Malaga, 10 de Junio de 2009

Informe Ejecutivo

TÍTULO: WSN-1.0

RESUMEN: En este informe se describe brevemente los resultados obtenidos en la resolución del problema de despliegue de nodos de una red de sensores desde un planteamiento multi-objetivo con restricciones. Además de ofrecer cobertura, la red debe formar una estructura conexa.

OBJETIVOS:

1. Presentar el problema de desplieque de nodos.
2. Describir las propiedades del problema: modelos empleados, restricciones asumidas.
3. Presentar las técnicas empleadas para la resolución del problema.
4. Mostrar y comentar los resultados obtenidos.

CONCLUSIONES:

1. Los resultados obtenidos confirman que las técnicas metaheurísticas permiten la resolución de instancias de gran dimensión.

Málaga, June 10th, 2009

Executive Summary

TITLE: WSN-1.0

ABSTRACT: This report briefly describes the results obtained solving the node deployment problem for wireless sensor networks using a multi-objective approach. Besides granting the coverage, we must produce a connected network.

GOALS:

1. Present the node deployment problem.
2. Describe the properties of the problem: models employed, constraints imposed.
3. Present the techniques employed for the resolution of the problem.
4. Display and comment the results obtained.

CONCLUSIONS:

1. The results obtained confirmed our expectation that metaheuristics are capable of solving large size instances.

Wireless Sensor Network Node Deployment Problem

DIRICOM

June 2009

1. Introduction

The Wireless Sensor Network Layout optimization problem (WSNL) is widely considered one of the fundamental problems in WSN. In its most basic form, WSNL amount to selecting the geographic locations for the deployment of each single node of the network.

There are two main concerns in the WSNL problem: coverage and connectivity. The coverage amounts to the basic quality of service offered by the sensor network, and it has to be maximized. The connectivity makes reference to the communication topology resulting from the node positioning. The main goal sought in the topology, besides the hard constraint of the network being fully connected with the HECN, is that the communication structure is such that the energy consumed for communications is minimized, hence maximizing the lifetime of the WSN. Additionally, the economical cost of the network (generally, the number of sensor nodes employed) is set as a third objective in WSNL.

The number of nodes and connectivity can in principle be considered as independent objectives; however we will see that they are in fact opposing objectives.

Some forms of the WSNL problem have been demonstrated to be NP-complete [3]. Additionally, the WSNL can be reduced to the set covering problem, by restricting the available positions of the sensor nodes to a set of discrete locations (for instance a regular point grid); and the set covering problem is well known to be a NP-complete problem [1]. Therefore, we state that the WSNL is NP-complete as well.

2. Problem Formulation and Models

In our formulation of the WSNL problem, coverage is treated as a constraint, with full coverage being required. The optimization objectives are then the cost of the network (which equals the number of nodes deployed and has to be minimized), and the lifetime of the network (that has to be maximized). The lifetime of the network will be defined as the Time to First Failure, that is, the time at which the first node of the network runs out of energy.

For the sensor nodes, we use **binary coverage** model, and *unit disk* model for the links, with respective radii values R_{SENS} and R_{COMM}. We are interested in **area coverage** at network level. In our formulation, a discrete **grid model** is used for the terrain, where each point in the grid represents one square meter of the terrain; the HECN is assumed to be located at the center of the terrain.

The formal definition of the problem is as follows. Let $\vec{x}$ be a vector of nodes x_i where each node is a 2D coordinate representing the node location; the lenght of $\vec{x}$ is non-fixed, and its nodes have to provide *full sensing coverage* $C(\vec{x}) = 100$ (Eq. 1). The number of sensor nodes and their locations have to be chosen in a way that minimizes the cost of the network which, in this case, is calculated as the number of deployed sensor nodes (Eq. 2), and the energy spent in communications by the most loaded node in the network (Eq. 3). The load in the most loaded node of the network is minimized since this node constitutes the bottleneck of the network with respect to the network lifetime; the most loaded node will be the first node to run out of energy, hence determinating the network lifetime according to the **TTFF criterion**. The two objectives are opposed, since the higher the number of nodes, the lower the share of retransmissions.

