We are IntechOpen, the world's leading publisher of Open Access books Built by scientists, for scientists

6,900
Open access books available

185,000
International authors and editors

200M
Downloads

Our authors are among the

154
Countries delivered to

TOP 1%
most cited scientists

12.2%
Contributors from top 500 universities



WEB OF SCIENCE™

Selection of our books indexed in the Book Citation Index in Web of Science™ Core Collection (BKCI)

Interested in publishing with us?
Contact book.department@intechopen.com

Numbers displayed above are based on latest data collected.
For more information visit www.intechopen.com



Chapter 4

A Novel Hybrid Methodology Applied Optimization Energy Consumption in Homogeneous Wireless Sensor Networks

Plácido Rogerio Pinheiro,
Álvaro Meneses Sobreira Neto,
Alexei Barbosa Aguiar and
Pedro Gabriel Calíope Dantas Pinheiro

Additional information is available at the end of the chapter

http://dx.doi.org/10.5772/intechopen.70207

Abstract

A wireless sensor network's lifetime is influenced directly by the sensors power management that composes the network. The models applied to the problem aims to optimize the energy usage managing the sensors activation in time intervals, activating only the minimum number of sensors respecting the coverage and connectivity restrictions. However, this problem's class has a significant computational complexity and many applications. It is necessary to implement methodologies to find the optimal solution, increasing the network's size, becoming closer to the real ones. This research's objective is to present a method based on a Partition Heuristic aggregating the Generate and Solve method, improving the results, and increasing the network's instances size, while maintaining the flexibility and reliability when applied to the homogeneous wireless sensors networks with coverage and connectivity restrictions.

Keywords: homogeneous models, wireless sensor network, optimize power consumption, Integer linear programming, hybrid methodology, random key genetic algorithm

1. Introduction

Wireless sensors networks (WSN) was initially used to monitor many phenomena, like temperature, atmospheric pressure, humidity, light, sound volume, solar radiation, and many others. These networks are standard, used in environments with challenging access and

INTECH
open science | open minds

© 2017 The Author(s). Licensee InTech. This chapter is distributed under the terms of the Creative Commons Attribution License (http://creativecommons.org/licenses/by/3.0), which permits unrestricted use, distribution, and reproduction in any medium, provided the original work is properly cited. CC BY

hostile. Due to this, methods that optimize the network power consumption to extend the network lifetime are always desired. The network's sensor nodes are typically deployed in the sensed area randomly due to the difficult local access. In these scenarios, some nodes might detect the same field or approximately the same area of another node. This characteristic can lead to redundant sensed data and the premature energy depletion. The solution proposed by [1, 8, 9] is created different network configurations associated with a time interval. In addition to the creation, of many time intervals, it is possible to activate only the necessary sensors to maintain the sensed coverage and connectivity each time interval. Furthermore, the integer linear programming model was proposed by [1, 12] to optimize the energy consumption considering the coverage and connectivity restriction, by allocating the necessary sensors to a pre-defined number of time intervals. As this problem is NP-Complete, it is complexity increases exponentially as the network number of sensors, time intervals, and demand increase.

The solution gets impracticable for the real system sizes. To get closer from real systems, an integrative collaboration of genetic algorithms and integer linear programming tries to merge their high points and has offered significant results improvements [3, 7]. However, its original implementation is shown some deficiencies which limit its performance, being one of them the density explosion. The correct use of a novel hybrid methodology to deal with a class of problem can show good results. The method mainly consists in reducing a problem, typically a high complexity issue, that the time to solve it is infeasible big, into subproblems.

In the wireless sensor network, the genetic algorithm (GA) is used as the Reduced Instances Generator. It is responsible for generating the sub-problems and evolving its population to find the good feasible solution with its value as close as possible from the best solution to the original problem. Although the GA is an excellent approach to solving complex problems in less time, the use of the pure GA might have problems to maintain the feasibility of solutions and execution time. In fact, some operators might be applied to control the chromosome's formation during the crossover operation. Adaptations can be made in the default GA to surpass this problem. The random-key genetic algorithm (RKGA) and biased random-key genetic algorithm (BRKGA) are adaptations of the default GA. They use random generated keys arrays to assist the operation, changing the default GA mutation method. In this work, the BRKGA was adapted attending the WSN problems characteristics. Hybrid approaches are often used to reduce the time spend to solve a problem when a network becomes closer to a real scenario; it finds a difficult time to deal with density explosion. So, the subproblems generated by the genetic algorithm alone (GA and BRKGA) aren't enough to maintain a good solution. Thus, a Partition heuristic is implemented in this problem to enable the more complexes scenarios executions. This new heuristic consists of dividing the whole network, generating subnetworks. This heuristic separates the network area to reduce the complexity. This division is made cautiously not to affect the results on the main problem. After the division, the hybrid methodology is applied over each sub-network. To reconstruct the original problem, a light algorithm, typically a tight one, must be implemented to acquire the first network's solution.

A dynamic model in integer linear programming with restrictions connectivity and coverage applied to the wireless sensor networks had been implemented in this work [4–7]. Moreover, Section 2 is given an outline about wireless sensor network. More details of the models in

http://dx.doi.org/10.5772/intechopen.70207

homogeneous wireless networks are provided is Section 3. On the other hand, in Section 4 provides more information on the applied methodologies. Moreover, in Section 5 the computational results obtained using the heuristic described the integer linear programming model implemented in this work. Finally, Section 6 presents conclusions and future works.

2. The wireless sensor network

The sensor nodes are small devices capable of measure different phenomena. They also can wirelessly communicate to each other to send the sensed data to another instrument. This other appliance, called sink, is a more powerful device responsible for receiving and processing the data from the sensor nodes to finally send it to a server computer. In the same network, more than one sink can be deployed. Furthermore, it has a significant impact on the network topology. A typical sensor node is formed mainly of four components, namely the processing module, the battery, the transceiver, and the sensor module as described [11]. The processor module is responsible for processing all the data sensed but in a simplified way in the microprocessor. The sensor nodes use a small operational system to manage its essential features. Provide power to the sensor; the battery must be present in it. Its lifetime depends on many aspects, like charge capacity and electric current necessary to make the node's components work. The transceiver is a device that sends and receives all the data through radio-frequency propagation. They frequently use public frequency bands to transmit the data. Consequently, external devices might interfere with the network communication. In the same way, communications between sensors nodes of the same network, when there are many close from each other might interfere among themselves, so, the smaller the number of active sensors in the system, more reliable the communication in the network will be.

