRNDP is an MO problem, while ABCLLS is a single-objective algorithm. Therefore, MOABCLLS is developed, which is an MO modification of ABCLLS. An outline of the algorithm is shown in **Algorithm 1**.

We assume a colony P_t of size N. The number of employed bees is as same as the number of onlooker bees, being $N/2$. Initially, random solutions of the optimization problem are generated and each bee of the colony is assigned one (line 1). At each iteration and so long as the stop criterion is not reached (line 2), the colony is divided into two groups: E_t and O_t, i.e. the group of employed bees and onlooker bees, respectively (line 3). Next, each $eB_i \in E_t$ produces a candidate solution in its surroundings, assuming a randomly selected individual $eB_k \in E_t$ (lines 5 and 6), that is

$$v_{ij} = eB_{ij} + \phi_{ij}(eB_{ij} - eB_{kj}) + l_f(G_{\tilde{i}j} - eB_{ij})\ \forall j \in [1, \cdots, N_r] , \tag{15}$$

where ϕ_{ij} is a random value in the interval [-1, 1], being eB_{ij} and eB_{kj} the routers in the jth component of eB_i and eB_k, respectively. G is so far the optimal front of the solutions, which are some best non-dominated solutions. $G_{\tilde{i}}$ that is closest to eB_i in Euclidean distance is selected from G to promote the convergence speed and effectiveness of the MO algorithm as y_j dose in Eq. (14) for single-objective algorithm. $G_{\tilde{i}j}$ is the jth element of $G_{\tilde{i}}$ and it represents a router's position in this approach. Then, a greedy selection between the new solution and the previous one is carried out. If new solution is dominated, then increase the attempt counter. While the new solution is not dominated, then execute **Algorithm 2** (line 7-12).

Onlooker bees take a probability-based selection process. As the ABCLLS is a single objective algorithm and the CRNDP is an MO problem, we consider the elitist crowded comparison operator $\geq_n$ defined in NSGA-II [10] to establish how good a food source is. The procedure of probability calculation refers to **Algorithm 3**. The formula in **Algorithm 3** means that the best food source of E_t is twice as likely as the second one. The second food source is as twice as the third one, and so on. Accordingly, each onlooker bee $oB_i \in O_t$ generates a new solution based on an employed forager eB_i, which is randomly selected according to these probabilities (line 14-18), that is

$$v_{ij} = oB_{ij} + \phi_{ij}(oB_{ij} - eB_{ij}) + l_f(G_{\tilde{i}j} - oB_{ij})\ \forall j \in [1,\cdots,N_r] \tag{16}$$

where oB_{ij} and eB_{ij} are the routers in the jth component of oB_i and eB_i, respectively. The onlooker bee takes the value via the greedy selection between both solutions (line 19-23).

If the food source of an employed bee or onlooker bee is exhausted, a solution is generated by the scout process. Thus, a solution $B_i \in P_{t+1}$ is randomly generated to the optimization problem. Then, the attempt counter of the food source is reset (line 28-33).

After the scout process, the optimal solutions in P_{t+1} will be saved to G if these solutions are not dominated by any solution of G. The solutions in G that are dominated by the ones in P_{t+1} will be eliminated (line 34). Notice that G only contains the global optimal solutions and its length is adjustable.

Algorithm 1. MOABCLLS

1. P_t=*initializePopulation*(N)
2. **while** not stop condition do
3. E_t, O_t=*selectGroupBees*(P_t)
4. **//****EMPLOYED BEE PHASE******
5.   **for** each $eB_i \in E_t$ **do**
6. v=*generateEmployedBee*(eB_i, $G_{\tilde{i}}$)   //Use Eq. (15)
7.   **if** eB_i dominates v then
8. eB_i .*attemptCounter++*
9.   **else**
10.   *candidateDeals* (v, eB_i)   //Use Algorithm 2
11.   **end if**
12.   **end for**

P_r=*calculateProbabilites*(E_t) //Use Algorithm 3
//******ONLOOKER BEE PHASE******
for each $oB_i \in O_t$ **do**
Produce random number r form the range [0,1]
if $p_i \le r$ **then** // $p_i \le P_r$
v=*generateOnlookerBee*(eB_i , oB_i , $G_{\tilde{i}}$) //Use Eq. (16) and $eB_i \in E_t$
if oB_i dominates v **then**
oB_i .*attemptCounter*++
else
candidateDeals (v, oB_i) //Use Algorithm 2
end if
end if
end for
$P_{t+1} \leftarrow E_t \cup O_t$
//******SCOUT BEE PHASE******
for each $B_i \in P_{t+1}$ do
if B_i .*attemptCounter*>*limit* **then**
B_i =*generateScoutBee*()
B_i .*attemptCounter*=0
end if
end for
G←*memorizeBestSource*(P_{t+1})
$t = t + 1$
end while

Algorithm 2. Candidate deals
Input(v, B_i)
if v dominates B_i **then**
$B_i = v$
else
if any one of G dose not dominate v **then**
$G \leftarrow G \cup \{v\}$
Eliminate the solutions of G dominated by v.
end if
end if

Algorithm 3. Probabilities calculation
Sort E_t by the operator $\ge_n$
for each $e_i \in E_t$ **do**
$p_i = \frac{2^{|E_t|-i}}{2^{|E_t|+1}-1}$ // $|E_t|$ is the cardinal of E_t and $p_i \in P_r$.
end for

Let us now take a view at the convergence of the search process and the complexity of one iteration of the algorithm. As for the convergence, it can be ensured that the optimal front G converge to the Pareto front with probability one based on the theories from [43]. For complexity, the basic operations being performed and the worst case complexities associated with are as follows:

1. The procedure of employed bee or onlooker bee is $O(N_r N|G|)$,
2. Probabilities calculation is $O(N_r N^2)$,
3. The procedure of scout bee is $O(N_r N)$, and
4. The best source memorization is $O(N_r N|G|)$.