$$
\begin{aligned}
C(\vec{x}) &= 100 \cdot \left(\frac{CoveredPoints(\vec{x})}{TotalPoints} \right) && (1)\\
Cost(\vec{x}) &= Length(\vec{x}) && (2)\\
Energy(\vec{x}) &= Max \left(\{EnergyConsumed(x_i)\}_{i=1}^{f_1(\vec{x})} \right) && (3)
\end{aligned}
$$

In order to determine the energy spent in communications by any node of the WSN, the number of transmissions performed is calculated. The WSN considered operates by rounds: in a round every node collects the data from its measurements and sends it to the HECN encapsulated in a packet; between rounds the nodes are in a low-energy state. It is assumed that the main source of energy consumption is packet transmission; besides, packet (re)transmission is the sole energy-consuming process of the WSN that is directly affected by node deployment (and its resulting topology), and thus susceptible of being optimized in order to extend network lifetime. Therefore, all sources of energy consumption are neglected besides packet transmissions in this work.

To calculate the energy spent by transmissions the simple wave propagation model shown in Eq. 4 is applied for the power required per data packet to be transmitted over from node n_i to node x_j. Assuming free-space path loss sets $\alpha = 2$. Since the β constant value does not affect the optimization problem results, it will be neglected. The total energy consumed by a node x_i is shown in Eq. 5, where $\beta = 1$ and $\alpha = 2$. The function $Sent(a, b)$ indicates the number of data packets sent from node a to node b (see Eq. 6).

$$LinkPower(x_i, x_j) = \beta \cdot ||x_i - x_j||^\alpha \tag{4}$$

$$EnergyConsumed(x_i) = \sum_{x_j \in neighbors(x_i)} Sent(x_i, x_j) \cdot ||x_i - x_j||^2 \tag{5}$$

A simple routing algorithm is considered: every node sends its (re)transmitted information packets to the HECN itself if it is within communication range, or distributes them among all neighbors that are closer (in hop count) to the HECN. When there are several neighbors closer to the HECN, each of them receives a traffic share proportional to the inverse of the link power (see equations 4 and 7). Every node has a traffic (number of packets to send) equal to the packets received form nodes farther from the HECN, and additionnally produces one data packet per round (corresponding to its own sensed data) (see Eq. 8).

$$Sent(x_i, x_j) = Traffic(x_i) \cdot ProbSend(x_i, x_j)) \tag{6}$$

$$ProbSend(x_i, x_j)) = \frac{\frac{1}{||x_i - x_j||^2}}{\sum_{x_k} \frac{1}{||x_i - x_k||^2}} \tag{7}$$

$$Traffic(x_i) = 1 + \sum_{x_j} Sent(x_j, x_i) \tag{8}$$

For this problem, a constrained multiobjective approach is adopted, by defining the objective functions f_1 and f_2, as follows:

$$f_1(\vec{x}) = Cost(\vec{x}) \tag{9}$$

$$f_2(\vec{x}) = Energy(\vec{x}) \tag{10}$$

subject to the constraint imposed be the penalty function P:

$$P(\vec{x}) = 100 - C(\vec{x}) \tag{11}$$

3. Algorithms

In this section we briefly describe the techniques we propose to tackle the WSN node deployment problem: NSGA-II, PAES, SPEA2 and MOCell.

3.1. NSGA-II

Deb et al. proposed in [2] the second Nondominated Sorting Genetic Algorithm (NSGA-II) as a multi-objective technique that dealt with the main problems existing in the field: high computational complexity of nondominated sorting, lack of elitism and need of a sharing parameter specification. The authors fixed these problems by using a fast non-dominated sorting, an elitist Pareto dominance selection and a crowding distance method.

NSGA-II is based on a genetic algorithm. Its behavior can be seen in Algorithm 1. The differences between this algorithm and mono-objective GAs lie within the fitness assignment strategy.

In NSGA-II, the solutions are first sorted according to restriction fulfillment. Feasible solutions come first, then unfeasible solutions are sorted by increasing degree of constraint violation. Feasible solutions and every set of solutions with the same violation degree are then respectively sorted according to Pareto dominance. This sorting is performed by successively extracting form the chosen subpopulation the current set of non-dominated solutions (fronts). All the solutions in a front are given the same rank value, beginning at 0 for the first front extracted, 1 for the second and so on. This way, solutions can be sorted according to rank, starting at 0. Finally, within every group of solutions having the same rank, solutions are sorted according to the *crowding distance*. This criterion places first those solutions whose closest neighbors are farther, thus enhancing diversity.

