Diseño Inteligente de Redes Inalámbricas de Comunicación
Conserjería de Innovación, Ciencia y Empresa
P07-TIC-03044
http://diricom.lcc.uma.es

Málaga, 2010

Informe Ejecutivo

TÍTULO: WSN-1.1: Un operador de búsqueda local para el problema de la disposición óptima de sensores

RESUMEN: Este informe presenta y analiza un operador de búsqueda local multiobjetivo denominado PACO (*Proximity Avoidance Coverage-preserving Operator*) específicamente diseñado para la resolución del problema de la disposición óptima de sensores (*Wireless Sensor Network Layout* o WSNL).

OBJETIVOS:

1. Definir en detalle el operador PACO.
2. Describir la integración de PACO en MOEAs del estado del arte: NSGA-II, SPEA2, PAES y MOCell.
3. Analizar los resultados obtenidos.

CONCLUSIONES:

1. PACO es un operador muy eficiente para el problema WSNL, mejorando en todos los casos la efectividad de los algoritmos en los que se integra.

RELACIÓN CON ENTREGABLES:

PRE: WSN-1.0

CO: —

Málaga, 2010

Executive Summary

TITLE: WSN-1.1: A local search operator for the WSN layout problem

ABSTRACT: This report presents and analyzes a multiobjective local search operator called PACO (*Proximity Avoidance Coverage-preserving Operator*), which has been specifically engineered for solving the Wireless Sensor Network Layout problem (WSNL).

GOALS:

1. Fully defining PACO.
2. Describing the integration of PACO within state-of-the-art MOEAs: NSGA-II, SPEA2, PAES, and MOCell.
3. Analyzing the obtained results.

CONCLUSIONS:

1. PACO offers a robust enhancement to the performances of the optimization algorithms.

RELATION WITH DELIVERABLES:

PRE: WSN-1.0

CO: —

A local search operator for the WSN layout problem

DIRICOM

2010

1. Introduction

Wireless Sensor Networks (WSNs) have become a hot topic in research ([1, 4]). Their capabilities for monitoring large areas, accessing remote places, reacting in real-time, together with their relative ease of use have brought scientists a whole new horizon of possibilities. WSNs have so far been employed in many applications ([5]): militar activities such as reconnaissance, surveillance and target acquisition, environmental activities such as forest fire prevention, geophysical activities such as volcano activity study, biomedical purposes such as health data monitoring or artificial retina, or civil engineering such as structural health measurement. Their uses increase by the day and their potential applications seem boundless. The wide variety of applications results in a wide variety of networks bearing different constraints and having different features, yet most of them share some common issues that allow them to be treated homogenously.

One of the main features of WSNs is their geographical ubiquity, this makes the deployment of the nodes a critical task ([18]). The coverage of the network, which depends directly on the positions of the nodes, is one such feature. For instance, in the countersniper system ([14]), the physical distribution of the nodes determine their capability to locate the shooter. In a forest fire prevention, the origin and evolution of the fire can also be known if the nodes are properly deployed ([16]).

Another feature of the uttermost importance in WSNs, that also depends on the node deployment, is node energy consumption. In most scenarios, it is practically unfeasible to substitute nodes or recharge their energy: the high number of nodes or the hostility of the environment they are deployed in makes the task impossible. However, WSNs should work for the longest possible time. This causes energy saving to be one of the principal policies in a WSN in order to increase the network lifetime. The main source of energy consumption for WSNs is widely considered to be wireless communication ([10, 15]), which depends on the communication structure of the network, which in turn depends on the node deployment. Therefore, an optimal layout of nodes involves considering several conflicting design objectives. In the adopted approach, these objectives are the network coverage, the lifetime, and the cost of the network (taken as the number of nodes). Thus, the problem at hand is a multi-objective optimization problem.

The WSN deployment or layout problem (WSNL problem for short), which was proven to be NP-complete in ([21]), is a complex task that has received much attention in the literature. The NP-completeness of the WSNL problem makes using metaheuristics ([2]) mandatory so as to deal with the increasingly-sized, real-world instances in affordable times. The point is that, even though metaheuristics have been used to some extent, the use of these optimization techniques is limited to almost canonical forms of the algorithms, with little-to-no adaptation to the problem particularities. Yet the use of specific problem knowledge is an important issue, and should not be overlooked.

Therefore, the main contributions of this work are as follows. First, a new local improvement operator, the "Proximity Avoidance Coverage-preserving Operator"(PACO) [17], is presented here that takes advantage of specific problem knowledge to solve the WSNL problem. The operator is characterized and its parameters are tuned. Second, it is combined with different state-of-the-art multi-objective optimization techniques, and its effectivity and robustness are assessed. Finally, a scalability study is carried out on the problem instance size (size of the terrain, number of nodes), demonstrating not only that the operator scales well, but also that its efficiency increases for larger problem instances. This last effect is specially important, since the number of nodes of a WSN is expected to grow in the future.

2. The PACO operator

This paper presents a new operator for local improvement in a WSN conceived to be integrated into an optimization algorithm: the "Proximity Avoidance Coverage-preserving Operator"(PACO). The basis of its functioning is identifying locally suboptimal configurations and trying to fix them.

2.1. Operator description

It is understandable that, for the purpose of an efficient WSN deployment, having nodes too close to one another produces inefficiency due to two reasons:

- An extra node is deployed (increased cost) that provides little-to-no coverage improvement (since most of its sensing area is already covered by the other node).
- An extra information packet (reduced energy efficiency) containing the extra node's data has to be relayed.

Thus, the purpose of PACO is to replace the pair of nodes that are close to one another by a single node, provided that this single node can safely replace them:

- The node guarantees that the area covered by the two initial nodes is still covered.
- The connectivity of the WSN is maintained.