$|G|$ is the cardinal of G. As can be seen, the overall complexity of the above algorithm is $O(\max(N_r N^2, N_r N|G|))$.

5. Experimental strategy

Other MO algorithms are introduced for comparison to evaluate the effectiveness of the MOABCLLS in this section. Thus, several questions arise about how to implement this comparison: What quantitative measures should be employed to present the quality of the results so that the metaheuristics used to CRNDP can be compared in a meaningful way? What is the outcome of an MO metaheuristics regarding a set of runs? What data set should be utilized to test our problem and algorithms? How can the parameters of the metaheuristics, regarding the CRNDP, be set appropriately? We treat these problems in the following.

5.1 Algorithms for comparison

First, two versions of MO ABC are assumed. We employ the MO strategies used in MOABCLLS to ABC and GABC respectively, obtaining MO ABC (MOABC) and MO GABC (MOGABC). They have the same complexity as MOABCLLS. Then, two standard genetic algorithms belonging to sub branch of EAs, NSGA-II [10] and SPEA2 [11], are involved in comparison. NSAG-II is an improved version of the previous NSGA and SPEA2 is a revised version of the previous SPEA. These two algorithms have same complexity $O(N_r N^2)$ and both have been employed to solve a wide variety of MO optimization problems, showing promising performance. In addition, we consider a comparison between MOABCLLS and two MO trajectory algorithms, MOVNS and MOVNSwP, employed in the reference [9]. MOVNS is an MO version of the variable neighborhood search algorithm, which is characterized by following a trajectory in the search space. MOVNSwP is an extension of the MOVNS by adding perturbation mechanism, in which the aim of the perturbation mechanism is to avoid local minima. The complexities of these two algorithms are the same, being $O(N_r N^3)$. They show good performance in many optimization problems and are employed to solve RNs deployment problem in [9]. They are close to our approach. However, the problem we focus is constrained and has different optimization objectives. Moreover, we propose a new MO algorithm to deal with the problem.

The complexity of our proposed scheme is same as MOABC and MOGABC, lower than MOVNS and MOVNSwP. For NSAG-II and SPEA2, MOABCLLS keeps the same complexity before the size of G reaches the value of N, but it gets more complex when the size of G increases beyond the value of N. However, our approach shows a better performance than other algorithms, which will be verified in Section 6.

5.2 Performance measures

We employ two complementary measures to evaluate the tradeoff fronts produced by the metaheuristics to CRNDP.

Hypervolume (HV): This metric calculates the portion of the objective space covered by members of a nondominated set of solutions *F*. Mathematically, for each solution $i \in F$, a hypercube μ_i is constructed with a reference point ω and the solution *i* as the diagonal corners of the hypercube. Then, the hypervolume of *F* is the union of all its hypercubes. That is,

$$HV = \text{volume}(\bigcup_{i=1}^{|F|} \mu_i) \ . \tag{17}$$

Coverage of two sets(CTS): This metric is based on the dominance concept. Let X_1, X_2 be two sets of phenotype decision vectors. The function CTS maps the ordered pair (X_1, X_2) to the interval [0, 1]. That is,

$$C(X_1, X_2) = \frac{|\{x_2 \in X_2 ; \exists x_1 \in X_1 : x_1 \succeq x_2\}|}{|X_2|} \tag{18}$$

If all individuals in X_1 dominate or are equal to all individuals in X_2, then $C(X_1, X_2)$=1 by definition. On the contrary, $C(X_1, X_2)$=0.

5.3 Strategy and data set used

For each algorithm and experiment, 30 independent runs is performed, which is a widely accepted value to reach statistical conclusions [44]. As stop condition, we assume several criteria in order to study the convergence of the algorithms. Accordingly, we assume 50000, 100000,200000, 300000, 400000 and 500000 evaluations.

The data set we used in this paper is proposed in the reference [9]. We configure this common framework for studying the CRNDP. The scenarios in our experiment are composed of three sizes: 100 × 100, 200 × 200, 300 × 300.

This WSN model considers some parameters as stated previously. It is assumed that α = 2, β = 1, *k* = 128 KB, r_s = 15m, r_c=30m and *amp* = 100 $pJ/bit/m^2$, from [45] and d_{min} is 0.1m in the model. Adding more RNs means the more network cost. Thus, we do not include more than 20% of these devices regarding the number of sensors as [9] did. The number of routers, which are added to optimize the network, is shown in **Table 1**. In addition, **Table 1** also indicates the value of the fitness functions without including RNs (N_r=0).

Table 1. Experimental cases considered

Test	N_s	Fitness(N_r =0)		Reference f_{aec}		Reference $\overline{f_{anr}}$		Experiment cases(N_r)
		f_{aec}	$\overline{f_{anr}}$	best	worst	best	worst	
100×100	15	0.1036	0.02851	0.054	0.098	0.0051	0.027	2,3
200×200	57	0.2288	0.06701	0.10	0.23	0.026	0.067	2,4,6,9
300×300	128	0.3488	0.1421	0.13	0.35	0.045	0.15	6,12,18,24

5.4 Parameters settings

Before conducting experiments, we consider the other metaheuristics for comparison using the same encoding as MOABCLLS used and all the parameters of the algorithms were sufficiently configured. As for population size, a same habitual value of 40 individuals is assumed for all metaheuristics. After that, a same value of parameter *limit* is set for MOABC, MOGABC, and MOABCLLS. The value of the *limit* is the number of RNs multiplied by half of the population size. In MOGABC and MOABCLLS, the value of parameter C is 1.5 [42]. For the other algorithms, the ranges of parameters default values are given in **Table 2**, where Mutation and Crossover stand for the probability of the mutation and crossover, while N_{ns} represents the number of neighbor structures and d_s delimits the displacement. In order to find the best parameters configuration, each configuration of parameters, i.e. a pair of values of any one of mutation and any one of crossover, was conducted 30 independent runs, considering a reduced stop condition (10000 evaluations). The best HV metric is selected as the quality indicator to choose the best parameters configuration, which provides the best performance on average as the value selected shown in **Table 2**. In addition, the reference points assumed to calculate the HV are listed in **Table 1**, where the terms "best" and "worst" are the best and the worst value of a fitness function, respectively. These values were obtained experimentally.