The sensors coverage radius is an issue that must be finely calibrated by the network manager before it is deployment [14]. For each sensor that should be monitored, a different coverage radius is necessary. On the other hand, this happens because of the phenomenon's variation ratio. For some phenomena, a large coverage space is applicable without denigrating the obtained data quality. Otherwise, some other events need a smaller coverage radius due to theirs high variation and significance on small areas as defined in [15]. The main impact on energy consumption is that the more sensors needed to cover the whole sensed area, the bigger is the current consumed across the network. So, a well-calibrated coverage radius is a critical aspect to be considered. Considering a minimum number of sensor nodes was used to cover a given and often unlimited area of observation a very unreliable and unstable network used to cover a given and often unlimited area of consideration. Furthermore, this is because the coverage areas of the redundant sensor nodes overlap too much, giving birth to redundant data.

3. Model in homogeneous wireless networks

The solution proposed by [7] is to create different schedules, each one associated with a time interval that activates only the set of sensor nodes necessary to satisfy the coverage and

connectivity restrictions. The employment of different schedules keeps from occurring the premature starvation from some of the nodes, bringing about a more homogeneous models energy consumption level across the whole network. Moreover, this is provided because the alternation of active nodes among the schedules is often a design outcome, as it optimizes the overall network energy consumption taking into account all time intervals, coverage, and connectivity restrictions. To accurately model the homogeneous wireless sensors networks setting some previous remarks:

- A demand point is a geographical moment in the region of monitoring where one the phenomenon is sensed. Considering the distribution of points across the area of control can be regular, like a grid and random in nature. At least one sensor must be active at a given moment to sense each demand point. Such restriction is implemented in the model.
- Commonly encountered, the sensors are connected with coverage areas that cannot be estimated accurately. To reduce fundamental parts modeling, we assume open areas without obstacles. Moreover, we think a circular coverage area. On the other hand, the coverage radius is determined by the spatial variation of the sensed phenomenon. The radio-frequency propagation in real wireless sensors networks is also irregular in nature.

The energy consumption is the electric current drawn by a circuit in each period. In what follows, the constants, variables, objective function and restrictions of the integer linear programming model Applied Energy Consumption in homogeneous is present in a step-by-step manner.

3.1. Constants

It presents sets aiming to outline the homogeneous model that stand for the constants.

S Set of sensors

D Set of demand points

M Set of sinks

T Set of n scheduling periods

AD_{ij} Matrix of arcs *ij*, $i \in S$, $j \in D$ *which indicate that* sensor i can cover to demand points j

A_{ij} Matrix of arcs *ij*, $i \in S$, $j \in S \cup M$ that interconnects sensors

EB_i Accumulated battery charge for sensor $i \in S$

EA_i Power needed to activate the sensor node $i \in S$

EM_i Power required to maintain a sensor $i \in S$ active for a period

ET_{ij} Power required for transmitting data from the sensor $i \in S$ to sensor $j \in S$.

ER_i Power needed in the reception of data for sensor $i \in S$

EH Penalty applied when any sensor does not cover a demand point in any time interval.

http://dx.doi.org/10.5772/intechopen.70207

3.2. Variables

The following decision variables are necessary to model the coverage and connectivity problem in wireless sensors networks.

x_{tij} If sensor $i \in S$ covers demand point $j \in D$ in period $t \in T$, *is assigned the value* 1, *otherwise the value* 0

z_{tlij} If arc ij belongs to the route from sensor $l \in S$ to a sink in period $t \in T$, *is assigned the value* 1, *otherwise the value* 0

w_{ti} If sensor $i \in S$ was activated in period $t \in T$ for at least on phenomenon, *is assigned the value* 1, *otherwise the value* 0

y_{ti} If sensor $i \in S$ is activated in period $t \in T$ *is assigned the value* 1, *otherwise the value* 0

h_{tj} If any sensor does not cover demand point $j \in D$ in period $t \in T$ *is assigned the value* 1, *otherwise the value* 0

e_i Electrical charge consumed by sensor $i \in S$ considering all time periods.

3.3. Objective function

The objective function (1) optimize the total energy used by the sensors all time periods. The second term penalizes the existence some uncovered demand points.

$$\min \sum_{i \in S} e_i + \sum_{t \in T} \sum_{j \in D} EH_{tj}\, h_{tj} \tag{1}$$

3.4. Restrictions

The constraint (2) imposes the activation of at least one sensor node i to cover the request point j in period t. Otherwise, the variable that indicates the penalty for not covering the request point is activated. Additionally, allows the model not to result in an unfeasible solution in case there is any point of demand that cannot be covered by any sensor node.

$$\sum_{i \in S} \sum_{j \in D} AD_{ij} x_{tij} + h_{tj} \geq 1, \forall j \in D, \forall t \in T \tag{2}$$

Moreover, the restriction (3) indicates to the model that maintenance energy is being consumed. The variable y indicates whether a sensor that monitors the demand point j is active in the interval t.

$$x_{tij} \leq y_{ti}, \forall i, j \in S, \forall t \in T \tag{3}$$

According to the flow conservation principle applied to the connectivity issue, if there is an incoming route to a sensor node, there should be an outgoing route from this same sensor node. Furthermore, this restriction (4) imposes, setting an outgoing route from the sensor node j to sensor node k if there is already an incoming route from sensor node i to the sensor node j.

$$\sum_{i \in (S-\{j\})} a_{ij} z^{t}_{lij} - \sum_{k \in (S \cup M - \{j\})} a_{jk} z_{tljk} = 0, \forall j, l \in S, \forall t \in T \quad (4)$$

If there is an active sensor, then there must be a path starting from it, as indicated in restriction (5).

$$\sum_{k \in (S \cup M - \{l\})} A_{ij} z_{tljk} = y_{ti}, \forall i \in S, \forall t \in T \quad (5)$$

On the other hand, the constraint (6) is necessary to create a path that reaches a sink if a sensor is active.

$$\sum_{i \in S} \sum_{j \in M} A_{ij} z_{tlij} = y_{ti}, \forall t \in T, \forall i \in S \quad (6)$$

Furthermore, in restriction (7), if there is an outgoing route passing through sensor node i, then this sensor node should be active.

$$A_{ij} z_{tlij} \leq y_{tj}, \forall j \in S, \forall l, i \in (S - \{k\}), \forall j, \forall t \in T \quad (7)$$

All data sensed must attain a sink node. On the other hand, the quantity sensors node has no direct connectivity to a sink node. Furthermore, others sensor nodes might be activated just to turn viable the route to the sink. Moreover, with restriction (8) if there is an incoming route passing through sensor i, then this sensor has to be active.

$$A_{ij} z_{tlij} \leq y_{ti}, \forall j \in S, \forall l, i \in (S - \{j\}), \forall t \in T \quad (8)$$

The total energy consumed by a sensor node is the sum of the parcels in the restriction (9).

$$\sum_{t \in T} \left(EM_i y_{ti} + EA_i w_{ti} + \sum_{l \in (S-\{i\})} \sum_{k \in S} ER_i z_{tlki} + \sum_{l \in S} \sum_{j \in (S \cup M)} ET_{ij} z_{tlij} \right) \leq e_i, \forall i \in S \quad (9)$$

