INTELLIGENT AUTOMATION & SOFT COMPUTING, 2017
http://dx.doi.org/10.1080/10798587.2017.1294873

A Multi-Objective Metaheuristics Study on Solving Constrained Relay Node Deployment Problem in WSNS

Wenjie Yu, Xunbo Li, Hang Yang and Bo Huang

School of Mechatronics Engineering, University of Electronic Science and Technology of China, Chengdu, P.R. China

ABSTRACT

This paper studies how to deploy relay nodes into traditional wireless sensor networks with constraint aiming to simultaneously optimize two important factors; average energy consumption and average network reliability. We consider tackling this multi-objective (MO) optimization problem with three metaheuristics, which employ greatly different evolutional strategies, and aim at an in-depth analysis of different performances of these metaheuristics to our problem. For this purpose, a statistical procedure is employed to analyse the results for confidence, in consideration of two MO quality metrics; hypervolume and coverage of two sets. After comprehensive analysis of the results, it is concluded that NSGA-II provides the best performance.

KEYWORDS

Wireless sensor networks; Constrained relay node deployment; metaheuristics; MO optimization; hypervolume; Coverage of two sets

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 (Akyildiz, Su, Sankarasubramaniam, & Cayirci, 2002).

Traditional WSNs typically consist of sensor nodes (SNs) and a base station (BS) (L. Cheng, Wang, Wu, & Han, 2015). Generally, SNs are powered by batteries with energy limitations (Ma et al., 2011). In this regard, it is hard to get balanced energy consumption in the whole network. To improve this situation, adding nodes with higher energy capacity, called relay nodes (RNs) has been considered in the literature (Hou, Yi Shi, Sherali, & Midkiff, 2005). The deployment of RNs has been employed to improve network properties, such as network connectivity and lifetime maximization (Xu, Hassanein, Takahara, & Wang, 2010).

In most approaches, the RN deployment problem (RNDP) is studied without any physical limitations. However, in reality, there may be forbidden regions or lower bounds on internode distances in most deployment regions-we cannot deploy nodes anywhere we want. Thus, we propose a more practical situation, the constrained RNDP (CRNDP). The CRNDP is an NP-hard optimization problem (X. Z. Xiuzhen Cheng, Narahari, Simha, Maggie Xiaoyan Cheng, & Liu, 2003), which cannot be solved with conventional methods. Instead, some works assume non-conventional methods, such as heuristics (Misra, Majd, & Huang, 2014). For RNDP or CRNDP, heuristics provide one unique solution, while metaheuristics usually show a good behaviour solving this kind of problem, providing a set of trade-off solutions, which provides the network designer more possibilities to deploy the network. Moreover, metaheuristics are used to solve a wide variety of problems (Gou, Wang, & Luo, 2015; Kang & He, 2013; Nandy, Yang, Sarkar, & Das, 2015; Rauf & A. Aleisa, 2015; Wei, Wang, Li, Zou, & Yang, 2015; Yazdani, Naderi, & Mousakhani, 2015).

This paper studies CRNDP with lower bounds on internode distances constraint by using metaheuristics, 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.

Three approaches are studied for solving CRNDP. One is an improved MO Particle Swarm Optimizer (MOPSO) (Sierra & Coello, 2005), which is based on Pareto dominance and the use of a crowding factor to filter out the list of available leaders. The second one is Archive-Based hYbrid Scatter Search (AbYSS) (Nebro et al., 2008), which follows the scatter search structure, but uses mutation and crossover operators from evolutionary algorithms. Moreover, we include one additional standard algorithm NSGA-II (K. Deb, Pratap, Agarwal, & Meyarivan, 2002), which belong to EC. Although these algorithms show good performance in many optimization problems, they employ greatly different evolutional strategies. The behaviour they perform on CRNDP is deeply studied.

We compare these metaheuristics with a widely accepted statistical methodology. The results acquired are analysed through two MO quality metrics: Hypervolume and coverage of two sets (Zitzler, 1999; Zitzler & Thiele, 1999).

The rest of this paper is organized as follows: In Section 2, we discussed related work concerning CRNDP and metaheuristics methods applied to WSNs. Section 3 is devoted to the description of the WSN model and the definition of the problem we propose. The metaheuristics considered are detailed in Section 4. Section 5 discusses the experimental configuration. The results obtained are analysed in Section 6. The conclusion is given in the last section.

CONTACT Wenjie Yu 201311080107@std.uestc.edu.cn

2. Related Work

In this section, we describe the research on how to efficiently deploy RNs into traditional WSNs without and with constraints. By routing structures, RNDPs can be classified into single-tiered and two-tiered. In the single-tiered RN deployment, SN forwards packets received from other sensor nodes or RNs. In the two-tiered RN deployment, SNs forward only their own sensed information to an RN or a base station (BS).

First, previous works on single-tiered RN deployment are analysed. Cheng et al. (X. Cheng, Du, Wang, & Xu, 2008) guaranteed global connectivity by placing a minimum of RNs in single-tiered WSN with heuristics. Han et al. (Han, Cao, Lloyd, & Shen, 2010) optimized the fault-tolerance in single-tiered network considering sensors with adjustable transmission radius with heuristics. Lanza-Gutierrez and Gomez-Pulido (Lanza-Gutierrez & Gomez-Pulido, 2015) studied how to use two MO variable neighbourhood search algorithms to deploy RNs into single-tiered WSN with the objective of optimizing average energy consumption and average sensitivity area of the network.

Next, we give a review of previous works on two-tiered RN deployment. Xu et al. (Xu et al., 2010) discussed the impacts of random node deployment on connectivity and lifetime in two-tiered WSN. Peiravi et al. (Peiravi, Mashhadi, & Hamed Javadi, 2013) proposed a clustering method using a genetic algorithm in homogeneous two-tiered WSN, optimizing the network lifetime with different delay values. Azharuddin and Jana (Azharuddin & Jana, 2015) intended to minimize the number of RNs and maximize network connectivity by using a genetic algorithm, in two-tiered WSN.

All of the above approaches concern RN 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 a single-tiered network, Misra et al. (Misra, Majd, & Huang, 2011) deployed minimum RNs into two-tiered WSN by ensuring connectivity, with a constraint that RNs were limited to place at a subset of candidate positions. By reaching out along this constrained approach, Misra et al. (Misra et al., 2014) ensure connectivity and survivability by deploying a minimum number of RNs in energy-harvesting single-tiered WSN. The candidate locations with the energy harvesting potential are pre-specified. Perez et al. (Perez, Labrador, & Wightman, 2011) employed an MO evolutionary algorithm to optimize both the energy cost and the number of routers in single-tiered WSN. Nigam et al. (Nigam & Agarwal, 2014) proposed a branch-and-cut algorithm to place the minimum number of RNs at a subset of candidate locations in single-tiered WSN, ensuring the sensors communicated with the sink node within a pre-specified delay bound. Yang et al. (Yang, Misra, Fang, Xue, & Zhang, 2012) studied the two-tiered CRNDP with heuristics, under both connectivity and survivability requirements, which means intend to deploy the minimum number of RNs into the pre-specified WSN.

