J Math Model Algor (2011) 10:145–162
DOI 10.1007/s10852-010-9146-9

A Linear Programming-based Evolutionary Algorithm for the Minimum Power Broadcast Problem in Wireless Sensor Networks

Roberto Montemanni · Parvaz Mahdabi

Received: 27 May 2010 / Accepted: 7 December 2010 / Published online: 23 December 2010

Abstract A matheuristic approach, where concepts from linear programming are integrated into an evolutionary algorithm, is proposed. It is tested on a problem arising in wireless sensor networks: a topology with minimum total power expenditure, that connects a source node to all the other nodes of the network, has to be identified. Experimental results are presented.

Keywords Matheuristics · Linear programming · Evolutionary algorithms · Wireless sensor networks · Minimum power broadcast

1 Introduction

Quantum computing was proposed in the early 1980s by Feynman and Benioff in order to gain more computational power over classical computers (Kaye et al. [13]). There has also been associated work in another direction in which certain principles of quantum mechanics serve as an inspiration for the design of novel evolutionary algorithms combined with computational intelligence (Han and Kim [11, 12], Mahdabi et al. [18, 19]). Using quantum computation concepts to improve the performance of the evolutionary algorithms on the digital computers led to the development of Quantum inspired Evolutionary Algorithms (QEAs). In Han and Kim [11] a QEA approach for solving optimization problems is proposed. This is the best known

R. Montemanni (✉) · P. Mahdabi
Istituto Dalle Molle di Studi sull'Intelligenza Artificiale (IDSIA), Galleria 2, 6928 Manno-Lugano, Switzerland
e-mail: roberto@idsia.ch

Present Address:
P. Mahdabi
Faculty of Informatics, Universitá della Svizzera Italiana, Via Buffi 13, 6905 Lugano, Switzerland

application of quantum-inspired concepts in evolutionary computing. QEA is based on the concepts and principles of quantum computing such as quantum bits, linear superposition of states, and quantum gates. It proposes the potential for using a quantum representation instead of traditional representations (binary, numeric, or symbolic) in an evolutionary algorithm.

The aim of the present paper is to show how concepts coming from linear programming can be integrated into the QEA framework, leading to effective algorithms. A problem arising in wireless sensor networks is considered to illustrate the procedure.

Wireless sensor networks are composed of a set of devices that communicate without using any permanently installed infrastructure by transmitting radio signals. The devices, called also nodes of the network, generally use omni-directional antennae and their transmission range is determined by the power they employ in the transmission of the messages. A device communicates directly with all the other devices which are located within its transmission range, but it can also reach terminals located out of its range using a multi-hop communication that simply consists of making use of intermediate devices, called routers, that relay the data packets. The devices, thus, are not only responsible for sending and receiving their own data, but they possibly forward the traffic of other terminals (see e.g. Oliveira and Pardalos [25]).

Since the very beginning of research in the area of wireless sensor networks, one of the major issues has been saving power. Such an high attention for this factor is easy to identify: the nodes of the network are typically equipped with low capacity, tiny batteries, and they have to stay alive in the longest possible time horizon in an environment which is usually characterized by reduced accessibility. Wireless sensor networks are typically used in commanding actuators, monitoring events or measuring values at locations difficult to be reached by people, or where a long term sensing task is required. A tight management of the power budget is imposed by all these factors. Examples of applications are habitat monitoring (Mainwaring et al. [20]), civil structural monitoring (Kim et al. [14]) and environmental monitoring (Doolin and Sitar [9]). Nodes can usually be characterized as low cost devices, and are expected to be deployed in a potentially inaccessible area. Recharging the sensors after the deployment might therefore not be an option, both for logistic and economical reasons. In this context, energy-efficiency becomes perhaps the most important design criteria for sensor networks, since it directly impacts on the time the network itself is kept in operation. Many sensor networking applications are intrinsically about dissemination of information from a well-identified *source* node to all the other nodes of the network, called *destinations*.

The total power consumption of a network is the sum of the powers assigned to all devices and thus the Minimum Power Broadcast (MPB) problem consists in minimizing this sum subject to the constraint that messages originated from the source are received by all the destinations.

The MPB problem is NP-hard (Cagalj et al. [4]) and has attracted a wide attention in the scientific literature. In Wieselthier et al. [30] (see also Wieselthier et al. [31]) it is first observed that the so called "node based" approach is more suitable for wireless environment than the previously adopted "link-based" algorithms. They developed the *Broadcast Incremental Power (BIP)* algorithm, which is a simple sub-optimal heuristic for constructing minimum power broadcast trees in wireless networks. In this algorithm, new nodes are added to the tree on a minimum incremental cost basis, until all intended destination nodes are included. Other techniques that have been suggested for solving this problem include an internal nodes based broadcasting produce (Stojmenovic et al. [29]), a localized algorithm (Cartigny et al. [5]), a cluster-merge method (Das et al. [7]) and a swarm based procedure (Das et al. [6]). Some refinement heuristic approaches, able to improve solutions provided by other methods, were presented in Das et al. [8]. A simulated annealing procedure was finally proposed in Montemanni et al. [23], where the new method is shown to outperform the previous approaches. Other contributions for variants of the MPB problem are in Althaus et al. [1], Altinkemer et al. [2], Montemanni and Gambardella [22], Montemanni et al. [24] and Yuan [32]. Specific studies for the MPB problem have been carried out in Guo and Yang [10], where a flow-based formulation expressed in terms of a mixed integer program has been proposed. In Leggieri et al. [16] (see also Leggieri [15]) a variant of the MPB problem has been expressed in terms of a set covering model and in Bauer et al. [3] a multi-commodity flow model and cut-based models have been considered. These methods are able to solve to optimality broadcasting problems with up to 30/40 nodes in one hour (on a modern computer). A further mixed integer programming model, the relaxation of which is used to produce lower bounds, is discussed together with some heuristic algorithms in Yuan et al. [33].

The rest of the paper is organized as follows. In Section 2 the MPB problem is formally defined. Section 3 describes a mixed integer linear programming formulation, that will be then exploited within the novel evolutionary algorithm presented in Section 4. Computational experiments are discussed in Section 5, while conclusions are drawn in Section 6.

2 Problem Definition

A static wireless sensor network can be modeled in terms of a graph, by considering the devices as nodes and the transmission links as arcs. Let $G(V, A)$ be a directed graph, where V represents the set of the devices and A the set of directed arcs which connect all the possible pairs (i, j), with $i, j \in V$. Notice that A also contains loops (i, i). A cost P_{ij} is associated with each arc (i, j): P_{ij} represents the minimum amount of power that has to be assigned to node i in order to establish a direct connection with node j. Following a simple signal propagation model proposed in Rappaport [27], this value is proportional to the power of the distance d_{ij} with an environment-dependent exponent γ whose value belongs to the interval [2,5]. Therefore, in the sequel $P_{ij} := (d_{ij})^{\gamma}$. We however observe that all the presented results remain valid also when more complex signal propagation models are considered.