Thus PACO has to find an "equivalent deployment area"for the node pair, such that any node placed inside this area is capable of maintaining both the coverage and connectivity of the network after the pair has been removed. This area is found as the intersection of two zones: the "coverage preserving zone", which is the area where a single node guarantees coverage, and the "connectivity preserving zone", which is the area where a single node maintains the network connectivity.

It has to be pointed out that node position and covered area points are subject to a reciprocity property. If a sensor node covers a disk-shaped area around it, then any given terrain point can be covered by a sensor node placed anywhere inside that same disk-shaped area around it. This property shall be used to define a reciprocal WSN whose coverage will identify the coverage equivalent area. The same property holds for the connectivity.

The operation of PACO can be summed up in the following steps:

1. Choose a pair of close nodes. The PACO operator first explores the whole WSN in search for all pairs of close nodes; this can be considered as a preliminary step. A *threshold* parameter defines which pairs of nodes are considered to be *close*: all nodes n_a, n_b whose Euclidean distance is below it. This threshold value should typically be some fraction of R_{SENS}.

2. Obtain the "coverage preserving zone"for that pair. To do so, PACO identifies the area that is exclusively covered by the selected pair (note that the connectivity constraint is not taken into account here). A reciprocal WSN is then created with a node in every terrain point of this area, and the coverage of this reciprocal network is computed; the area that is covered by all the nodes in the reciprocal WSN is the "coverage preserving zone". Thus, a single node placed in this zone can effectively replace the selected pair in terms of coverage.

3. Obtain the "connectivity preserving zone"for that pair. Regarding the connectivity, the node has to fulfill the following constraints:

   - All children nodes of the two nodes removed must be within communicating range of the placed node.
   - At least one of the parent nodes must be within communicating range of the placed node.

   To locate this "connectivity preserving zone"the same principle as before is applied: each child and each parent defines a disk-shaped connectivity zone around itself (with radius R_{COMM}). The intersection or ovelap zone (if any) of all the zones defined by the children guarantees that a single node will keep all the children connected. The union of all the zones defined by the parent nodes guarantees that at least one parent one is connected. The final "connectivity preserving zone"is the intersection of the children and parent zones.

4. Obtain the "equivalent deployment area.ªs the intersection of both the coverage and connectivity preserving zones.

5. If the "equivalent deployment area"is empty, i.e., no overlap is found between the two previous zones, the two removed nodes must be restored and the operator does nothing. Otherwise, when there is an overlap zone (non-empty "equivalent deployment area"), then a single node is placed inside it that effectively replaces the two initially chosen nodes.

The general PACO procedure is an iterative procedure (Algorithm 1). The steps above are performed for each pair of close nodes found in the WSN.

2.2. PACO formal specification

A formal description of PACO's operation is as follows. Let T be the set of terrain points p (the discretized terrain grid), and let WSN be the points where a sensor node is deployed ($WSN \subseteq T$). Assume the functions $converage()$ that for each node $n \in WSN$ returns the set of points in T covered by that node, $parentNodes()$ that for each node $n \in WSN$ returns the set of nodes in WSN that are parent nodes of n, and $childNodes()$ that for each node $n \in WSN$ returns the set of nodes in WSN that are children nodes of n. Select a pair of nodes n_a and n_b such that $n_a, n_b \in WSN$ and $||n_a - n_b|| < threshold$.

Algorithm 1 Pseudocode for PACO.

```
input: a WSN layout wsn = n_1 n_2 ... n_k, n_i ∈ WSN, a threshold value th
wsnBackup ← wsn // Store a copy of the current layout
stop ← false
for All (n_a, n_b) ← NodePair(wsn) do
  if NearbyNodes(n_a,n_b, th) then
    CovEq ← ComputeCovEq(wsn,n_a,n_b) // Step 1
    ConEq ← ComputeConEq(wsn,n_a,n_b) // Step 2
    CovConEq ← CovEq ∩ ConEq          // Step 3
    if CovConEq ≠ ∅ then
      n_p ← ChooseNode(wsn,CovConEq)
      wsn ← Remove(wsn,n_a,n_b)
      wsn ← Deploy(wsn,n_p)
      Evaluate(wsn)
    end if
  end if
  if NodesDeployed(wsn) < NodesDeployed(wsnBackup) &
    EnergyConsumption(wsn) < EnergyConsumption(wsnBackup) then
    wsnBackup ← wsn // wsn dominates wsnBackup
  else
    wsn ← wsnBackup // restore the assignment
  end if
end for
return wsnBackup
output: a possibly improved WSN layout
```

- Step 1 Define E as the set of points covered only by $\{n_a, n_b\}$, i.e. $p \in E \equiv p \in coverage(n_a) \cup coverage(n_b)$; $\forall n \in WSN, n \neq n_a, n_b$, $p \notin \cup coverage(n)$. Find the set of points $CovEq$ that guarantee coverage to the set E: $n \in CovEq \equiv \forall p \in T : p \in E \rightarrow p \in coverage(n)$.

- Step 2 Define the sets P and C such that: $P = parentNodes(n_a) \cup parentNodes(n_b)$ and $C = childNodes(n_a) \cup childNodes(n_b)$. Then find the set $ConEq$ that maintains the connectivity of the network: $n \in ConEq \equiv \forall n_c \in WSN : n_c \in C \rightarrow n_c \in childNodes(n), \exists n_p \in P : n_p \in parentNodes(n)$.

- Step 3 Define $CovConEq$ as the set of points that guarantee both coverage and connectivity: $CovConEq = CovEq \cap ConEq$.

Then as long as $CovConEq \neq \varnothing$, a single sensor placed in any $n \in CovConEq$ may replace the pair $\{n_a, n_b\}$ without loss of coverage or connectivity.

