A HYBRID METHODOLOGY FOR COVERAGE AND CONECTIVITY IN WIRELESS SENSOR NETWORK DYNAMIC PLANNING

Alexei B. Aquiar
Universidade de Fortaleza - UNIFOR
Av. Washington Soares 1321, Fortaleza, CE, Brasil, 60811-905, Brasil
alexei@verde.com.br

Alvaro M. S. Neto[1], Ruddy P. P. Cunha[2], Plácido R. Pinheiro[3], Andre Coelho[4]
Universidade de Fortaleza UNIFOR
Av. Washington Soares 1321, Fortaleza, CE, Brasil, 60811-905, Brasil
({netosobreira[1], ruddypaz[2]}@edu.unifor.br), ({placido[3], acoelho[4]}@unifor.br)

ABSTRACT

High power consumption efficiency in wireless sensor networks is always desirable. However it is not achievable easily. One way to deal with this issue is using a mathematical model based upon a schedule of sensor allocation plans in multiple time intervals subject to coverage and connectivity constraints. The use of pure linear integer programming approach is limited to a certain level of complexity that sometimes is not enough for a real size network. The use of a new hybrid methodology that gathers the strengths of metaheuristics and exact methods has obtained good results in other problem domains. The adoption of this tool provided solutions to problem instances larger than found in previous works related to the issue discussed before, proving that it is adaptable to a broad range of problem classes. Its adaptation from its original form to the particularities found in this problem is discussed here as a continuous methodological evolution.

KEYWORDS. Wireless Sensor Network, Hybridization, Linear Integer Programming, Metaheuristics.

1 The wireless sensor network

A Wireless Sensor network typically consist of a large number of small, low-power, and limited-bandwidth computational devices, named sensor nodes. These nodes can frequently interact with each other, in a wireless manner, in order to relay the sensed data towards one or more processing machines (a.k.a. sinks) residing outside the network. For such a purpose, special devices, called gateways, are also employed, in order to interface the WSN with a wired, transport network. To avoid bottleneck and reliability problems, it is pertinent to make one or more of these gateways available in the same network setting, a strategy that can also reduce the length of the traffic routes across the network and consequently lower the overall current consumption. A typical sensor node is composed of four modules, namely the processing module, the battery, the transceiver module and the sensor module as described in Loureiro (2002). Besides the packet building processing, a dynamic routing algorithm runs over the sensor nodes, in order to discover and configure in runtime the “best”' network topology in terms of number of retransmissions and waste of current. Due to the limited resources available to the microprocessor, most devices make use of a small operating system that supplies basic features to the application program. To supply the power necessary to the whole unit, there is a battery, whose lifetime duration depends on several aspects, among which, its storage capacity and the levels of electrical current employed in the device. The transceiver module, conversely, is a device that transmits and receives data using radio-frequency propagation as media, and typically involves two circuits, viz. the transmitter and the receiver. Due to the use of public-frequency bands, other devices in the neighborhood can cause interference during sensor communication. Likewise, the operation/interaction among other sensor nodes of the same network can cause this sort of interference. So, the lower is the number of active sensors in the network, the more reliable tends to be the radio-frequency communication among these sensors. The last component, the sensor module, is responsible to gauge the phenomena of interest; the ability of concurrently collecting data pertaining to different phenomena is a property already available in some models of sensor nodes.

For each application scenario, the network designer has to consider the rate of variation for each sensed phenomenon in order to choose the best sampling rate of each sensor device. Such decision is very important to be pursued with precision as it surely has a great impact on the amount of data to be sensed and delivered, and, consequently, on the levels of current consumed prematurely by the sensor nodes. This is the temporal aspect to be considered in the network design.

Another aspect to be considered is the spatial one. Megerian et al. (2002) define coverage as a measure of the ability to detect objects within a sensor field. The lower the variation of the physical variable being measured across the area, the shorter has to be the radius of coverage for each sensor while measuring the phenomenon. This will have an influence in the number of active sensors to be employed to cover all demand points related to the given phenomenon. The fact is: the more sensors are active in a given moment, the bigger is the overall current consumed across the net. WSNs are usually deployed in hostile environments, with many restrictions of access. In such cases, the network would be very unreliable and unstable if the minimum number of sensor nodes was effectively used to cover the whole area of observation. If some sensor node fails to operate, its area of coverage would be out of monitoring, preventing the correlation of data coming from this area with others coming from other areas.