We select a node s to be the source of the communication. The remaining nodes are the so called destination nodes, which have to receive the messages periodically generated in s. The nodes belonging to $V \setminus \{s\}$ may act as routers or just being reached by the message without propagating it.

Our purpose is to optimally allocate transmission powers to the nodes in such a way that a connected topology emerges. It guarantees that the messages broadcasted by the source node are reception by all the other nodes. Thus, for solving the MPB problem we define a *range assignment* function ρ, which assigns to each node $i \in V$ a

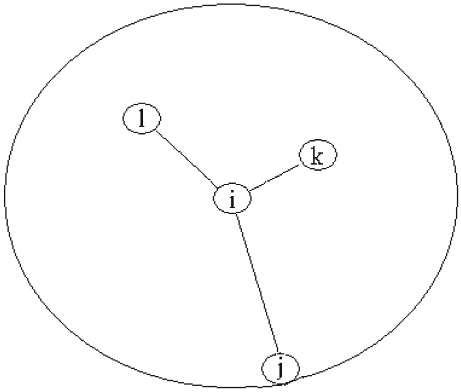

Fig. 1 The Wireless Multicast Advantage. Power requirements are taken proportional to the distances between nodes in this example

transmitting power $\rho(i)$. We aim at minimizing the amount $\sum_{i \in V} \rho(i)$ while fulfilling the constraint that the topology implied by the range assignment function ρ connects the source to all the other nodes. Notice that in any efficient solution, $\rho(i)$ must be either zero or equal to P_{ij} for some j (i.e., either node i does not transmit or uses exactly the amount of power necessary to reach a target node j).

Since nodes are equipped with omni-directional antennae, any signal transmitted by a node $i \in V$ to a node $j \in V$ is also received by all the nodes that are in the transmission range of i i.e., if $\rho(i) = P_{ij}$ then every node $k \in V$ such that $P_{ik} \leq P_{ij}$ receives the signal. This is the so-called Wireless Multicast Advantage (WMA) property (Wieselthier et al. [30]) and is summarized in the example of Fig. 1, where nodes k and l are both reached when node i is transmitting to node j. Notice that power requirements are proportional to the distances between nodes in the example. However, the WMA property holds also when more complex signal propagation models, with power requirements not proportional to distances, are adopted.

Notice that both the model and the approach we propose work also for problems sharing the same structure, but with a different objective function. For example, it is straightforward to adapt the model and the algorithm described in the remainder of this paper to the problem where information has to be sent from some nodes of the network to a unique destination node (data gathering instead of data dissemination), or to the problem where the objective is $\min \max_{i \in V} \rho(i)$ (see Montemanni [21]). The choice of focusing the problem described in the beginning of the section is motivated by the previous literature.

3 A Mixed Integer Linear Programming Formulation

We can define for each $i \in V$, v^i as the array whose components are the nodes of the network ordered in non-decreasing order of power requirement, for a transmission from node i (see Montemanni and Gambardella [22] and Leggieri [15]). In other words, if j and l are two indices in $\{1, 2, \ldots, |V|\}$, with $j \leq l$, then v^i_j and v^i_l are two nodes in V whose power requirements from i are related by $P_{iv^i_j} \leq P_{iv^i_l}$. Note that $v^i_1 = i$, with $P_{iv^i_1} = 0$, by definition. We refer to v_i as the power levels array for node i. We also define $p_{ij} = P_{iv^i_j}$ $\forall i \in V, \forall j \in \{1, 2, \ldots, |V|\}$. The introduction of p_{ij}s is necessary to have a representation of the problem suitable for the evolutionary algorithm. Notice that two consecutive power levels of a node i can share the same power requirements, i.e. $P_{iv^i_j} = P_{iv^i_{j+1}}$. In this case the order of v^i_j and v^i_{j+1} is arbitrary chosen.

The basic idea of the formulation that we adopt for the MPB problem is that each connecting structure must contain an arborescence, which is represented in our case through a network flow model (see Magnanti and Wolsey [17]). One unit of flow is sent from node s (the root of the arborescence) to each of the remaining $|V|-1$ nodes of the network. The variable $0 \leq y_{ij} \leq |V|-1$ (with $i \neq j$) represents the flow on arc (i, j) connecting nodes i and j. Variables y will model the arborescence that guarantees connectivity for the topology emerging from the power assignments. Binary x_{ij} variables regulate transmission powers:

$$x_{ij} := \begin{cases} 1 & \text{if } \rho(i) = p_{ij} \\ 0 & \text{otherwise.} \end{cases} \quad \forall i \in V, \forall j \in \{1, 2, \ldots, |V|\}$$

that is, $x_{ij} = 1$ if the node i is assigned its j-th power level, i.e. it transmits to a power such that it reaches node v^i_j exacly. The following mixed integer linear programming formulation arises (see Montemanni et al. [23]):

$$(F) \quad \min \sum_{i \in V} \sum_{j=1}^{|V|} p_{ij} x_{ij} \tag{1}$$

$$\text{s.t.} \sum_{j=1}^{|V|} x_{ij} = 1 \qquad \forall i \in V \tag{2}$$

$$(|V|-1) \sum_{k=l:v^i_l=j}^{|V|} x_{ik} \geq y_{ij} \qquad \forall i, j \in V \tag{3}$$

$$\sum_{j=1}^{|V|} y_{ji} - \sum_{j=1}^{|V|} y_{ij} = \begin{cases} |V|-1 & \text{if } i = s \\ 1 & \text{otherwise} \end{cases} \qquad \forall i \in V \tag{4}$$

$$0 \leq y_{ij} \leq |V|-1 \qquad \forall (i, j) \in A \tag{5}$$

$$x_{ij} \in \{0, 1\} \qquad \forall i \in V, \forall j \in \{1, 2, \ldots, |V|\} \tag{6}$$

Constraints (2) state that each node has to transmit at exactly one transmission level (level 1 corresponds to power 0). Constraints (3) connect flow variables y with power variables x (the WMA property appears here). Equations 4 define the flow problem to define the arborescence, while Eqs. 5 and 6 are domain definition constraints.

In the context of the work presented in this paper, we are interested in the linear relaxation of formulation F, which will be referred to as F_{LR}. It is obtained by changing constraints (6) with the following ones:

$$0 \leq x_{ij} \leq 1 \quad \forall i \in V, \forall j \in \{1, 2, \ldots, |V|\} \tag{7}$$

The optimal solution of the linear relaxation F_{LR} not only guarantees a lower bound for the cost of the original problem, but also provides potentially useful information about promising network topologies that will be exploited by the evolutionary algorithm described in Section 4.