3. Experimentation

A set of experiments are conducted to assess the performance of PACO on different scenarios. The base problem instance is defined with the following properties:

- Square terrain $250 \times 250\,m^2$
- Maximum number of nodes 250
- Initial node deployment probability 50 %, uniform distribution
- Sensor node features: $R_{SENS} = 30\,m$, $R_{COMM} = 30\,m$

3.1. Algorithms

Four multi-objective evolutionary algorithms (EAs, [3]) are used in this study, namely NSGA-II, SPEA2, PAES, and MOCell. The first two algorithms are the two most widely used ones in the literature, while PAES is a simple trajectory-based algorithm, and MOCell is a fairly new proposal that achieves state-of-the-art performances for some problems. The implementation of these algorithms provided by jMetal ([9]), an object-oriented Java-based framework aimed at the development, experimentation, and study of metaheuristics for solving multi-objective optimization problems[1], is used in this work.

[1] jMetal is freely available for download at the following URL: `http://jmetal.sourceforge.net/`.

Algorithm 2 Pseudocode for a generic multi-objective EA.

1: $P(0) \leftarrow$ **GenerateInitialPopulation**()
2: **EvaluateObjectives**($P(0)$)
3: $PF \leftarrow$ **CreateParetoFront**() //Create an empty front
4: $t \leftarrow 0$
5: **while not Termination_Condition() do**
6: parents $\leftarrow$ **Selection**($P(t)$);
7: offspring$\leftarrow$**EvolutionaryOperators**(parents);
8: **Improvement**(offspring); /* PACO goes here */
9: **EvaluateObjectives**(offspring);
10: $P(t+1) \leftarrow$**UpdatePopulation**($P(t)$, offspring)
11: **UpdateFront**($PF, P(t+1)$)
12: $t \leftarrow t+1$
13: **end while**

Starting from the pseudocode of a generic multi-objective EA included in Algorithm 2, the main features of the algorithms used in this work are outlined. For a detailed description, interested readers are referred to the references provided for each solver.

Both NSGA-II ([8]) and SPEA2 ([22]) use the scheme of Algorithm 2. They differ one each other in the mechanism used to keep a diverse approximated Pareto front. PAES ([12]) in turn has a population with one single solution that it is iteratively modified by using a mutation operator only (no crossover is required). MOCell ([19]) is a structured (cellular) EA, where each solution has a neighborhood of solutions inside with which it can cross. Though none of these algorithms includes an improvement operator (line 8 of Algorithm 2) in their canonical definition, the position where PACO comes in has been indicated in the pseudocode nonetheless.

In order to deal with constrained optimization problems such as the WSNL problem, all the algorithms have used the constraint domination principle presented in ([6]). It is based on considering feasible solutions as better solutions than non-feasible ones. Among non-feasible solutions, those with a smaller overall constraint violation are better (constraints are normalized to be greater than or equal to zero).

3.2. Operators

This section presents the different crossover and mutation operators used to evaluate the suitability of PACO under different operating conditions.

3.2.1. Crossover operators

Two crossover operators are used: SBX ([7]) crossover and a geographic crossover ([21]). Whereas the former is the most widely applied operator in the evolutionary multi-objective community, the latter is engineered to capture the particularities of the WSNL problem. In a crossover, two solutions called parents produce one or more new solutions called offsprings by exchanging information with some probability (the *crossover probability*, p_c).

The main issue when adapting the SBX crossover to the solution encoding presented in Section **??** concerns the management of deployed vs. non-deployed sensors. Let p_1 and p_2 be the individuals to be crossed and let $s_i^{p_1}$ and $s_i^{p_2}$ be the sensors at position i of each individual, at which SBX is operating. Let o_1 and o_2 also be the two generated offsprings and $s_i^{o_1}$ and $s_i^{o_2}$ be the corresponding sensors at the same position (i). The following cases may arise:

- Neither $s_i^{p_1}$ nor $s_i^{p_2}$ is deployed: neither $s_i^{o_1}$ nor $s_i^{o_2}$ are deployed either.
- Either $s_i^{p_1}$ or $s_i^{p_2}$ is deployed, but not both: the deployed sensor in the parent ($s_i^{p_1}$ or $s_i^{p_2}$) is independently copied to each offspring with a chance of 50 %.
- Both $s_i^{p_1}$ and $s_i^{p_2}$ are deployed: the coordinates of $s_i^{o_1}$ and $s_i^{o_2}$ are computed by using the coordinates of $s_i^{p_1}$ and $s_i^{p_2}$ and the standard SBX operations. The distribution index is set to $\eta_c = 20$, a widely used value in the literature.

The other crossover operator used is the geographic crossover, called RGX (Rectangular Geographic Crossover, [21]). In it, nodes are exchanged among solutions based on their geographic locations. A rectangular-shaped area is defined, and all nodes belonging to that area are exchanged between the two solutions (see Figure 1).

3.2.2. Mutation operators

Two different mutation operators have been used: a fully random mutation and a geographic mutation which is based on the polynomial mutation defined in ([7]). Both mutation operators modify each potential node (which can be either deployed or not) of a given solution with some probability (the *mutation probability*, p_m); different nodes

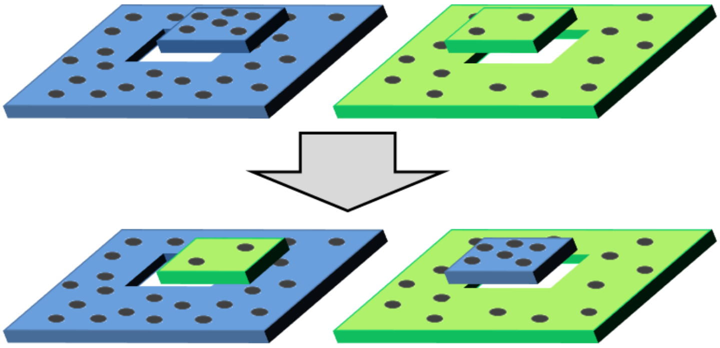


Figura 1: Example rectangular geographical crossover. All nodes in the extracted rectangles are exchanged between solutions.

are affected by the mutation independently. When a node is chosen to be modified, the performed procedure differs, depending on the mutation operator that is used. Both first check whether the node is deployed or not. If not, it is placed in a random location. Otherwise, it is either removed or repositioned with equal probability:

- Random mutation: The node is moved to any terrain point with uniformly distributed probability.
- Geographic mutation: The node is moved to a point in the surrounding area of the node's current position. This bounded movement is computed by using the polynomial mutation operator separately on the two coordinates of the node.

