A Trajectory-Based Heuristic to Solve a Three-Objective Optimization Problem for Wireless Sensor Network Deployment

Jose M. Lanza-Gutiérrez(✉), Juan A. Gómez-Pulido, and Miguel A. Vega-Rodríguez

Department of Computers and Communications Technologies, Polytechnic School, University of Extremadura, Campus Universitario s/n, 10003 Caceres, Spain
{jmlanza,jangomez,mavega}@unex.es

Abstract. Nowadays, wireless sensor networks (WSNs) are widely used in more and more fields of application. However, there are some important shortcomings which have not been solved yet in the current literature. This paper focuses on how to add relay nodes to previously established static WSNs with the purpose of optimizing three important factors: energy consumption, average coverage and network reliability. As this is an NP-hard multiobjective optimization problem, we consider two well-known genetic algorithms (NSGA-II and SPEA2) and a multiobjective approach of the variable neighborhood search algorithm (MO-VNS). These metaheuristics are used to solve the problem from a freely available data set, analyzing all the results obtained by considering two multiobjective quality indicators (hypervolume and set coverage). We conclude that MO-VNS provides better performance on average than the standard algorithms NSGA-II and SPEA2.

Keywords: Coverage · Energy efficiency · Multiobjective optimization · NSGA-II · SPEA2 · Relay node · Reliability · VNS · Wireless sensor network

1 Introduction

At the moment, Wireless Sensor Networks (WSNs) are one of the most emerging wireless technologies. They are applied in many fields, such as precision agriculture, industrial control, robotic, rescue operations or forest fire detection [18].

A traditional WSN is composed of a set of sensors capturing information (i.e. physical variables), and a sink node collecting all this information [4]. There are some important factors that encourage the use of WSNs, where for other technologies the deployment of the network would be more expensive or impossible. Some of them are the use of power-autonomous low-cost devices and the absence of wires. However, WSNs also have important shortcomings affecting important factors like energy costs and Quality of Service (QoS).

A.I. Esparcia-Alcázar et al. (Eds.): EvoApplications 2014, LNCS 8602, pp. 27–38, 2014.
DOI: 10.1007/978-3-662-45523-4_3

Because of sensors are often powered by batteries, WSNs are particulary sensitive to energy expenditure. The sensors send all the information captured to the collector node, implying an energy cost. In a star topology, this energy consumption is similar in all the sensors. However, in a multi-hop topology is habitual the existence of bottlenecks: some sensors are subject to a higher energy cost. These bottlenecks adversely affect the behavior of the network. With the aim of avoiding this situation, a new type of device specialized in communication tasks called router or relay node was added to WSNs recently [16].

The efficient design of WSNs is defined in the literature as an NP-hard optimization problem [22]. Consequently, non-conventional techniques are often used, such as heuristics and metaheuristics. Heuristics are techniques designed to solve an specific problem. Metaheuristics are procedures to solve very general types of problems. We find two main lines of research for WSNs, works optimizing traditional WSNs, and works adding relay nodes to traditional WSNs, the so-called Relay node Placement Problem (RNPP). Taking the first approach, there are some relevant contributions using heuristics. Cardei et el. [1] split WSNs into disjoint set of sensors, deciding which must be active to optimize the network lifetime. Cheng et al [2] assigned different power transmission levels to the sensors to reduce the energy consumption. Other authors considered metaheuristics from the Evolutionary Computation (EC) for the same purpose. In this line, Konstantinidis and Yang assigned power transmission levels to the sensors as in [11], but optimizing network lifetime and coverage. Hu et al. [10] maximized the network lifetime splitting WSNs (as do [1]). However, this research line has two main shortcomings. Firstly, it is habitual the use of redundant sensors to maximize the network lifetime, implying costly networks. Secondly, network size is limited because of more sensors implies a higher energy cost.

The works taking the second approach try to overcome these shortcomings by adding routers. Beginning with heuristics, Wang et al. [22] considered routers with processing limitations to optimize the energy cost and Han et al. [9] optimized the fault-tolerance. On the other hand, other authors considered EC. Perez et al. [19] optimized the number of routers and the energy expenditure and Zhao and Chen [23] optimized both average path length and energy cost.

Our work follows this second line of research. We add relay nodes to previously established static WSNs in order to optimize three important factors: average energy consumption, average coverage and network reliability. The following contributions are presented in the curse of this paper:

- The three-objective approach for the RNPP is solved by using three different metaheuristics: two well-known genetic algorithms NSGA-II [6] and SPEA2 [24], and a multiobjective version of the Variable Neighborhood Search algorithm (MO-VNS) [8].
- All the results obtained are analyzed in depth thought a widely recognized statistical methodology. Using as quality indicators two multiobjective metrics: hypervolume and set coverage.