Differing from the approaches described before in a few aspects, our approach studies the problems of single-tiered CRNDP. As described above, several works explored this problem (Misra et al., 2011, 2014; Nigam & Agarwal, 2014; Perez et al., 2011). However, there are some significant differences between the study of ours and those of the related literature's. First, the studies of the literature mainly focus on improving the performance of the network via reasonable deployment of the minimum number of RNs, which are placed in some pre-specified candidate locations. However, we study how to place some fixed number RNs in WSN where the deployment space is continuous, i.e. there are countless possible candidate positions for an RN to be deployed at. Moreover, in our study, there is the minimum Euclidean distance between any pair of nodes. Second, in (Misra et al., 2011, 2014; Nigam & Agarwal, 2014), they studied single-objective CRNDP by heuristics. Our problem is MO and methods are metaheuristics. Although Perez et al. (Perez et al., 2011) studied an MO CRNDP with a metaheuristic, their optimization objectives and method used are different with ours.

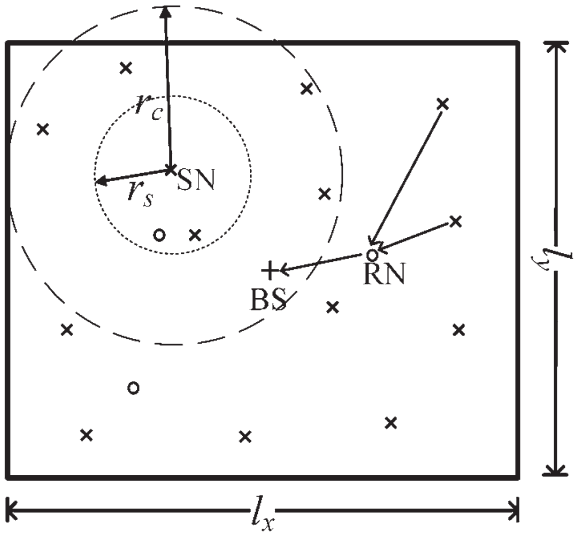

Figure 1. Wireless sensor network

3. System Models and Problem Formulation

In this section, we first describe the single-tiered architecture for WSN applied in the CRNDP. Next, we introduce the energy model and lifetime model considered in detail. Finally, we formally define the problem in this paper.

3.1. Network Model

We assume that the network on a two-dimensional 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 Figure 1. Only SNs are powered by batteries and the reminders have an 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 only connection point of the WSNs to the outside. Any two devices can communicate if they are at a Euclidean distance lower than communication radius r_c. All SNs are in the same energy charge initially. If a SN runs out of energy, it cannot be linked. For the sake of the simplicity, a perfect synchronization is assumed among all the devices and we consider S-MAC as the medium access protocol (Ye, Heidemann, & Estrin, 2002). The routing protocol based on shortest path, for all the devices, is provided by SPFA algorithm (Fanding, 1994). To reduce interference in practice, the minimum Euclidean distance d for any pair of devices is more than d_{min}.

3.2. Energy Model

We employ the energy model as discussed in (Konstantinidis, Yang, & Zhang, 2008), to simulate the energy cost of these devices. In this model, the packets' sending is the most expensive task of the energy expenditure. To this end, we ignore the energy cost by receiving, processing and sensing tasks.

The number of packets relayed by the sensor and the packets generated by the sensing mission of the sensor decide the number of information packets $P_i(t)$ sent by a sensor i, with $i \in S_s(t)$, for $t > 0$ and $t \in \tau$. The information packet is generated by a sensor per time unit. Thus, we get

$$P_i(t) = 1 + \sum_{j \in \{S_s(t) - i\}} \kappa_{j,i}^{s}(t) \quad t > 0, \tag{1}$$

Where $S_s(t)$ is the set of sensor coordinates with an energy charge more than 0 at time $t > 0 \in \tau$, $S_s(t) \subseteq S_s$, S_s is the set of initial sensor coordinates and $\kappa_{j,i}^{s}(t)$ is 1, if $i \in S_s(t)$ is in the minimum path between $j \in S_s(t)$ and the collector node at $t > 0$.

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

$$EP_i(t) = P_i(t) \cdot amp \cdot k \cdot ||i - \varsigma_i^s(t)||^{\alpha} \cdot \beta \tag{2}$$

Where 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 $\beta$is the transmission quality parameter ($\beta > 0$). This equation simulates extra cost due to packet loss. Finally, at time t, the residual energy of a sensor i, is shown as;

$$EL_i(t) = \begin{cases} EL_i(t-1) - EP_i(t) & if\ t > 0 \\ i.e. & if\ t = 0 \end{cases} \tag{3}$$

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

3.3. Lifetime Model

We define the lifetime as the number of time units. The network stop works when the energy of any sensor turns to be zero. To this end, the lifetime of the network is;

$$lt = |\{t > 0 \in \tau / EL_i = 0\}| \quad i \in S_s \tag{4}$$

where lt is the lifetime and $|.|$ is the cardinal of the set.

3.4. Problem Formulation

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

$$f_{aec} = \frac{\sum_{t=1}^{lt} \left(\sum_{i \in S_s} EP_i(t) \right)}{N_s \cdot lt}, \tag{5}$$

where N_s, the number of initial sensors, is the cardinal of S_s, lt and $EP_i(t)$ are given by (4) and (2), respectively.

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

$$f_{anr} = \frac{1}{N_s} \sum_{i \in S_s} r_i \tag{6}$$

where $f_{anr} \in [0, 1]$and r_i is the reliability of the sensor i, defined in (B. Deb, Bhatnagar, & Nath, 2003) as;

$$r_i = 1 - \prod_{k=1}^{edp_i^s} \left(1 - (1 - err)^{h_k^{i,s}} \right), \tag{7}$$

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 kth 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 (Ford & Fulkerson, 1956).

The $\overline{f_{anr}}$ is the complementation of the f_{anr}. That is,

$$\overline{f_{anr}} = \frac{1}{N_s} \sum_{i \in S_s} (1 - r_i) \tag{8}$$

This way, CRNDP is defined as an NP-hard MO problem, where given a traditional wireless sensor network, i.e. a set of sensors $S_s(t)$ and a sink node, the objective is to deploy a set of RNs S_r to

$$\min(f_{aec}), \max(f_{anr}) \tag{9}$$

This is equivalent to;

$$\min \left(f_{aec}, \overline{f_{anr}} \right), \tag{10}$$

subject to

$$\begin{cases} m = (x_1, y_1) \quad \forall m \in S_a, x_1 \in [0, l_x], y_1 \in [0, l_y] \\ n = (x_2, y_2) \quad \forall n \in S_a, x_2 \in [0, l_x], y_2 \in [0, l_y] \\ d(m, n) \geq d_{\min} \end{cases} \tag{11}$$

and

$$z = (x, y) \quad \forall z \in S_r, x \in [0, l_x], y \in [0, l_y] \tag{12}$$

where S_a is the set of all nodes coordinates (including SNs, RNs and the BS), S_r is the set of router coordinates, $\forall r \in S_r, r = (x, y)$ where $x \in [0, l_x]$ and $y \in [0, l_y]$.