It is worth to observe that formulations with tighter linear relaxations, but with longer solution times (see Leggieri [15]) could have been adopted instead of F. However, in the evolutionary framework we will propose in Section 4, the linear relaxation of the formulation has to be repeatedly solved with different cost coefficients, and therefore a trade-off had to be made between quality of the linear relaxation and the solution time.

4 A Linear Programming-based Evolutionary Algorithm

The algorithm we propose, the Linear Programming Evolutionary Algorithm (LPEA) works by embedding principles from mathematical programming (in the case of the MPB problem from the linear formulation F_{LR} described in Section 3) into a framework inspired by Quantum inspired Evolutionary Algorithms. The resulting method is an evolutionary algorithm where operators are based on linear programming. For this reason we will first introduce the main principles of QEAs, then we will describe how operators from mathematical programming can be used within a QEA-like algorithm to obtain the LPEA algorithm we propose for the MPB problem.

4.1 Quantum Inspired Evolutionary Algorithms

Quantum inspired Evolutionary Algorithms are based on the concepts and principles of quantum computing such as quantum bits, linear superposition of states, and quantum gates (see Han and Kim [11]). The method belongs to the class of Estimation Distribution Algorithms (EDAs, see Platel et al. [26]). A pseudo-code of QEAs is provided in Fig. 2. The elements of the algorithm appearing in the pseudo-code will be briefly summarized in the next sections.

4.1.1 Representation of the Problem

A *Q-bit* is defined as the smallest unit of information which may be in the “1” state, in the “0” state, or in a linear superposition of the two. With some approximations, and according to our purposes, a Q-bit can be defined as the probability β that the bit associated is “1”.[1] The probability for “0” will be therefore $1 - \beta$.

A *Q-bit individual* is defined as a string of Q-bits, and represents the solutions of the combinatorial optimization problem under investigation. The Q-bit representation enables a Q-bit individual to probabilistically represent a linear superposition of states (binary solutions). Indeed, each Q-bit individual can be viewed as a distribution of promising solutions in the search space. In order to make the evaluation

[1] Note that in the original Quantum Inspired framework the probability of “1” is given by the square of β. The simplified notation we use is functional to our purposes.

```
Initialize the population of Q-bit individuals;
Initialize the best solution b to NULL;
While (termination criteria not met)
  For each Q-bit individual q(i) of the population
     Apply the evaluation operator to generate solution c(i);
     If c(i) is better than b
        Update b;
     EndIf;
  EndFor;
  For each Q-bit individual q(i) of the population
     Apply the evolutionary operator;
  EndFor;
EndWhile;
```

Fig. 2 A pseudo-code for QEAs

phase possible, each Q-bit individual will be evaluated to form binary solutions (see Section 4.1.3). A Q-bit individual provides therefore the representation of a combinatorial optimization problem within a QEA.

4.1.2 Initialization Operator

Before starting the evolutionary phase of the algorithm, the population of Q-bit individuals has to be initialized. Random values, equally likely values or values coming from solutions obtained by fast-heuristics are usually adopted.

4.1.3 Evaluation Operator

In order to derive a solution of the original problem from a Q-bit individual, it is necessary to transform each Q-bit into a binary value. This is usually done according to the probabilities associated with each Q-bit. The cost of the solutions so obtained is typically used to evaluate the quality of a Q-bit individual.

4.1.4 Evolutionary Operator

An evolutionary operator has to be defined to evolve the values of the Q-bits. The rationale behind the evolutionary operator is that we would like Q-bits (and Q-bit individuals with them) to evolve towards the bitwise representation of the best solution b retrieved so far.

4.2 The LPEA for the MPB Problem

The idea behind the Linear Programming Evolutionary Algorithm is that the solution of the linear relaxation of a mixed integer linear programming formulation (F_{LR} in our case) typically provides useful information about the characteristics of the optimal solution of the integer problem (F in our case). In our approach we will try to exploit the information provided by the linear relaxation within an evolutionary algorithm.

4.2.1 Representation of the Problem

A parallel between x variables of the linear program F_{LR} and Q-bits β^k, associated with individual k of the population, is introduced. Given the *kth Q-bit individual* of the population, its Q-bit β^k_{ij} will be "1" if node i transmits at its jth power level, i.e. it reaches node v^i_j exactly. With reference to the formulation discussed in Section 3, we formally have:

$$\beta^k_{ij} = x_{ij} \tag{8}$$

Moreover, since Q-bits β^k_{ij} correspond to x_{ij} variables, and thanks to constraints (2), β^k_{ij} values can be regarded as probabilities for a given individual k.

The evolving population is composed of N_{IND} Q-bit individuals. The tuning of parameter N_{IND} will be analysed in Section 5.1.

4.2.2 Initialization Operator

The population of Q-bits individuals is initialized as follows.

One third of the individuals has the Q-bits initialized to the values of x variables in the optimal solution of F_{LR}, according to Eq. 8. One third of the individuals is initialized at random values such that $\sum_{j=1}^{|V|} \beta^k_{ij} = 1$ for each $i \in V$. One third of the individuals is finally initialized in such a way that probabilities are evenly spread for each node over the possible power levels: $\beta^k_{ij} = \frac{1}{|V|}$ $\forall i \in V$, $j \in \{1, 2, \ldots, |V|\}$.

The aim of the initialization phase we propose is to have the population starting from different positions in the search space, with an emphasis to the most promising region, which is likely to be represented by the optimal solution of the linear program F_{LR}. Notice that during the very first iteration (before the evolutionary algorithm is applied for the first time) the individuals initialized at random or uniformly do not necessarily correspond to feasible solutions of F_{LR}.

Some computational experiments reported in Section 5.2 will support the initialization strategy adopted.

4.2.3 Evaluation Operator

The probabilities defined by β^k_{ij}s are used to generate heuristic solutions starting from Q-bit individuals. The procedure is based on a probabilistic modification of the Broadcast Incremental Power (BIP) constructive algorithm originally described in Wieselthier et al. [30]. Notice that in our case a probabilistic framework is desirable in order to generate different evaluations starting from a same Q-bit individual.