3.3. PACO integration

The approach used to include PACO in the multi-objective algorithms frame is straightforward, as can be seen in the Algorithm 2. Whenever a new solution is produced by the evolutionary operators (line 7), PACO may be applied onto it (line 8). An elitist criterion is applied: the solution produced by PACO is kept if and only if both the number of deployed sensors and the energy consumption are reduced, i.e., the new solution is said to dominate the older one. Otherwise it is rejected and the previous one is kept.

Another important remark has to be done here. Note that each replacement operation by PACO consumes one function evaluation. As explained in Section 2, the function evaluations consumed by PACO are properly accounted for into the computational effort, to ensure fairness in the comparisons between configurations using PACO and not using PACO.

3.4. Experimental methodology

3.4.1. Hypervolume indicator

HV calculates the (hyper)volume (in the objective space of solutions) covered by members of a nondominated set of solutions Q for problems where all objectives are to be optimized in the same direction (either minimized or maximized). Since the problem is a minimization one, the minimization version of HV is described. For each solution $i \in Q$, a hypercube v_i is constructed with the solution i and a reference point W as the diagonal corners. This reference point is common to all the hypercubes, and is generated here as a vector containing the worst objective function values per objective found in the global pool of non-dominated solutions of each problem instance. Only those points that dominate the reference point are computed for the HV. Thereafter, the union of all hypercubes is computed and its hypervolume (HV) is calculated as $HV = volume\left(\bigcup_{i=1}^{|Q|} v_i\right)$.

Finally, a normalization procedure is performed that translates the hypervolume values to the range [0.0,1.0]. For this, the set of globally non-dominated solutions found among all executions is produced, this set is called the reference Pareto front. Let $f_{max} = [f_1^{max}, f_2^{max}, \ldots, f_k^{max}]$ and $f_{min} = \left[f_1^{min}, f_2^{min}, \ldots, f_k^{min}\right]$ be the vectors of maximum and minimum values for the k objectives in the reference Pareto front. Then every nondominated solution $f = [f_1, f_2, \ldots, f_k]$ is normalized, assuming a minimization problem, as follows: $f_i^{norm} = \frac{f_i - f_i^{min}}{f_i^{max} - f_i^{min}}$ if $f_i^{min} \leq f_i \leq f_i^{max}, i = 1, \ldots, k$. Higher values of the hypervolume metrics are desirable.

3.4.2. Attainment function and surface

From the point of view of a decision maker, knowing the HV value gives little information, because it indicates nothing about the shape of the front. However, there is a need of knowing the general shape of the front, and thus of a way of representing the expected non-dominated front. For this, the concept of empirical attainment function is used (EAF, see ([13]). In short, the EAF is a function α from the objective space $\mathbb{R}^n$ to the interval $[0, 1]$ that estimates for each n-dimensional point in the objective space the probability of being dominated by the Pareto front from a single run of the multi-objective algorithm. Given r approximate Pareto fronts obtained in that same number of different runs, the EAF is defined as:

$$\alpha(z) = \frac{1}{r} \sum_{i=1}^{r} I(A^i \preceq \{z\}) \tag{1}$$

where A^i is the i-th approximate Pareto optimal set and I is an indicator function that takes value 1 when A^i dominates solution z, and 0 otherwise. From the attainment function it is possible to define the concept of $k\,\%$-attainment surface ([11]): the level curve with α value $k/100$. Informally, the 50 %-attainment surface in the multi-objective domain is analogous to the median in the mono-objective one.

The attainment surface provides the decision maker with a tool for quick evaluation of the variability of an algorithm. When the number of objectives of the MOP does not surpass three, the attainment surfaces can be represented graphically and constitute a helpful visual tool.

3.4.3. Statistical analysis

Since EAs are stochastic algorithms their results have to be given statistical significance. The following statistical procedure is used. First, 30 independent runs for test case (an algorithm with a given configuration used on a problem instance) are performed. The HV indicator and the attaiment surfaces are then computed. In the case of HV, the following statistical analysis are carried out ([20]). First a Kolmogorov-Smirnov test is performed in order to check whether the samples are distributed according to a normal distribution or not. If so, an ANOVA I test is performed; otherwise a Kruskal-Wallis test is performed. Since more than two algorithms are involved in the study, a post-hoc testing phase which allows for multiple comparison of samples (multicompare) is been performed. Because of room constraints, the full details of the statistical analysis are not displayed in the paper. However, their results will be properly discussed when needed.

3.5. Performance of PACO with different genetic operators

The set of experiments in this section will test the effect of using PACO with different algorithms, genetic operators and parametric configurations, in order to assess the general robustness and performance of the operator. The genetic operators used are two crossover operators (SBX and RGX), and two mutation operators (random and polynomial mutation), previously described in sections 3.2.1 and 3.2.2, respectively. The parametric configurations are the different combinations of crossover probability with values $p_c = 0.0$, $p_c = 0.1$, $p_c = 0.5$ and $p_c = 0.9$, and mutation probability such that on average 1 node, 5 nodes and 10 nodes are modified (for conveniency referred to as $p_m = 1.0$, $p_m = 5.0$ and $p_m = 10.0$, respectively). Table 1 displays all the results obtained in this study; in this table, algorithms and parametric configurations are sorted by rows, while genetic operators and application/not of PACO, by columns. Again, the best results obtained (per line) are highlighted with grey background color.