Table 2. Values of Parameters selected about NSGA-II, SPEA2, MOVNS and MOVNSwP

Algorithm	Parameter	Selected	Range
NSGA-II	*Mutation*	0.1	[0.05,0.1,0.15,…0.95]
	Crossover	0.9	[0.05,0.1,0.15,…0.95]
SPEA2	*Mutation*	0.2	[0.05,0.1,0.15,…0.95]
	Crossover	0.8	[0.05,0.1,0.15,…0.95]
MOVNS	N_{ns}	6	[4,5,6,…,14]
	d_s	3	[1,2,3,4,5,6,7,8,9]
MOVNSwP	*Mutation*	0.15	[0.05,0.1,0.15,…0.95]
	N_{ns}	10	[4,5,6,…,14]
	d_s	5	[1,2,3,4,5,6,7,8,9]

6. Performance evaluation

The advantages provided by the addition of RNs has been analyzed in various literatures [4, 9, 17, 23, 30, 34]. Therefore, we chiefly address the CRNDP based on the data set with MO metaheuristics in this section. Regarding the simulation experiment, JDK 1.7 is employed to code the process. We acquire the optimization results for the industry parameters: AEC and ANR. Instead of showing the detailed optimization results of the industrial parameters, we pay attention to use some classical statistical methods for analyzing the quality of the solutions obtained, which could verify the effectiveness of our algorithm.

6.1 Analysis based on HV metric

Initially, we evaluate the quality of the MO algorithms based on the metric of HV. As shown in **Fig. 3** to **Fig. 5**, the data included in the figures are the average HV for each algorithm, test case and stop iteration. It is clear about the behavior and the differences among the algorithms over the number of iterations. In these figures, we notice that MOABCLLS has a good convergence rate. Moreover, it seems that MOABCLLS provides the best performance among the algorithms. Since our experiments are dealing with some stochastic analysis with MO metaheuristics to verify the effectiveness of our algorithm and we want to show some results with confidence, the following statistical analysis is employed to further analyze the results through this approach.

First, we consider the Kolmogorov–Smirnov–Lilliefor's [46] and Shapiro–Wilk's [47] tests in order to analyze whether the results come from a normal distribution. In this regarding, we have the following hypothesis: H_0: if results follow a normal distribution and H_1: the opposite. We consider in this work a confidence level of 95% (i.e. *p*-value under 0.05). For all the cases, we get the *p*-values more than 0.05. Therefore, the assumption of H_0 fails. Thus, the results do not follow a Gaussian distribution and the samples are independent. The median is written simply as *M* in this experiment.