The maintenance energy is attributed when the sensor is active for any reason. The activation energy is summed only when there was an effective activation through time intervals. The reception and transmission energy are specified when there are incoming and outgoing routes respectively passing from a sensor node. Additionally, the sum has been less or equals than the battery's energy. Additionally, the restriction (10) enforces that each sensor node should consume at most the capacity limit of its battery.

$$0 \leq e_i \leq EB_i, \forall i \in S \quad (10)$$

Also, if a sensor is active in the first time interval, it means it consumed energy to activate. The w variable indicates this activation. In addition to, the variable's value is set to 1. By the other hand, if the sensor is kept off in the first-time interval, the value is set to 0. Restriction (11) ensures that.

$$w_{0i} - y_{0i} \geq 0, \forall i \in S \quad (11)$$

http://dx.doi.org/10.5772/intechopen.70207

Also, in restriction (12) the sensor's past and current activation states are compared. If the sensor node was active from period t − 1 to period t, then w is set to 1, 0 otherwise.

$$w_{ti} - y_{ti} + y_{(t-1)i} \geq 0, \forall i \in S, \forall t \in T,\ t > 0 \tag{12}$$

Finally, the restriction (13) only indicates that the decision variables x, w, y, z, and h are binary. Moreover, the decision variable e belongs to the set of real numbers.

$$x, w, y, z, h \in \{0, 1\},\ e \in R \tag{13}$$

4. Methodologies applied

This section presents the methods that have been applied in this work. The Generate and Solve method, described in [2], controls the application of genetic algorithm and Partition heuristics. Partition heuristics, in turn, is applied during the evaluation process of individuals of the genetic algorithm.

4.1. The Generate and Solve Methodology

Exact algorithms and metaheuristics are distinct, approaches to solving combinatorial optimization problems efficiently, each presenting advantages and disadvantages. Some of these hybrid algorithms aim to obtain optimal solutions with smaller execution times, while others seek to achieve better heuristic solutions [18]. The solutions of the instances generated by metaheuristics are determined when the subproblems are solved by an exact solver, which functions as a decoder. To achieve better heuristic solutions, the proposed hybrid methodology [19] suggests the integration of an accurate method to a metaheuristic generating small instances of the problem addressed, according to **Figure 1**. The hybrid mechanism is intended to determine a subproblem that can be solved correctly and provide a good solution to the original problem. The reductions made to the problem seek to limit the number of possible

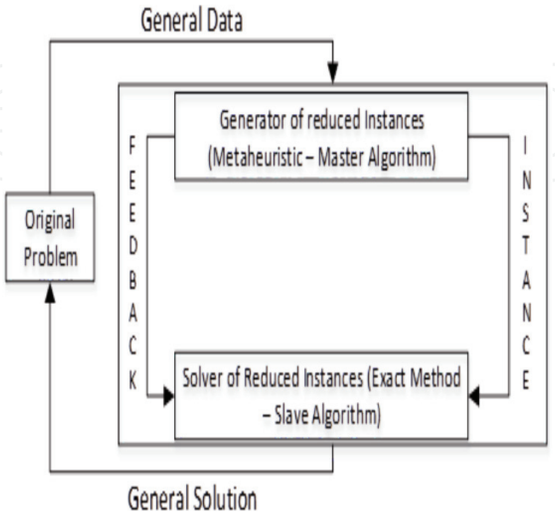


Figure 1. The hybrid framework.

solutions, always respecting all the restrictions imposed to the initial problem. In this way, the optimal solution found for each generated instance is also a viable solution for the problem to be solved. Also, the value of the objective function for the optimal solution of each subproblem, obtained from the exact method (encapsulated in the Solved of Reduced Instances (SRI) component), defines the quality of the generated instance. As shown in **Figure 1**, the information can be used to guide the search process of Generator of Reduced Instances (GRI).

4.2. Density control

When the genetic algorithm is applied to generate and solve methodology may occur a problem known as explosion density. To the extent that the generations are evolved, there is the possibility that the algorithm converges to a great place where the number of sensors participating chromosomes increases significantly, causing an alignment of subproblems of the original problem undermining the execution in time feasible and proper implementation of the methodology. To prevent this issue was introduced by [10] a density control operator. This operator is applied while generating reduced levels, after executing the intersection operator, replacing the mutation operator, as shown in **Figure 2**. The more disabling this sensor impair the value of the objective function, checking aspects of coverage, connectivity, and the penalty for not covering demand points, the more chances it must be maintained. This amount is credited only to chromosomes that are active after crossing operation. Inactive genes, i.e., equal to "0" did not receive credit.

Its purpose is to measure the density of the chromosomes; it is necessary to use a parameter called density ideal (DI). When the method is executed, a check of each chromosome density is made by comparing the credit value with value density ideal. If the density is less than or equal to DI, the chromosome remains unchanged. Otherwise, the genes with zero credits values that are active are disabled randomly. If the chromosome density is still higher than the DI even disabling genes with a null value, genes with lesser value become disabled until the density meets the density parameter ideal. In the latter case, there is a possibility of damaging the final solution, because genes that have some value for the objective function are disabled. Thus, this last step becomes optional and should be activated according to the application scenario. If time is less relevant, it is interesting to keep this step.

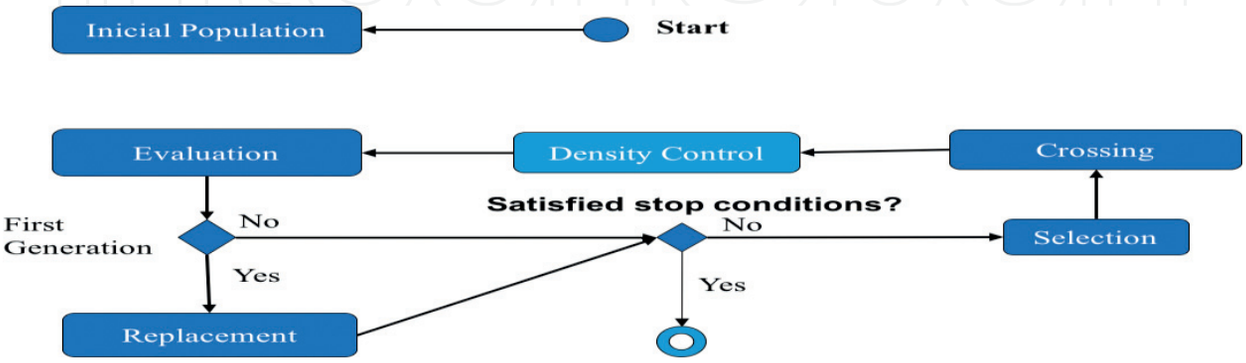


Figure 2. Genetic algorithm with density control.

http://dx.doi.org/10.5772/intechopen.70207

4.3. Random key genetic algorithm (RKGA)

The genetic algorithm (GA) is chosen to be the Generator of Reduced Instances. To adapt the problem so the GA can be used, a reducible structure must be set maintaining the original's problem characteristics. Moreover, the main goal of this question is to indicate which sensors will be active in which time interval. The reducible structure matrix describes if a sensor can be activated or not in each time interval. **Table 1** presents the alleles that compose a chromosome for a network instance with three-time intervals and four sensors.