Each time the evaluation operator is run, some artificial power requirements, that will be used by the probabilistic BIP method and are based on the values of β^k_{ij}s are created as follows:

$$\overline{p_{ij}} := \begin{cases} 0 & \text{if } i = 1 \\ p_{i(j-1)} + (1 - \beta^k_{ij})(p_{ij} - p_{i(j-1)}) & \text{otherwise} \end{cases} \quad \begin{array}{l} \forall i \in V, \\ \forall j \in \{1, 2, \ldots, |V|\} \end{array} \tag{9}$$

We perturb (sometimes heavily) the power requirements according to the probabilities (Q-bits) associated with the Q-bit individuals. For example, if $\beta^k_{ij} \to 1$ for some i, j, the corresponding power level $\overline{p_{ij}}$ will be squeezed towards the previous (original) power level $p_{i(j-1)}$. An example of application of the evaluation operator is presented in Fig. 3. Notice that the rankings of power levels are still fulfilled: $\overline{p_{ij}} \leq \overline{p_{i(j+1)}}$ $\forall i \in V, \forall j \in \{1, 2, \ldots, |V|-1\}$.

The artificial power requirements $\overline{p_{ij}}$ defined in (9) are used within the following probabilistic BIP-like construction heuristics, to evaluate a Q-bit individual. An evolving set R, that contains nodes that are connected to s in the structure under construction, is defined. In the beginning we set $R := \{s\}$, and we will stop when $R = V$. We also initialize the current power $Pow(i)$ of each node i at 0. At each iteration, all possible extensions of the current partial solution Sol (which is empty in the beginning) are considered, and a new arc $(\hat{i}, \hat{j})$, with $\hat{i} \in R$ and $\hat{j} \in V \setminus R$ is selected according to a Monte Carlo sampling technique (Robert and Casella [28]), where probabilities are obtained by normalizing at 1 the sum of the following values:

$$e_{ij} := \frac{1}{\overline{p_{il}} - Pow(i) + 1} \quad \forall i \in R,\ j \in V \setminus R : v_l^i = j \tag{10}$$

Observe that the definition of e_{ij} is based on the artificial power requirements, in turn defined by β_{ij}^k values. The selected arc is inserted into the partial solution Sol. All the nodes in $V \setminus R$ which are covered (thanks to the Wireless Multicast Advantage) by the newly selected arc are inserted into R and the power of the node $\hat{i}$ is updated to $\overline{p_{\hat{i}\hat{j}}}$.

A pseudo-code for the probabilistic BIP procedure is provided in Fig. 4.

After each run of the probabilistic BIP procedure, the Sweep local search (Wieselthier et al. [30]) is executed on the solution obtained, in order to bring it down to a local optimum. The sweep post-optimization procedure iteratively examine nodes and reduce their transmission power by one level in case this does not disconnect the topology. The choice of running the Sweep local search so often, which might appear time-consuming, is important in the economy of the algorithm we propose, as suggested by the experimental analysis reported in Section 5.3. This conclusion is also intuitively supported by the observation that the Sweep procedure is extremely fast in practice.

At each iteration the evaluation operator is run N_{OBS} times for each Q-bit individuals, and the best among the heuristic solutions obtained is selected as the evaluation of the the Q-bit individual (see Mahdabi et al. [19]). A discussion about the tuning of parameter N_{OBS} will be discussed in Section 5.1.

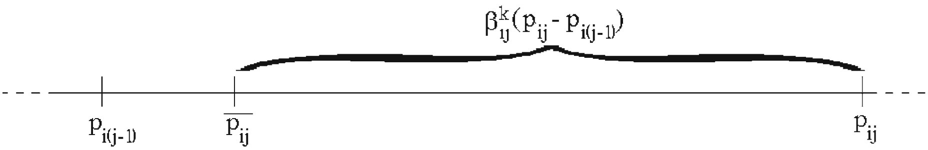

Fig. 3 Evaluation operator. The artificial power requirement $\overline{p_{ij}}$ is created by adjusting the original power requirement p_{ij} according to the Q-bit β_{ij}^k: large values of β_{ij}^k correspond to lower values for $\overline{p_{ij}}$

```
Initialize set R to {s};
Initialize arc set Sol to NULL;
While (R ≠ V)
   Select arc (î, ĵ) according to probabilities derived from (10);
   Insert arc (î, ĵ) in Sol;
   R := R ∪ {k : p̄_îk ≤ p̄_îĵ};
EndWhile;
```

Fig. 4 A pseudo-code of the probabilistic BIP procedure

4.2.4 Evolutionary Operator

The linear formulation F_{LR} is at the basis of the evolutionary operator we propose. Given a Q-bit individual k, for each node i we define the desired new power level through the weighted average of β_{ij}^k Q-bits as follows:

$$W_i^k := \sum_{j=1}^{n} j \cdot \beta_{ij}^k + \left(\text{best}_i - \sum_{j=1}^{n} j \cdot \beta_{ij}^k \right) (\Theta + \text{Rand}(-\Theta \cdot 0.1; \Theta \cdot 0.1)) \tag{11}$$

where best_i is the power level assigned to node i in the best solution retrieved so far, Θ is a Q-bit evolutionary parameter that needs to be tuned (see Section 5.1). Rand(A, B) is finally a function returning a random number in $[A, B]$. The range adopted for random numbers was found experimentally, and is used to introduce noise into the evolutionary framework. In the sum in the right hand side of Eq. 11, each power level j of node i is weighted by its respective Q-bit β_{ij}^k. The sum itself returns the average power level for node i, according to the weights represented by the Q-bits. An example of calculation of the desired value W_i^k is provided in Fig. 5.

W_i^k represents the target value for node i in the evolutionary phase of Q-bit individual k. It is used to modify the objective functions (1) of F_{LR} as follows:

$$\min \sum_{i \in V} \sum_{j=1}^{|V|} \left(p_{ij} \left| W_i^k - j \right| \right) x_{ij} \tag{12}$$

The costs are modified in such a way that the desired power levels W_i^ks are most likely to be selected, while the other power levels are less and less attractive, proportionally

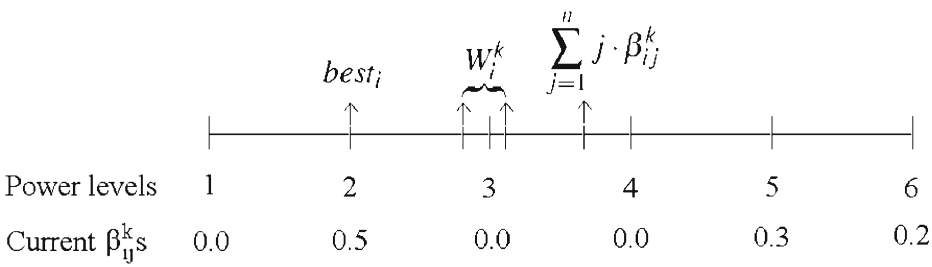

Fig. 5 Evolutionary operator. The desired power level W_i^k is obtained by moving the current average power level from $\sum_{j=1}^{n} j \cdot \beta_{ij}^k$ towards best_i by a factor given by $\Theta + \text{Rand}(-\Theta \cdot 0.1; \Theta \cdot 0.1)$

to the distance from the wanted power levels. The power levels corresponding to $\lfloor W_i^k \rfloor$ and $\lceil W_i^k \rceil$ will typically have the cost coefficient reduced in Eq. 12, since the absolute value will be below one. The remaining power levels will have the cost coefficient increased, proportionally to the distance of their index from the desired value W_i^k.