Another worst-case scenario occurs when we have sensor nodes as network bottlenecks, being responsible for routing all data coming from the sensor nodes in the neighborhood. In this case, a failure in such nodes could jeopardize the whole network deployment. To avoid these problems and make a robust design of the WSN, extra sensor nodes are usually employed in order to introduce some sort of redundancy. By this means, the routing topology needs to be dynamic and adaptive: When a sensor node that is routing data from other nodes fails, the routing algorithm discovers all its neighbor nodes and then the network reconfigures its own topology dynamically. One problem with this approach is that it entails unnecessary current consumption. This is because the coverage areas of the redundant sensor nodes overlap too much, giving birth to redundant data. And these redundant data bring about extra current consumption in retransmission nodes. The radio-frequency interference is also stronger, which can cause unnecessary retransmissions of data, increasing the levels of current expenditure. Megerian and Potkonjak (2003) present many integer linear programming models to minimize current consumption but not consider the dynamic time scheduling.

2 Model

The solution proposed by Nakamura et al. (2004) is to create different schedules, each one associated with a given time interval, that activate only the minimum set of sensor nodes necessary to satisfy the coverage and connectivity constraints. The employment of different schedules prevents the premature starvation from some of the nodes, bringing about a more homogeneous level of consumption of battery across the whole network. This is because the alternation of active nodes among the schedules is often an outcome of the model, as it optimizes the current consumption of the whole network taking into account all time intervals and coverage and connectivity constraints.

In order to properly model the WSN setting, some previous remarks are necessary:

1. A demand point is a geographical point in the region of monitoring where one or more phenomena are sensed. The distribution of such points across the area of monitoring can be regular, like a grid, but can also be random in nature. The density of such points varies according to the spatial variation of the phenomenon under observation. At least one sensor must be active in a given moment to sense each demand point. Such constraint is implemented in the model;

2. Usually, the sensors are associated with coverage areas that cannot be estimated with accuracy. To simplify the modeling, we assume plain areas without obstacles. Moreover, we assume a circular coverage area with a radius determined by the spatial variation of the sensed phenomenon. Within this area, it is assumed that all demand points can be sensed. The radio-frequency propagation in real WSNs is also irregular in nature. In the same way, we can assume a circular communication area. The radius of this circle is the maximum distance at which two sensor nodes can interact;

3. A route is a path from one sensor node to a sink possibly passing through one or more other sensor nodes by retransmission. Gateways are regarded as special sensor nodes whose role is only to interface with the sinks. Each phenomenon sensed in a node has its data associated with a route leading to a given sink, which is independent from the routes followed by the data related to other phenomena sensed in the same sensor node;

4. The electric charge consumption is actually the electric current drawn by a circuit in a given time period.

S	Set of sensors
D	Set of demand points
M	Set of sinks
$T=\{1..N\}$	Set of n scheduling periods
AD_{ij}	Set of arcs $ij, i \in S, j \in D$ that link sensors to demand points
A_{ij}	Set of arcs $ij, i \in S, j \in S \cup M$ that interconnects sensors
EB_i	Accumulated battery charge for sensor $i \in S$
EAi	Electrical charge dissipated while activating sensor $i \in S$
EM_i	Electrical charge dissipated while sensor $i \in S$ is activated (effectively sensing)
ET_{ij}	Electrical charge dissipated when transmitting data from sensor $i \in S$ to sensor $j \in S$. Such values can be different for each arc ij if a sensor can have its

	transmitter power adjusted based on the distance to the destination sensor
ER_l	Electrical charge expended in the reception of data for sensor $i \in S$
EH	Penalty applied when a demand point in any time interval is not covered by any sensor
x^t_{ij}	If sensor i covers demand point $j \in D$ in period $t \in T$
z^t_{lij}	If arc ij belongs to the route from sensor $i \in S$ to a sink in period $t \in T$
w^t_l	If sensor i was activated in period t for at least on phenomenon
y^t_i	If sensor i is activated in period t
h^t_j	If demand point j is not covered by any sensor in period t
e_i	Electrical charge consumed by sensor i considering all time periods

The objective function (1) minimizes the total electrical charge consumption through all time periods. The second term penalizes the existence some not-covered demand points, but the solution continues feasible. It penalizes unnecessary activation for phenomenon too.