The genetic algorithms are applied in many solutions in problem classes. However, accordingly, to [1], the basic Genetic Algorithm has a difficult time to maintain feasible solutions when generating the offspring chromosomes. So, this might reduce the final result's quality. To mitigate this problem introduced a new genetic algorithm class called random key genetic algorithm. Those keys belong to real numbers set ($n \in \mathbb{R}$). Additionally, store the values; an array is used. A decoder, frequently a deterministic method, uses the random-key array as input data to return a problem's feasible solution. As the RKGA is an evolutionary process, it evolves a chromosome population, randomly generating the alleles into [0, 1] interval. After the decoder calculates the fitness value for each individual, the population is partitioned into two individual groups, the elite ones, and the non-elite ones. The first one, with fewer individuals, with higher fitness value, and the other ones, a bigger group, with smaller fitness value. Typically, the RKGA uses the elitism to create the next generation. Thus, all elite individuals are cloned from k generation to k+1. Following this, a mutation operator must be applied. The mutants are created using a random-key array like happens with the first generation. After the mutation, it is obtained a set of $P - Pe - Pm$, where P is the current population, Pe is the set of elite individuals, and Pm is the set of mutant individuals. Finally, the next generation is created using the elite, non-elite, and mutated individuals, generating the necessary number of individuals to reach the population's size. The crossover operation is executed by an operator introduced by [2], called parameterized uniform crossover. Given to parents A and B, stochastically selected from the current population, the child chromosome's alleles are chosen from parent A or B, respecting a probability P, for choose from one of them. **Figure 3** represents the RKGA generation k to generation k+1.

The application was proposed in two phases. The first step consists of constructing stables blocks and boxes and a second one that installs these blocks on the container floor. Conducted by Generate and Solve as its Reduced Instances Decoder. The experiments were made using library test cases, and the technique exceeded the best methodologies found in the literature for several cases. The decoding is made in three phases. First, a packing sequence of the type of the item is created, using a random-key array. Additionally, a rotation variance array is generated, and finally, it is decoded an allocation procedure array. Promising results are applying the RKGA comparing with approaches to solving the problem.

Array index	0	1	2	3	4	5	6	7	8	9	10	11
Alleles	1	0	0	1	0	1	1	0	0	1	0	1

Table 1. A chromosome alleles representing sensors by time intervals.

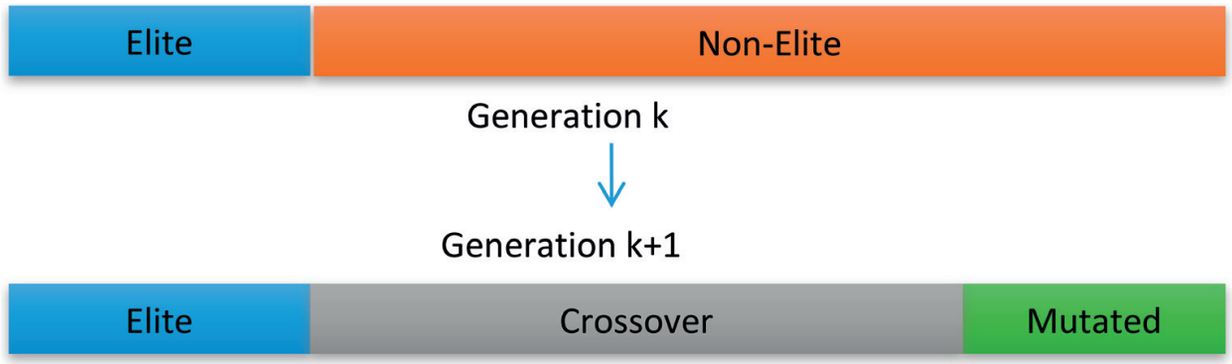


Figure 3. RGKA generation k to k+1.

4.4. Biased random-key genetic algorithm (BRKGA)

The main difference of the BRKGA to RGKA is that BRKGA has different ways to select parents to create the offspring during the crossover operation. In this one, the key array is composed of various genes, which are coded into the real number interval [0, 1]. Following this, as the RKGA and the conventional GA, a deterministic method is used to calculate the chromosomes fitness values and executing the remaining operations and generations until a stop condition is satisfied. In this work, three different decoders were used. First, it was used a decoder with job allocation's priority equals to the gene's value. A second decoder, where the combination of gene's value and ideal priority value give the priority. Finally, a third decoder is used. This one is a hybrid decoder that combines the other two, obtaining two solutions for one chromosome. It was concluded that the hybrid decoding method is appropriate to be applied to BRKGA to solve multi-objective problems, which many feasible solutions are necessary. Basically, for each crossover operation, it is generated a new array of random keys. The keys generated work as a probability value P_i. When executing the crossover operation, two random parents are chosen and, for each allele of the child chromosome, the parent to heir the gene is chosen accordingly to the random-key array. Each parent has a 50% probability to be selected for each gene. **Table 2** represents the process described.

4.5. Division of observation period

They are presented by [16] many of integer linear programming to optimize energy consumption but do not consider the use of dynamic schedule. The solution proposed by [1] is to create

Rk-array	0.58	0.42	0.11	0.83	0.38	0.75	0.68	0.39	0.21	0.94	0.46	0.88
Parent A	0	1	1	0	1	1	1	0	0	0	1	0
Child	0	1	0	0	1	1	1	0	0	0	1	0
Parent B	1	1	0	0	1	0	1	0	0	1	1	1

Table 2. BRGKA child generation accordingly to a random-key array.

http://dx.doi.org/10.5772/intechopen.70207

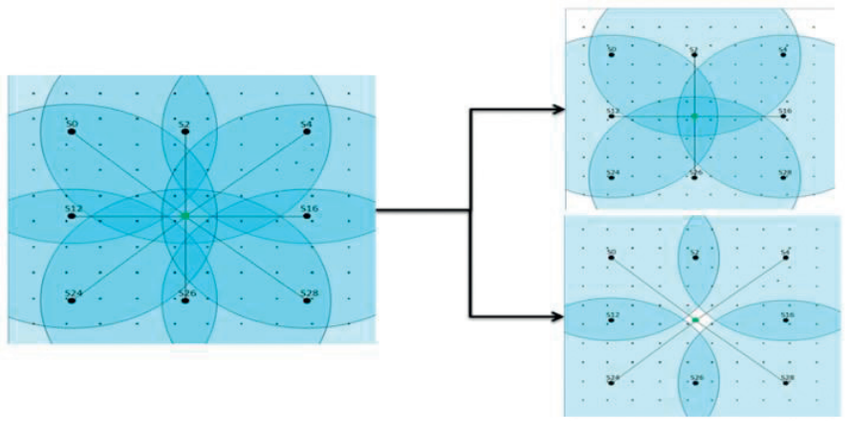


Figure 4. Division of the observation period.