4. Multi-objective Optimization Algorithms

The MO metaheuristics considered to solve the CRNDP are introduced in this section. It is impossible to solve these NP-hard optimization problems with exact techniques. Approximation algorithms are often applied to solve this kind of problem. The Evolutionary Algorithms (EAs) play an important role in approximation algorithms (Bhattacharyya & Goswami, 2007; Johnston, 2008; Li & Liu, 2015; Liu, Wang, & Cheung, 2009; Xue, Zhong, Ma, & Cao, 2015). In our approach, one standard genetic algorithm NSAG-II is considered. This algorithm encodes its individuals as chromosomes and belongs to sub branch of EAs. Next, we consider MOPSO, an improved MO optimization algorithm developing from standard particle swarm optimization (PSO). Finally, AbYSS, a MO algorithm employing scatter search template, is also taken into account. This algorithm searches the resolution space in a systemic way. These algorithms employ greatly different evolutional strategy, but they all show good performance in many optimization problems.

Before detailed description of these algorithms, we consider the same encoding for all the metaheuristics. The representation is defined as: Every chromosome is composed of N_r genes. A gene is the two-dimension-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. Non-dominated Sorting Genetic Algorithm (NSGA-II)

Deb et al. (K. Deb et al., 2002) proposed this algorithm, which is one of the most efficient and famous MO evolutionary algorithms. NSGA-II maintains a population P_t of size N at generation t. Q_t, keeping the same size with P_t, is the offspring population of P_t. The characteristic feature of NSGA-II is that it employs a fast non-dominated sorting and crowding-distance estimation process for ranking the population into different fronts.

Based on the features above, the authors defines an elitist crowded-comparison operator to guide the evolutionary process. i and j are two possible solutions of an optimization problem. i_{rank} and j_{rank} are the sorting of i and j. If i_{dis} and j_{dis} are the crowding distance of i and j, the crowded-comparison operator $\geq_n$ is defined as:

$$i \geq_n j \ \ if((i_{rank} < j_{rank}) or((i_{rank} = j_{rank}) and(i_{dis} > j_{dis}))) \quad (13),$$

Which means i dominates j if one of the conditions stands. The outline of NSGA-II is described in Algorithm 1.

Algorithm 1: NSGA-II

1. Initially, Q_t and P_t are empty.
2. N individual in P_t are randomly generated.
3. While not stop condition (A number of iterations are performed).
4. P_t and Q_t are combined into a new set called R_t, whose size is $2N$.
5. R_t is sorted into non-dominated fronts F by fast non-dominated sort.
6. Set P_t empty and set i=1.
7. While $|P_t|+|F_i| \leq N$. $|.|$ is the cardinal of the set.
8. Employ crowding-distance-assignment process on F_i.
9. Set $P_t = P_t \cup F_i$ and $i = i+1$.
10. End while.
11. If $|P_t| < N$ then sort F_i by $\geq_n$ defined in Equation (13) and set $P_t = P_t \cup F_i[1:N-|P_t|]$.
12. Set Q_t empty.
13. While $|Q_t| \leq N$.
14. Select two individuals, p_1 and p_2, from P_t via binary tournament.
15. p_1' and p_2' are generated by simulated binary crossover and the probability of crossover is *cross*.
16. Two new individuals, p_1" *and* p_2", are generated from p_1' and p_2' through polynomial mutation, and the probability of mutation is *mut*.
17. *Set* $Q_t = Q_t \cup p_1" \cup p_2"$.
18. End while.
19. Set $t = t+1$.
20. End while.

4.2. A New Multi-Objective Particle Swarm Optimization (MOPSO)

This algorithm was proposed by Margarita et al. (Sierra & Coello, 2005) as an improved version of the multiple objective particle swarm optimization algorithm (MOPSO). MOPSO is characterized by using a crowding factor in order to establish a second discrimination criterion (additional to Pareto dominance). This criterion is also adopted to decide what leaders to keep over generations when the maximum list size has been exceeded. Other important features are the use of three kinds of mutation strategies and ε-dominance.

Three kinds of mutation strategies contain; uniform mutation, non-uniform mutation and no mutation at all. To this end, MOPSO subdivides the swarm to three parts (of equal size). Each sub-part of the swarm will adopt a different mutation scheme, respectively. The concept of ε-dominance adopted in this algorithm is used to fix the size of the external archive that contains the non-dominated solutions. A decision vector x_1 is said to ε-dominance a decision vector x_2 for some ε>0 if: $f_i(x_1)/(1+\varepsilon) \leq f_i(x_2)$, $\forall i = 1, ..., m$ and $f_i(x_1)/(1+\varepsilon) < f_i(x_2)$, for at least one i = 1, ..., m. In order to perform the flight of each particle, the changes to the velocity vector are formulated in the following way:

$$v_i(t) = Wv_i(t-1) + C_1 r_1(x_{pbest_i} - x_i(t)) + C_2 r_2(x_{gbest} - x_i(t)) \quad (14)$$

Where W = random (0.1, 0.5), C_1, C_2 = random (1.5, 2.0), and r_1, r_2=random (0.0, 1.0). The pseudocode is shown in Algorithm 2.

Algorithm 2: MOPSO

1. Set the swarm population P_t empty and set the set of leaders, L_t, empty. The set of leaders is composed of non-dominated solutions. P_t and L_t have the same maximum size.
2. Initialize the population P_t, i.e. the position and the speed of each individual are randomly generated.
3. *Evaluate* P_t. The non-dominated particles found in P_t are introduced into L_t. If the size of L_t exceeds the maximum list size, the crowding factor is used to decide what leaders to keep over generations.
4. *Use* ε-dominance to save leaders from L_t to archive εArc, which contains the non-dominated solutions that will be reported by the algorithm.
5. Update the x_{pbest} of the P_t.
6. While $g < g_{max.}$
7. For each particle.
8. Select the global best solution x_{gbest} by means of a binary tournament based on the crowding value of the leaders L_t.
9. Compute the speed of the particle by Equation (14).
10. Compute the new position of the particle.
11. Separate the population into three equal parts and execute three kinds of mutation strategies on them, respectively.
12. Update the x_{pbest} of the particle.
13. End for.
14. Evaluate P_t. The non-dominated particles found in P_t are introduced into L_t.
15. *Use* ε-dominance to save leaders from L_t to archive εArc.
16. Set $g = g+1$.
17. End while.

4.3. Archive-Based hYbrid Scatter Search (AbYSS)

This algorithm was proposed by Antonio et al. (Nebro et al., 2008) as a multiple algorithm based on scatter search template. It first adapts the well-known scatter search template for single-objective optimization to the MO domain. This algorithm follows the scatter search structure, but uses mutation and crossover operators from evolutionary algorithms. AbYSS also uses typical concepts from the MO field, such as Pareto dominance, density estimation, and an external archive to store the non-dominated solutions. The flow path is as depicted in Algorithm 3.

Table 1. Experimental Cases Considered.