$$\min \sum_{i \in S} e_i + \sum_{t \in T} \sum_{j \in D} EH h^t_j \tag{1}$$

These are the constraints adopted:

$$\sum_{i \in S} \sum_{j \in D} AD_{ij} x^t_{ij} + h^t_j \geq 1, \forall j \in D, \forall t \in T \tag{2}$$

Constraint (2) enforces the activation of at least one sensor node i to cover the demand point j in period t. Otherwise, the penalty variable h is set to one. This last condition will occur only in those cases when no sensor node can cover the demand point.

$$x^t_{ij} \leq y^t_i, \forall i \in S, \forall j \in S, \forall t \in T \tag{3}$$

Constraint (3) turns on variable y (which means that a sensor node is actively sensing in period t) if its associated sensor node is indeed allocated to cover any demand point.

$$\sum_{i \in (S-j)} A_{ij} z^t_{lij} - \sum_{k \in (SUM-j)} A_{jk} z^t_{ljk} = 0, \forall j \in S, \forall l \in S, \forall t \in T \tag{4}$$

Constraint (4) relates to the connectivity issue using the flow conservation principle. This constraint enforces that an outgoing route exists from sensor node j to sensor node k if there is already an incoming route from sensor node i to sensor node j.

$$\sum_{k \in (SUM-l)} A_{ij} z^t_{ljk} = y^t_l, j = l, \forall l \in S, \forall t \in T \tag{5}$$

Constraint (5) enforces that a route is created if a sensor node is already active.

$$\sum_{i \in S} \sum_{j \in M} A_{ij} z_{itj}^{t} = y_i^t, \forall t \in T, \forall i \in S \tag{6}$$

Constraint (6) is necessary to create a route that reaches a sink if a sensor is active.

$$A_{ij} z_{itj}^{t} \leq y_j^t, \forall j \in S, \forall i \in (S - j), \forall i \in (S - j), \forall t \in T \tag{7}$$

In Constraint (7), if there is an outgoing route passing through sensor node i, then this sensor node has to be necessarily active.

$$A_{ij} z_{itj}^{t} \leq y_i^t, \forall j \in S, \forall i \in (S - j), \forall i \in (S - j), \forall t \in T \tag{8}$$

In the same way, with Constraint (8) if there is an incoming route passing through sensor i, then this sensor has to be active.

$$\sum_{t \in T} BM_i y_i^t + BA_i w_i^t + \sum_{i \in (S-i)} \sum_{k \in S} BR_i z_{ikt}^{t} + \sum_{i \in S} \sum_{j \in (S \cup M)} BT_{ij} z_{itj}^{t} \leq e_i, \forall i \in S \tag{9}$$

The total electrical charge consumed by a sensor node is the sum of the parcels given in Constraint (9).

$$0 \leq e_i \leq BB_i, \forall i \in S \tag{10}$$

Constraint (10) enforces that each sensor node should consume at most the capacity limit of its battery.

$$w_i^0 - y_i^0 \geq 0, \forall i \in S \tag{11}$$

Constraint (11) determines when the sensor node should start to sense (parameter w). If a sensor is active in the first period, its corresponding w should be set to 1.

$$w_i^t - y_i^t + y_i^{t-1} \geq 0, \forall i \in S, \forall t \in T, t > 0 \tag{12}$$

In Constraint (12), the past and current activation states of a sensor node are compared. If the sensor node was active from period $t - 1$ to period t, then w is set to 1.

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

3 The base hybrid methodology

Although distinct, both the exact and metaheuristic approaches have pros and cons when dealing with hard combinatorial optimization problems. But their hybridization, when properly done, may allow the merging of their strong points in a complementary manner. For instance, it is well-known that the direct application of exact methods is only possible for limited-sized instances. However, the size and complexity of the optimization problems faced nowadays have increased a lot, demanding for the development of new methods and solutions that can find acceptable results within a reasonable amount of time.

In Aguiar et al. (2007, 2008) WSN problems were explored regarding the heterogeneity of the phenomena. This model however suffers a shortage of variables due to the increase of complexity as many matrices had gained one more dimension.