The HV values displayed in Table 1 vary from 0.0 to 0.721 (they are normalized to unity). The configurations integrating PACO produce higher HV than the same configurations without PACO in 113 of 132 test configurations, that is improved efficiency with 85.61 % probability. But some of these test configurations correspond to poor performing configurations, and their results are not extremely meaningful; if the comparison is restricted to the upper half (the best performing half) of the test configurations, then PACO yields improved performance with 98.48 % probability. Hence PACO is a robust technique, and its performance improves for high performing configurations.

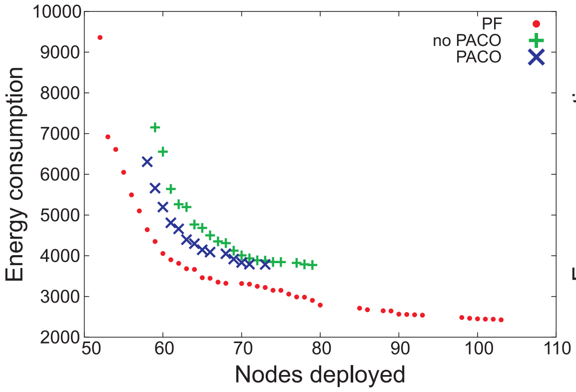
Regarding the mutation operator, it is clear that random mutation does not bring high performances and is largely outperformed by polynomial mutation: for the 132 test configurations, the HV obtained with polynomial mutation was never lower thant the one obtained with random mutation (97.73 % improved efficiency). For the crossover operator, it is RGX that produces the best results: in the 108 test configurations (excluding the ones with PAES and the ones having $p_c = 0$ since there is no crossover involved there), RGX always obtained higher HV than SBX (again 100 % improved efficiency). Furthermore, for the three algorithms including crossover (NSGA-II, SPEA2 and MOCell), the best configuration with crossover outperforms 100 % of the time the one without crossover.

In the parametric configuration, the dominant factor seems to be the mutation probability, with the highest HV values obtained for $p_m = 1.0$. For the crossover, the rate does not have such a big influence, but the best results are generally obtained with $p_c = 0.5$. The statistical analysis results prove that the algorithmic configurations with SBX, random mutation, and PACO/noPACO are statistically similar, and they are statistically worse than the rest. At the same time, the configuration the uses RGX, polynomial mutation and PACO is statistically better than the rest.

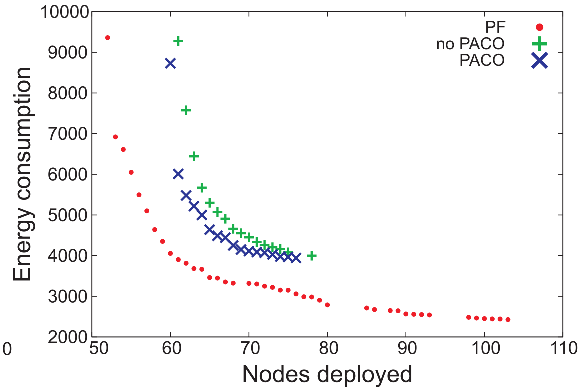
Table 1: Performance of PACO with different genetic operators: median and IQR of the HV indicator.