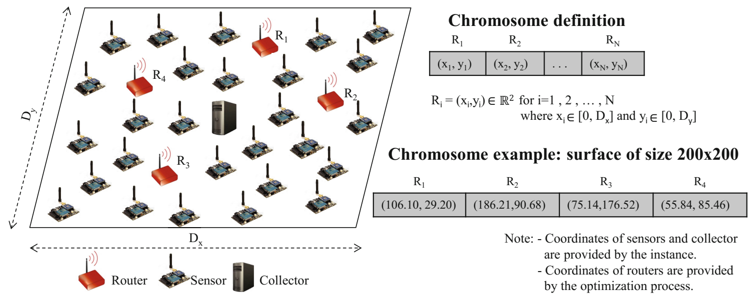

Fig. 1. Network definition considered in the RNPP

– In the current literature, some papers use randomly generated data set or non-public ones. In this work, we consider a freely available data set, implying that this work can be replicated and improved by other authors.

The remainder of this paper is structured as follows. In Section 2, a formal statement of the RNPP is provided. Algorithms used appear in Section 3. Experimental results are discussed in Section 4. Finally, our concluding remarks are left for Section 5.

2 A Realistic Approach for the Relay Node Placement Problem

The WSN considered in the RNPP is composed of three types of wireless static devices placed on the same 2D-surface of size $D_x \times D_y$: a sink node (also called collector node), M sensors and N routers or relay nodes (see Fig. 1). Each sensor obtains information about the environment with a sensibility radius R_s on a regular basis. This information is sent to the sink node, being this node the only connection point of the WSN to the outside. The routers only relay all the received information to the collector node. All the devices communicate among them with a same communication radius R_c. The routers and the collector node have an unlimited power supply, and the sensors are powered by batteries. Thus, a sensor is alive if its battery is not exhausted.

The routing protocol used by sensors and routers is the same. It is based on the minimum-distance path between devices provided by Dijkstra's algorithm [3]. In addition, we consider a perfect synchronization and a perfect medium access, ensuring that there are no collisions among devices.

Let C and S_r be the collector node and the set of routers, respectively, and let $S_s(t)$ be the set of alive sensors at time t. With the aim of modeling the energy expenditure suffered by the sensors, the energy model proposed by A. Konstantinidis et al. [11] is considered. Then, according to this model, the transmission power needed by a sensor $i \in S_s(t)$ to reach another device $j \in S_s(t) \cup S_r \cup C$ at time t is given by

$$P_i(t) = \beta \cdot d_{i,j}^{\alpha} \qquad t > 0, \tag{1}$$

where $\beta > 0$ is the transmission quality parameter, $d_{i,j}$ is the Euclidean distance between i and j, and $\alpha > 0$ is the path loss exponent. Thus, the residual energy of the sensor i at time t is given by

$$E_i(t) = E_i(t-1) - [(r_i(t)+1) \cdot P_i(t) \cdot amp \cdot K], \qquad t > 0, \tag{2}$$

where $r_i(t)$ is the number of packets that the sensor i receives and relays to the collector node at time t, the $+1$ term is the information packet that the sensor i captures at this time and sends, amp is the energy consumption per bit of the power amplifier, and K is the information packet size. Initially, all the sensors start with the same energy charge IEC in their batteries. Hence,

$$E_i(t) = IEC \qquad \forall i \in S_s(t),\ t = 0. \tag{3}$$

When the residual energy of a sensor equals 0, the device cannot capture more information or be linked again. Following this energy model, we assume the energy expenditure depends only on the most expensive task: the sending. The receiving, processing and sensing tasks are considered negligible.

The network lifetime (LF) is an important concept in this type of network. It is the amount of time units over which a WSN is able to provide enough information about its environment. For this purpose, a coverage threshold (CV) is often used. If the coverage provided by the alive sensors is lower than CV, we consider that the network lifetime has come to its end.

In a previous work two important factors were optimized [15]: average energy consumption and average coverage . Such as in [14], in this paper we include a third factor which provides a better realism to this problem definition: network reliability. These three factors are defined as:

- Average energy consumption(AEC, to minimize): It is the average energy expenditure of the sensors over LF (in Joules), that is

$$f_1 = LF^{-1} \left[\sum_{t=1}^{LF} \sum_{i \in S_s(t)} \left(\frac{E_i(t-1) - E_i(t)}{|S_s(t)|} \right) \right], \tag{4}$$

  where $|S_s(t)|$ is the cardinal of the set $S_s(t)$.
- Average coverage(AC, to maximize): It is the percentage of the surface area covered by the sensors over LF. There are two main ways to obtain this value in the literature [21]. Some authors consider that a sensor covers a circumference of radius R_s. Hence the global coverage is the union of the M areas. Other authors place a matrix of binary demand points on the surface, where a demand point equals 1 if there is some alive sensor at a distance lower than R_s, and 0 otherwise. Finally the activated points are

counted. We consider the second approach. Although the first one is a little bit accurate, the second one is less hard to compute. Thus, AC is given by

$$f_2 = LF^{-1}\left[\sum_{t=1}^{LF}\sum_{x=1}^{\lceil D_x\rceil}\sum_{y=1}^{\lceil D_y\rceil}\left(\frac{R_{x,y}(t)}{\lceil D_x\rceil \times \lceil D_y\rceil}\right)\right], \tag{5}$$

where $R_{x,y}(t)$ is the demand point placed at the coordinates (x,y) of the matrix of $\lceil D_x\rceil \times \lceil D_y\rceil$ binary demand points at time t.