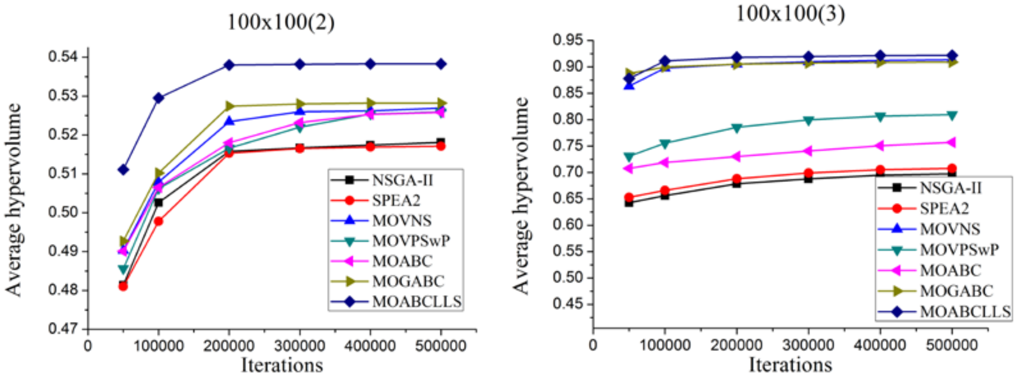


Fig. 3. Average HV of 100x100 tests

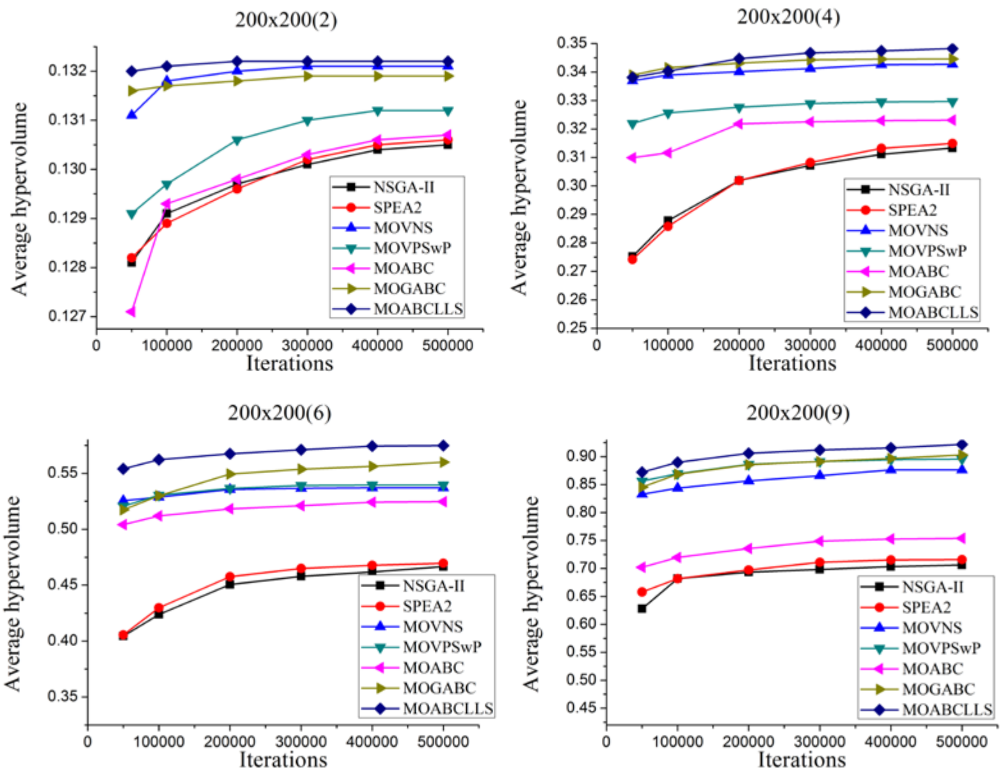


Fig. 4. Average HV of 200x200 tests

Next, we consider the Wilcoxon-Mann-Whitney's [48] test to study if some significant differences are shown among these algorithms. In this test, we have following hypotheses: H_0: M_i is smaller than M_j or equal to M_j and H_1: M_i is bigger than M_j (i=a, b, c, d, e, f, g j=b, c, d, e, f, g a is NSGA-II, b is SPEA2, c is MOVNS, d is MOVNSwP, e is MOABC, f is MOGABC and g is MOABCLLS). We consider the p-values with a significance level of 0.05. Based on this test method, we compare MOABCLLS with other four other algorithms for figuring out which one provides the best significant performance with each stop iteration and test case.

Along with the above statistical procedure, the results about the percentage of test cases are depicted in **Fig. 6.**

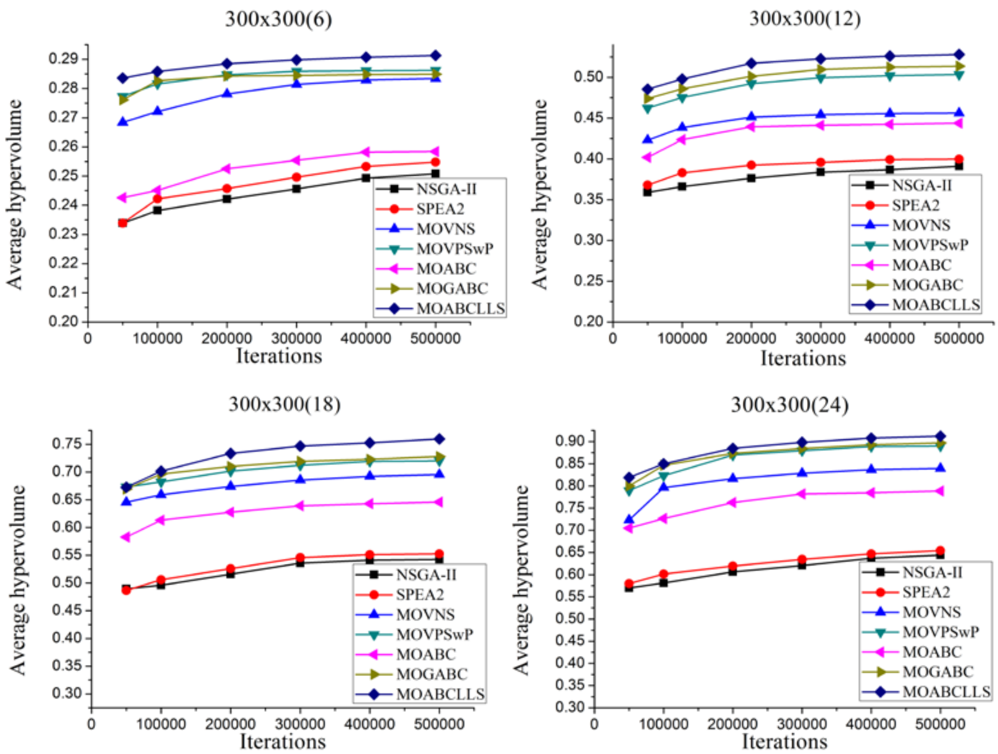
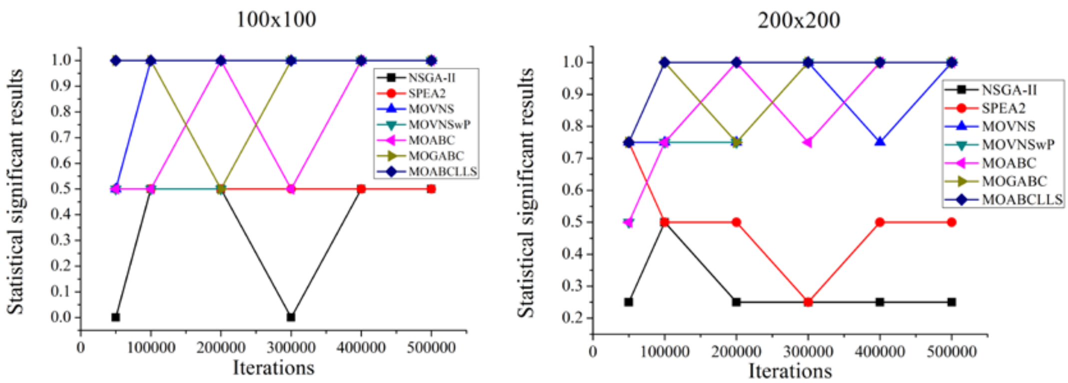