Algorithm 1 Pseudocode of NSGA-II.

```
t ← 0
Initialize(P_a)
while not EndingCondition(t,P_a) do
    Parents ← SelectionParents(P_a)
    Offspring ← Crossover(Parents)
    Offspring ← Mutate(Offspring)
    P_i ← Merge(P_a,Offspring)
    RankingCrowding(P_i)
    P_n ← ElitistSelection(P_i)
    t ← t + 1
    P_a ← P_n
end while
```

3.2. PAES

The Pareto Archived Evolutionary Strategy (PAES) is a multi-objective evolutionary strategy that does not use self-adaptation, or recombination (crossover). Hence, PAES is a trajectory-based technique. Despite PAES handles a single solution at a time, a full Pareto optimal set is required as the execution's output; to generate such a set, PAES uses an external *archive* in which non-dominated solutions found are stored, and returns that archive upon execution completion. Algorithm 2 sketches the operation of PAES.

Algorithm 2 Pseudocode of PAES.

```
1: Archive ← ∅
2: Initialize(c)
3: t ← 0
4: Insert(Archive, c)
5: while not EndingCondition(t) do
6:     m ← Mutate(c)
7:     Evaluate(m)
8:     if Dominate(m,c) then
9:         Discard (m)
10:    else if Dominate(c,m) then
11:        Insert(Archive, m)
12:    else if Dominate(Archive,m) then
13:        Discard(m)
14:    else
15:        Test(c, m, Archive)
16:    end if
17:    t ← t + 1
18: end while
```

This algorithm maintains a single solution, and mutates it at each iteration to generate a new candidate solution (line 5). This new candidate solution either replaces the current one or not, and either enters the archive or not, subject to a Pareto-dominance criterion (lines 8, 10, 12, 15). Since the archive has bounded size, not all non-dominated solutions may be stored; a criterion based on the distribution of solutions over the front determines which solutions are accepted into the archive. Specifically, PAES employs a diversity measure based on an adaptive grid to uniformly distribute the non-dominated solutions in the front.

3.3. SPEA2

The Strength Pareto Evolutionary Algorithm (SPEA2) is a multi-objective evolutionary algorithm. SPEA2 was proposed by Zitler et al. in [4]. We show the algorithm's pseudocode in Algorithm 3.

SPEA2 uses a population and an archive simultaneously in its operation. In it, each individual is assigned a fitness value that is the sum of its strength raw fitness and a density estimation. The strength value of a solution i represents the number of solutions (in either the population or the archive) that are dominated by that solution, that is $S(i) = |\{j | j \in P_t \cup \overline{P_t} \wedge i > j\}|$. The strength raw fitness value of a given solution i, on the contrary, is the sum of strengths of all the solutions that dominate it, and is subject to minimization, that is, $R(i) = \sum_{j \in P_t \cup \overline{P_t}, j > i} S(j)$. The algorithm applies the selection, crossover, and mutation operators to fill an archive of individuals; then, the nondominated individuals of both the original population and the archive are copied into a new population. If the number of non-dominated individuals is greater than the population size, a truncation operator based on calculating the distances to the k-th nearest neighbor is used (a typical value is $k = 1$), $D(i) = \frac{1}{\sigma_i^k + 2}$, where σ_i^k is the distance from solution i to its k-th nearest neighbor. This way, the individuals having the minimum distance to any other

Algorithm 3 Pseudocode of SPEA2.

```
t ← 0
Initialize(P_0, \overline{P_0})
while not EndingCondition(t, \overline{P_t}) do
    FitnessAssignment(P_t, \overline{P_t})
    \overline{P_{t+1}} ← NonDominated(P_t ∪ \overline{P_{t+1}})
    if |\overline{P_{t+1}}| > \overline{N} then
        \overline{P_{t+1}} ← Truncate(\overline{P_{t+1}})
    else
        \overline{P_{t+1}} ← FillWithDominated(\overline{P_t})
    end if
    Parents ← BinaryTournament(\overline{P_{t+1}})
    Offspring ← Crossover(Parents)
    \overline{P_{t+1}} ← Mutate(Offspring)
    t ← t + 1
end while
```

individual are chosen.