– Network reliability(NR, to maximize): It is the average network fault-tolerance, showing the probability that the sensors successfully send information to the sink node. Let Re_i be the reliability of the sensor i defined in [5] as

$$Re_i = 1 - \prod_{l=1}^{P}(1-(1-Err)^{h_l}), \tag{6}$$

where P is the number of disjoint paths between i and the sink node given by Suurballe's Algorithm [20], h_l is the number of hops in the l-th disjoint path, and Err is the local channel error. Thus, NR is defined as

$$f_3 = \sum_{i\in S_s(t)}\left(\frac{Re_i}{M}\right) \qquad t=0. \tag{7}$$

To summarize, the RNPP is defined as an NP-hard multiobjective optimization problem. The objective is to place N routers to optimize a traditional WSN defined by the parameters D_x, D_y, R_s, R_c, IEC, K, CV, α, β, amp, Err and the positions of the collector node and the M sensors.

3 Multiobjective Optimization: The Algorithms Used

As stated before, the RNPP is an NP-hard optimization problem. This type of problem is solved through approximated techniques. Accordingly, we consider three different metaheuristics. NSGA-II and SPEA2 belong to genetic algorithms, a subtype of evolutionary algorithm characterized by encoding their individuals as chromosomes. An individual is a possible solution to the optimization problem. The remainder is a trajectory algorithm, solving methods whose search process follows a trajectory in the search space.

NSGA-II uses two populations P_t and Q_t of the same size PS. P_t saves the parents of generation t, and Q_t saves the offspring generated by individuals in P_t. Initially, P_t is randomly generated and Q_t is empty. So long as the stop condition is not reached, both populations are combined in a new set R_t of size $2PS$. Then, according to both rank and crowding measures, the best PS solutions of R_t are inserted into the new parent population P_{t+1}. Next, a new Q_{t+1} is generated based on P_{t+1}. To this end, and so long as Q_{t+1} is not filled, a pair of individuals are selected from P_{t+1} though binary tournament method. Then, a new individual is generated and inserted into Q_{t+1} through crossover

Algorithm 1. MO-VNS with perturbation mechanism

1: *add a random solution to the emply population* P_v
2: *generate the set of neighborhood structures* N_s
3: **while** *not stop condition* **do**
4:     **while** *there are solutions non − used during the search in* P_v **do**
5:         $a \leftarrow$ *randomly pick a non − used solution from* P_v
6:         $n_{s_k} \leftarrow$ *randomly pick a neighborhood structure,* $k \in 1, \ldots, k_{max},\ n_{s_k} \in N_s$
7:         **while** $k <= k_{max}$ **do**
8:             $\tilde{a} \leftarrow$ *generate a neighborhood solution of* a *in* n_{s_k}*, marking* a *as used*
9:             *add* $\tilde{a}$ *to* P_v *and remove all the dominated solutions*
10:             **if** $\tilde{a} \in P_v$ **then**
11:                 $k \leftarrow 1$ *and* $a \leftarrow \tilde{a}$
12:             **else**
13:                 $k \leftarrow k + 1$
14:             **end if**
15:         **end while**
16:     **end while**
17:     *perform perturbation in* P_v *to avoid local minima*
18:     *reset all the marks of* P_v
19: **end while**

and mutation operators,. As crossover operator, we consider the usual one-point crossover. As mutation operator, we assume a greedy strategy: router coordinates are randomly changed, but only changes that provide a better individual are accepted. The same encoding is used for the three algorithms. A chromosome is a 2D-coordinate list of M routers (see Fig. 1).

SPEA2 uses an auxiliary population $\overline{P_t}$ where the best solutions are saved along generations, and a regular population P_t with sizes $\overline{PS}$ and PS respectively. Initially, P_t is randomly generated and $\overline{P_t}$ is empty. So long as the stop condition is not reached, the fitness value for each individual in $P_t \cup \overline{P_t}$ is obtained. This fitness is based on the Pareto dominance concept and additional density information. The best solutions according to this fitness are inserted into the new $\overline{P_{t+1}}$. Next, a new P_{t+1} is generated based on $\overline{P_{t+1}}$, using the binary tournament, mutation and crossover strategies as discussed for NSGA-II.