Fig. 5. Average HV of 300x300 tests



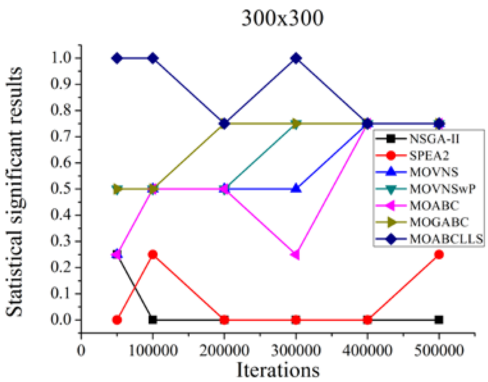


Fig. 6. Analysis of the *p*-values obtained considering the HV metric

For 100 × 100 scenario, MOABCLLS provides the best behavior for all iterations. Then, for 200 × 200 scenario, MOABCLLS shows the better performance for reduced stop evaluations, but MOGABC and MOVNSwP perform as good as MOABCLLS do for a high number of iterations. Finally, we consider 300 × 300 scenario, MOABCLLS provides better behavior for reduced stop conditions. The performance of MOABCLLS is lightly weakened for high stop conditions. And MOGABC, MOVNSwP, MOVNS and MOABC show an increasing performance for high iterations. In addition, SPEA2 show better performance than NSGA-II in this simulation and for all test cases comprehensively. Regarding this analysis, we conclude that MOABCLLS is the best algorithm for CRNDP even with the enlarged terrain and the more added nodes. The order of the other algorithms is MOGABC, MOVNSwP, MOVNS, MOABC, SPEA2 and NSGA-II.

Table 3. Average CTS among the algorithms for 50000 iterations

NSGA-II							
l_x×l_y	SPEA2	MOVNS	MOVNSwP	MOABC	MOGABC	MOABCLLS	Average
100×100	32.44%	0.65%	0.96%	1.23%	0.09%	0.23%	5.93%
200×200	37.24%	16.11%	15.27%	39.62%	13.55%	1.43%	20.54%
300×300	33.15%	10.17%	8.43%	32.30%	4.73%	4.78%	15.59%
Average	34.28%	8.98%	8.22%	24.38%	6.12%	2.15%	
SPEA2							
l_x×l_y	NSGA-II	MOVNS	MOVNSwP	MOABC	MOGABC	MOABCLLS	Average
100×100	53.02%	3.07%	6.95%	7.18%	3.19%	2.69%	12.68%
200×200	46.51%	19.64%	14.32%	42.42%	15.49%	2.31%	23.45%
300×300	58.52%	20.79%	17.88%	37.11%	17.36%	17.55%	28.20%
Average	52.68%	14.50%	13.05%	28.90%	12.01%	7.52%	
MOVNS							
l_x×l_y	NSGA-II	SPEA2	MOVNSwP	MOABC	MOGABC	MOABCLLS	Average
100×100	95.03%	88.29%	42.38%	53.93%	51.87%	21.86%	58.89%
200×200	53.83%	49.74%	39.18%	54.28%	41.67%	8.37%	41.18%
300×300	56.71%	46.89%	31.09%	50.18%	39.00%	29.54%	42.24%
Average	68.52%	61.64%	37.55%	52.80%	44.18%	19.92%	
MOVNSwP							
l_x×l_y	NSGA-II	SPEA2	MOVNS	MOABC	MOGABC	MOABCLLS	Average
100×100	96.32%	81.16%	37.76%	41.32%	39.54%	15.62%	51.95%
200×200	62.15%	56.54%	41.06%	36.91%	51.25%	12.55%	43.41%
300×300	69.61%	52.35%	35.94%	40.85%	37.62%	31.56%	44.66%
Average	76.03%	63.35%	38.25%	39.69%	42.80%	19.91%	

MOABC							
$l_x \times l_y$	NSGA-II	SPEA2	MOVNS	MOVNSwP	MOGABC	MOABCLLS	Average
100×100	86.18%	63.83%	42.22%	48.54%	71.25%	17.71%	54.96%
200×200	32.06%	25.31%	24.98%	21.62%	23.91%	5.45%	22.22%
300×300	72.60%	57.95%	45.01%	39.30%	46.30%	27.95%	48.19%
Average	63.61%	49.03%	37.40%	36.49%	47.15%	17.04%	
MOGABC							
$l_x \times l_y$	NSGA-II	SPEA2	MOVNS	MOVNSwP	MOABC	MOABCLLS	Average
100×100	93.41%	86.32%	40.02%	47.28%	54.69%	32.36%	59.01%
200×200	59.63%	56.39%	54.65%	42.19%	61.58%	4.06%	46.42%
300×300	89.89%	76.12%	41.73%	49.52%	34.92%	26.32%	53.08%
Average	80.98%	72.94%	45.47%	46.33%	50.40%	20.91%	
MOABCLLS							
$l_x \times l_y$	NSGA-II	SPEA2	MOVNS	MOVNSwP	MOABC	MOGABC	Average
100×100	99.46%	99.23%	75.38%	83.39%	97.90%	96.10%	**91.91%**
200×200	87.97%	86.09%	86.14%	85.21%	85.35%	85.09%	**85.98%**
300×300	75.92%	93.21%	49.68%	43.36%	43.93%	48.84%	**59.16%**
Average	87.78%	92.84%	70.40%	70.65%	75.73%	76.68%	

6.2 Analysis based on CTS

In addition to HV, the CTS metric is employed to analyze the quality of the MOABCLLS algorithm comparing to other algorithms, according to the size of the area. The values of this metric are calculated by considering the median Pareto fronts from previous 30 samples. The CTS results are shown for 50000 and 500000 iterations. The results are listed in **Table 3** and **Table 4**. Higher average values for each scenario have a gray background. Accordingly, MOABCLLS is the best algorithm for 50000 and 500000 evaluations.