In this regard, it has become ever more evident that a skilled combination of concepts stemming from different metaheuristics can be a very promising strategy one should resort to when having to deal with complicated optimization tasks. The hybridization of metaheuristics with other operations research techniques has been shown great appeal as well, as they typically

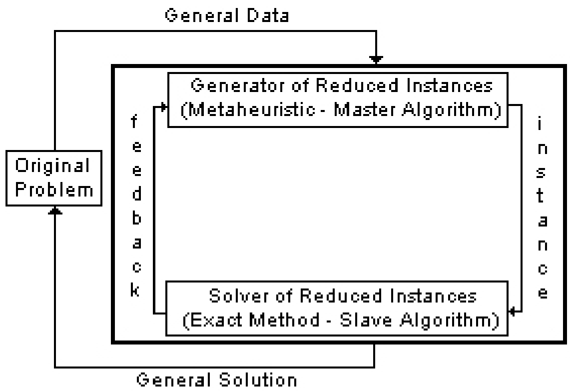

Figure 1: The hybrid framework under investigation

represent complementary perspectives over the problem solving process as a whole. In general, combinations of components coming from different metaheuristics and/or from more conventional exact methods into a unique optimization framework have been referred to by the label of "hybrid metaheuristics" by Blum e Roli (2003), Dumitrescu e Stützle (2003), Talbi (2002), Raidl (2006).

In this context, a hybrid methodology has been recently introduced in the literature by Nepomuceno et al. (2006, 2007a, 2007b, 2008), trying to push forward the boundaries that limit the application of an exact method through the decomposition of the original problem into two conceptual levels. According to the framework underlying this approximative methodology (see Figure1) the exact method (encapsulated in the Solver of Reduced Instances (SRI) component) works no more with the original problem but with reduced instances (i.e. subproblems) of it that still preserve its conceptual structure. By this means, an optimal solution to a given subproblem will also be a feasible solution to the original problem. On the other hand, the metaheuristic component of the framework works on a complementary optimization problem, that is, the design of reduced instances of the original problem formulated as mathematical programming (viz., integer linear programming (ILP) models. It is referred to as the Generator of Reduced Instances (GRI), whose goal is to determine the subset of points of the reducible structure that could derive the best subproblem instance; that is, the subproblem which, when submitted to the SRI, would bring about the feasible solution with the highest possible objective function value. In this scenario, the objective function values of the solutions that could be realized by the solver are used as figure of merit (fitness) of their associated subproblems, thus guiding the metaheuristic search process. The interaction between GRI and SRI is iterative and repeats until a given stopping condition is satisfied.

So far, the metaheuristic chosen to implement the generator of reduced instances has been a Genetic Algorithm as explained by Eiben and Smith (2003). This option is due mainly to the good levels of flexibility and adaptability exhibited by the class of evolutionary algorithms when dealing with a wide range of optimization problems as presented by Back et at. (1997). The genetic representation of the individuals (chromosomes) follows a binary encoding that indicates which decision variables belonging to the reducible structure will be kept in the new subproblem to be generated. That is, those genes having '1' as alleles define the subset of variables that generates the reduced instance. Conversely, the exact method is assumed to be any state-of-theart algorithm used to solve mixed integer-linear problems, such as *Branch-and-bound* or *Branch-and-cut* described in Wolsey (1998). Usually, the solver libraries available incorporate sets of strategies, heuristics, and problem reduction techniques that complement the main exact method and enhance its performance.

According to the classification proposed in Puchinger e Raidl (2005), the methodology falls into the category of integrative combinations. The quality of the solutions to the instances generated by the metaheuristic is determined when the sub-problems are solved by the exact method, and the best solution obtained throughout the whole metaheuristic process is deemed to be the final solution to the original problem.

Although showing remarkable levels of performance for some case problems studied in the realm of cutting & packing problems in Nepomuceno (2006, 2007a, 2007b, 2008), the original version of the aforementioned hybrid methodology has drawbacks, some of which are circumvented with the adoption of the mechanisms discussed here. Other impacting factor that must be noticed is that the original version addressed only the cutting and packing problem class. One consequence of this particularity is that it requires some changes in order to be adapted to new optimization problem classes, described as follows in section 4.

4 Improvements for the dynamic coverage and connectivity in wireless sensor network problem

Adapting the base hybrid methodology to be suitable for a totally different class of problem is a challenge. Even the direction of optimization is opposite and requires changing since genetic algorithms natively maximizes, while this problem is a minimization one. Although this issue is easy to solve it shows how distinct problem classes can be even right in the beginning.