MO-VNS performs local searches by using neighborhood structures. Let a neighborhood structure be the maximum displacement that a router experiences during the local search. Thus, the set of neighborhood structures N_s is given by

$$N_s = \left\{ n_{s_k} \in \mathbb{R} \ / \ n_{s_k} = \frac{min(D_x, D_y) * k}{dv * k_{max}} \right\} \qquad n_{s_k} < n_{s_{k+1}}, \tag{8}$$

for $k = 1, \ldots, k_{max}$, where k_{max} is the number of neighborhood structures, dv is a factor which delimites the displacement, and $min(D_x, D_y)$ provides the minimum value between D_x and D_y.

As outlined in Algorithm 1, MO-VNS uses a population P_v where only non-dominated individuals are kept. Each individual in P_v has a flag which determines if the solution was used during the search. Initially, a random solution is added to P_v (line 1). Then, so long as the stop condition is not reached, a non-used solution $a \in P_v$ and a neighborhood structure $n_{s_k} \in N_s$ are randomly selected (lines 5-6). Next, a new solution is generated through a local search using $a \in P_v$ as base solution (line 8), marking $a \in P_v$ as used. The local search

Table 1. Instances used in this paper

Instance	$D_x \times D_y$	*M*	*HO_AEC*	*HO_AC*	*HO_NR*
100x100_15_30	100x100	15	0.1091	89.24%	95.67%
200x200_15_30	200x200	57	0.2791	87.10%	93.23%
300x300_15_30	300x300	128	0.4225	76.44%	85.28%

Table 2. Hypervolume reference points

Instance	*Ref_AEC*		*Ref_AC*		*Ref_NR*	
	ideal	*nadir*	*ideal*	*nadir*	*ideal*	*nadir*
100x100_15_30	0.02	0.10	1.00	0.60	1.00	0.50
200x200_15_30	0.10	0.30	1.00	0.60	1.00	0.50
300x300_15_30	0.04	0.50	1.00	0.60	1.00	0.50

Table 3. Parametric sweep

NSGA-II		
Parameter	Value	Range
Mutation	0.80	0.05,0.10,0.15,...,0.95
Crossover	0.80	0.05,0.10,0.15,...,0.95
SPEA2		
Parameter	Value	Range
Mutation	0.70	0.05,0.10,0.15,...,0.95
Crossover	0.60	0.05,0.10,0.15,...,0.95
MO-VNS		
Parameter	Value	Range
Mutation	0.10	0.05,0.1,0.15,...,0.95
k_{max}	10	3,4,5,6,7,8,...,14
dv	2	1,1.5,2,2.5,3,3.5,...,6.5

is given by

$$R_{\tilde{a}_z} = R_{a_z} + \left(\frac{n_{s_k}}{2} - rand(n_{s_k})\right) \qquad n_{s_k} \in N_s,\ k \in 1, \ldots, k_{max}, \tag{9}$$

for $z = 1, \ldots, N$, where R_{a_z} and $R_{\tilde{a}_z}$ are the routers placed on the z-th gene of the solutions a and $\tilde{a}$ respectively, and $rand(n_{s_k})$ is a random number between 0 and n_{s_k}. Next, the new solution is added to P_v, removing all the dominated solutions (line 9). If $\tilde{a} \in P_v$, the local search provided a good solution, and then the local search is repeated again using a k value of 1 and taking $\tilde{a}$ as base solution (line 11). Otherwise, k is increased, so long as k takes the maximum value k_{max} (line 13). Once all the solutions are explored, the marks are reset, and then all the individuals are eligible for a new selection again (line 18). Before starting the search process again, a perturbation mechanism is performed to avoid local minima (line 17). To this end, the greedy mutation operator discussed for NSGA-II and SPEA2 is used for each solution in P_v.

4 Experimental Methodology

As stated before, non-public data set was found that fit this problem definition. Hence, in order to study the performance of the metaheuristics, we consider a data set defined by ourselves in [13]. This data set is composed of three traditional WSNs (a set of sensors and a collector node). The number of sensors is the minimum value to cover the whole surface, being placed by a monoobjective genetic algorithm optimizing the coverage offered by the sensors (see Table 1). The collector node is placed in the center of the scenario. We assume the following network parameters: R_c = 30m and $R_s = 15m$ from [17], $K = 128KB$, $CV = 70\%$, $Err = 10\%$, and the energy parameters $EC = 5J$, $\alpha = 2$, $\beta = 1$ and $amp = 100pJ/bit/m^2$ from [12]. In a previous work [15], two different R_c values were assumed, 30 and 60 meters. However, it makes no sense to consider $R_c = 60m$ for our problem definition, since the network reliability is almost 100% for all the cases.