Table 4. Average CTS among the algorithms for 500000 iterations

NSGA-II							
$l_x \times l_y$	SPEA2	MOVNS	MOVNSwP	MOABC	MOGABC	MOABCLLS	Average
100×100	67.40%	2.45%	8.12%	17.69%	1.63%	5.41%	17.12%
200×200	21.39%	28.46%	24.94%	31.84%	27.60%	8.32%	23.76%
300×300	15.46%	21.92%	19.49%	30.12%	6.34%	15.97%	18.22%
Average	34.75%	17.61%	17.52%	26.55%	11.86%	9.90%	
SPEA2							
$l_x \times l_y$	NSGA-II	MOVNS	MOVNSwP	MOABC	MOGABC	MOABCLLS	Average
100×100	78.90%	7.36%	9.75%	13.83%	0.73%	3.96%	19.09%
200×200	37.03%	38.49%	25.59%	41.52%	42.20%	8.13%	32.16%
300×300	12.41%	35.21%	30.63%	4.23%	61.50%	15.99%	26.66%
Average	42.78%	27.02%	21.99%	19.86%	34.81%	9.36%	
MOVNS							
$l_x \times l_y$	NSGA-II	SPEA2	MOVNSwP	MOABC	MOGABC	MOABCLLS	Average
100×100	83.84%	86.52%	57.51%	60.53%	41.27%	36.52%	61.03%
200×200	42.48%	39.33%	43.24%	54.29%	40.58%	22.46%	40.40%
300×300	65.99%	68.41%	46.66%	66.62%	72.12%	68.55%	64.73%
Average	64.10%	64.75%	49.14%	60.48%	51.32%	42.51%	
MOVNSwP							
$l_x \times l_y$	NSGA-II	SPEA2	MOVNS	MOABC	MOGABC	MOABCLLS	Average
100×100	77.59%	68.60%	50.13%	49.10%	37.44%	28.13%	51.83%
200×200	48.04%	47.08%	51.47%	61.66%	43.02%	29.77%	46.84%
300×300	74.85%	71.27%	48.28%	69.54%	77.19%	73.40%	69.09%
Average	66.83%	62.32%	49.96%	60.10%	52.55%	43.77%	

MOABC							
$l_x \times l_y$	NSGA-II	SPEA2	MOVNS	MOVNSwP	MOGABC	MOABCLLS	Average
100×100	54.51%	56.44%	18.52%	27.72%	15.19%	16.47%	31.48%
200×200	54.96%	52.11%	39.98%	25.9%	37.98%	4.41%	35.89%
300×300	62.18%	12.11%	43.4%	40.82%	76.85%	39.07%	45.74%
Average	57.22%	40.22%	33.97%	31.48%	43.34%	19.98%	
MOGABC							
$l_x \times l_y$	NSGA-II	SPEA2	MOVNS	MOVNSwP	MOABC	MOABCLLS	Average
100×100	81.76%	83.67%	37.04%	42.81%	52.61%	28.95%	54.47%
200×200	47.65%	43.17%	42.11%	35.35%	55.62%	8.62%	38.75%
300×300	68.94%	74.32%	77.64%	73.24%	80.73%	72.19%	74.51%
Average	66.12%	67.05%	52.26%	50.47%	62.99%	36.59%	
MOABCLLS							
$l_x \times l_y$	NSGA-II	SPEA2	MOVNS	MOVNSwP	MOABC	MOGABC	Average
100×100	94.36%	92.38%	59.84%	66.52%	62.13%	51.65%	**71.15%**
200×200	85.84%	86.48%	83.48%	76.33%	79.85%	77.30%	**81.55%**
300×300	78.83%	85.33%	77.99%	64.41%	81.51%	66.57%	**75.77%**
Average	86.34%	88.06%	73.77%	69.09%	74.50%	65.17%	

7. Conclusion

In this paper, we consider how to solve CRNDP with the objective of optimizing two important factors in industry: AEC of SNs and ANR. CRNPD is an NP-hard optimization problem proved in several literatures. Metaheuristics usually show good performance on solving this kind of problems by providing a set of trade-off solutions, which provides the network designer more possibilities to design the network. In this case, we proposed a novel metaheuristic, namely MOABCLLS, to solve this problem. In order to verify the effectiveness of the algorithm, we present a comparison among MOABCLLS and a wide range of other MO metaheuristics including MOABC, MOGABC, NSGA-II, SPEA2, MOVNS and MOVNSwP. These metaheuristics are employed to optimize a data set obtained from the literature, assuming three different scenarios. The results obtained are analyzed considering two standard MO metrics: HV and CTS, through a widely accepted statistical methodology. The simulation results show that MOABCLLS performs well on solving CRNDP and provides better performance than the other algorithms for all cases in this investigation.

As part of our future work, we intend to involve other key fitness functions and more realistic constraints in the industry into our MO calculations. Conducting our experiments with some standard simulation tools, such as OMNET++ and NS2, is also in our plan.

References

[1] J. Yick, B. Mukherjee, and D. Ghosal, "Wireless sensor network survey," *Computer Networks*, vol. 52, pp. 2292-2330, Aug., 2008. Article (CrossRef Link)

[2] L. Wei, Y. W. Yang, Z. Wei, and W. Lei, "Practical Node Deployment Scheme Based on Virtual Force for Wireless Sensor Networks in Complex Environment," *KSII Transactions on Internet and Information Systems*, vol. 9, pp. 990-1013, Mar., 2015. Article (CrossRef Link)

[3] S. Vikrant, R. B. Patel, H. S. Bhadauria, and D. Prasad, "Policy for planned placement of sensor nodes in large scale wireless sensor network," *KSII Transactions on Internet and Information Systems*, vol.10, pp. 3213-3230, Jul., 2016. Article (CrossRef Link)

[4] Y. T. Hou, Y. Shi, H. D. Sherali, and S. F. Midkiff, "On energy provisioning and relay node placement for wireless sensor networks," *IEEE Transactions on Wireless Communications*, vol. 4,

pp. 2579-2590, Sep., 2005. Article (CrossRef Link)

[5] K. Xu, H. Hassanein, G. Takahara, and Q. Wang, "Relay node deployment strategies in heterogeneous wireless sensor networks," *IEEE Transactions on Mobile Computing*, vol. 9, pp. 145-159, 2010. Article (CrossRef Link)

[6] D. Satria, D. Park, and M. Jo, "Recovery for overloaded mobile edge computing," *Future Generation Computer Systems*, vol. 70, pp. 138-147, May, 2016. Article (CrossRef Link)