A drawback that has limited the effectiveness of the base hybrid methodology as presented in Section 3 relates to its propensity for bringing about an uncontrolled density explosion over the individuals (i.e. reduced instances of the original problem) produced by the GRI. We define "density of an individual" as the ratio between the number of genes having '1' as allele (referred to as activated) and its total length. The fact is that an increase in density tends to generate subproblems more closer to the original problem, thus possibly yielding better solutions. This situation can be better pictured as if having some sort of an "attractor" pushing the overall population density up as the GRI (GA) evolves. Although expected, this phenomenon may have an undesirable side-effect if it occurs prematurely. This is because, usually, high densities imply higher complexity to be dealt with by the SRI, which indirectly affects the search process conducted by the GRI as the time spent in each generation tends to become progressively higher. This may cause a drastic limitation over the number of search iterations performed by the SRI, hindering both the effectiveness and efficiency of the whole optimization.

Other undesirable characteristic of the original version of this methodology is that its binary chromosome encoding can be prohibitively long, depending on the chosen reducible matrix. Long chromosomes can lead to problems.

4.1 Compact chromosome encoding

According to Eiben and Smith (2003), the right representation of the individuals is one of the most difficult parts of designing a good evolutionary algorithm.

The binary chromosome encoding was used in the original version of the hybrid methodology. Each gene represents the inclusion of the equivalent element of the reducible structure that will be considered in the generation of the new subproblem. It is well suited for the cutting and packing problem class for which the methodology was designed. This type of chromosome encoding however is not appropriated for other problem domains like the one treated in this work. It would generate too large chromosomes (i.e. 10 time intervals × 36 sensors). Table 1 shows a possible chromosome with the binary encoding. Each color represents a set of 16 genes associated to its respective sensors of the each time interval.

The proposed new encoding (table 2) represents the integer indexes of the sensors that must be taken in the subproblem generation. So there is no need of representing all sensors. Only a small amount of sensors has to be considered and the length of this chromosome can be down to 17% of the binary encoding one.

1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	…
0	0	1	0	1	0	0	0	0	0	1	0	0	0	0	1	0	1	0	0	…

Table 1: Part of a chromosome with binary encoding

3	5	11	16	2	5	10	13	4	8	9	15	1	7	…

Table 2: Part of a chromosome with the new compact encoding

In the original version of the hybrid methodology, the density rise was a problem as described in section 4. The first resource created to avoid this undesirable effect was the density control operator that effectively accomplished its goal and is expected to be published soon. Here the issue is solved with a much more controlled expedient: Constant density.

This new compact chromosome encoding has a side effect of turning the density constant, since the ratio of sensors considered in the subproblem and the total number of sensors is always fixed. Now the solver can work in its best range of operation, balancing efficiency and effectiveness.

5 Computational results

Experiments were made for the dynamic coverage and connectivity in wireless sensor networks problem using the mentioned hybrid methodology.

It follows most premises of section 2. The grid sensor placement was used for simplicity sake because the random scenario did not present significant variation of the problem complexity which is the main concern of these experiments. The machine used on this test was an Intel Core 2 Quad 64 bits with 8 GB of RAM machine with OpenSuse Linux 11.0 64 bits. As Linear Integer Programming (LIP) solver, the Ilog Cplex 10.1 dynamic library (ILOG 2006) was used attached to the Java program implementation of the methodology. The pure LIP approach is a particular case of the methodology and is obtained by a proper parameter set in a XML script.

Table 3 presents the comparison of Hybrid Methodology (HM) and LIP approaches. On these experiments, the demand points are disposed in a grid. Due to the stochastically nature of the HM, it is presented the average and standard deviation of results found in a batch of 10 problem instances. The notation used here is the average value followed by the ± sign and the standard deviation value.

The objective function is composed of two parts: The summation of electrical charge consumption in all sensor nodes and the penalties. The penalties are an artifact that allow uncovered demand points giving flexibility to the model, but at the same time, avoid the unnecessary use of this resource. Thus, the real objective is calculated by subtracting the artificial coverage penalties of the objective function or just calculating the first part (summation) of the objective expression.

Table 4 shows the results of similar experiments. However, this time the demand points are spread randomly through the sensed area.