This data set is optimized by adding relay nodes. We assume the addition of these devices increases the network cost. Hence, we decide not to include

Table 4. Hypervolume and standard deviation for each algorithm and test case

NSGA-II ($\overline{Hyp}$ %, *std.dev*)					
Test case	Evaluations (Stop condition)				
Instance (routers)	50 000	100 000	200 000	300 000	400 000
100x100_15_30(2)	41.01%, 0.0030	41.25%, 0.0024	41.47%, 0.0002	41.48%, 0.0001	41.48%, 0.0000
100x100_15_30(3)	53.54%, 0.0050	54.15%, 0.0018	54.46%, 0.0019	54.56%, 0.0011	54.63%, 0.0005
200x200_15_30(2)	32.49%, 0.0100	33.22%, 0.0042	33.53%, 0.0025	33.64%, 0.0018	33.74%, 0.0021
200x200_15_30(4)	41.46%, 0.0180	43.21%, 0.0167	45.07%, 0.0109	45.57%, 0.0134	45.96%, 0.0116
200x200_15_30(6)	48.75%, 0.0345	53.12%, 0.0193	55.65%, 0.0161	57.00%, 0.0168	57.68%, 0.0156
200x200_15_30(9)	57.14%, 0.0254	61.82%, 0.0223	65.57%, 0.0211	67.45%, 0.0194	68.31%, 0.0174
300x300_15_30(6)	28.35%, 0.0074	29.44%, 0.0068	30.42%, 0.0061	30.81%, 0.0060	31.05%, 0.0057
300x300_15_30(12)	29.84%, 0.0068	31.53%, 0.0100	32.86%, 0.0098	33.81%, 0.0107	34.37%, 0.0112
300x300_15_30(18)	31.26%, 0.0061	32.92%, 0.0088	34.30%, 0.0107	34.99%, 0.0097	35.41%, 0.0099
300x300_15_30(24)	33.40%, 0.0060	34.99%, 0.0137	36.51%, 0.0157	37.22%, 0.0133	37.86%, 0.0132
SPEA2 ($\overline{Hyp}$ %, *std.dev*)					
Test case	Evaluations (Stop condition)				
Instance (routers)	50 000	100 000	200 000	300 000	400 000
100x100_15_30(2)	41.07%, 0.0021	41.24%, 0.0016	41.31%, 0.0015	41.46%, 0.0002	41.46%, 0.0002
100x100_15_30(3)	53.76%, 0.0038	54.27%, 0.0029	54.56%, 0.0011	54.61%, 0.0007	54.64%, 0.0007
200x200_15_30(2)	32.56%, 0.0054	32.88%, 0.0053	33.21%, 0.0032	33.38%, 0.0031	33.47%, 0.0026
200x200_15_30(4)	42.41%, 0.0150	44.03%, 0.0148	45.03%, 0.0153	45.54%, 0.0141	45.72%, 0.0130
200x200_15_30(6)	53.35%, 0.0180	55.98%, 0.0179	57.53%, 0.0072	58.57%, 0.0124	**59.09**%, 0.0084
200x200_15_30(9)	61.49%, 0.0179	65.42%, 0.0200	67.85%, 0.0184	68.99%, 0.0165	**69.70**%, 0.0132
300x300_15_30(6)	29.45%, 0.0062	30.55%, 0.0071	31.19%, 0.0072	31.54%, 0.0068	**31.78**%, 0.0055
300x300_15_30(12)	31.58%, 0.0071	33.19%, 0.0106	34.62%, 0.0116	35.41%, 0.0113	**36.00**%, 0.0115
300x300_15_30(18)	33.44%, 0.0089	35.22%, 0.0086	36.73%, 0.0092	37.68%, 0.0080	38.34%, 0.0093
300x300_15_30(24)	35.43%, 0.0077	37.04%, 0.0094	38.63%, 0.0076	39.45%, 0.0082	40.20%, 0.0093
MO-VNS ($\overline{Hyp}$ %, *std.dev*)					
Test case	Evaluations (Stop condition)				
Instance (routers)	50 000	100 000	200 000	300 000	400 000
100x100_15_30(2)	41.76%, 0.0003	41.79%, 0.0002	41.81%, 0.0002	41.82%, 0.0002	**41.82**%, 0.0001
100x100_15_30(3)	54.96%, 0.0037	55.21%, 0.0037	55.31%, 0.0019	55.56%, 0.0033	**55.61**%, 0.0033
200x200_15_30(2)	31.76%, 0.0241	34.04%, 0.0088	34.60%, 0.0126	35.22%, 0.0080	**35.92**%, 0.0017
200x200_15_30(4)	42.81%, 0.0189	44.38%, 0.0184	45.24%, 0.0165	45.78%, 0.0155	**46.14**%, 0.0166
200x200_15_30(6)	54.46%, 0.0197	56.37%, 0.0146	56.99%, 0.0127	57.27%, 0.0139	57.47%, 0.0136
200x200_15_30(9)	63.48%, 0.0155	64.21%, 0.0116	65.33%, 0.0104	65.87%, 0.0109	66.45%, 0.0102
300x300_15_30(6)	30.36%, 0.0043	30.93%, 0.0057	31.19%, 0.0050	31.34%, 0.0058	31.40%, 0.0057
300x300_15_30(12)	33.82%, 0.0063	34.56%, 0.0071	35.31%, 0.0070	35.68%, 0.0056	35.83%, 0.0056
300x300_15_30(18)	37.04%, 0.0068	37.83%, 0.0061	38.48%, 0.0056	38.83%, 0.0038	**39.01**%, 0.0048
300x300_15_30(24)	40.14%, 0.0098	40.91%, 0.0072	41.48%, 0.0067	41.79%, 0.0054	**41.95**%, 0.0048

more than 20% of routers regarding to the number of sensors. Thus, 10 different test cases are defined as shown Table 4. Each test case follows the notation *instance_name(number of routers)*.