different time intervals, which only activate the minimum number of sensors needed to cover all points of demand and to satisfy the connectivity restrictions. The use of different time intervals prevents premature depletion of some sensors, thus bringing a more homogenous level of consumption of the battery power of sensor nodes of the network. Considering only a distance of coverage for all monitored phenomena implies that the radius used shall correspond to the phenomenon that has the smallest radius of coverage. In the same way, just take a sample rate for all phenomena implies that the rate used should be the phenomenon that varies more often. Furthermore, in **Figure 4**, it corresponds to the left to the deployment of a network without using multiple time intervals, i.e., all the sensors are activated and are monitoring the region corresponding to its coverage radius to drain the battery power. So, this creates a large volume of redundant data, which should be treated not to harm analysis. In this case, when a sensor node is activated, it must be active to drain the battery power. Also, in **Figure 4** on the right shows the application's network division into periods, activating only the required nodes sensors to cover the maximum possible demand points. In this example, the lifetime of the network is doubled.

4.6. Implementation of genetic algorithm in WSN

Apply efficiently to Generate and Solve Methodology; it is necessary to find a way to reduce the problem size and represent it in a chromosome. Implementation of WSN applied in this study, the values of the genes indicate that the sensors in time and similar phenomena participate in the sub-problem. Therefore, the implementation of the methodology shows what restrictions or variables must be taken from the mathematical model according to the indications of the chromosome. The data structure used to represent the genes is a vector of binary numbers {0, 1}, where "0" indicates that the sensor node cannot be activated in the corresponding time interval and the value "1" indicates that a sensor node can be enabled in the relevant range. Many vector indexes are corresponding to the number of sensor nodes multiplied by the number of time intervals. **Table 3** shows a chromosome relating to an instance with three times intervals and four sensors. The indices 0 through 3 correspond to

0	1	2	3	4	5	6	7	8	9	10	11
0	1	1	0	1	0	0	1	1	0	1	0

Table 3. Chromosome, for instance, WSN homogeneous.

the genes about sensors 4 in time interval 1. Similarly, there is an indication of the values of the genes in subsequent time intervals.

The second way is to keep the chromosomes as is done in homogeneous networks, indicating the possibility of activation of the sensor at time intervals. The selection of individuals for the crossover operation is made through a technique called roulette. The chances that an individual should be selected corresponds to a share of proportional roulette to their fitness value. The fitness values of all individuals are joined in the total amount of roulette. Each individual has an equal share to its fitness value. It then generates a random number within the range comprising the total value of roulette. This value indicates which individual was selected. The crossover operation applied this GA was the Crossover one point. For each pair chosen to participate in the operation, is selected at random a cut-off point. The genetic material from each parent is divided and exchanged with each other, creating two new chromosomes. Each individual who will participate in the next generation undergoes the mutation procedure. Each chromosome gene has a probability of 0.5% of having its value altered. A random value between 0 and 1 is generated. If this value is less than 0.05, the allele undergoes the mutation.

4.7. Application of BRKGA Wireless Sensor Network Problems

Among the differences between the GA and BRKGA, one is the intersection operation. In execution the crossing of individuals applying the random key algorithm, a key vector is generated, and the operation is performed structured on the values of this vector. First, it was implemented RKGA to carry out the crossover operation. Through a standard acceptance value of 0.5, for example. It is decided whether the sensor will participate in the reduced problem instance. If the value of the random key is greater than the acceptance value, then the sensor part of the reduced instance, as exemplified in **Table 4**.

Enhance application of the algorithm in WSN problem; the BRKGA algorithm was modified. This time, instead of indicating whether a sensor may be activated at a time interval only if the value of a key is greater than the acceptance value, the algorithm generates, for each allele a random value and compares it with the corresponding key. Thus, the higher the value of your key, the more chances you have a sensor to participate in the sub-problem. Thus, even if one allele has a random key with a little value. It can still be selected to participate in the reduced instance, even if their chances are lower. **Table 5** demonstrates the application.

4.8. Heuristic Partition

The application Generate and Solve Methodology significantly reduces the original problem into smaller problems maintaining its characteristics, allowing solutions to be found next to the solutions of the original problem. However, even with the improvement obtained by the

http://dx.doi.org/10.5772/intechopen.70207

Index	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23
Gene	0	1	1	0	0	1	1	1	0	0	1	1	1	0	1	0	1	1	1	1	0	0	0	1
Key	0.32	0.73	0.54	0.28	0.13	0.84	0.56	0.73	0.25	0.41	0.55	0.68	0.56	0.18	0.92	0.37	0.76	0.51	0.65	0.85	0.08	0.36	0.22	0.89

Table 4. Random keys BRKGA.

Index	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23
Gene	0	1	1	0	0	1	1	1	0	0	1	1	1	0	1	0	1	1	1	1	0	0	0	1
Key	0.72	0.3	0.44	0.28	0.13	0.84	0.56	0.73	0.65	0.41	0.55	0.68	0.56	0.18	0.92	0.37	0.36	0.48	0.65	0.85	0.08	0.36	0.22	0.89

Table 5. Random keys adapted BRKGA.

methodology are instances where even higher, the runtime ends up again impractical. The Generate and Solve Methodology is the reduction of the original problem into subproblems, less complex, preserving the original features of the problem. Similarly, the Heuristic Partitioning is also to reduce the size of the problems. It subdivides the sensing region of the wireless network into smaller pieces, creating subnets of an original network, generating partitions, the Generate and Solve Methodology should be applied. Additionally, variables that do not belong to those partitions are discarded during the execution of each party. This fact reduces the complexity of the problem. After the solution of each partition, it must perform the junction of the results of each partition to get the results for the original problem. This heuristic must have low run-time, as a real solution had been obtained previously. The larger the sensor the coverage radius for a phenomenon, fewer sensors are required to cover the entire region of interest. So, this should be considered to calculate the optimal size of the partitions. When partitions are generated based on the phenomenon that has a greater distance to cover, the size of them may end up too big, getting very close to the size of the original network. Therefore, the time to solve large problem instances will not be feasible. However, if the partition sizes are too small, when the junction of the partitions is done there will be many active sensors in a small area, especially when there are phenomena whose coverage rays are large.