	HM	**HM**	**LIP**
Time intervals	6	10	10
Demand points	400	400	400
Sensor nodes	36	36	16
Sinks	1	1	1
Time (minutes)	151.78 ± 14.61	189.80 ± 61.01	298.55
Time for first final solution	79.84 ± 54.65	104.14 ± 75.06	298.55
Uncovered demand points (%)	0.93 ± 0.39	2.35 ± 0.54	0.33
Real Objective	22,223.27 ± 2,614.63	33,374.04 ± 2,389.76	26,665.91

Table 3: Simulation results for demand point in grid.

	HM	**HM**
Time intervals	6	10
Demand points	400	400
Sensor nodes	36	36
Sinks	1	1
Time (minutes)	146.77 ± 32.45	366.28 ± 13.54
Time for first final solution	91.55 ± 47.14	196.05 ± 78.30
Uncovered demand points (%)	1.56 ± 0.61	2.10 ± 0.53
Real Objective	20,472.51 ± 4,135.35	32,522.51 ± 2,628.33

Table 4: Simulation results for demand point in aleatory positions.

The real purpose of this model is to extend the WSN lifetime as far as possible, preserving the WSN cost. So, lower electrical charge consumption is not necessarily an important issue if it does not reflect in more time slots. The number of time slots multiplied by the duration of each time slot represents this WSN lifetime.

Given this explanation is reasonable to say that both solutions found by HM and LIP are equivalent in effectiveness. However, the HM approach can handle an amount of sensors 325% times larger, extending the working range of this application.

The only drawback here is the uncovered demand point rate which is worse than LIP value. Despite this small imperfection of 2.35% many real applications tolerates some lack of coverage by the nature of the observed phenomenon and other aspects. Even though this uncovered demand points are often situated at the periphery of the observed area. The coverage radius does not reflect necessarily a sharp threshold of sensing.

Figure 1 shows the evolution of the best individual fitness in plain line and population fitness average in line with points as well.

Figures 2 to 11 are graphical representations of 10 time slots of an solution example. It shows the active sensor nodes, its coverage radios, the covered and uncovered demand points and the routes from sensor nodes to the sink in the center.

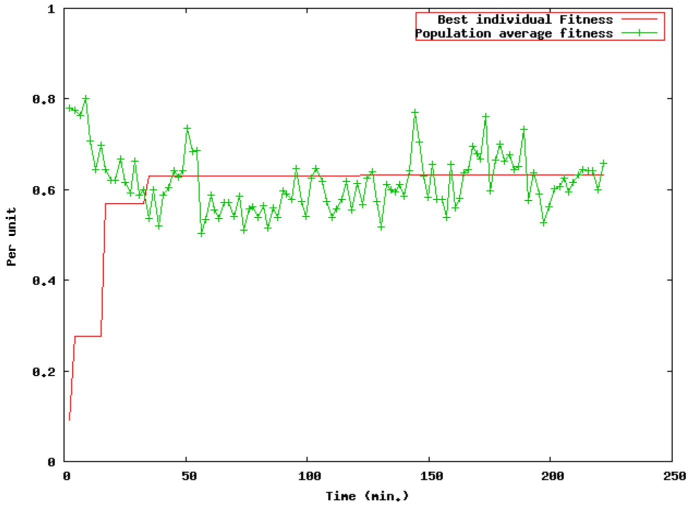

Figure 1: Best individual and population average evolution.