				Fitness (N_r =0)		Referencf_{aec}		Reference $\overline{f_{anr}}$		
Test	r_s	r_c	N_s	f_{aec}	$\overline{f_{anr}}$	best	worst	best	worst	Experiment cases (N_r)
100×100	15	30	15	0.1036	0.02851	0.054	0.098	0.0051	0.027	1,2,3
200×200	15	30	57	0.2288	0.06701	0.10	0.23	0.026	0.067	1,2,4,6,9
300×300	15	30	128	0.3488	0.1421	0.13	0.35	0.045	0.15	2,4,6,12,18,24

Algorithm 3: AbYSS

1. Set the population P_t empty.
2. Use diversification method to generate the set P_t' including N initial solutions. N is the population size.
3. Each solution in P_t' is passed to the improvement method. Then the solution set P_t" is generated. The improvement method is mainly based on a polynomial mutation operator to improve the quality of the solution.
4. The solutions in P_t" are inserted to P_t.
5. While not stop condition.
6. The reference sets $RefSet_1$ and $RefSet_2$ are built based on P_t. $RefSet_1$ contains the best quality solutions from P_t, while $RefSet_2$ are filled with solutions promoting diversity.
7. Use subset generation method to generate subset P_s based on $RefSet_1$ and $RefSet_2$. The subset generation method is mainly based on a binary crossover operator.
8. While $|P_s|$>0.
9. For each individual p_s in P_s.
10. p_s', an improved individual, is generated by using improve method to work on *ps*.
11. Use reference set update method to decide whether p_s' should be added to $RefSet_1$ or $RefSet_2$ or *archive* or be discarded. *archive* is the external archive for storing a record of the non-dominated individuals found during the search process.
12. End for.
13. Use subset generation method to regenerate subset P_s.
14. End While.
15. *Set* P_t empty.
16. The individuals in $RefSet_1$ are inserted to P_t.
17. *The best n* individuals from archive are inserted to P_t, according to the crowding distance.
18. The diversification and improvement methods are used to produce new solutions for filling up the set P_t. The value of n is the minimum of the size of the archive and half of the size of P_t.
19. Increment t.
20. End While.

5. Experimental Strategy

In this section, several questions arise: 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 a MO metaheuristics regarding a set of runs? What data-set will be used 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. Performance Measures

We employ two complementary measures to evaluate the trade-off fronts produced by the metaheuristics to CRNDP.

Hypervolume (HV): This metric calculates the portion of the objective space covered by members of a non-dominated 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 HV of F is the union of all its hypercubes. That is,

$$HV = \text{volume}(\bigcup_{i=1}^{|F|} \mu_i) \tag{15}$$

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,

$$CTS(X_1, X_2) = \frac{|\{x_2 \in X_2; \exists x_1 \in X_1 : x_1 \succeq x_2\}|}{|X_2|} \tag{16}$$

5.2. Strategy and Data-set Used

In our approach, we perform 30 independent runs for each algorithm and experiment, being 30 runs a widely accepted value to reach statistical conclusions (William & Hays, 1975). As stop condition, we assume several criteria in order to study the convergence of the algorithms. Accordingly, we assume 50,000, 100,000, 200,000, 300,000, 400,000 and 500,000 evaluations.

The data-set we used in this paper is proposed in (Lanza-Gutierrez & Gomez-Pulido, 2015). 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 WSNs model considers some parameters as stated previously. It is assumed that $\alpha = 2$, $\beta = 1$, $k = 128$ KB, $r_s = 15$ m, r_c=30 m and $amp = 100\ pJ/bit/m^2$, from (Konstantinidis & Yang, 2011) and the d_{min} is 0.1 m in the WSN model. Too many RNs imply that the network overhead will increase. Thus, in this approach, we do not include more than 20% of these devices regarding the number of sensors as (Lanza-Gutierrez & Gomez-Pulido, 2015) did. The number of routers, which we will add to optimize the network, is shown in Table 1. In addition, Table 1 also shows the value of the fitness functions without including RNs (N_r=0).

5.3. Parameters Settings

Before conducting experiments, all the parameters of the algorithms are configured. In the NSGA-II algorithm, possible values of *mut* are defined as $mut=\{m|m=0.05{\cdot}i,\ i\in n\}$, n=1,2,..19. And cross is divided into $cross=\{c \mid c=0.05{\cdot}i,\ i\in n\}$, n=1,2,…19. Then, the configurations of NSGA-II parameters are derived as

Table 2. $\overline{HV}$ and *IQR*: 100 × 100, 200 × 200, 300 × 300 Tests (NSGA-II).

Test(N_r)	50,000		100,000		200,000		300,000		400,000		500,000	
100×100–30 (1)	0.1059	0.0108	0.1113	0.0001	0.1114	0.0000	0.1114	0.0000	0.1114	0.0000	0.1114	0.0000
100×100–30 (2)	0.5101	0.0348	0.5265	0.0025	0.5280	0.0000	0.5282	0.0000	0.5283	0.0000	0.5283	0.0000
100×100–30 (3)	0.8976	0.0120	0.8988	0.0107	0.9002	0.0123	0.9006	0.0124	0.9007	0.0124	0.9070	0.0000
200×200–30(1)	0.0419	0.0000	0.0419	0.0000	0.0419	0.0000	0.0419	0.0000	0.0419	0.0000	0.0419	0.0000
200×200–30 (2)	0.1317	0.0009	0.1317	0.0009	0.1318	0.0009	0.1318	0.0009	0.1318	0.0009	0.1318	0.0009
200×200–30 (4)	0.3399	0.0099	0.3416	0.0123	0.3418	0.0122	0.3421	0.0122	0.3423	0.0125	0.3427	0.0119
200×200–30 (6)	0.5541	0.0161	0.5620	0.0119	0.5672	0.0074	0.5690	0.0066	0.5741	0.0063	0.5746	0.0060
200×200–30 (9)	0.8721	0.0197	0.8897	0.0201	0.9056	0.0004	0.9088	0.0018	0.9127	0.0035	0.9239	0.0020
300×300–30 (2)	0.0829	0.0019	0.0835	0.0009	0.0840	0.0002	0.0848	0.0001	0.0848	0.0001	0.0848	0.0001
300×300–30(4)	0.1943	0.0111	0.1989	0.0081	0.2003	0.0090	0.2006	0.0093	0.2013	0.0105	0.2014	0.0105
300×300–30 (6)	0.2760	0.0057	0.2829	0.0020	0.2841	0.0033	0.2844	0.0033	0.2847	0.0036	0.2848	0.0038
300×300–30 (12)	0.4737	0.0450	0.4859	0.0485	0.5012	0.0401	0.5093	0.0395	0.5123	0.0381	0.5135	0.0391
300×300–30 (18)	0.6720	0.0268	0.7015	0.0246	0.7232	0.0136	0.7359	0.0019	0.7427	0.0025	0.7495	0.0064
300×300–30 (24)	0.8199	0.0210	0.8489	0.0141	0.8843	0.0029	0.8951	0.0095	0.9072	0.0048	0.9120	0.0030

Table 3. $\overline{HV}$ and *IQR*: 100 × 100, 200 × 200, 300 × 300 Tests (MOPSO).