Notice that solving F_{LR} will not necessarily take the desired power level W_i^k for each node, but will produce a solution in which the power level stick as much as possible to this desired values, while generating a "feasible" solution, according to the (relaxed) problem constraints.

A variation of F_{LR} is therefore solved for each Q-bit individual k, and the solution of the linear program provides the new value β_{ij}^k for each Q-bit. The choice of using the linear formulation F_{LR} for evolutionary purposes is also supported by some experimental evidences that will be presented in Section 5.2.

4.2.5 Further Parameters

The LPEA procedure is interrupted after N_{IT} consecutive iterations without any improvement to the best known solution. We also adopt a multi-start approach: the LPEA procedure is repeated (each time after a new initialization), but always keeping the best solution retrieved so far, which is used by the evolutionary operator (see Section 4.2.4). This helps to differentiate the search. The algorithm stops when a maximum computation time allowed is elapsed.

5 Computational Results

The LPEA has been tested on networks with 10, 15, 20, 25, 30, 40, 50, 75 and 100 nodes. In each case, 50 networks have been randomly generated on a 5×5 grid[2] with the signal propagation parameter γ set to 2 and the total power requirements of the structures retrieved by each algorithm are averaged to obtain what we will refer to as the *mean tree power* in the remainder of the section.

The algorithms have been coded in ANSI C, IBM ILOG CPLEX[3] 12.1 has been used to solve linear programs. All the tests reported were carried out on a computer equipped with an Intel Core 2 Duo 2.40 GHz processor and 4 GB of memory.

The termination criterion for the LPEA algorithm adopted in this study is a maximum computation time which depends on the number of nodes considered, as reported in Fig. 1. These computation times are the same required by the SA algorithm described in Montemanni et al. [23], here used as a reference algorithm for comparison purposes. As explained in Section 4.2.5, for the LPEA we will adopt a multi-start mechanism for the allowed computation time (Table 1). Notice that preliminary experiments suggested that the longer computation times (in the same order of magnitude) would not lead to improvements both for the SA and for the LPEA approaches.

[2]The instances can be obtained upon request to the corresponding author.

[3]http://www.cplex.com

Table 1 Maximum computation times allowed

$\lvert V \rvert$	10	15	20	25	30	40	50	75	100
Max time (sec)	2	3	4	5	6	7	9	11	14

5.1 Parameters Tuning

We take the following settings as reference values for the parameters of the LPEA algorithm (see Section 4.2): $N_{\mathrm{IND}} = 6$, $N_{\mathrm{OBS}} = 3$, $N_{\mathrm{IT}} = 10$, $\Theta = 0.5$. We will modify some of these values to see the reaction of the algorithm itself. Parameter N_{IND}, regulating the number of individuals of the population, is not considered in the analysis because preliminary tests suggested that it is not crucial: its role can be compensated by operating on the remaining parameters.

The first parameters analyzed are N_{OBS}, regulating the number of observation carried out for each individual during each iteration (Section 4.2.3), and N_{IT}, modeling the number of iterations of each restart of the LPEA procedure (Section 4.2.5). These parameters were considered together in order to maintain the execution time of each run of the algorithm comparable with that of the reference settings, leading therefore to a fair comparison. We consider a configuration with $N_{\mathrm{OBS}} = 5$ and $N_{\mathrm{IT}} = 6$ and another with $N_{\mathrm{OBS}} = 10$ and $N_{\mathrm{IT}} = 3$, and we compare the mean tree powers obtained with those of the reference settings. The results obtained are summarized in Fig. 6, from which it emerges that the reference configuration is the most promising one for these parameters (lower mean tree power for all the network sizes considered).

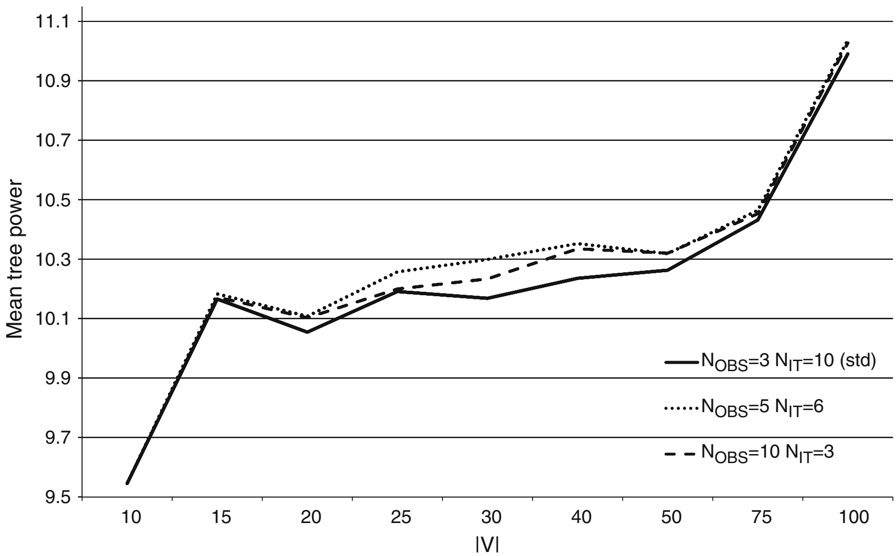

Fig. 6 Tuning for N_{OBS} and N_{IT}. Mean tree powers (averages over 50 networks)

A second set of experiments concerned parameter Θ, regulating the Q-bit evolution speed during the evolutionary phase (see Section 4.2.4). A configuration with $\Theta = 0.2$ and one with $\Theta = 0.8$ where considered together with the standard one ($\Theta = 0.5$). The results are plotted in Fig. 7. Also in this case, the standard configuration clearly emerges as the most promising one.