4.8.1. Application forms

Two ways of application of Partition heuristics were tested. In the first case, the wireless network is divided into partitions. Thus, each partition is treated in isolation or is regarded as a different network and the Generate and Solve then executed one time for each partition generated. Thus, when the chromosomes of the genetic algorithm are generated, some alleles correspond only to the number of variables relating to the partition that is being resolved. **Figure 5** is the problem of the execution flow by applying this way:

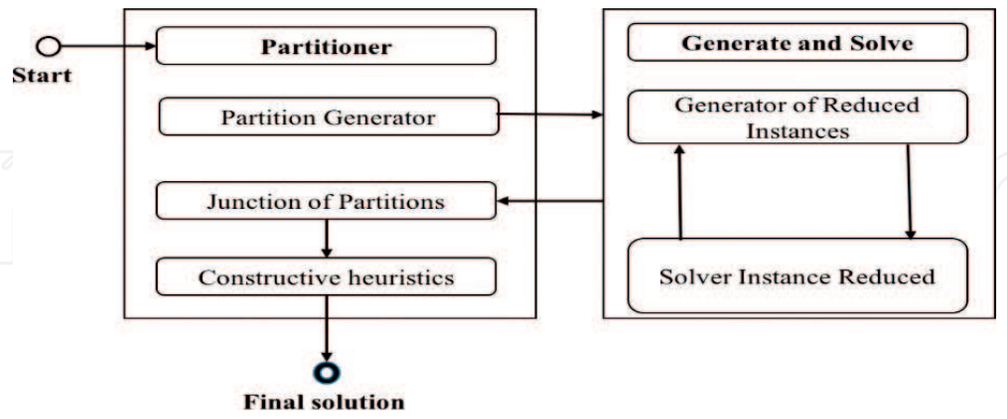


Figure 5. Partition Heuristic flow applied before Generate and Solve.

This approach is more efficient in problems in that all phenomena network coverage rays are identical. The size of the partitions is too large, it is very close to the original problem, making it impractical. It was then implemented another form of application of Partition heuristics, to solve this issue. This time, it is executed within the Generate and Solve Methodology. When the

http://dx.doi.org/10.5772/intechopen.70207

genetic algorithm performs the assessment of the chromosome, it runs the exact method to get the result of the sub-problem. Only this time the partitions are generated for the problem. So, every individual assessed, the exact method is executed n times, where n is the number of partitions. In an other manner, the chromosomes generated for each individual had the size corresponding to the variables of the corresponding partition. In this case, the treated chromosomes are the same size as the original problem, but each partition only considers the values of the genes found on chromosomes that match your partition. The key to this approach is precisely at this point. For after the junction of the partition, the value obtained will be the result of the chromosome in question. Then, in the case of phenomena with a large radius of coverage, the result of the separate partition may contain time intervals in which there is no active sensor. For the separate partition that corresponds to a bad result, injuring coverage restriction. However, there are high chances of a sensor on another partition cover the region due to its large radius coverage. So, **Figure 6** shows the flow for implementing.

Another key point to obtain a good solution implement the second approach is to facilitate the partitions exist solutions without active sensor at some time intervals. At this point, density control operator has fundamental importance. It will generate instances where there is no possibility of any sensor cover a region of the monitored space. For this, the operator must be applied separately to chromosome regions representing each phenomenon, on a chromosome where allele 1–8, the values correspond to the temperature phenomenon, which has smaller coverage radius, and allele 9–16. The values correspond to the allele light phenomenon, which has a larger radius of coverage. After solving the problems related to the partitions, it must perform the junction of the partitions back to the original state. At this point, it is necessary to calculate the net energy consumption. The power consumption of each sensor of each partition should be added since no sensors were sharing between partitions. The calculation of the penalty for not demand coverage points should be made after the joint because in some cases, demand points that were not covered by a sensor of a partition can be covered by other sensors which were in another partition. In some cases, sensors can be activated very close to each other between different partitions when they are close to the circumferences of the partitions. A constructive heuristic should be executed to perform the junction of the partitions to optimize this problem. This heuristic adds a new layer in wireless sensor networks problem by improving the performance of the implementation of the problem, trying to solve it in time

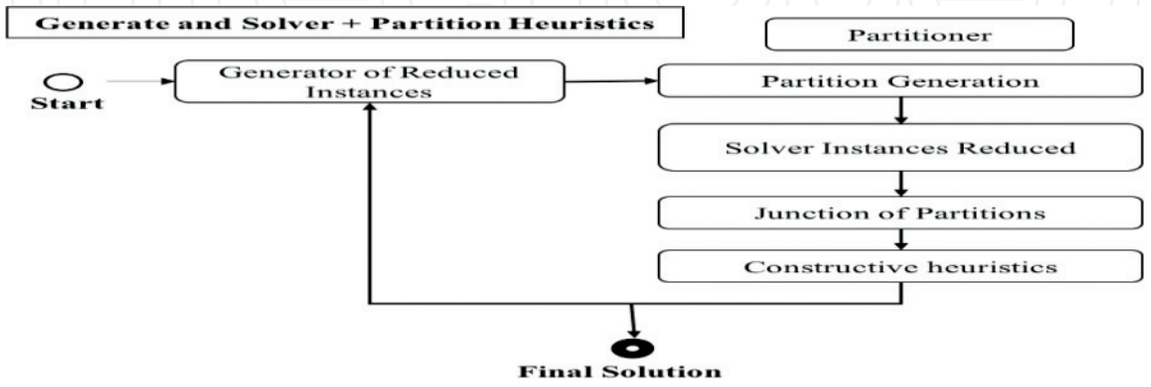


Figure 6. The flow of Partition Heuristics applied internally in Generate and Solve.

viable, keeping the restrictions of coverage and connectivity. The method selected to apply the WSN problem was nearest neighbor, due to its low complexity and the initial solution is the best solution that the methodology applied encountered so far. Neighbors were generated according to the possibility of sensor activation according to the remaining power of their respective batteries following the following rules:

- Sensor activation combinations are generated that still have enough energy at different time intervals. In some cases, the result obtained by the hybrid method can end up leaving the region of a sub-area of interest with the possibility of finding a sensor cover it. So, neighbors are generated starting from this premise, where sensors still have energy are activated and evaluated. This rule holds just one activation neighbor. That is, if more sensors can be activated either at different time intervals or the same interval, only one will be enabled. The remaining sensors will generate new neighbors. In such cases, the activation of more than one sensor on the first individual is performed in subsequent generations.
- It is made of a sensor off when there are two active sensors very close to each other in the same time interval. Sensors monitor tend to close redundant areas, consuming energy that could be used in other time intervals. Disabling these sensors involves the reduction of the objective function value and, as the goal is to minimize the total consumption of the network, the neighbors generated with this rule have a higher value of fitness. This rule is checked whether the candidate sensor to deactivate the route is part of another sensor in the same time interval. To recalculate the objective function, it needs to check in what point interval each sensor is active and which phenomena each is monitoring. Thus, it is possible to identify how much energy each sensor starts to consume according to data consumption, MOTE Battery Life Calculator [17].