Test(N_r)	50,000		100,000		200,000		300,000		400,000		500,000	
100×100–30 (1)	0.1107	0.0007	0.1110	0.0000	0.1113	0.0000	0.1113	0.0000	0.1113	0.0000	0.1113	0.0000
100×100–30 (2)	0.4910	0.0029	0.5028	0.0251	0.5253	0.0018	0.5255	0.0018	0.5259	0.0011	0.5261	0.0014
100×100–30 (3)	0.8429	0.0051	0.8559	0.0047	0.8783	0.0101	0.8850	0.0086	0.8951	0.0215	0.8976	0.0210
200×200–30 (1)	0.0419	0.0000	0.0419	0.0000	0.0419	0.0000	0.0419	0.0000	0.0419	0.0000	0.0419	0.0000
200×200–30 (2)	0.1275	0.0014	0.1283	0.0026	0.1293	0.0018	0.1299	0.0014	0.1304	0.0008	0.1307	0.0004
200×200–30 (4)	0.2642	0.0246	0.2738	0.0148	0.2919	0.0163	0.2942	0.0178	0.2995	0.0075	0.3073	0.0018
200×200–30 (6)	0.4045	0.0010	0.4288	0.0271	0.4565	0.0331	0.4579	0.0321	0.4627	0.0388	0.4685	0.0275
200×200–30 (9)	0.6589	0.0204	0.6927	0.0272	0.6964	0.0250	0.7002	0.0201	0.7062	0.0185	0.7160	0.0167
300×300–30 (2)	0.0804	0.0012	0.0811	0.0019	0.0817	0.0007	0.0818	0.0008	0.0818	0.0008	0.0818	0.0008
300×300–30 (4)	0.1696	0.0205	0.1758	0.0172	0.1797	0.0146	0.1854	0.0051	0.1873	0.0043	0.1900	0.0082
300×300–30 (6)	0.2359	0.0052	0.2382	0.0019	0.2400	0.0022	0.2416	0.0044	0.2493	0.0083	0.2508	0.0066
300×300–30 (12)	0.3670	0.0119	0.3850	0.0167	0.3913	0.0096	0.3928	0.0083	0.3978	0.0139	0.4070	0.0307
300×300–30 (18)	0.4856	0.0184	0.5069	0.0034	0.5259	0.0032	0.5459	0.0139	0.5513	0.0159	0.5528	0.0144
300×300–30 (24)	0.5788	0.0161	0.6002	0.0280	0.6153	0.0197	0.6226	0.0097	0.6459	0.0343	0.6540	0.0419

configs={(*a*,*b*)|*a*∈ *mut*, *b*∈ *cross*}. After that, each configuration of the parameter is conducted by 30 independent runs, considering a reduced stop condition (10,000 evaluations). Then, assuming the best HV metric as the quality indicator, the configuration, which provides the best performance on average is *mut*=0.1 and *cross*=0.9. By employing the same configuration method, the crossover probability in AbYSS is 0.95 and the mutation probability is $1/N_r$ as the algorithm is proposed. The mutation probability in MOPSO is $1/N_r$ proposed by the algorithm and the ε is 0.0075 as (Sierra & Coello, 2005) did for simplicity. Note that the reference points, "Reference f_{aec}" and "Reference f_{anr}", assumed to calculate the HV are also shown in Table 1. These values were obtained experimentally where best and worst are the best and the worst value of a fitness function, respectively. As population size, *N*, the same habitual value of 100 individuals is assumed for NSGA-II and MOPSO. However, AbYSS whose individuals are generated systematically is a structured strategy, and its size of population is 20 but size of archive is 100.

6. Performance Evaluation

The advantages provided by the addition of RNs to traditional WSNs has been analysed in many literatures (X. Cheng et al., 2008; Misra et al., 2011; Lanza-Gutierrez & Gomez-Pulido, 2015; Yang et al., 2012). Therefore, we chiefly address the CRNDP based on the data-set with MO metaheuristics in this section. In the simulation experiment, JDK 1.7 is employed to code the simulation process. We acquire the optimization results about the industry parameters; AEC of SNs and ANR. However, due to the space limitations, we focus on using some classical statistical analysis methods to analyse the quality of the solutions obtained in MO metrics instead of showing the optimization results of the industrial parameters.

6.1. Analysis based on HV Metric

Initially, we evaluate the quality of the MO algorithms based on the metric of HV. As is shown in Table 2, Table 3 and Table 4, "$l_x \times l_y$-$r_c(N_r)$" means there are N_r RNs being deployed in a two dimensional areas with size $l_x \times l_y$ and the communication radius of r_c. The data included in the tables are the average HV ($\overline{HV}$) and the interquartile range (*IQR*) for each algorithm, test case and stop evaluation. The best ones for each test case and stop evaluation have a grey background.

NSGA-II seems to provide the best performance among the algorithms. Since our experiments are dealing with some stochastic analyses with MO metaheuristics so as to show some results with confidence, the following statistical analyses are employed through this approach.

First, we consider the Kolmogorov–Smirnov–Lilliefor's (Lilliefors, 1967) and Shapiro–Wilk's (Villasenor Alva & Estrada, 2009) tests in order to analyse whether results come from a normal distribution. In this regard, the following hypothesis is presented: H_0: 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). In all the cases, we get the *p*-values more than 0.05. Therefore, the assumption of H_0 fails and the median is the average value. The median is written simply as *M*. Thus, the results do not follow the Gaussian distribution and the samples are independent.

Next, we consider the Wilcoxon-Mann-Whitney's (Mann & Whitney, 1947) test to study if some significant differences are shown among the algorithms. In this test, we have following

Table 4. $\overline{HV}$ and *IQR*: 100 × 100, 200 × 200, 300 × 300 Tests (AbYSS).

Test(N_i)	50,000		100,000		200,000		300,000		400,000		500,000	
100×100–30 (1)	0.1111	0.0003	0.1112	0.0004	0.1114	0.0000	0.1114	0.0000	0.1114	0.0000	0.1114	0.0000
100×100–30 (2)	0.5089	0.0324	0.5100	0.0336	0.5275	0.0006	0.5280	0.0001	0.5281	0.0000	0.5282	0.0001
100×100–30 (3)	0.8463	0.0196	0.8736	0.0647	0.8890	0.0361	0.9132	0.0123	0.9135	0.0129	0.9137	0.0131
200×200–30 (1)	0.0419	0.0000	0.0419	0.0000	0.0419	0.0000	0.0419	0.0000	0.0419	0.0000	0.0419	0.0000
200×200–30 (2)	0.1306	0.0002	0.1309	0.0002	0.1312	0.0001	0.1313	0.0000	0.1314	0.0000	0.1314	0.0000
200×200–30 (4)	0.3166	0.0023	0.3200	0.0003	0.3329	0.0172	0.3343	0.0158	0.3349	0.0168	0.3354	0.0159
200×200–30 (6)	0.4896	0.0111	0.5048	0.0075	0.5150	0.0169	0.5182	0.0231	0.5286	0.0435	0.5292	0.0447
200×200–30 (9)	0.7467	0.0200	0.7915	0.0442	0.8245	0.0461	0.8476	0.0499	0.8668	0.0155	0.8693	0.0198
300×300–30 (2)	0.0813	0.0024	0.0831	0.0004	0.0836	0.0002	0.0836	0.0002	0.0836	0.0002	0.0836	0.0002
300×300–30 (4)	0.1830	0.0114	0.1869	0.0150	0.1880	0.0147	0.1900	0.0137	0.1902	0.0135	0.1946	0.0053
300×300–30 (6)	0.2626	0.0273	0.2651	0.0310	0.2725	0.0212	0.2754	0.0240	0.2782	0.0247	0.2784	0.0247
300×300–30 (12)	0.4520	0.0467	0.4737	0.0485	0.4895	0.0435	0.4912	0.0433	0.4925	0.0449	0.4939	0.0473
300×300–30 (18)	0.5828	0.0038	0.6132	0.0242	0.6277	0.0327	0.6392	0.0400	0.6431	0.0415	0.6529	0.0430
300×300–30 (24)	0.7747	0.0425	0.7966	0.0213	0.8225	0.0182	0.8520	0.0384	0.8646	0.0368	0.8686	0.0327