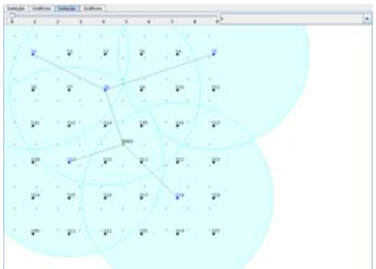
Figure 2: Solution in time slot 1

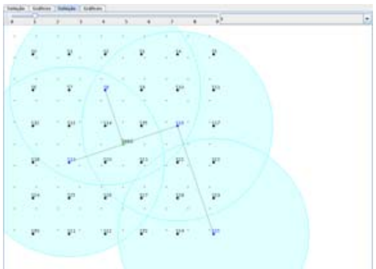
Figure 3: Solution in time slot 2

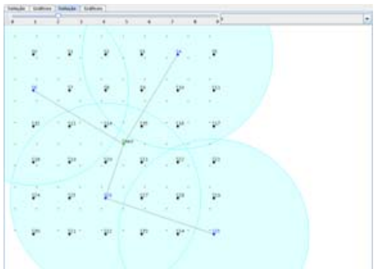
Figure 4: Solution in time slot 3

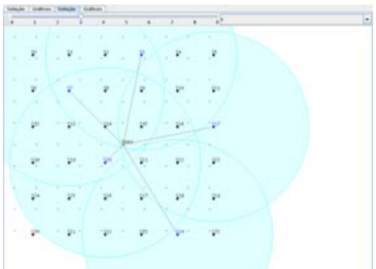
Figure 5: Solution in time slot 4

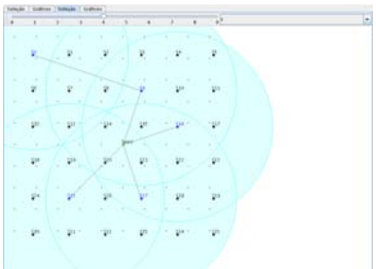
Figure 6: Solution in time slot 5

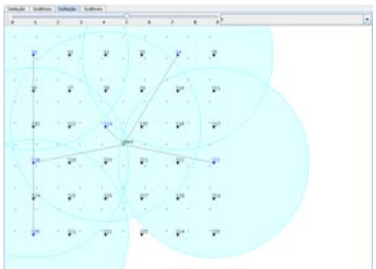
Figure 7: Solution in time slot 6

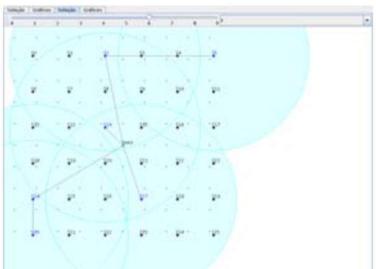
Figure 8: Solution in time slot 7

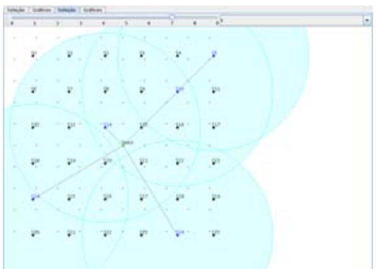
Figure 9: Solution in time slot 8

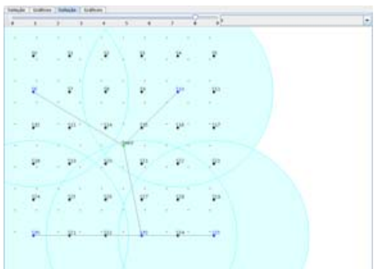
Figure 10: Solution in time slot 9

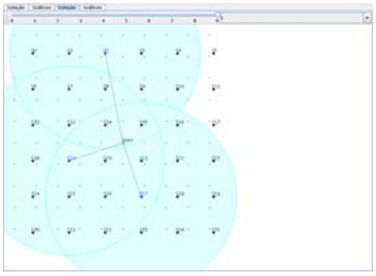

Figure 11: Solution in time slot 10

6 Conclusion

This hybrid methodology is not only suitable for solving complex instances in the domain of cutting and packing problems. It can be adapted to tackle other problem classes like WSN as shown here.

The key point in this adaptation is finding the best or at least a good reducible structure. This analysis is very linked to the chromosome encoding choice as it represents a trade of between subproblem complexity range width and chromosome size. A good reducible structure allows a wide range of subproblem complexity from very light and fast subproblems to the actual real problem. On the other hand the reducible matrix size affects the chromosome size and a large chromosome size reduces the GA effectiveness.

In this problem a good reducible structure was found but it is much larger than the ones found in the cutting and packing problem instance. That is the reason why a new chromosome encoding was developed. This new encoding makes the matrix choice viable.

The result found are far better than reference literature and leaves opportunities of future enhancements as new supplementary algorithms and heuristics are aggregated to this methodology.

Acknowledgments. The first author gratefully acknowledges the financial support received from CAPES, through a M.Sc. scholarship.

References

Aguiar, A. B. de; Pinheiro, P. R.; Coelho, A. L. V. (2007), Optimizing energy consumption in heterogeneous wireless sensor networks: A novel integer programming model. Proceedings of the IV International Conference on Operational Research for Development – ICORD. pp. 496-505.