Before optimizing the data set, the three algorithms were configured by a parametric sweep [15]. The range of values considered for each parameter is shown in Table 3, as well as the configuration obtained through this tuning. After this step, 31 independent runs are performed for each algorithm in order to obtain statistical validity. With the purpose of studying the convergence of the algorithms, five different stop conditions are considered: 50 000, 100 000, 200 000, 300 000 and 400 000 evaluations. The solutions obtained are evaluated through hypervolume metric, considering the experimental reference points shown in Table 2. Thus, average hypervolumes and standard deviation for each

Table 5. P-values obtained through Wilcoxon-Mann-Whitney's test comparing among hypervolumes

	MO-VNS vs SPEA2					SPEA2 vs NSGAII				
Instance (routers)	50 000	100 000	200 000	300 000	400 000	50 000	100 000	200 000	300 000	400 000
100x100_15_30(2)	0.0000	0.0000	0.0000	0.0000	0.0000	0.2505	0.9060	1.0000	1.0000	1.0000
100x100_15_30(3)	0.0000	0.0000	0.0000	0.0000	0.0000	0.0486	0.0182	0.0188	0.0370	0.1032
200x200_15_30(2)	0.3431	0.0000	0.0000	0.0000	0.0000	0.3920	0.9938	0.9999	0.9995	0.9999
200x200_15_30(4)	0.1376	0.2843	0.3086	0.2215	0.0871	0.0273	0.0410	0.5530	0.6136	0.7949
200x200_15_30(6)	0.0094	0.1815	0.9750	0.9996	1.0000	0.0000	0.0000	0.0000	0.0001	0.0001
200x200_15_30(9)	0.0000	0.9920	1.0000	1.0000	1.0000	0.0000	0.0000	0.0000	0.0008	0.0005
300x300_15_30(6)	0.0000	0.0099	0.3646	0.8787	0.9953	0.0000	0.0000	0.0000	0.0000	0.0000
300x300_15_30(12)	0.0000	0.0000	0.0079	0.2257	0.7012	0.0000	0.0000	0.0000	0.0000	0.0000
300x300_15_30(18)	0.0000	0.0000	0.0000	0.0000	0.0010	0.0000	0.0000	0.0000	0.0000	0.0000
300x300_15_30(24)	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000
	MO-VNS vs NSGA-II					**SUMMARY**				
Instance (routers)	50 000	100 000	200 000	300 000	400 000	50 000	100 000	200 000	300 000	400 000
100x100_15_30(2)	0.0000	0.0000	0.0000	0.0000	0.0000	MO-VNS	MO-VNS	MO-VNS	MO-VNS	MO-VNS
100x100_15_30(3)	0.0000	0.0000	0.0000	0.0000	0.0000	MO-VNS	MO-VNS	MO-VNS	MO-VNS	MO-VNS
200x200_15_30(2)	0.4691	0.0000	0.0000	0.0000	0.0000	NONE	MO-VNS	MO-VNS	MO-VNS	MO-VNS
200x200_15_30(4)	0.0038	0.0148	0.2997	0.2215	0.2299	NONE	NONE	NONE	NONE	NONE
200x200_15_30(6)	0.0000	0.0000	0.0006	0.2223	0.6455	MO-VNS	NONE	SPEA2	SPEA2	SPEA2
200x200_15_30(9)	0.0000	0.0000	0.6764	0.9996	1.0000	MO-VNS	SPEA2	SPEA2	SPEA2	SPEA2
300x300_15_30(6)	0.0000	0.0000	0.0000	0.0009	0.0203	MO-VNS	MO-VNS	NONE	NONE	SPEA2
300x300_15_30(12)	0.0000	0.0000	0.0000	0.0000	0.0000	MO-VNS	MO-VNS	MO-VNS	NONE	NONE
300x300_15_30(18)	0.0000	0.0000	0.0000	0.0000	0.0000	MO-VNS	MO-VNS	MO-VNS	MO-VNS	MO-VNS
300x300_15_30(24)	0.0000	0.0000	0.0000	0.0000	0.0000	MO-VNS	MO-VNS	MO-VNS	MO-VNS	MO-VNS

test case, stop condition and algorithm are shown in Table 4. The highest hypervolumes for 400 000 evaluations are in bold.