3.4. MOCell

MOCell is a recent proposal based on the cellular model (that structures the population of solutions), for multi-objective optimization. Algorithm 4 shows its pseudocode.

Algorithm 4 Pseudocode of MOCell.

```
 1: Initialize(P)
 2: ParetoFront ← CreateEmptyFront()
 3: t ← 0
 4: while not EndingCondition(t,P) do
 5:     for all i in P do
 6:         Neighbors ← GetNeighborhood(i)
 7:         Parents ← Selection(Neighbors, ParetoFront)
 8:         Offspring ← Recombination(Parents)
 9:         Offspring ← Mutate(Offspring)
10:         Evaluate(Offspring)
11:         Insert(P,i,Offspring)
12:         InsertParetoFront(i, ParetoFront)
13:     end for
14:     t ← t + 1
15: end while
```

MOCell first creates an empty Pareto front (line 2). The individuals are placed in a 2D toroidal grid, and undergo the reproductive cycle iteratively (lines 5 to 14) until the stopping condition is met (line 4). This way, for each individual, the algorithm selects two parents, each through a binary tournament, one is taken from the grid neighborhood, and the other from the external archive. The winner of each tournament is determined by its crowding distance inside the neighborhood and the archive, respectively. The selection of a parent from the archive introduces front solutions (intensity), thus guiding the search towards promising regions. The selected parents are then recombined, and the resulting offspring is mutated and evaluated. This newly produced individual is then inserted in the population, replacing the solution in the current neighborhood with the worst crowding distance. The new individual may be inserted in the external archive as well, using a similar procedure as in PAES, but with NSGA-II's crowding distance as the diversity measure instead of the adaptive grid.

MOCell can also handle restrictions in the problem in the same way NSGA-II does. When comparing two solutions, if both are feasible, Pareto-dominance is used. If only one is feasible, this one dominates the other. When none is feasible, the one with the less restriction violation dominates the other.

4. Experiments

In this section we briefly present the results obtained for the node deployment problem with the four algorithms previously described. We define the instance for the WSNL problem as follows:

- Terrain: square grid of $250 \times 250m^2$.
- HECN located at the centre of the terrain.

- Maximum number of sensor nodes: 250.
- Initial node deployment probability 50 %, uniform distribution
- $R_{SENS} = 30m$.
- $R_{COMM} = 30m$.
- Wave propagation model: $P = d^2$, where P is the required power to send, d is the distance traveled by the signal.
- We neglect the energy consumption associated with sensing, processing and signal reception.

And for this instance, full coverage is required (100 %).

Cuadro 1: HV obtained for the WSNL by the different algorithmic configurations (median and IQ range).

