Applying the Generate and Solve Methodology in the Problem of Dynamic Coverage and Connectivity in Wireless Sensor Networks

Placido R. Pinheiro[1], Andre L. V. Coelho[1], Alexei B. Aguiar[1], Alvaro M. S. Neto[1]

[1] University of Fortaleza - UNIFOR, Applied informatics pos-graduation program - PPGIA, Av. Washington Soares, 1321 60811-905 Fortaleza, Brazil
{netosobreira, alexei}@verde.com.br, {placido, acoelho}@unifor.br

Abstract. High power consumption efficiency in wireless sensor networks is always desirable. One way to deal with this issue is using a linear integer programming model based upon a schedule of sensor allocation plans in multiple time intervals subject to coverage and connectivity constraints. The Generate-and-Solve (GS) methodology is a hybrid approach that combines a metaheuristic component with an exact solver. GS has been recently introduced in the literature in order to solve Problem of Dynamic Coverage and Connectivity in Wireless Sensor Networks, showing promising results. The GS framework includes a metaheuristic engine (e.g., a genetic algorithm) that works as a generator of reduced instances of the original optimization problem, which are, in turn, formulated as mathematical programming problems and solved by an integer programming solver. The use of linear integer programming approach is limited to a certain level of complexity that sometimes is not enough for a real size network.

Keywords: Wireless Sensor Network, Combinatorial optimization, Hybrid Metaheuristics, Linear Integer Programming, Genetic algorithms.

1 Introduction

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 [1]. Besides the packet building processing, a dynamic routing algorithm runs through the sensor nodes network, 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. 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.

2 The Model

The solution proposed by [2] 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. This approach prevents the premature starvation from some of the nodes, bringing about a more homogeneous level of battery's consumption across the network.

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.
2. To simplify the modeling, we assume plain areas without obstacles. Moreover, we assume a circular coverage and transmission radius.
3. A route is a path from a sensor node to a sink. It might pass through one or more other sensor nodes. Gateway is a special sensor node to interface with the sinks.

The Model is described below:

= Set of Sensors

= Set of demand points

= Set of Sinks

= 1.. = Set of n scheduling periods

11, Set of arcs , ∈ , ∈ that link sensors to demand points

11 Set of arcs , ∈ , ∈ ∪ that interconnects sensors

1 Accumulated battery charge for sensor ∈

1 Electrical charge used while activating sensor ∈ 1

Electrical charge used while sensor ∈ is activated

11 Electrical charge used to transmit data from sensor ∈ to ∈ .

1 Electrical charge expended in the reception of data for sensor ∈

Penalty applied when a demand point in any time interval is not covered.

!"# If sensor covers demand point ∈ in period ∈

!"#$ If arc belongs to the route from sensor ∈ to a sink in period ∈

!1 If sensor was activated in period for at least one phenomenon

ti If sensor is activated in period

h!" If demand point is not covered by any sensor in period

The electrical charge consumed by sensor considering all the periods

The objective function (1) minimizes the total electrical charge consumption through all time periods considering the penalties for uncovered demand points.

These are the constraints adopted: I∈S 1 + t∈T f∈D !1 (1)

Σ I∈S f∈D 1"!1 + h!1 ≥ 1, ∀ ∈ , ∀ ∈ (2)

!"# ≤ t", ∀ ∈ , ∀ ∈ , ∀ ∈ (3)

Σ I∈(S—j) 1"!"1 − k∈(S∪M—j) jktlkj = 0, ∀ ∈ , ∀ ∈ , ∀ ∈ (4)

Σ Σ k∈(S∪M_l)Ijtljk = ti, = , ∀ ∈ , ∀ ∈ (5)

( ) ( )

I∈S J∈M 1"!"1 = ti , ∀ Σ ∈ , ∀ Σ ∈ (6) Σ Σ ( )

1"!"1 ≤ t", ∀ ∈ , ∀ ∈ − , ∀ ∈ − , ∀ ∈ (7)

1"!"1 ≤ t", ∀ ∈ , ∀ ∈ − , ∀ ∈ − , ∀ ∈ (8)

t∈Titi + !!" + I∈(S—i) IC∈SitIICi + I∈S j∈ S∪M !1!1#1 < !, ∀ ∈ (9)

0 ≤ 1 ≤ 1, ∀ ∈ (10)

01 − oi ≥ 0, ∀ ∈ (11)

ti − ti + t_i ≥ 0, ∀ ∈ , ∀ ∈ , > 0 (12)

, h ∈ 0, 1 , , ∈ (13)

The constraints (2), (3), (11) and (12) are about activating or not the sensor nodes. The constraints (4), (5), (6), (7) and (8) are responsible for routing. The constraint (10) is responsible for sensor's battery's electrical charge usage limits.