AGUIAR, Alexei Barbosa de ; PINHEIRO, P. R. ; COELHO, A. L. V. ; Alvaro Neto; Ruddy P. P. Cunha.(2008) Scalability Analisys of a Novel Integer Programming Model to Deal with Energy Consumption in Heterogeneous Wireless Sensor Networks. Communications in Computer and Information Science, v. 14, p. 11-20.

Back, T., Fogel, D. B., and Michalewicz, Z., Handbook of Evolutionary Computation. IOP Publishing Ltd., Bristol, UK, UK, 1997.

Blum, C., and Roli, A., (2003) Metaheuristics in combinatorial optimization: Overview and conceptual comparison. ACM Comput. Surv. 35, 3, 268-308.

Dumitrescu, L., and Stützle, T. (2003) Combinations of Local Search and Exact Algorithms, vol. 2611. Springer Verlag, Berlin, Applications of Evolutionary Computing, LNCS, pp. 211-224.

Eiben, A. E., and Smith, J. E., (2003) Introduction to Evolutionary Computing. Springer-Verlag.

ILOG, ILOG CPLEX 10.0 User's Manual, 2006.

Loureiro, A., Ruiz, L., Mini, R., and Nogueira, J. (2002)**,** Redes de sensores sem fio. Simpósio Brasileiro de Computação Jornada de Atualização de Informática.

Megerian, S., Koushanfar, F., Qu, G., Veltri, G., and Potkon-jak, M., (2002) Exposure in wireless sensor networks: Theory and practical solutions. Wireless Networks 8, 443-454.

Megerian, S., and Potkonjak, M., Lower power 0/1 coverage and scheduling techniques in sensor networks. University of California, Los Angeles, Technical Reports 030001, January, 2003.

Nakamura, F. G., Quintão, F. P., Menezes, G. C., and Mateus, G. R. (2004)**,** Planejamento dinâmico para controle de cobertura e conectividade em redes de sensores sem fio. Workshop de Comunicação sem Fio e Computação Móvel 1, 182-191.

Nepomuceno N., Pinheiro P. R., and Coelho A. L. V.(2006) Aplicação de uma Metodologia Híbrida ao Problema de Carregamento de Contêiner. In: XXXVIII Simpósio Brasileiro de Pesquisa Operacional, 2006, Goiânia. Anais do XXXVIII Simpósio Brasileiro de Pesquisa Operacional. Rio de Janeiro, 2006. p. 1596-1603

Nepomuceno, N. V., Pinheiro, P. R., and Coelho, A. L. V. (2007a)**,** Tackling the container loading problem: A hybrid approach based on integer linear programming and genetic algorithms. In EvoCOP 2007 - Valência, Spain. LNCS, vol. 4446, Springer-Verlag, pp. 154-165.

Nepomuceno N., Pinheiro P. R., and Coelho A. L. V. (2007b), Combining Metaheuristics and Integer Linear Programming: A Hybrid Methodology Applied to the Container Loading Problem. In: XXVII Congresso da Sociedade Brasileira de Computação, 2007, Rio de Janeiro. Anais do XXVII Congresso da Sociedade Brasileira de Computação, p. 2028-2032.

Nepomuceno, N. V., Pinheiro, P. R., and Coelho, A. L. V. (2008)**,** A Hybrid Optimization Framework for Cutting and Packing Problems: Case Study on Constrained 2D Non-guillotine Cutting. In C. Cotta and J. Van Hemert (Org.). Recent Advances in Evolutionary Computation for Combinatorial Optimization. Berlin / Heidelberg: Springer- Verlag, v. 153, p. 87-99.

Puchinger, J., and Raidl, G. (2005)**,** Combining Metaheuristics and Exact Algorithms in Combinatorial Optimization: A Survey and Classification, vol. 3562. Springer Verlag, ch. Proceedings of the 1st International Work Conference on the Interplay Between Natural and Artificial Computation, LNCS, pp. 41-53.

Raidl, G. R. (2006)**,** A Unified View on Hybrid Metaheuristics, vol. 4030. Springer-Verlag, ch. Proceedings of the 3rd International Hybrid Metaheuristics Workshop, LNCS, pp. 1-12.

Talbi, E. G., A Taxonomy of Hybrid Metaheuristics, Journal of Heuristics, pp. 541-564, 2002.

Wolsey, L. A., Integer Programming. John Wiley & Sons, 1998.