Algorithm	p_m	p_c	**SBX** **Polynomial**	**SBX** **Random**	**RGX** **Polynomial**	**RGX** **Random**
NSGAII	1.0	0,0	$0{,}629_{0{,}043}$	$0{,}489_{0{,}045}$	$0{,}629_{0{,}043}$	$0{,}489_{0{,}045}$
		0,1	$0{,}640_{0{,}041}$	$0{,}498_{0{,}045}$	$0{,}698_{0{,}040}$	$0{,}550_{0{,}071}$
		0,5	$0{,}636_{0{,}050}$	$0{,}501_{0{,}054}$	$0{,}677_{0{,}041}$	$0{,}520_{0{,}068}$
		0,9	$0{,}606_{0{,}059}$	$0{,}478_{0{,}065}$	$0{,}671_{0{,}035}$	$0{,}523_{0{,}030}$
	5.0	0,0	$0{,}508_{0{,}029}$	$0{,}463_{0{,}036}$	$0{,}508_{0{,}029}$	$0{,}463_{0{,}036}$
		0,1	$0{,}504_{0{,}045}$	$0{,}448_{0{,}040}$	$0{,}528_{0{,}059}$	$0{,}478_{0{,}047}$
		0,5	$0{,}456_{0{,}042}$	$0{,}398_{0{,}055}$	$0{,}565_{0{,}039}$	$0{,}532_{0{,}058}$
		0,9	$0{,}339_{0{,}059}$	$0{,}289_{0{,}072}$	$0{,}570_{0{,}052}$	$0{,}528_{0{,}073}$
	10.0	0,0	$0{,}149_{0{,}036}$	$0{,}087_{0{,}027}$	$0{,}149_{0{,}036}$	$0{,}087_{0{,}027}$
		0,1	$0{,}123_{0{,}032}$	$0{,}088_{0{,}036}$	$0{,}152_{0{,}032}$	$0{,}104_{0{,}037}$
		0,5	$0{,}084_{0{,}035}$	$0{,}048_{0{,}030}$	$0{,}171_{0{,}045}$	$0{,}134_{0{,}042}$
		0,9	$0{,}008_{0{,}017}$	$0{,}001_{0{,}010}$	$0{,}201_{0{,}053}$	$0{,}171_{0{,}053}$
SPEA2	1.0	0,0	$0{,}569_{0{,}060}$	$0{,}450_{0{,}036}$	$0{,}569_{0{,}060}$	$0{,}450_{0{,}036}$
		0,1	$0{,}569_{0{,}043}$	$0{,}475_{0{,}060}$	$0{,}608_{0{,}048}$	$0{,}510_{0{,}064}$
		0,5	$0{,}579_{0{,}061}$	$0{,}459_{0{,}063}$	$0{,}626_{0{,}036}$	$0{,}489_{0{,}068}$
		0,9	$0{,}559_{0{,}070}$	$0{,}450_{0{,}063}$	$0{,}610_{0{,}044}$	$0{,}490_{0{,}055}$
	5.0	0,0	$0{,}490_{0{,}028}$	$0{,}447_{0{,}049}$	$0{,}490_{0{,}028}$	$0{,}447_{0{,}049}$
		0,1	$0{,}487_{0{,}032}$	$0{,}436_{0{,}032}$	$0{,}515_{0{,}037}$	$0{,}463_{0{,}048}$
		0,5	$0{,}447_{0{,}033}$	$0{,}404_{0{,}060}$	$0{,}539_{0{,}044}$	$0{,}502_{0{,}044}$
		0,9	$0{,}353_{0{,}068}$	$0{,}313_{0{,}077}$	$0{,}564_{0{,}052}$	$0{,}518_{0{,}037}$
	10.0	0,0	$0{,}137_{0{,}029}$	$0{,}094_{0{,}034}$	$0{,}137_{0{,}029}$	$0{,}094_{0{,}034}$
		0,1	$0{,}119_{0{,}041}$	$0{,}087_{0{,}032}$	$0{,}147_{0{,}037}$	$0{,}111_{0{,}033}$
		0,5	$0{,}071_{0{,}028}$	$0{,}047_{0{,}021}$	$0{,}171_{0{,}035}$	$0{,}128_{0{,}031}$
		0,9	$0{,}004_{0{,}009}$	$0{,}000_{0{,}000}$	$0{,}207_{0{,}054}$	$0{,}173_{0{,}053}$
PAES	1,0	0,0	$0{,}568_{0{,}086}$	$0{,}470_{0{,}107}$	$0{,}568_{0{,}086}$	$0{,}470_{0{,}107}$
	5,0	0,0	$0{,}535_{0{,}060}$	$0{,}445_{0{,}062}$	$0{,}535_{0{,}060}$	$0{,}445_{0{,}062}$
	10,0	0,0	$0{,}229_{0{,}041}$	$0{,}188_{0{,}056}$	$0{,}229_{0{,}041}$	$0{,}188_{0{,}056}$
MOCell	1.0	0,0	$0{,}648_{0{,}028}$	$0{,}492_{0{,}043}$	$0{,}648_{0{,}028}$	$0{,}492_{0{,}043}$
		0,1	$0{,}636_{0{,}034}$	$0{,}508_{0{,}050}$	$0{,}728_{0{,}047}$	$0{,}605_{0{,}079}$
		0,5	$0{,}603_{0{,}054}$	$0{,}478_{0{,}083}$	$0{,}724_{0{,}043}$	$0{,}566_{0{,}059}$
		0,9	$0{,}559_{0{,}056}$	$0{,}428_{0{,}087}$	$0{,}700_{0{,}048}$	$0{,}547_{0{,}051}$
	5.0	0,0	$0{,}448_{0{,}048}$	$0{,}375_{0{,}036}$	$0{,}448_{0{,}048}$	$0{,}375_{0{,}036}$
		0,1	$0{,}423_{0{,}045}$	$0{,}368_{0{,}051}$	$0{,}441_{0{,}042}$	$0{,}402_{0{,}066}$
		0,5	$0{,}372_{0{,}071}$	$0{,}337_{0{,}036}$	$0{,}516_{0{,}042}$	$0{,}501_{0{,}040}$
		0,9	$0{,}374_{0{,}055}$	$0{,}351_{0{,}059}$	$0{,}566_{0{,}046}$	$0{,}523_{0{,}051}$
	10.0	0,0	$0{,}079_{0{,}021}$	$0{,}056_{0{,}037}$	$0{,}079_{0{,}021}$	$0{,}056_{0{,}037}$
		0,1	$0{,}077_{0{,}038}$	$0{,}046_{0{,}030}$	$0{,}095_{0{,}032}$	$0{,}054_{0{,}026}$
		0,5	$0{,}040_{0{,}029}$	$0{,}019_{0{,}018}$	$0{,}115_{0{,}041}$	$0{,}081_{0{,}030}$
		0,9	$0{,}021_{0{,}023}$	$0{,}000_{0{,}007}$	$0{,}159_{0{,}049}$	$0{,}127_{0{,}024}$