Table 5. Average CTS among the Algorithms for 100 × 100 Instances.

NSGAII	50,000	100,000	200,000	300,000	400,000	500,000	Average
MOPSO	100.00%	100.00%	100.00%	99.23%	97.90%	96.10%	98.87%
AbYSS	90.07%	85.19%	87.97%	86.09%	85.35%	85.09%	86.63%
Average	95.03%	92.59%	93.99%	92.66%	91.63%	90.60%	92.75%
MOPSO							
NSGAII	71.52%	70.25%	68.99%	69.75%	73.29%	73.78%	71.26%
AbYSS	75.50%	74.07%	72.15%	68.87%	70.06%	71.43%	72.01%
Average	73.51%	72.16%	70.57%	69.31%	71.68%	72.60%	71.64%
Abyss							
NSGAII	89.70%	95.57%	91.77%	91.36%	90.68%	90.24%	91.55%
MOPSO	100.00%	100.00%	100.00%	100.00%	100.00%	100.00%	100.00%
Average	94.85%	97.78%	95.89%	95.68%	95.34%	95.12%	95.78%

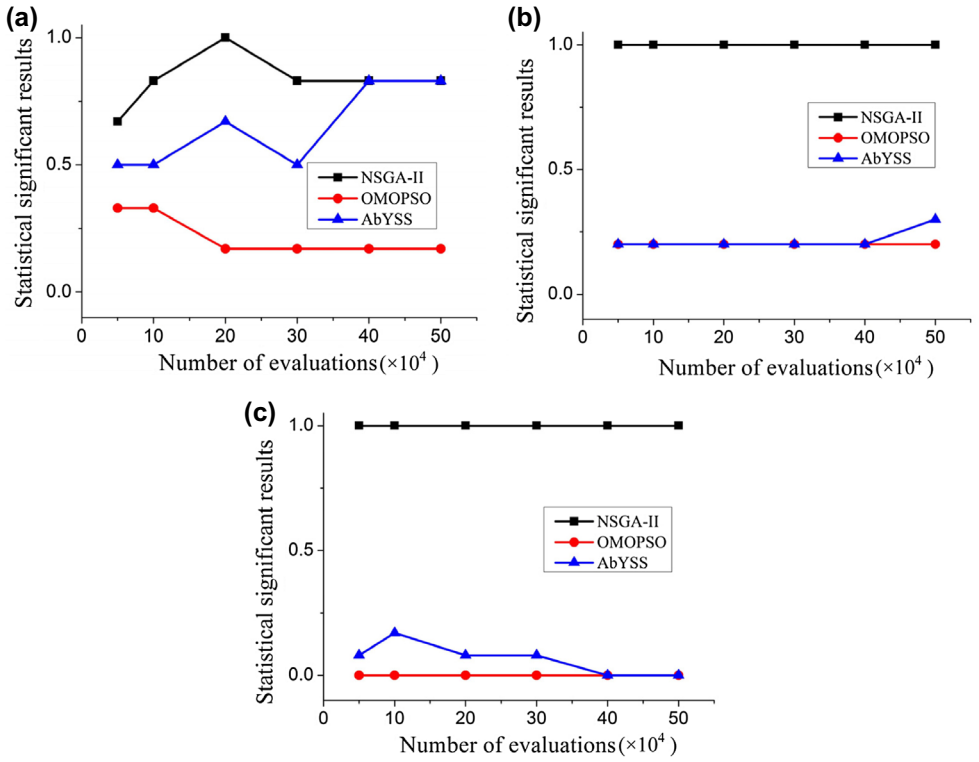

Figure 2. Analyses of the *p*-values obtained in different scenarios

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, j=b,c, a is NSGA-II, b is MOPSO, c is AbYSS). We consider the p-values with a significance level of 0.05. Based on this test method, we compare the MO algorithms for figuring out which one provides the best significant performance with each stop evaluation and test case.

Along with above-mentioned statistical procedure, the result of the percentage of test cases is depicted in Figure 2. In three different cases, it is clear that which algorithm provides the best significant performance. First, for 100 × 100 scenario, we note that NSGA-II provides the best behaviour for reduced stop evaluations, while NSGA-II and AbYSS both perform greatly for a high number of evaluations. Then, for 200 × 200 scenario, NSGA-II shows the best performance for all the stop evaluations, followed by MOPSO and AbYSS. However, AbYSS shows a better behaviour than MOPSO in a high number of

Table 6. Average CTS among the Algorithms for 200 × 200 Instances.

NSGAII	50000	100,000	200,000	300,000	400,000	500,000	Average
MOPSO	100.00%	100.00%	100.00%	99.23%	97.90%	96.10%	98.19%
AbYSS	90.07%	85.19%	87.97%	86.09%	85.35%	85.09%	86.05%
Average	95.03%	92.59%	93.99%	92.66%	91.63%	90.60%	92.12%
MOPSO							
NSGAII	71.52%	70.25%	68.99%	69.75%	73.29%	73.78%	72.22%
AbYSS	75.50%	74.07%	72.15%	68.87%	70.06%	71.43%	71.61%
Average	73.51%	72.16%	70.57%	69.31%	71.68%	72.60%	71.92%
Abyss							
NSGAII	89.70%	95.57%	91.77%	91.36%	90.68%	90.24%	91.30%
MOPSO	100.00%	100.00%	100.00%	100.00%	100.00%	100.00%	100.00%
Average	94.85%	97.78%	95.89%	95.68%	95.34%	95.12%	95.65%

Table 7. Average CTS among the Algorithms for 300 × 300 Instances.