Crossover operator			SBX				RGX			
Mutation operator			**Polynomial**		**Random**		**Polynomial**		**Random**	
Algorithm	p_m	p_c	no PACO	**PACO**	no PACO	**PACO**	no PACO	**PACO**	no PACO	**PACO**
NSGAII	1.0	0.0	$0.591_{0.031}$	$0.621_{0.053}$	$0.442_{0.040}$	$0.529_{0.047}$	$0.591_{0.031}$	$0.621_{0.053}$	$0.442_{0.040}$	$0.529_{0.047}$
		0.1	$0.592_{0.041}$	$0.631_{0.027}$	$0.459_{0.052}$	$0.543_{0.040}$	$0.651_{0.047}$	$0.686_{0.060}$	$0.508_{0.083}$	$0.619_{0.066}$
		0.5	$0.592_{0.050}$	$0.618_{0.034}$	$0.457_{0.054}$	$0.530_{0.069}$	$0.624_{0.057}$	$0.688_{0.070}$	$0.472_{0.087}$	$0.578_{0.057}$
		0.9	$0.548_{0.067}$	$0.569_{0.049}$	$0.434_{0.032}$	$0.515_{0.065}$	$0.615_{0.044}$	$0.660_{0.054}$	$0.475_{0.061}$	$0.574_{0.052}$
	5.0	0.0	$0.469_{0.027}$	$0.482_{0.059}$	$0.420_{0.037}$	$0.432_{0.038}$	$0.469_{0.027}$	$0.482_{0.059}$	$0.420_{0.037}$	$0.432_{0.038}$
		0.1	$0.466_{0.043}$	$0.486_{0.034}$	$0.408_{0.046}$	$0.412_{0.037}$	$0.483_{0.044}$	$0.499_{0.035}$	$0.435_{0.046}$	$0.453_{0.042}$
		0.5	$0.416_{0.050}$	$0.431_{0.041}$	$0.354_{0.043}$	$0.370_{0.034}$	$0.524_{0.041}$	$0.541_{0.032}$	$0.489_{0.048}$	$0.501_{0.037}$
		0.9	$0.295_{0.066}$	$0.264_{0.067}$	$0.238_{0.058}$	$0.231_{0.072}$	$0.529_{0.065}$	$0.558_{0.033}$	$0.476_{0.067}$	$0.524_{0.069}$
	10.0	0.0	$0.109_{0.031}$	$0.126_{0.036}$	$0.066_{0.023}$	$0.091_{0.034}$	$0.109_{0.031}$	$0.126_{0.036}$	$0.066_{0.023}$	$0.091_{0.034}$
		0.1	$0.093_{0.016}$	$0.110_{0.035}$	$0.058_{0.030}$	$0.070_{0.029}$	$0.118_{0.030}$	$0.123_{0.030}$	$0.074_{0.031}$	$0.078_{0.027}$
		0.5	$0.056_{0.029}$	$0.057_{0.025}$	$0.024_{0.020}$	$0.027_{0.022}$	$0.131_{0.036}$	$0.148_{0.033}$	$0.104_{0.047}$	$0.102_{0.051}$
		0.9	$0.000_{0.007}$	$0.000_{0.000}$	$0.000_{0.000}$	$0.000_{0.000}$	$0.154_{0.039}$	$0.166_{0.040}$	$0.130_{0.044}$	$0.126_{0.034}$
SPEA2	1.0	0.0	$0.521_{0.057}$	$0.587_{0.037}$	$0.402_{0.044}$	$0.484_{0.064}$	$0.521_{0.057}$	$0.587_{0.037}$	$0.402_{0.044}$	$0.484_{0.064}$
		0.1	$0.527_{0.051}$	$0.592_{0.034}$	$0.425_{0.068}$	$0.493_{0.055}$	$0.560_{0.044}$	$0.613_{0.086}$	$0.462_{0.054}$	$0.541_{0.072}$
		0.5	$0.536_{0.053}$	$0.575_{0.064}$	$0.417_{0.067}$	$0.502_{0.074}$	$0.583_{0.044}$	$0.628_{0.061}$	$0.439_{0.041}$	$0.540_{0.054}$
		0.9	$0.518_{0.066}$	$0.542_{0.047}$	$0.418_{0.058}$	$0.456_{0.056}$	$0.566_{0.050}$	$0.623_{0.070}$	$0.444_{0.071}$	$0.505_{0.088}$
	5.0	0.0	$0.446_{0.036}$	$0.478_{0.029}$	$0.400_{0.052}$	$0.408_{0.055}$	$0.446_{0.036}$	$0.478_{0.029}$	$0.400_{0.052}$	$0.408_{0.055}$
		0.1	$0.439_{0.045}$	$0.472_{0.036}$	$0.395_{0.030}$	$0.409_{0.035}$	$0.468_{0.042}$	$0.497_{0.041}$	$0.424_{0.054}$	$0.437_{0.055}$
		0.5	$0.405_{0.038}$	$0.417_{0.050}$	$0.364_{0.050}$	$0.379_{0.049}$	$0.499_{0.048}$	$0.518_{0.076}$	$0.465_{0.045}$	$0.482_{0.056}$
		0.9	$0.318_{0.069}$	$0.270_{0.110}$	$0.279_{0.071}$	$0.243_{0.072}$	$0.512_{0.054}$	$0.538_{0.044}$	$0.464_{0.052}$	$0.502_{0.053}$
	10.0	0.0	$0.112_{0.023}$	$0.129_{0.021}$	$0.082_{0.029}$	$0.091_{0.021}$	$0.112_{0.023}$	$0.129_{0.021}$	$0.082_{0.029}$	$0.091_{0.021}$
		0.1	$0.103_{0.033}$	$0.123_{0.025}$	$0.068_{0.026}$	$0.078_{0.022}$	$0.126_{0.042}$	$0.144_{0.024}$	$0.084_{0.023}$	$0.105_{0.031}$
		0.5	$0.059_{0.024}$	$0.062_{0.023}$	$0.039_{0.023}$	$0.032_{0.022}$	$0.145_{0.027}$	$0.156_{0.031}$	$0.105_{0.018}$	$0.125_{0.029}$
		0.9	$0.005_{0.011}$	$0.000_{0.005}$	$0.000_{0.001}$	$0.000_{0.000}$	$0.177_{0.049}$	$0.174_{0.041}$	$0.145_{0.051}$	$0.151_{0.036}$
PAES	1.0	N/A	$0.535_{0.083}$	$0.632_{0.083}$	$0.398_{0.110}$	$0.500_{0.052}$	$0.535_{0.083}$	$0.632_{0.083}$	$0.398_{0.110}$	$0.500_{0.052}$
	5.0	N/A	$0.496_{0.058}$	$0.545_{0.084}$	$0.394_{0.095}$	$0.455_{0.077}$	$0.496_{0.058}$	$0.545_{0.084}$	$0.394_{0.095}$	$0.455_{0.077}$
	10.0	N/A	$0.186_{0.051}$	$0.211_{0.062}$	$0.158_{0.065}$	$0.162_{0.062}$	$0.186_{0.051}$	$0.211_{0.062}$	$0.158_{0.065}$	$0.162_{0.062}$
MOCell	1.0	0.0	$0.598_{0.044}$	$0.657_{0.033}$	$0.448_{0.055}$	$0.539_{0.063}$	$0.598_{0.044}$	$0.657_{0.033}$	$0.448_{0.055}$	$0.539_{0.063}$
		0.1	$0.595_{0.034}$	$0.649_{0.033}$	$0.465_{0.058}$	$0.531_{0.053}$	$0.687_{0.054}$	$0.717_{0.044}$	$0.557_{0.066}$	$0.630_{0.039}$
		0.5	$0.555_{0.059}$	$0.605_{0.041}$	$0.435_{0.071}$	$0.517_{0.037}$	$0.685_{0.045}$	$0.721_{0.060}$	$0.521_{0.064}$	$0.625_{0.049}$
		0.9	$0.506_{0.074}$	$0.549_{0.050}$	$0.391_{0.079}$	$0.472_{0.062}$	$0.658_{0.055}$	$0.710_{0.046}$	$0.513_{0.070}$	$0.619_{0.090}$
	5.0	0.0	$0.397_{0.044}$	$0.412_{0.055}$	$0.343_{0.044}$	$0.363_{0.044}$	$0.397_{0.044}$	$0.412_{0.055}$	$0.343_{0.044}$	$0.363_{0.044}$
		0.1	$0.375_{0.041}$	$0.406_{0.068}$	$0.332_{0.036}$	$0.358_{0.055}$	$0.398_{0.050}$	$0.430_{0.048}$	$0.357_{0.037}$	$0.369_{0.067}$
		0.5	$0.317_{0.080}$	$0.337_{0.070}$	$0.282_{0.027}$	$0.292_{0.054}$	$0.469_{0.048}$	$0.501_{0.049}$	$0.451_{0.042}$	$0.446_{0.059}$
		0.9	$0.334_{0.051}$	$0.329_{0.066}$	$0.309_{0.084}$	$0.296_{0.089}$	$0.521_{0.046}$	$0.522_{0.067}$	$0.475_{0.051}$	$0.512_{0.054}$
	10.0	0.0	$0.045_{0.016}$	$0.062_{0.026}$	$0.023_{0.018}$	$0.040_{0.018}$	$0.045_{0.016}$	$0.062_{0.026}$	$0.023_{0.018}$	$0.040_{0.018}$
		0.1	$0.040_{0.022}$	$0.055_{0.016}$	$0.020_{0.021}$	$0.032_{0.014}$	$0.052_{0.020}$	$0.062_{0.024}$	$0.025_{0.024}$	$0.042_{0.027}$
		0.5	$0.015_{0.014}$	$0.018_{0.030}$	$0.003_{0.005}$	$0.003_{0.010}$	$0.082_{0.042}$	$0.096_{0.030}$	$0.040_{0.025}$	$0.059_{0.019}$
		0.9	$0.004_{0.013}$	$0.001_{0.005}$	$0.000_{0.000}$	$0.000_{0.000}$	$0.115_{0.045}$	$0.112_{0.036}$	$0.090_{0.030}$	$0.098_{0.025}$