5.2 Role of the Linear Program F_{LR}

The aim of this section is to show that the linear relaxation F_{LR} has a clear role in the results obtained by the LPEA approach. For this purpose, we consider some variants of the LPEA method. In the variant *No LR init* the linear program F_{LR} is not used during the initialization phase: the first lot of individuals is not anymore initialized to the solution of F_{LR}, but half of it is initialized like the second lot of individuals, and half like the third one (Section 4.2.2). In the variant *BIP init* the first set of individuals ($\frac{1}{3}$) is initialized to the solution provided by the BIP heuristic (see Wieselthier et al. [30]) instead of to the solution of the linear relaxation F_{LR}. This should help to understand if the information brought by the linear relaxation are more promising than those provided by a fast heuristic algorithm, or not. The variant *No LR evol* implements a simplified evolutionary phase with respect to that described in Section 4.2.4: Q-bits β_{ij}^ks corresponding to node i and individual k take the following values, based on W_i^ks:

$$\beta_{i,\lfloor W_i^k \rfloor} = \left(W_i^k - \lfloor W_i^k \rfloor\right);\ \beta_{i,\lfloor W_i^k \rfloor+1} = 1 - \beta_{i,\lfloor W_i^k \rfloor};\ \beta_{ij} = 0\ \forall j \neq \lfloor W_i^k \rfloor, \lfloor W_i^k \rfloor + 1 \quad (13)$$

In this case there is no feasibility adjustment guaranteed by F_{LR} (β_{ij}^ks are not guaranteed to fulfill the relaxed problem constraints). Finally, a variant where no

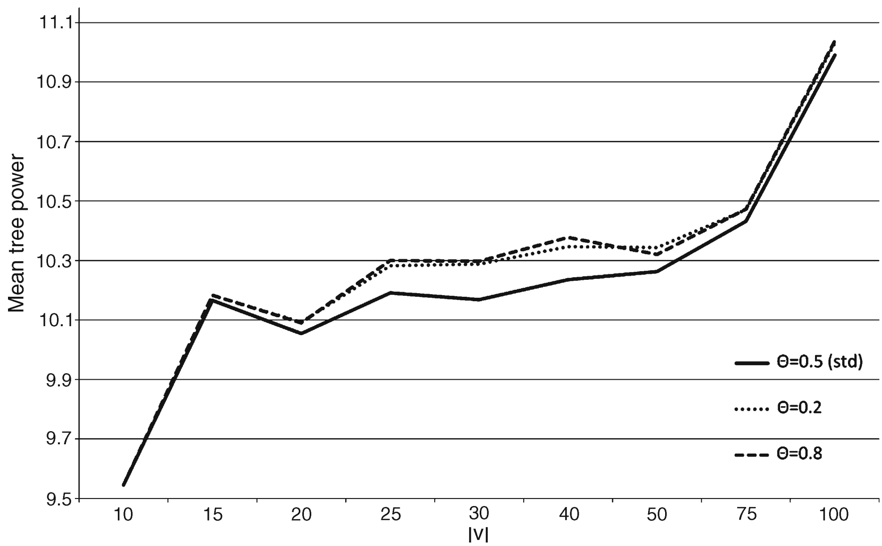

Fig. 7 Tuning for Θ. Mean tree powers (averages over 50 networks)

evolutionary phase is carried out (*No Evo (RRLS)*) is considered. The resulting method can be seen as a Random Restart Local Search (RRLS).

It is important to stress that some of the variants of the LPEA method considered will be intuitively able to perform more iterations than the original one in the given computation time.

The results are plotted in Fig. 8, from which the importance of F_{LR} both in the initialization and evolutionary phases emerges. It is interesting to observed that *BIP init* seems to preclude a satisfactory exploration of the search space. *RRLS* implementation presents similar exploration problems, confirming the importance of the evolutionary phase.

5.3 Role of the Sweep Local Search

The aim of this section is to highlight the importance in the economy of the LPEA approach of the Sweep local search, running during the evaluation phase (see Section 4.2.3). We consider a variant of the LPEA approach where the Sweep algorithm is not run after the probabilistic BIP procedure in the evaluation phase (*No Sweep*). It is important to stress that the version without Sweep will be intuitively able to perform more iterations in the given computation time than the original LPEA algorithm.

In Fig. 9 we compare the results obtained by the modified algorithm with those of the standard LPEA method. The plot highlights the role of the Sweep procedure within the framework we propose. Notice that gap between the two algorithms

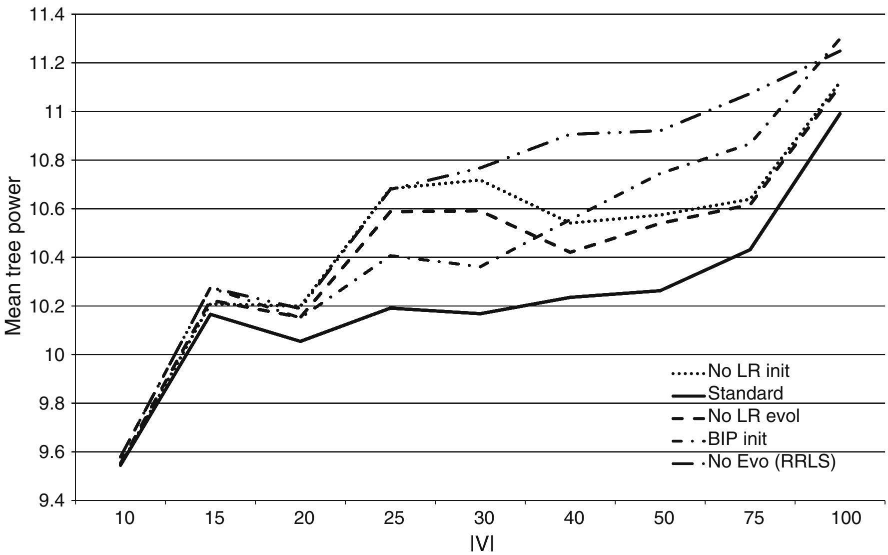

Fig. 8 Role of the Linear Relaxation F_{LR}. Mean tree powers (averages over 50 networks)

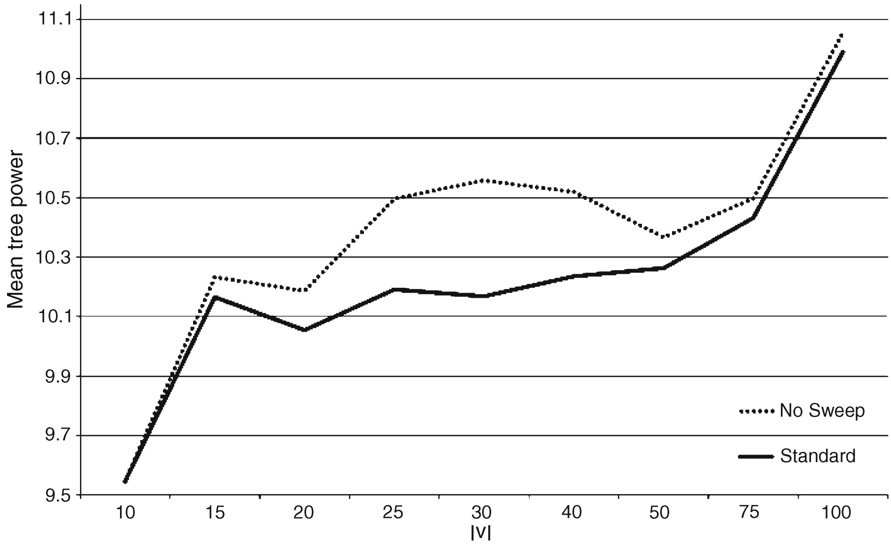

Fig. 9 Role of the Sweep local search. Mean tree powers (averages over 50 networks)

analyzed reduces for the largest problems, but statistical tests confirm that using the Sweep algorithm leads to better results.