[7] X. Z. Cheng, B. Narahari, R. Simha, M. X. Y. Cheng, and D. Liu, "Strong minimum energy topology in wireless sensor networks: NP-completeness and heuristics," *IEEE Transactions on Mobile Computing*, vol. 2, pp. 248-256, Jul.-Sep., 2003. Article (CrossRef Link)

[8] S. Misra, N. E. Majd, and H. Huang, "Approximation Algorithms for Constrained Relay Node Placement in Energy Harvesting Wireless Sensor Networks," *IEEE Transactions on Computers*, vol. 63, pp. 2933-2947, Dec., 2014. Article (CrossRef Link)

[9] J. M. Lanza-Gutierrez and J. A. Gomez-Pulido, "Studying the multiobjective variable neighbourhood search algorithm when solving the relay node placement problem in Wireless Sensor Networks," *Soft Computing*, vol. 20, pp. 67-68, 2016. Article (CrossRef Link)

[10] K. Deb, A. Pratap, S. Agarwal, and T. Meyarivan, "A fast and elitist multiobjective genetic algorithm: NSGA-II," *IEEE Transactions on Evolutionary Computation*, vol. 6, pp. 182-197, Apr., 2002. Article (CrossRef Link)

[11] E. Zitzler, M. Laumanns, and L. Thiele, "SPEA2: Improving the strength Pareto evolutionary algorithm," *Technical report, Computer Engineering and Networks Laboratory (TIK), ETH Zurich*, 2001.

[12] E. Zitzler, "Evolutionary algorithms for multiobjective optimization: Methods and applications," *Doctoral thesis ETH NO. 13398, Zurich: Swiss Federal Institute of Technology (ETH), Aachen, Germany: Shaker Verlag*, 1999.

[13] E. Zitzler and L. Thiele, "Multiobjective evolutionary algorithms: a comparative case study and the strength Pareto approach," *IEEE Transactions on Evolutionary Computation*, vol. 3, pp. 257-271, 1999. Article (CrossRef Link)

[14] X. Han, X. Cao, E. L. Lloyd, and C.-C. Shen, "Fault-tolerant relay node placement in heterogeneous wireless sensor networks," *IEEE Transactions on Mobile Computing*, vol. 9, pp. 643-656, 2010. Article (CrossRef Link)

[15] E. L. Lloyd and G. Xue, "Relay node placement in wireless sensor networks," *IEEE Transactions on Computers*, vol. 56, pp. 134-138, Jan., 2007. Article (CrossRef Link)

[16] C. Zhao and P. Chen, "Particle swarm optimization for optimal deployment of relay nodes in hybrid sensor networks," in *Proc. of IEEE Congress on Evolutionary Computation*, pp. 3316-3320, 2007. Article (CrossRef Link)

[17] X. Cheng, D.-Z. Du, L. Wang, and B. Xu, "Relay sensor placement in wireless sensor networks," *Wireless Networks*, vol. 14, pp. 347-355, 2008. Article (CrossRef Link)

[18] X. Han, X. Cao, E. L. Lloyd, and C.-C. Shen, "Fault-Tolerant Relay Node Placement in Heterogeneous Wireless Sensor Networks," *IEEE Transactions on Mobile Computing*, vol. 9, pp. 643-656, May, 2010. Article (CrossRef Link)

[19] V. Ranga, M. Dave, and A. K. Verma, "Relay node placement to heal partitioned wireless sensor networks," *Computers & Electrical Engineering*, vol. 48, pp. 371-388, Nov., 2015. Article (CrossRef Link)

[20] T. T. Truong, K. N. Brown, and C. J. Sreenan, "Multi-objective hierarchical algorithms for restoring Wireless Sensor Network connectivity in known environments," *Ad Hoc Networks*, vol. 33, pp. 190-208, Oct., 2015. Article (CrossRef Link)

[21] B. Hao, J. Tang, and G. Xue, "Fault-tolerant relay node placement in wireless sensor networks: formulation and approximation," in *Proc. of High Performance Switching and Routing*, pp. 246-250, 2004. Article (CrossRef Link)

[22] H. Liu, P.-J. Wan, and X. Jia, "Fault-tolerant relay node placement in wireless sensor networks," in *Proc. of Computing and combinatorics*, pp. 230-239, 2005. Article (CrossRef Link)

[23] J. Tang, B. Hao, and A. Sen, "Relay node placement in large scale wireless sensor networks," *Computer communications*, vol. 29, pp. 490-501, 2006. Article (CrossRef Link)