Additionally, we use two types of genetic operator for both the mutation and recombination processes (alternatively). The first type of operator are “random” operators, in which all modifications are randomly performed; belonging to this type, we have the Simulated Binary Crossover (SBX) in which random nodes are exchanged among solutions, and the Random mutation operators, in which nodes are randomly relocated. The second type of operator are geographic operators, that are location-aware; belonging to this second type, we have the Rectangular Geographic Crossover (RGX) in which nodes belonging to a certain rectangular area are exchanged among solutions, and the Polynomial mutation, in which nodes are moved through paths.

The four algorithms are used with all different combinations of the previous operators, using different parametric configurations. For the mutation operator, the *degree* of mutation p_m, taken as the expected number of modified nodes per mutation application is tuned, whereas for the crossover, the probability p_c, taken as the probability to apply the crossover operator to each selected pair of solutions during an iteration, is tuned. The hypervolume (HV) is calculated for each execution as a measure of the "quality" of the output (the larger the HV, the better the output). The results are displayed in table 1. We highlight all results where the HV value surpasses 0,650 by using a gray background color.

The results obtained clearly point out that NSGA-II and MOCell produce higher performances than SPEA2 and PAES. Additionally, the geographic operators clearly outperform the random ones (in fact, the best configuration is the one that combines RGX with polynomial mutation). Regarding the parametric configuration, a low value for p_m, namely 1,0, is clearly superior to the rest; for p_c the differences are more subtle, the biggest one being the difference between not using crossover ($p_c = 0{,}0$) and using it, which clearly favors using the operator.

5. Conclusions

We have defined an energy-aware problem instance for the WSN node deployment problem, in which sensor nodes have to be deployed in a terrain field to meet a given coverage objective, while producing a connected network and minimizing the energy consumption at the network bottleneck (in order to maximize the lifetime, taken as the *Time To First Failure*). A multi-objective approach with constraint handling is adopted, and four different state-of-the-art metaheuristic algorithms are chosen to solve the problem: NSGA-II, SPEA2, PAES, and MOCell. Additionally, two different types of genetic operator are used, namely random and geographic operators, and several parametric configurations are tested for each possible combination. The results have shown that NSGA-II and MOCell offer better performances than the two other algorithms, and that geographic operators produce best results (in terms of hypervolume) than random operators.

Referencias

[1] Maggie X. Cheng and Lu Ruan and Weili Wu. Achieving minimum coverage breach under bandwidth constraints in wireless sensor networks. *Proceedings of the IEEE INFOCOM 2005. 24th Annual Joint Conference.* 4:2636–2645.

[2] K. Deb and A. Pratap and S. Agarwal and T. Meyarivan. A fast and elitist multiobjective genetic algorithm: NSGA-II. *Evolutionary Computation, IEEE Transactions on*, 6(2):187–197, 2002.

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

[4] Eckart Zitzler and Marco Laumanns and Lothar Thiele. SPEA2: Improving the Strength Pareto Evolutionary Algorithms. *EUROGEN 2001. Evolutionary Methods for Design, Optimization and Control with Applications to Industrial Problems*, 95–100, 2002.