5.4 Comparison with Other Algorithms

The new algorithm we propose is compared with some reference methods previously appeared in the literature. The results of the experiments are summarized in Table 2. In the first column of the table the different networks considered are identified by their number of nodes. In the remaining columns the mean tree power obtained by different algorithms are presented. Exact solutions obtained by the exact method discussed in Leggieri et al. [16] are reported when it is possible to retrieve them within one hour for each instance (in fact, up to 30 nodes), otherwise the table

Table 2 Mean tree powers obtained by different algorithms (averages over 50 networks)

$\|V\|$	Exact	BIP	LPEA	SA	SA+LPEA
10	9.5450	11.1000	9.5450	9.5949	9.5450
15	10.1498	12.4973	10.1656	10.2179	10.1535
20	9.9337	12.1599	10.0543	10.0778	10.0072
25	9.9795	12.4551	10.1910	10.2516	10.1476
30	9.7356	12.0240	10.1682	10.0890	10.0178
40	n.a.	11.7506	10.2354	10.0609	10.0189
50	n.a.	11.6618	10.2627	10.1141	10.1086
75	n.a.	11.6265	10.4316	10.2901	10.2541
100	n.a.	11.5955	10.9905	10.6007	10.5796
Average	n.a.	11.8745	10.2272	10.1441	10.0925

entry is marked as "n.a." (not available). We then consider BIP (see Wieselthier et al. [30]), often regarded as a reference algorithm, the LPEA method discussed in Section 4 (LPEA), SA+Sweep (see Montemanni et al. [23]), here referred as SA for short, and SA+LPEA which means a variant of the LPEA where the initial best solution is given by the solution returned by the SA+Sweep method. Notice that the computation time required by SA+LPEA is twice the time required by SA and LPEA. The last line of the table is devoted to the average tree powers over all the instances considered.

The results summarized in Table 2 suggest that LPEA an effective approach. In particular, it is able to retrieve solutions of quality comparable with those obtained with SA, which can be regarded as state-of-the-art. It is however important to observe that the quality of the solution provided by LPEA decreases (with respect to those of SA) as the dimension of the networks increases. It is interesting to observe that all the methods considered present performance degradation (with respect to exact solutions) as the number of nodes increases, suggesting the hardness of the problem itself. On the other hand, SA+LPEA is able to improve the solutions provided by SA, even for the largest instances, on which LPEA alone was less effective. This shows that LPEA is very effective in locally exploring the search space around its starting solution, even when this starting solution is already an optimized local minimum (like those provided by SA). In this context it is important to observe that running either SA or LPEA for twice the time reported in Table 1 does not lead to any improvement over those reported in Table 2. For this reason the SA+LPEA approach appears particularly appealing.

To evaluate the robustness of the heuristic methods considered, we took the first instance of each network size considered, and we run each algorithm fifty times on it. The average and best results obtained (together with standard deviation) are reported in Table 3 for methods LPEA, SA and SA+LPEA. The results suggest that LPEA is more consistent than SA in retrieving quality solutions (comparable average results, lower standard deviation), but SA is able to retrieve higher quality best solutions over the fifty runs considered. This indicates that sometimes SA is probably able to explore a wider search space than LPEA, although in a less accurate way. The method LPEA+SA is confirmed to be able to take the best out of methods SA and LPEA, leading to improved solutions.

Table 3 Mean tree powers obtained by different algorithms on one random instance for each network size (statistics over 50 runs)

$\|V\|$	LPEA			SA			SA+LPEA		
	Average	StDev	Best	Average	StDev	Best	Average	StDev	Best
10	9.4661	0.0134	9.4557	9.4563	0.0039	9.4557	9.4557	0.0000	9.4557
15	12.8586	0.0497	12.8374	13.4654	0.6210	12.8374	12.8412	0.0188	12.8374
20	13.5103	0.1802	13.3779	13.8953	0.5050	13.3779	13.4518	0.0396	13.3779
25	9.7807	0.0000	9.7807	9.8119	0.1665	9.7807	9.7807	0.0000	9.7807
30	11.9339	0.0730	11.8645	11.7876	0.1814	11.6471	11.6554	0.0843	11.5399
40	12.0005	0.2595	11.6261	11.7395	0.3465	11.3563	11.5532	0.1208	11.2439
50	9.4391	0.3100	9.2391	9.3928	0.3127	9.0130	9.1722	0.1039	8.9373
75	9.4956	0.1442	9.4154	9.5755	0.2995	9.2213	9.3690	0.0765	9.1968
100	11.3742	0.0000	11.3742	11.1476	0.5377	10.3982	10.7290	0.4236	10.3881
Average	11.0954	0.1144	10.9968	11.1413	0.3305	10.7875	10.8898	0.0964	10.7509

6 Conclusion

A novel matheuristic approach, the Linear Programming-based Evolutionary Algorithm has been proposed for the minimum power broadcast problem in wireless sensor networks. The new method uses linear programming within the operators of an evolutionary algorithm.

Computational experiments show that the new algorithm is able to retrieve good quality solutions when used alone. Moreover, the method proposed can be used as a refinement tool, able to successfully explore the search space around already optimized solutions, retrieving improved solutions.

It has been shown that linear programming can be effectively integrated within an evolutionary algorithm for the problem considered. Future research will investigate the application of similar paradigms to other optimization problems.

Acknowledgements Parvaz Mahdabi was supported by the Swiss National Science Foundation through project 200021-119720: "Robust Optimization: theory and applications". The authors would like to thank the reviewers for the many useful suggestions provided.