Analyzing Table 4, we may note that MO-VNS seems to provide better results. However, we do not known if the differences are significant. To this end, we assume a widely used statistical methodology. The first step is to study if the data follow a normal distribution through $Shapiro$ - $Wilk's$ and $Kolmogrov$ - $Smirnov$ - $Lilliefors's$ tests with the hypothesis: H_0 if data follow a normal distribution, and H_1 otherwise. P-values lower than 0.05 were obtained for all the cases. Hence, we cannot assume data follow a gaussian distribution. Consequently, the median (Me) must be used as average value. The second step is to check if there are differences among the algorithms. To this end, $Wilcoxon$ - $Mann$ - $Whitney$'s test (samples do not follow a normal distribution and are independent) is used with the hypothesis: H_0 Me_i is worse or equal than Me_j, and H_1 Me_i is better than Me_j, with $i = 1, 2, 3$, $j = 2, 3$, $i < j$, 1=MO-VNS, 2=SPEA2 and 3=NSGA-II. The P-values obtained are shown in Table 5. Values exceed 0.05 are shaded, because of differences are considered not significant.

Based on these p-values, the algorithm which provide the best performance in each case appears in the part summary of Table 5. Analyzing this summary, we observe as MO-VNS provides the best results in complex and simple test cases, but it does not in medium ones. Furthermore, we check as MO-VNS is quicker than NSGA-II and SPEA2 on average. It is necessary a less number of evaluations to get similar results, but when the number of evaluations is

Table 6. Average set coverage C(A,B) among algorithms

A	MO-VNS		NSGA-II		SPEA2	
Instance (routers) B	NSGA-II	SPEA2	SPEA2	MO-VNS	NSGA-II	MO-VNS
100x100_15_30(2)	98.56%	98.29%	63.26%	0.00%	75.81%	0.00%
100x100_15_30(3)	87.89%	89.89%	39.95%	3.17%	33.10%	1.72%
200x200_15_30(2)	72.29%	76.06%	49.24%	12.36%	42.85%	14.50%
200x200_15_30(4)	70.57%	72.56%	43.05%	9.76%	43.68%	8.17%
200x200_15_30(6)	76.56%	45.83%	17.04%	15.43%	77.67%	30.40%
200x200_15_30(9)	40.41%	17.38%	5.89%	35.33%	77.39%	63.40%
300x300_15_30(6)	85.74%	56.88%	17.67%	4.67%	61.09%	18.19%
300x300_15_30(12)	73.02%	50.02%	11.89%	12.70%	71.56%	31.04%
300x300_15_30(18)	92.48%	67.69%	8.70%	5.53%	75.78%	16.91%
300x300_15_30(24)	96.86%	86.30%	17.94%	0.60%	67.81%	13.51%
Partial average	**79.44%**	**66.09%**	**27.46%**	**9.95%**	**62.68%**	**19.78%**
Average	**72.76%**		**18.71%**		**41.23%**	

increased, this advantage is reduced. On average, MO-VNS is the best a 62%, SPEA2 a 16%, NSGA-II a 0%, and none of them a 22%.

In addition to hypervolume, we consider the set coverage C(A,B). That is the percentage of solutions from the algorithm B that are weakly dominated by A. To this end, we obtain the set coverage between each pair of algorithms, test case and stop condition. For this purpose, we use the medium front of the distribution of 31 samples. The average set coverage between each pair of algorithms during the 400 000 evaluations is shown in Table 6. Analyzing this table, we reach similar conclusions as for hypervolume. MO-VNS provides the best coverage relation (72.76%), followed by SPEA2 (41.23%) and in the tail NSGA-II (18.71%).

Finally, some implementation details. The algorithms were programmed by ourselves in C++, using the Lemon library for graphs (*http://lemon.cs.elte.hu*). The IBM SPSS software was used to get the *Shapiro-Wilk's* and *Kolmogrov-Smirnov-Lilliefors's* tests. Finally, the *Wilcoxon−Mann − Whitney's* test and hypervolume were taken from [7].

5 Final Remarks

In this paper, we study the addition of relay nodes to previously established WSNs, with the aim of optimizing three important factors: average energy consumption, average coverage and network reliability. This is the so-called relay node placement problem, which is an NP-hard optimization problem. To solve this problem, we consider three different metaheuristics, two well-known genetic algorithms (NSGA-II and SPEA2), an a novel multiobjective approach of the VNS. These algorithms are used to optimize a freely available data set. Analyzing all the obtained results in depth, and using two known multiobjective indicator: hypervolume and set coverage. As a result, MO-VNS provides the best behavior on average, followed by SPEA2, and in the tail NSGA-II.

As future lines of research, it would be interesting to consider other metaheuristics. One of our aim is to find an algorithm providing good results in general terms. In addition, it would be a good idea to consider a greater number of test cases, and conduct real world-experiments.

Acknowledgments. This work was partially funded by the Spanish Ministry of Economy and Competitiveness and the ERDF (European Regional Development Fund), under the contract TIN2012-30685 (BIO project), and by the Government of Extremadura, with the aid GR10025 to the group TIC015.