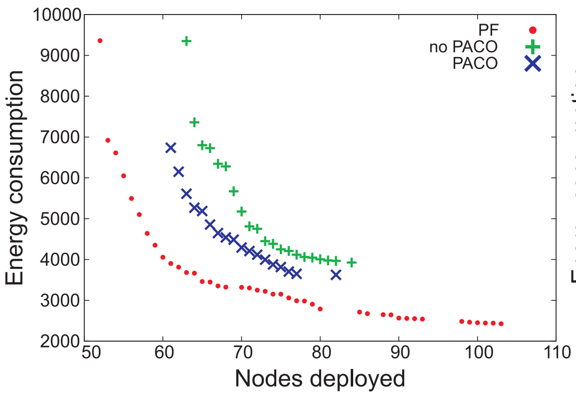
Finally, for the comparison of the algorithms, the results are less clear. MOCell obtains the highest HV values (the 10 best performing configurations obtain their highest HV values if MOCell is used), but is quite sensitive to the operator configuration and is outperformed by NSGA-II in the big picture (of the 84 test configurations, NSGA-II outperforms MOCell in 57, hence in 67.86 %). SPEA2 and PAES produce lower HV values. In the statistical tests, the best configuration of MOCell is the one that most often outperforms any other configuration; as a matter of fact, for any combination of mutation, crossover and PACO operator, MOCell with $p_m = 1.0$ and $p_c = 0.5$ systematically obtains the highest number of wins against other algorithms and/or configurations, or is at least tied for highest number of wins.



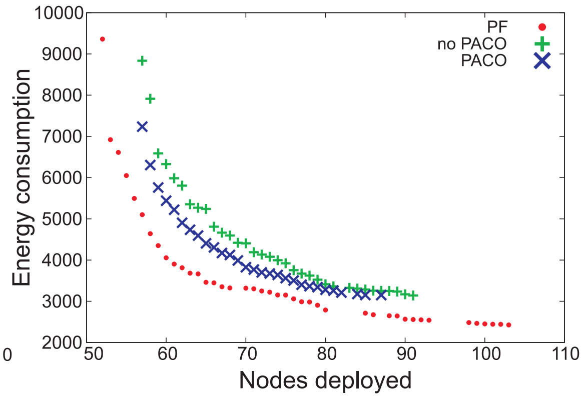
(a) NSGA-II



(b) SPEA2



(c) PAES



(d) MOCell

Figura 2: 50 %-attainment surfaces of the optimization algorithms with and without PACO. The global non-dominated fronts are represented for comparison, labeled as 'PF'.

The 50 %-attainment surfaces obtained by the best configuration of each algorithm both with PACO and without PACO are plotted in Figure 2. For all the four algorithms the attainment surfaces when PACO is used completely dominate the ones where PACO is not used. For NSGA-II the region where the number of nodes is below 70 is clearly improved with PACO, where the one where the number of nodes is beyond 70 differences are tighter. Finally, the same behavior emerges for MOCell: the attainment surface of PACO when the number of nodes is below 80 clearly dominates the one without PACO and, beyond 80 nodes, the surfaces get closer each other. From the problem's perspective, this means that for a given number of nodes (i.e., for a fixed cost), the algorithmic configuration with PACO achieves full terrain coverage with lower energy consumption than the configuration without PACO, these differences being more apparent as the number of nodes is reduced.

Finally, the 50 %-attainment surfaces obtained by the best configuration of each algorithm with PACO are shown in Figure 3. NSGA-II and MOCell dominate each other below and beyond configurations with 70 nodes, respectively, and both noticeably dominate SPEA2 and PAES.

As a result of the experiments in this section, the following conclusions are drawn:

- PACO offers a robust enhancement to the performances of the optimization algorithms.
- The polynomial mutation and RGX crossover are best suited for the WSNL problem.
- NSGA-II and MOCell outperform SPEA2 and PAES.