NSGAII	50,000	100,000	200,000	300,000	400,000	500,000	Average
MOPSO	100.00%	100.00%	100.00%	100.00%	100.00%	100.00%	100.00%
AbYSS	96.75%	99.83%	99.95%	100.00%	100.00%	99.95%	99.41%
Average	98.37%	99.92%	99.97%	100.00%	100.00%	99.97%	99.71%
MOPSO							
NSGAII	71.81%	70.26%	75.60%	76.23%	77.40%	76.73%	74.67%
AbYSS	71.21%	73.09%	77.53%	78.29%	78.87%	79.36%	76.39%
Average	71.51%	71.67%	76.56%	77.26%	78.13%	78.04%	75.53%
AbYSS							
NSGAII	87.43%	86.61%	85.91%	87.39%	87.55%	87.05%	86.99%
MOPSO	100.00%	100.00%	100.00%	100.00%	100.00%	100.00%	100.00%
Average	93.71%	93.31%	92.95%	93.70%	93.77%	93.53%	93.50%

evaluations. Finally, for 300 × 300 scenario, NSGA-II still provides the best behaviour for all stop evaluations. AbYSS shows better performance than MOPSO for reduced stop conditions. However, AbYSS decreases its performance as same as MOPSO for a high number of evaluations. Based on these analyses, we conclude that NSGA-II is the best algorithm for all instances, followed by AbYSS, and the last one is MOPSO.

6.2. Analysis based on CTS Metric

In addition to HV, the CTS metric is employed to analyse the quality of the MO algorithms according to the size of the area. We calculated the value of this metric by considering the median Pareto fronts from previous 30 samples. First, for 100 × 100 scenarios, Table 6 shows the average set coverage for each stop condition, providing AbYSS with the best behaviour on average, followed by NSGA-II. For 200 × 200 instances and 300 × 300 instances, as is shown in Table 5 and Table 7, NSGA-II performs an overwhelming superiority over the other algorithms, followed by AbYSS.

6.3. Comprehensive Evaluation

With the objective of solving CRNDP and according to two MO metrics used for algorithm evaluation, AbYSS provides the similar behaviour as NSGA-II in instances with a small number of RNs. However, NSGA-II shows an overwhelming performance in all instances and stop conditions. To this end, NSGA-II is the best algorithm for CRNDP, followed by AbYSS, and MOPSO is the most inefficient.

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. We find many works assuming employing heuristics to this problem. However, metaheuristics usually perform a good behaviour solving this kind of problems, providing a set of trade-off solutions, which provides the network designer more possibilities to deploy the network. In this case, we assume several MO metaheuristics employing greatly different evolutional strategies to solve this problem. MOPSO is based on Pareto dominance and the use of a crowding factor to filter out the list of available leaders. AbYSS follows the scatter search structure, but uses mutation and crossover operators from evolutionary algorithms. Moreover, NSGA-II is a standard genetic algorithm. These metaheuristics are employed to optimize a data-set obtained from the literature, assuming three different scenarios. The results obtained are analysed considering two standard MO metrics; HV and CTS. Through a widely accepted statistical methodology, we conclude that AbYSS provides the similar behaviour as NSGA-II in the instances with a small number of RNs. However, NSGA-II shows the best performance for all instances and stop conditions, and MOPSO is the last.

As future extensions of this work, it would be interesting to assume CRNDP to three-dimensional sensing field. In addition, other key fitness functions, more realistic constraints in industry and more different kinds of metaheuristics will be included in our further research.

Disclosure statement

No potential conflict of interest was reported by the authors.

Notes on contributors

Wenjie Yu is pursuing a Ph.D. degree in Mechanical Engineering from University of Electronic Science and Technology of China, in a successive postgraduate and doctoral program. Having studied the mechanical and electronic engineering in Master's stage, his current research involves wireless sensor networks, multi-objective optimization and network coding.

Xunbo Li received an M.S. degree in Machinery Manufacturing from University of Electronic Science and Technology of China in 1991 and his Ph.D. degree in Measurement and Instrument from the same university in 2001. Currently, he is a professor and the director of the Institute of Electrical and Mechanical Intelligent Control. His current research involves intelligent measurement and control technology research, as well as mechanical and electrical application.

Hang Yang is pursuing a Ph.D. degree in Mechatronic Engineering at University of Electronic Science and Technology of China, working in the field of wireless communication and signal processing, and focusing on the Total Factors Sensing Technology in Industry.

Bo Huang is a Ph.D. student with the School of Mechatronics Engineering of the University of Electronic Science and Technology of China. His research includes mechanical dynamics, electronic packaging reliability, fatigue life prediction.

References

Akyildiz, I.F., Su, W., Sankarasubramaniam, Y., & Cayirci, E. (2002). Wireless sensor networks: A survey. *Computer Networks, 38*, 393–422. doi:10.1016/S1389-1286(01)00302-4

Azharuddin, M., & Jana, P.K. (2015). A GA-based approach for fault tolerant relay node placement in wireless sensor networks. In Proceedings of the Third International Conference on Computer, Communication, Control and Information Technology (C3IT).

Bhattacharyya, B., & Goswami, S.K. (2007). Reactive power optimization through evolutionary techniques: A comparative study of the Ga, De and PSO. *Intelligent Automation and Soft Computing, 13*, 461–469. doi:10.1080/10798587.2007.10642976

Cheng, L., Wang, Y., Wu, C., & Han, Q. (2015). A Pso-based Maintenance Strategy in Wireless Sensor Networks. *Intelligent Automation and Soft Computing, 21*, 65–75. doi:10.1080/10798587.2014.919716

Cheng, X., Du, D.-Z., Wang, L., & Xu, B. (2008). Relay sensor placement in wireless sensor networks. *Wireless Networks, 14*, 347–355. doi:10.1007/s11276-006-0724-8

Deb, B., Bhatnagar, S., & Nath, B. (2003). ReInForM: Reliable information forwarding using multiple paths in sensor networks. In Proceedings of the 28th Annual IEEE International Conference on Local Computer Networks, 2003. LCN'03.

Deb, K., Pratap, A., Agarwal, S., & Meyarivan, T. (2002). A fast and elitist multiobjective genetic algorithm: NSGA-II. *IEEE Transactions on Evolutionary Computation, 6*, 182–197. doi:10.1109/4235.996017

Fanding, D. (1994). A Faster Algorithm for Shortest-Ptath — SPFA. *Journal of Southwest Jiaotong University, 29*, 207–212.

Ford, L.R., & Fulkerson, D.R. (1956). Maximal flow through a network. *Canadian Journal of Mathematics, 8*, 399–404.

Gou, J., Wang, F., & Luo, W. (2015). Mining Fuzzy Association Rules Based on Parallel Particle Swarm Optimization Algorithm. *Intelligent Automation and Soft Computing, 21*, 147–162. doi:10.1080/10798587.2014.957482

Han, X., Cao, X., Lloyd, E.L., & Shen, C.-C. (2010). Fault-tolerant relay node placement in heterogeneous wireless sensor networks. *IEEE Transactions on Mobile Computing, 9*, 643–656. doi:10.1109/TMC.2009.161

Hou, Y.T., Yi Shi, S., Sherali, H.D., & Midkiff, S.F. (2005). On energy provisioning and relay node placement for wireless sensor networks. *IEEE Transactions on Wireless Communications, 4*, 2579–2590. doi:10.1109/TWC.2005.853969

Johnston, M.D. (2008). An evolutionary algorithm approach to multi-objective scheduling of space network communications. *Intelligent Automation and Soft Computing, 14*, 367–376. doi:10.1080/10798587.2008.10643001

Kang, Q., & He, H. (2013). Honeybee Mating Optimization Algorithm For Task Assignment In Heterogeneous Computing Systems. *Intelligent Automation & Soft Computing, 19*, 69–84. doi:10.1080/10798587.2013.771438

Konstantinidis, A., & Yang, K. (2011). multi-objective k-connected deployment and power assignment in WSNs using a problem-specific constrained evolutionary algorithm based on decomposition. *Computer Communications, 34*, 83–98. doi:10.1016/j.comcom.2010.08.011

Konstantinidis, A., Yang, K., & Zhang, Q. (2008). An evolutionary algorithm to a multi-objective deployment and power assignment problem in wireless sensor networks. In Proceedings of the Global Telecommunications Conference, 2008. IEEE GLOBECOM 2008. IEEE.