References

1. Cardei, M., Du, D.Z.: Improving wireless sensor network lifetime through power aware organization. Wireless Networks **11**, 333–340 (2005)
2. Cheng, X., Narahari, B., Simha, R., Cheng, M., Liu, D.: Strong minimum energy topology in wireless sensor networks: Np-completeness and heuristics. IEEE Transactions on Mobile Computing **2**, 248–256 (2003)
3. Cormen, T.H., Leiserson, C.E., Rivest, R.L., Stein, C.: Introduction to Algorithms, 3rd edn. The MIT Press (2009)
4. Dargie, W., Poellabauer, C.: Fundamentals of Wireless Sensor Networks: Theory and Practice. Wiley (2010)
5. Deb, B., Bhatnagar, S., Nath, B.: Reliable information forwarding using multiple paths in sensor networks. In: Proceedings of IEEE LCN, pp. 406–415 (2003)
6. Deb, K., Pratap, A., Agarwal, S., Meyarivan, T.: A fast elitist multi-objective genetic algorithm: Nsga-ii. IEEE Transactions on Evolutionary Computation **6**, 182–197 (2000)
7. Fonseca, C., Knowles, J., Thiele, L., Zitzler, E.: Performance assessment tool suite. http://www.tik.ee.ethz.ch/pisa/?page=assessment.php
8. Geiger, M.J.: Randomised variable neighbourhood search for multi objective optimisation. In: Proceedings of the 4th EU/ME Workshop 0809.0271, pp. 34–42 (2008)
9. Han, X., Cao, X., Lloyd, E.L., Shen, C.C.: Fault-tolerant relay node placement in heterogeneous wireless sensor networks. IEEE Transactions on Mobile Computing **9**, 643–656 (2010)
10. Hu, X.M., Zhang, J., Yu, Y., Chung, H.H., Li, Y.L., Shi, Y.H., Luo, X.N.: Hybrid genetic algorithm using a forward encoding scheme for lifetime maximization of wireless sensor networks. IEEE Transactions on Evolutionary Computation **14**, 766–781 (2010)
11. Konstantinidis, A., Yang, K., Zhang, Q.: An evolutionary algorithm to a multi-objective deployment and power assignment problem in wireless sensor networks. In: Proceedings of IEEE GLOBECOM, pp. 1–6 (2008)
12. Konstantinidis, A., Yang, K.: 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 (2011)
13. Lanza-Gutierrez, J.M., Gomez-Pulido, J.A., Vega-Rodriguez, M.A.: Instance sets for optimization in wireless sensor networks. http://arco.unex.es/wsnopt (2011)
14. Lanza-Gutierrez, J.M., Gomez-Pulido, J.A., Vega-Rodriguez, M.A.: A new realistic approach for the relay node placement problem in wireless sensor networks by means of evolutionary computation. Ad Hoc and Sensor Wireless Networks (2013) (accepted)

15. Lanza-Gutiérrez, J.M., Gómez-Pulido, J.A., Vega-Rodríguez, M.A., Sánchez-Pérez, J.M.: Relay Node Positioning in Wireless Sensor Networks by Means of Evolutionary Techniques. In: Kamel, M., Karray, F., Hagras, H. (eds.) AIS 2012. LNCS, vol. 7326, pp. 18–25. Springer, Heidelberg (2012)
16. Lloyd, E.L., Xue, G.: Relay node placement in wireless sensor networks. IEEE Transactions on Computers **56**, 134–138 (2007)
17. Martins, F., Carrano, E., Wanner, E., Takahashi, R., Mateus, G.: A hybrid multiobjective evolutionary approach for improving the performance of wireless sensor networks. IEEE Sensors Journal **11**, 545–554 (2011)
18. Mukherjee, J.Y.B., Ghosal, D.: Wireless sensor network survey. Computer Networks **52**, 2292–2330 (2008)
19. Perez, A., Labrador, M., Wightman, P.: A multiobjective approach to the relay placement problem in wsns. Proceedings of IEEE WCNC **1**, 475–480 (2011)
20. Suurballe, J.W.: Disjoint paths in a network. Networks **4**, 125–145 (1974)
21. Wang, B.: Coverage problems in sensor networks: A survey. ACM Comput. Surv. 43, 32:1–32:53 (2011)
22. Wang, Q., Xu, K., Takahara, G., Hassanein, H.: Device placement for heterogeneous wireless sensor networks: Minimum cost with lifetime constraints. IEEE Transactions on Wireless Communications **6**, 2444–2453 (2007)
23. Zhao, C., Chen, P.: Particle swarm optimization for optimal deployment of relay nodes in hybrid sensor networks. Proceedings of IEEE CEC. **1**, 3316–3320 (2007)
24. Zitzler, E., Laumanns, M., Thiele, L.: Spea 2: Improving the strength pareto evolutionary algorithm. Tech. rep., Computer Engineering and Networks Laboratory (TIK), ETH Zurich (2001)