5. Computational results

The intentionally of the model is to optimize the energy consumption without homogeneous wireless sensor networks. The number of time intervals has fundamental importance for this period of the total time of network life is obtained by multiplying the number of intervals for the duration of each of them. However, networks of wireless sensors are usually allocated in severe environments, so it is necessary to extend the maximum time that the network is active without compromising its reliability. The combination of the number of time intervals, sensors, demand points, phenomena, an array of sensors and types of genetic algorithms make different scenarios. Since the purpose of the model is to optimize the lifetime of the network, have been generated scenarios first structured in the number of time intervals of 2, 4, 6, 8, 10, 12, 14, 16, 18 and 20 times intervals. For each of these scenarios, instances were generated structured in a combination of features: Sensor Nodes Quantity: 16 sensors, Phenomena: Temperature, Sensor allocation: in a grid or randomly and Genetic Algorithm Type: AG or BRKGA. The instances with 16 sensors nodes were created, and to instances of 16 sensors were allocated, 100 points of demand. In turn, for each one of these bodies were set other sub-sets with the sensors arranged in grid form and randomly arranged sensors. Then, for each of these subsets, others were performed by applying the AG and the other with BRKGA. The values shown in the tables is the arithmetic mean of the values found in 10 executions. The temperature phenomenon was used in scenarios with just a phenomenon,

with an 8.8 m coverage radius. The demand points were arranged in a regular grid in all scenarios, in an area of 20 × 20 m with a demand point per m^2. Each demand point is associated with a phenomenon. Then a demand point can be served to a phenomenon and not the other, in the same time interval. The sensor nodes when distributed in grid form, are allocated analogously to demand points. Otherwise, the position of each sensor is chosen at random using a uniform distribution function for generating the network. The sensor transmission range, which allows the exchange of information between them to arrive at a sink node, was 11 m. All sensor nodes have the same characteristics of transmission, reception, coverage, connectivity, and battery capacity. Only one sink node has been allocated to the centers of the monitoring area in all instances. All network elements were generated by applying concepts of geographic coordinates. The power consumption values were calculated based on the values found in the sensor manual. The total transmission and reception relating to monitoring data have been computed based on the amount of data transmitted by the devices. It was awarded a penalty considerably high value not to allow the model disable sensors unnecessarily increasing the number of demand points not covered. The results are displayed actual values with the values of the penalty and the values representing the energy used by the sensors. Moreover, the model of integer linear programming was implemented in the Java programming language 7, using the CPLEX library 12.1 academic licenses, responsible for integrating with the solver used. The computer used has an Intel Core i7 fourth generation processor, 8GB of RAM, running Windows 7 operating system 64-bit. It is emphasized that the value of the objective function is the sum of the energy used by all sensor nodes during all time intervals with the values of the penalties applied when a demand point is not covered. However, for practical purposes, the value that is only the net energy expenditure is also important. Thus, tables shows the total value as "Value Obj" and the value representing only the energy used by the sensors as "Energy." The "% not covered" represents the average percentage of demand points not covered. To emphasize the need to apply heuristics to the problem, **Table 6** shows the results obtained by performing only the exact method for scenarios with 16 sensors, both cases one allocation grid sensors and random. The time required to carry out the scenarios is impractical. So, it is because the only difference between the instances is some sensor nodes.

It is a remarkable increase in the size of cases 100% about some time intervals. However, in cases with a phenomenon, 12, 14 and 16 times intervals, it is remarkable that the value of the objective function remains with similar values across instances, and some demand points not

16 Sensors/1 phenomenon

Time intervals	Variables	Constraints	Grid			Random		
			Value Obj	% not covered	Time	Value Obj	% not covered	Time
2	12217.00	19648.00	5376.22	0.00	0.03	5376.22	0.00	0.03
4	48689.00	78384.00	10752.43	0.00	0.08	13203.82	0.02	0.12
6	73025.00	117552.00	15698.30	0.00	0.12	15698.30	0.04	0.13
8	97361.00	156720.00	21074.52	0.01	1.97	21358.16	0.04	25.18

Table 6. Execution with ILP, 16 sensor nodes.

covered increases dramatically. Moreover, this is due to the total collapse of the network energy. However, as in such cases, the energy used tends to be higher, depletion of energy is already noticeable on executions with eight times intervals on, therefore, is not necessary to run instances with larger amounts of time intervals. In such cases, the number of demand points increases according to the number of time intervals, since there would be no more sensors to meet the request. It is possible to observe the instances 12, 14 and 16 times intervals implemented with BRKGA that the energy expended values are identical in **Table 7**. So, this was due to the depletion of the energy of all the receivers.

1 Phenomenon/16 sensors/allocation grid								
Time intervals	**Partition + GA**				**Partition + BRKGA**			
	Value Obj	**Energy**	**Not covered**	**Time**	**Value Obj**	**Energy**	**Not covered**	**Time**
2	5376.22	5376.22	0.00	1.41	5376.22	5376.22	0.00	1.33
4	15831.43	10752.43	2.75	2.15	14256.43	10752.43	2.92	1.61
6	23078.30	15698.30	3.50	3.09	22028.30	15698.30	3.52	2.60
8	30800.52	21074.52	4.05	7.77	30692.52	21074.52	4.01	4.51
10	40991.74	26450.74	4.85	6.58	40910.74	26450.74	4.82	4.70
12	68753.18	29483.18	10.91	6.33	54565.61	31396.61	6.44	4.59
14	164963.53	30278.53	32.07	7.89	130714.61	31396.61	23.65	5.50
16	226349.95	31826.95	40.53	9.30	173134.61	31396.61	29.53	6.17
1 Phenomenon/16 sensors/allocation random								
2	8273.52	8273.52	0.00	1.95	8273.52	8273.52	0.00	1.06
4	14802.82	13203.82	0.00	2.49	14913.82	13203.82	0.00	1.68
6	23633.30	15698.30	3.50	3.63	23744.30	15698.30	3.50	3.07
8	55332.23	23637.23	13.21	6.24	54979.36	23242.36	12.86	5.45
10	84676.68	23938.68	20.25	7.74	83176.50	24664.50	19.50	7.47
12	131714.63	23762.63	29.99	7.91	114270.74	25509.74	24.66	7.46
14	179129.22	27128.22	36.19	9.64	178187.74	26150.74	34.87	6.87
16	228660.20	27549.20	41.90	14.79	216423.20	27687.20	39.32	10.18

Table 7. One phenomenon, 16 sensors, and 100 demand points.