3 The base hybrid methodology

Both the exact and meta-heuristic 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. 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 this context, a hybrid methodology has been recently introduced in the literature by [3], 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 approximate methodology, the exact method works no more with the original problem, but with reduced instances of it. That preserves its conceptual structure. By this means, an optimal solution to a given sub-problem will also be a feasible solution to the original problem. On the other hand, the meta-heuristic 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 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 sub-problem instance; that is, the sub-problem 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 sub-problems, thus guiding the meta-heuristic search process. The interaction between GRI and SRI is iterative and repeats until a given stopping condition is satisfied.

So far, the meta-heuristic chosen to implement the reduced instances generator has been a Genetic Algorithm explained by [4]. This option is due 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 [5]. 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 sub-problem 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-the-art algorithm used to solve mixed integer-linear problems, such as *Branch-and-bound* or *Branch-and-cut* described in [6].

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. 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 numbers 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 sub-problems 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.

4.1 Compact chromosome encoding

According to [4], 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 sub-problem. This type of chromosome encoding is not appropriated for other problem domains like the one treated in this work. It would generate too large chromosomes.

The proposed new encoding represents the integer indexes of the sensors that must be taken in the sub-problem 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.

5 Computational Results

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 IlogCplex 10.1 dynamic library [7] was used attached to a Java program that implements the methodology.

Table 1 presents the comparison of Hybrid Methodology (HM) and LIP approaches. On these experiments, the demand points are disposed in a grid and the last two columns present the results when demand points are spread randomly. Due to the stochastically nature of the HM, it is presented the average and standard deviation, separated by the symbol ±, of results found in a batch of 10 problem instances.

The objective function is composed by the summation of electrical charge consumption in all sensor nodes and the penalties. The penalties 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. The amount of demand points and sinks were 400 and 1 respectively.

	HM	HM	LIP	Random HM	Random HM
Time intervals	6	10	10	6	10
Sensor nodes	36	36	16	36	36
Time (minutes)	*151.78* ± 14.61	189.80 ± 61.01	298.55	146.77 ± 32.45	366.28 ± 13.54
Time for first solution	*79.84* ± 54.65	104.14 ± 75.06	298.55	91.55 ± 47.14	196.05 ± 78.30
Uncovered [illegible]	[illegible]	[illegible]	[illegible]	[illegible]	[illegible]
Real objective	*22,223.27* ± 2,614,63	33.374,04 ± 2,389.76	26,665.91	20,452.51 ± 4,135,35	32.522,51 ± 2,628.33

Table 1: Simulation results for demand point in grid and randomly 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 is the uncovered demand point rate, which is worse than LIP value. Despite this imperfection of 2.35%, many real applications tolerate some lack of coverage by the nature of the observed phenomenon and other aspects.

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 sub-problem complexity range width and chromosome size. A good reducible structure allows a wide range of sub-problem complexity from very light and fast sub-problems 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 making the matrix choice viable.

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

References

1. Loureiro, A., Ruiz, L., Mini, R., and Nogueira, J. (2002), Redes de sensores sem fio. Simpósio Brasileiro de Computação Jornada de Atualização de Informática. Technical Reports 030001, January, 2003.
2. Nakamura, F. G.,Quintão, F. P.,Menezes,G.C.,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, 182-191.
3. 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
4. Eiben, A. E., and Smith, J. E., (2003) Introduction to Evolutionary Computing. Springer-Verlag.
5. Back, T., Fogel, D. B., and Michalewicz, Z., Handbook of Evolutionary Computation. IOP Publishing Ltd., Bristol, UK, UK, 1997.
6. Wolsey, L. A., Integer Programming. John Wiley& Sons, 1998.
7. ILOG, ILOG CPLEX 10.0 User's Manual, 2006.