[24] Q. Wang, K. Xu, G. Takahara, and H. Hassanein, "Transactions papers-device placement for heterogeneous wireless sensor networks: Minimum cost with lifetime constraints," *IEEE Transactions on Wireless Communications*, vol. 6, pp. 2444-2453, 2007. Article (CrossRef Link)
[25] W. Zhang, G. Xue, and S. Misra, "Fault-tolerant relay node placement in wireless sensor networks: Problems and algorithms," in *Proc. of 26th IEEE International Conference on Computer Communications*, pp. 1649-1657, 2007. Article (CrossRef Link)
[26] A. Peiravi, H. R. Mashhadi, and S. Hamed Javadi, "An optimal energy-efficient clustering method in wireless sensor networks using multi-objective genetic algorithm," *International Journal of Communication Systems*, vol. 26, pp. 114-126, 2013. Article (CrossRef Link)
[27] M. Azharuddin and P. K. Jana, "A GA-based approach for fault tolerant relay node placement in wireless sensor networks," in *Proc. of Third International Conference on Computer, Communication, Control and Information Technology (C3IT)*, pp. 1-6, 2015. Article (CrossRef Link)
[28] C. C. Chen, C. Y. Chang, and P. Y. Chen, "Linear Time Approximation Algorithms for the Relay Node Placement Problem in Wireless Sensor Networks with Hexagon Tessellation," *Journal of Sensors*, 2015. Article (CrossRef Link)
[29] H. A. Hashim, B. O. Ayinde, and M. A. Abido, "Optimal placement of relay nodes in wireless sensor network using artificial bee colony algorithm," *Journal of Network and Computer Applications*, vol. 64, pp. 239-248, Apr., 2016. Article (CrossRef Link)
[30] S. Misra, N. E. Majd, and H. Huang, "Constrained relay node placement in energy harvesting wireless sensor networks," in *Proc. of IEEE 8th International Conference on Mobile Adhoc and Sensor Systems (MASS)*, pp. 25-34, 2011. Article (CrossRef Link)
[31] A. J. Perez, M. Labrador, and P. M. Wightman, "A multiobjective approach to the relay placement problem in wsns," in *Proc. of Wireless Communications and Networking Conference (WCNC)*, pp. 475-480, 2011. Article (CrossRef Link)
[32] A. Nigam and Y. K. Agarwal, "Optimal relay node placement in delay constrained wireless sensor network design," *European Journal of Operational Research*, vol. 233, pp. 220-233, 2014. Article (CrossRef Link)
[33] O. Ozkan and M. Ermis, "Nature-inspired relay node placement heuristics for wireless sensor networks," *Journal of Intelligent & Fuzzy Systems*, vol. 28, pp. 2801-2809, 2015. Article (CrossRef Link)
[34] D. Yang, S. Misra, X. Fang, G. Xue, and J. Zhang, "Two-Tiered Constrained Relay Node Placement in Wireless Sensor Networks: Computational Complexity and Efficient Approximations," *IEEE Transactions on Mobile Computing*, vol. 11, pp. 1399-1411, Aug., 2012. Article (CrossRef Link)
[35] W. Ye, J. Heidemann, and D. Estrin, "An energy-efficient MAC protocol for wireless sensor networks," in *Proc. of Twenty-First Annual Joint Conference of the IEEE Computer and Communications Societies. Proceedings.*, pp. 1567-1576, 2002. Article (CrossRef Link)
[36] D. Fanding, "A Faster Algorithm for Shortest-Ptath─SPFA," *Journal of Southwest Jiaotong University*, vol. 29, pp. 207-212, 1994.
[37] A. Konstantinidis, K. Yang, and Q. Zhang, "An Evolutionary Algorithm to a Multi-Objective Deployment and Power Assignment Problem in Wireless Sensor Networks," in *Proc. of IEEE Global Telecommunications Conference*, pp. 1-6, 2008. Article (CrossRef Link)
[38] B. Deb, S. Bhatnagar, and B. Nath, "ReInForM: Reliable information forwarding using multiple paths in sensor networks," in *Proc. of 28th Annual IEEE International Conference on Local Computer Networks, LCN'03. Proceedings.* , pp. 406-415, 2003. Article (CrossRef Link)
[39] L. R. Ford and D. R. Fulkerson, "Maximal flow through a network," *Canadian journal of Mathematics*, vol. 8, pp. 399-404, 1956.
[40] D. Karaboga, "An idea based on honey bee swarm for numerical optimization," *Technical report TR06, Computer Engineering Department, Engineering Faculty, Erciyes university*, 2005.
[41] D. Karaboga and B. Basturk, "A powerful and efficient algorithm for numerical function optimization: artificial bee colony (ABC) algorithm," *Journal of Global Optimization*, vol. 39, pp. 459-471, Nov., 2007. Article (CrossRef Link)

[42] G. Zhu and S. Kwong, "Gbest-guided artificial bee colony algorithm for numerical function optimization," *Applied Mathematics and Computation*, vol. 217, pp. 3166-3173, 2010. Article (CrossRef Link)
[43] C. C. Coello, G. B. Lamont, and D. A. Van Veldhuizen, "Evolutionary algorithms for solving multi-objective problems", *Springer Science & Business Media*, pp. 283-288, 2007.
[44] L. H. William and W. L. Hays, "Statistics: Probability, inference, and decision (2nd Edn)," *Holt, Rinehart and Winston*, pp. 203-270, 1975.
[45] A. Konstantinidis and K. Yang, "Multi-objective k-connected deployment and power assignment in wsns using a problem-specific constrained evolutionary algorithm based on decomposition," *Computer Communications*, vol. 34, pp. 83-98, 2011. Article (CrossRef Link)
[46] H. W. Lilliefors, "On the Kolmogorov-Smirnov test for normality with mean and variance unknown," *Journal of the American Statistical Association*, vol. 62, pp. 399-402, 1967. Article (CrossRef Link)
[47] S. Shaphiro and M. Wilk, "An analysis of variance test for normality," *Biometrika*, vol. 52, pp. 591-611, 1965. Article (CrossRef Link)
[48] H. B. Mann and D. R. Whitney, "On a test of whether one of two random variables is stochastically larger than the other," *The annals of mathematical statistics*, pp. 50-60, 1947. Article (CrossRef Link)

Wenjie Yu is pursuing Ph.D. degree in Mechanical Engineering from University of Electronic Science and Technology of China. He takes a successive postgraduate and doctoral program. He studied the mechanical and electronic engineering in Master's stage. His research interests include wireless sensor networks, multi-objective optimization and swarm intelligence.

Xunbo Li received his M.S. degree in Machinery Manufacturing from University of Electronic Science and Technology of China in 1991. He received his Ph.D. degree in Measurement and Instrument from University of Electronic Science and Technology of China in 2001. Currently, he is a Professor and Director of the Institute of Electrical and Mechanical Intelligent Control. His current research interests include intelligent measurement and control technology research and application of mechanical and electrical.

Xiang Li received B.S, M.S. and Ph.D. degree in mechanical engineering in 2005, 2007 and 2012 respectively. His research interests include ultrasonic nondestructive testing and its signal processing.

Zhi Zeng received B.S., M.S. and Ph.D. degree in materials science and engineering in 2004, 2006 and 2009 respectively. Currently, he is an associate professor in the Department of Mechatronics Engineering, University of Electronic Science and Technology of China (UESTC). His research activities have been concerned with laser soldering and welding, modeling and simulation in materials science and engineering.