6. Conclusions and future work

Despite the growing advancement of the processing power of today's hardware, it is not possible to perform exact methods to solve the problem of coverage and connectivity in Wireless Sensor Networks. The Generate and Solve Methodology was implemented to solve the problem in homogeneous networks [10]. However, as the network grows instances, the execution time tends to be impractical again. The Heuristic Partitioning layer adds a further problem, further reducing

http://dx.doi.org/10.5772/intechopen.70207

the time required to find a feasible solution with good instances 100% higher in both types of networks. Not only it is remarkable the possibility of growth scenarios, but also the quality of the results, in turn, exceeded the results found in the literature on total consumption of network energy and demand points coverage. There are scenarios in which some sensor nodes are far away from the sink node. Thus, they cannot communicate directly and require others node (s) sensor (s) to provide the route to the sink. However, particularly in instances with many time intervals, there is the tendency of all the sensors that could promote this route does not possess sufficient energy to do so. In such cases, these sensors end up being little use or even not used, in general, leaving demand points not covered. One possibility to solve this problem is to allocate some extra sensors to the network. The weakness of this method is that the inclusion of sensors leads to increased complexity, which should not significantly increase the runtime by applying the methodology set. It has been seen that with the applied genetic algorithms solutions can be found in less time than with BRKGA. Tests were performed with a larger battery capacity of sensor nodes to test how far it can increase the complexity of the scenarios solving them infeasible time. A significant increase in the number of time intervals was observed. In some situations, it was found an increase of 300% compared with the amount of execution time intervals in the literature. Therefore the ability of the methodology to be even closer to real network instances; they are allocated some sensor nodes greater than the amount ranges of the tested scenarios. Also, used results found in [10] for comparison. In it, the coding of chromosomes occurs differently, using a fixed number of alleles. These alleles are represented indices of sensors participating in the problem.

As future work, we highlight the application of coding the problem of heterogeneous wireless sensor networks and as well as the Partition Heuristic. Another future work would consider Heuristic partition in the container loading problem in parallel to Generate and Solve Methodology which was initially, applied to this problem in [2, 13]. The challenge lies in how to produce partitions without hurting the quality of results and maintains the possibility of allocating all kinds of items required within the container.

Acknowledgements

The authors are thankful to National Counsel of Technological and Scientific Development (CNPq), Foundation for Support of Scientific and Technological Development Ceara State (FUNCAP) and Coordination for the Improvement of Higher Level or Education-Personnel (CAPES) for the support received on this project.

Author details

Plácido Rogerio Pinheiro*, Álvaro Meneses Sobreira Neto, Alexei Barbosa Aguiar and Pedro Gabriel Calíope Dantas Pinheiro

*Address all correspondence to: placidrp@uol.com.br

University of Fortaleza – UNIFOR, Graduate Program in Applied Informatics, Fortaleza, CE, Brazil

References

[1] Aguiar AB. Tackling the problem of dynamic coverage and connectivity in wireless sensor networks with an extended version of the generate and solve methodology generate and solve methodology [Master's dissertation]. Graduate Program in Applied Informatics, University of Fortaleza, Brazil; 2009

[2] Aguiar A., Pinheiro PR, Coelho ALV, Nepomuceno N, Neto A, Cunha RPP. Scalability analysis of a novel integer programming model to deal with energy consumption in heterogeneous wireless sensor networks. Communications in Computer and Information Science. 2008;14:11-20

[3] Aguiar AB, Pinheiro PR, Coelho ALV. On the concept of density control and its application to a hybrid optimization framework: An investigation into cutting problems. Computers & Industrial Engineering. 2011;61:463-472

[4] Aguiar AB, Pinheiro PR, Coelho ALV. Optimizing energy consumption in heterogeneous wireless sensor networks: A novel integer programming model. In: Proceedings of the IV International Conference on Operational Research for Development – ICORD; 2007. pp. 496-505

[5] Amaro Junior B, Pinheiro PR, Saraiva RD, Pinheiro PGCD. Handing the random-key genetic algorithm for solving the nesting problem. In: Anais do XLVI Simpósio Brasileiro de Pesquisa Operacional; 2014

[6] Blum C, Roli A. Metaheuristics in combinatorial optimization: Overview and conceptual comparison. ACM Computing Surveys. 2003;3:268-308

[7] Eiben AE, Smith JE. Introduction to Evolutionary Computing. Springer; Berlin, Heidelberg, 2003

[8] Quintão FP, Nakamura FG, Mateus GR. Evolutionary algorithms for the dynamic coverage problem applied to wireless sensor networks design. Proceedings of the IEEE Congress on Evolutionary Computation. 2005;2:1589-1596

[9] Talbi G. A taxonomy of hybrid metaheuristics. Journal of Heuristics. 2002;8:541-564.

[10] Harold Ishebabi, Philipp Mahr, Christophe Bobda, Martin Gebser, and Torsten Schaub, "Answer Set versus Integer Linear Programming for Automatic Synthesis of Multiprocessor Systems from Real-Time Parallel Programs," International Journal of Reconfigurable Computing, vol. 2009, Article ID 863630, 11 pages, 2009.

[11] Zhou H, Liang T, Xu C, Xie J. Multiobjective coverage control strategy for energy-efficient wireless sensor networks. International Journal of Distributed Sensor Networks. vol. 2012, Article ID 720734, 10 pages, 2012

[12] Gonçalves JF, Almeida J. A hybrid genetic algorithm for assembly line balancing. Journal of Heuristics. 2002;8:629-642

[13] Puchinger J, Raidl G. Combining metaheuristics and exact algorithms in combinatorial optimization: A survey and classification. In: Proceedings of the 1st International Work-Conference on the Interplay Between Natural and Artificial Computation, Lecture Notes in Computer Science. Vol. 3562; 2005. pp. 41-53

[14] Nguyen K, Nguyen T, Cheung S-C. On reducing communication energy using cross-sensor coding technique. International Journal of Distributed Sensor Networks. 2011;7:12, Article ID 837128.

[15] Wolsey LA. Integer Programming. John Wiley & Sons; USA, 1998

[16] Dumitrescu L, Stützle T. Combinations of local search and exact algorithms. In: Applications of Evolutionary Computing, Eds: Stefano Cagnoni, Colin G. Johnson, Juan J. Romero Cardalda, Elena Marchiori, David W. Corne, Jean-Arcady Meyer, Jens Gottlieb, Martin Middendorf, Agnès Guillot, Günther R. Raidl, Emma Hart, Lecture Notes in Computer Science. Vol. 2611; 2003. pp. 211-2247

[17] MOTE Battery Life Calculator [Internet]. May 2007. Available in: <http://www.xbow.com/Support/Sypport_pdf_files/PowerManagement.xls>

[18] Pinheiro PR, Coelho ALV, Sobreira Neto AM, Aguiar AB. Towards aid by generate and solve methodology: Application to the problem of coverage and connectivity in wireless sensor networks. International Journal of Distributed Sensor Networks. 2012;8:1-11.

[19] Pinheiro PR, Sobreira Neto AM, Aguiar AB. Handing optimization energy consumption in heterogeneous wireless sensor networks. International Journal of Distributed Sensor Networks. 2013;2013:1-9

IntechOpen

IntechOpen

IntechOpen