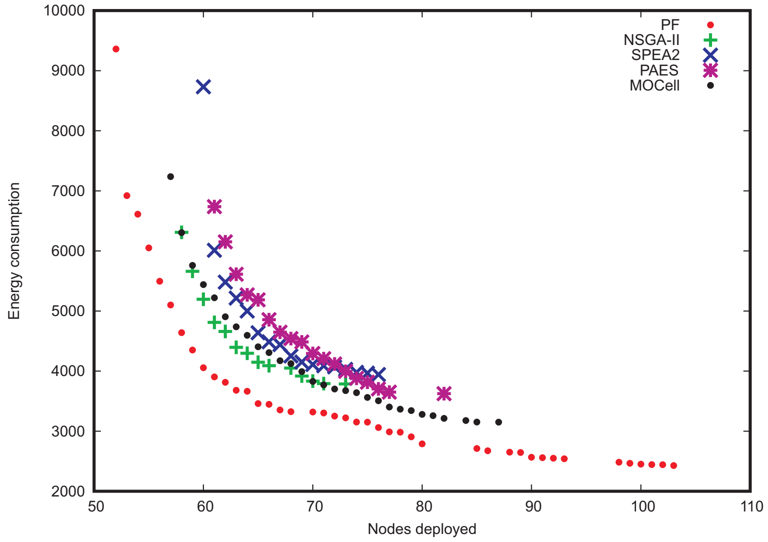


Figura 3: 50 %-attainment surface comparison of the optimization algorithms using PACO.

Referencias

[1] I.F. Akyildiz, W. Su, Y. Sankasubramaniam, and E. Cayirci. A survey on sensor networks. *IEEE Communications Magazine*, 2002.

[2] C. Blum and A. Roli. Metaheuristics in combinatorial optimization: Overview and conceptual comparison. *ACM Computing Surveys*, 35(3):268–308, 2003.

[3] C. A. Coello Coello, G. B. Lamont, and D. A. Van Veldhuizen. *Evolutionary Algorithms for Solving Multi-Objective Problems*. Springer, New York, second edition, September 2007. ISBN 978-0-387-33254-3.

[4] David Culler, Deborah Estrin, and Mani Srivastava. Overview of sensor networks. *IEEE Computer*, 37(8):41–49, 2004.

[5] W. Dargie and C. Poellabauer. *Fundamentals of Wireless Sensor Networks: Theory and Practice*. Wiley, 2010.

[6] K. Deb. *Multi-objective optimization using evolutionary algorithms*. John Wiley & Sons, 2001.

[7] K. Deb and R.B. Agrawal. Simulated binary crossover for continuous search space. *Complex Systems*, 9:115–148, 1995.

[8] K. Deb, A. Pratap, S. Agarwal, and T. Meyarivan. A fast and elitist multiobjective genetic algorithm: Nsga-ii. *Evolutionary Computation, IEEE Transactions on*, 6(2):182–197, Apr 2002.

[9] J.J. Durillo, A.J. Nebro, and E. Alba. The jmetal framework for multi-objective optimization: Design and architecture. In *IEEE Congress on Evolutionary Computation, CEC'2010*, pages 4138 – 4325, 2010.

[10] Deepak Ganesan, Răzvan Cristescu, and Baltasar Beferull-Lozano. Power-efficient sensor placement and transmission structure for data gathering under distortion constraints. *ACM Trans. Sen. Netw.*, 2(2):155–181, 2006.

[11] J. Knowles. A summary-attainment-surface plotting method for visualizing the performance of stochastic multiobjective optimizers. In *5th International Conference on Intelligent Systems Design and Applications (ISDA'05)*, pages 552 – 557, Washington, DC, USA, 2005. IEEE Computer Society.

[12] J. Knowles and D. Corne. Approximating the nondominated front using the pareto archived evolution strategy. *Evolutionary Computation*, 8(2):149 – 172, 2000.

[13] J. Knowles, L. Thiele, and E. Zitzler. A tutorial on the performance assessment of stochastic multiobjective optimizers. Technical Report 214, Computer Engineering and Networks Laboratory, ETH Zurich, February 2006.

[14] Á. Lédeczi, A.Ñádas, P.r Völgyesi, G.y Balogh, B. Kusy, J. Sallai, G. Pap, S. Dóra, K. Molnár, M. Maróti, and G. Simon. Countersniper system for urban warfare. *ACM Trans. Sen. Netw.*, 1(2):153–177, 2005.

[15] Xiang-Yang Li, Wen-Zhan Song, and Yu Wang. Localized topology control for heterogeneous wireless sensor networks. *ACM Trans. Sen. Netw.*, 2(1):129–153, 2006.

[16] N. Mladineo and S. Knezic. Optimisation of forest fire sensor network using gis technology. *Information Technology Interfaces, 2000. ITI 2000. Proceedings of the 22nd International Conference on*, pages 391–396, 13-16 June 2000.

[17] G. Molina, F. Luna, A. J. Nebro, and E. Alba. An efficient local improvement operator for the multi-objective wireless sensor network deployment problem. *Engineering Optimization*, 2011 (to appear).

[18] Guofang Nan and Minqiang Li. Evolutionary based approaches in wireless sensor networks: A survey. In *Natural Computation, 2008. ICNC '08. Fourth International Conference on*, volume 5, pages 217 –222, oct. 2008.

[19] A. J. Nebro, J. J. Durillo, F. Luna, B. Dorronsoro, and E. Alba. A cellular genetic algorithm for multiobjective optimization. *International Journal of Intelligent Systems*, 24(7):723 – 725, 2009.

[20] D. J. Sheskin. *Handbook of Parametric and Nonparametric Statistical Procedures*. Chapman & Hall/CRC; 4 edition, 2007.

[21] Qishi Wu, Nageswara S. V. Rao, Xiaojiang Du, S. Sitharama Iyengar, and Vijay K. Vaishnavi. On efficient deployment of sensors on planar grid. *Comput. Commun.*, 30(14-15):2721–2734, 2007.

[22] Eckart Zitzler, Marco Laumanns, and Lothar Thiele. SPEA2: Improving the strength Pareto evolutionary algorithms. In *EUROGEN 2001. Evolutionary Methods for Design, Optimization and Control with Applications to Industrial Problems*, pages 95–100, 2002.