Lanza-Gutierrez, J.M., & Gomez-Pulido, J.A. (2015). Studying the multiobjective variable neighbourhood search algorithm when solving the relay node placement problem in Wireless Sensor Networks. *Soft Computing, 20*, 67–86. doi:10.1007/s00500-015-1670-0

Li, Z., & Liu, H-L. (2015). Integrating Preference by Means of Desirability Function with Evolutionary Multi-objective Optimization. *Intelligent Automation and Soft Computing, 21*, 197–209. doi:10.1080/10798587.2014.961313

Lilliefors, H.W. (1967). On the Kolmogorov-Smirnov test for normality with mean and variance unknown. *Journal of the American Statistical Association, 62*, 399–402.

Liu, H.-L., Wang, Y., & Cheung, Y.-M. (2009). A multi-objective evolutionary algorithm using min-max strategy and sphere coordinate transformation. *Intelligent Automation and Soft Computing, 15*, 361–384. doi:10.1080/10798587.2009.10643036

Ma, C., Liu, H-W., Zhou, H.-Y., Wu, Z.-B., Yang, X.-Z., & Xiao, X.-B. (2011). A fault-tolerant algorithm of wireless sensor network based on recoverable nodes. *Intelligent Automation and Soft Computing, 17*, 737–747. doi:10.1080/10798587.2011.10643183

Mann, H.B., & Whitney, D.R. (1947). On a test of whether one of two random variables is stochastically larger than the other. *The annals of mathematical statistics, 18*, 50–60.

Misra, S., Majd, N.E., & Huang, H. (2011). Constrained relay node placement in energy harvesting wireless sensor networks. In Proceedings of the IEEE 8th International Conference on Mobile Adhoc and Sensor Systems (MASS).

Misra, S., Majd, N.E., & Huang, H. (2014). Approximation Algorithms for Constrained Relay Node Placement in Energy Harvesting Wireless Sensor Networks. *IEEE Transactions on Computers, 63*, 2933–2947. doi:10.1109/tc.2013.171

Nandy, S., Yang, X.-S., Sarkar, P.P., & Das, A. (2015). Color Image Segmentation by Cuckoo Search. *Intelligent Automation and Soft Computing, 21*, 673–685. doi:10.1080/10798587.2015.1025480

Nebro, A.J., Luna, F., Alba, E., Dorronsoro, B., Durillo, J.J., & Beham, A. (2008). AbYSS: Adapting scatter search to multiobjective optimization. *IEEE Transactions on Evolutionary Computation, 12*, 439–457. doi:10.1109/Tevc.2007.913109

Nigam, A., & Agarwal, Y.K. (2014). Optimal relay node placement in delay constrained wireless sensor network design. *European Journal of Operational Research, 233*, 220–233. doi:10.1016/j.ejor.2013.08.031

Peiravi, A., Mashhadi, H.R., & Hamed Javadi, S. (2013). An optimal energy-efficient clustering method in wireless sensor networks using multi-objective genetic algorithm. *International Journal of Communication Systems, 26*, 114–126. doi:10.1002/dac.1336

Perez, A.J., Labrador, M., & Wightman, P.M. (2011). A multiobjective approach to the relay placement problem in WSNs. In Proceedings of the Wireless Communications and Networking Conference (WCNC).

Rauf, A., & A. Aleisa, E.A. (2015). PSO based Automated Test Coverage Analysis of Event Driven Systems. *Intelligent Automation and Soft Computing, 21*, 491–502. doi:10.1080/10798587.2014.966479

Sierra, M.R., & Coello, C.A.C. (2005). Improving PSO-based multi-objective optimization using crowding, mutation and∈-dominance. *Proceedings of the Third International Conference on Evolutionary Multi-Criterion Optimization, EMO 2005, Berlin* (pp. 505–519). Germany: Springer Verlag.

Villasenor Alva, J.A. & Estrada, E.G. (2009). A generalization of Shapiro–Wilk's test for multivariate normality. *Communications in Statistics—Theory and Methods, 38*, 1870–1883. doi:10.1080/03610920802474465

Wei, X., Wang, Y., Li, Z., Zou, T., & Yang, G. (2015). Mining Users Interest Navigation Patterns Using Improved Ant Colony Optimization.

Intelligent Automation and Soft Computing, 21, 445–454. doi:10.1080/10798587.2015.1015778

William, L.H., & Hays, W.L. (1975). *Statistics: Probability, Inference, and Decision (2nd Edn)* (pp. 203–270). New York, NY: Holt, Rinehart and Winston.

Xiuzhen Cheng, X.Z., Narahari, B., Simha, R., Maggie Xiaoyan Cheng, M.X.Y., & Liu, D. (2003). Strong minimum energy topology in wireless sensor networks: np-completeness and heuristics. *IEEE Transactions on Mobile Computing, 2*, 248–256. doi:10.1109/TMC.2003.1233530

Xu, K., Hassanein, H., Takahara, G., & Wang, Q. (2010). Relay node deployment strategies in heterogeneous wireless sensor networks. *IEEE Transactions on Mobile Computing, 9*, 145–159. doi:10.1109/TMC.2009.105

Xue, Y., Zhong, S., Ma, T., & Cao, J. (2015). A Hybrid Evolutionary Algorithm for Numerical Optimization Problem. *Intelligent Automation and Soft Computing, 21*, 473–490. doi:10.1080/10798587.2014.962239

Yang, D., Misra, S., Fang, X., Xue, G., & Zhang, J. (2012). Two-tiered Constrained Relay Node Placement in Wireless Sensor Networks: Computational Complexity and Efficient Approximations. *IEEE Transactions on Mobile Computing, 11*, 1399–1411. doi:10.1109/tmc.2011.126

Yazdani, M., Naderi, B., & Mousakhani, M. (2015). A Model and Metaheuristic for Truck Scheduling in Multi-door Cross-dock Problems. *Intelligent Automation and Soft Computing, 21*, 633–644. doi:10.1080/10798587.2015.1014620

Ye, W., Heidemann, J., & Estrin, D. (2002). An energy-efficient MAC protocol for wireless sensor networks. In Proceedings of the Twenty-First Annual Joint Conference of the IEEE Computer and Communications Societies (INFOCOM 2002).

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

Zitzler, E., & Thiele, L. (1999). Multiobjective evolutionary algorithms: A comparative case study and the strength Pareto approach. *IEEE Transactions on Evolutionary Computation, 3*, 257–271. doi:10.1109/4235.797969