References

1. Althaus, E., Călinescu, G., Măndoiu, I.I., Prasad, S., Tchervenski, N., Zelikovsky, A.: Power efficient range assignment in ad-hoc wireless networks. In: Proceedings of the Wireless Communications and Networking Conference (WCNC), pp. 1889–1894 (2003)
2. Altinkemer, K., Salman, F.S., Bellur, P.: Solving the minimum energy broadcasting problem in ad hoc wireless networks by integer programming. In: Proceedings of the International Network Optimization Conference (INOC), pp. B2.635–B2.642 (2005)
3. Bauer, J., Haugland, D., Yuan, D.: Analysis and computational study of several integer programming formulations for minimum-energy multicasting in wireless ad hoc networks. Networks **52**(2), 57–68 (2008)
4. Cagalj, M., Hubaux, J.P., Enz, C.: Minimum-energy broadcast in all-wireless networks: NP-completeness and distribution issues. In: Proceedings of the Mobicom Conference (2002)
5. Cargigny, J., Simplot, D., Stojmenović, I.: Localized minimum-energy broadcasting in ad-hoc networks. In: Proceedings of the Infocom Conference (2003)
6. Das, A.K., Marks, R.J., El-Sharkawi, M., Arabshahi, P., Gray, A.: The minimum power broadcast problem in wireless networks: an ant colony system approach. In: Proceedings of the Workshop on Wireless Communications and Networking (WWCN) (2002)
7. Das, A.K., Marks, R.J., El-Sharkawi, M., Arabshahi, P., Gray, A.: A cluster-merge algorithm for solving the minimum power broadcast problem in large scale wireless networks. In: Proceedings of the Milcom 2003 Conference, Boston, MA, pp. 13–16 (2003)
8. Das, A.K., Marks, R.J., El-Sharkawi, M., Arabshahi, P., Gray, A.: *r*-shrink: a heuristic for improving minimum power broadcast trees in wireless networks. In: Proceedings of the Globecom Conference (2003)
9. Doolin, D.M., Sitar, N.: Wireless sensors for wildfire monitoring. In: Proceedings of the SPIE Symposium on Smart Structures and Materials, pp. 477–484 (2005)
10. Guo, S., Yang, O.: Optimal tree construction for minimum-energy multicast in wireless ad hoc networks. In: Proceedings of the International Conference on Network Protocols (ICNP) (1999)
11. Han, K.-H., Kim, J.-H.: Quantum-inspired evolutionary algorithm for a class of combinatorial optimization. IEEE Trans. Evol. Comput. **6**(6), 580–593 (2002)
12. Han, K.-H., Kim, J.-H.: Quantum-inspired evolutionary algorithms with a new termination criterion, h_{epsilon} gate, and two-phase scheme. IEEE Trans. Evol. Comput. **8**(2), 156–169 (2004)
13. Kaye, P., Laflamme, R., Mosca, M.: An Introduction to Quantum Computing. Oxford University Press (2007)

14. Kim, S., Pakzad, S., Culler, D.E., Demmel, J., Fenves, G., Glaser, S., Turon, M.: Health monitoring of civil infrastructures using wireless sensor networks. In: Proceedings of the International Conference on Information Processing in Sensor Networks (IPSN), pp. 254–263 (2007)
15. Leggieri, V.: Multicast Problems in Telecommunication Networks. PhD Thesis, Università del Salento (2007)
16. Leggieri, V., Nobili, P., Triki, C.: Minimum power multicasting problem in wireless networks. Math. Meth. Oper. Res. **68**, 295–311 (2008)
17. Magnanti, T.L., Wolsey, L.: Optimal trees. In: Ball, M.O., et al. (eds.) Network Models. Handbook in Operations Research and Management Science, vol. 7, pp. 503–615. North-Holland (1995)
18. Mahdabi, P., Abadi, M., Jalili, S.: A novel quantum-inspired evolutionary algorithm for solving combinatorial optimization problems. In: Proceedings of the Genetic and Evolutionary Computation Conference (GECCO), pp. 1807–1808 (2009)
19. Mahdabi, P., Jalili, S., Abadi, M.: A multi-start quantum-inspired evolutionary algorithm for solving combinatorial optimization problems. In: Proceedings of the Genetic and Evolutionary Computation Conference (GECCO), pp. 613–614 (2008)
20. Mainwaring, A., Culler, D., Polastre, J., Szewczyk, R., and Anderson, J.: Wireless sensor networks for habitat monitoring. In: Proceedings of the International Workshop on Wireless Sensor Networks & Applications (WSNA), pp. 88–97 (2002)
21. Montemanni, R.: Maximum lifetime broadcasting topologies in wireless sensor networks: advanced mathematical programming models. In: Proceedings of the 42nd Hawai'i International Conference on System Sciences (HICSS 42) (2009)
22. Montemanni, R., Gambardella, L.M.: Exact algorithms for the minimum power symmetric connectivity problem in wireless networks. Comput. Oper. Res. **32**(11), 2891–2904 (2005)
23. Montemanni, R., Gambardella, L.M., Das, A.K.: The minimum power broadcast problem in wireless networks: a simulated annealing approach. In: Proceedings of the Wireless Communications and Networking Conference (WCNC) (2005)
24. Montemanni, R., Gambardella, L.M., Das, A.K.: Models and algorithms for the MPSCP: an overview. In: Wu, J. (ed.) Handbook on Theoretical and Algorithmic Aspects of Sensor, Ad Hoc Wireless, and Peer-to-Peer Networks, pp. 133–146. Auerbach Publications (2006)
25. Oliveira, C.A.S., Pardalos, P.M.: A survey of combinatorial optimization problems in multicast routing. Comput. Oper. Res. **32**, 1953–1981 (2005)
26. Platel, M.D., Schliebs, S., Kasabov, N.: Quantum-inspired evolutionary algorithm: a multimodel EDA. IEEE Trans. Evol. Comput. **13**(6), 1218–1232 (2009)
27. Rappaport, T.: Wireless Communications: Principles and Practices. Prentice Hall (2001)
28. Robert, C., Casella, G.: Monte Carlo Statistical Methods. Springer Texts in Statistics. Springer (2004)
29. Stojmenovic, I., Seddigh, M., Zunic, J.: Internal nodes based broadcasting in wireless networks. In: Proceedings of Proceedings of the 34nd Hawai'i International Conference on System Sciences (HICSS 34) (2001)
30. Wieselthier, J., Nguyen, G., Ephremides, A.: On the construction of energy-efficient broadcast and multicast trees in wireless networks. In: Proceedings of the Infocom Conference, pp. 585–594 (2000)
31. Wieselthier, J., Nguyen, G., Ephremides, A.: Algorithms for energy-efficient multicasting in static ad hoc networks. Mobile Netw. Appl. **6**, 251–263 (2001)
32. Yuan, D.: An integer programming approach for the minimum-energy broadcast problem in wireless networks. In: Proceedings of the International Network Optimization Conference (INOC), pp. B2.643–B2.650 (2005)
33. Yuan, D., Bauer, J., Haugland, D.: Minimum-energy broadcast and multicast in wireless networks: an integer programming approach and improved heuristic algorithms. Ad Hoc Netw. **6**(5), 696–717 (2008)