WIRELESS SENSOR DEPLOYMENT DECISION SUPPORT USING GENETICS ALGORITHM AND DEVS FORMALISM

Bastien POGGI [(a)], Jean-François SANTUCCI [(b)], Thierry ANTOINE-SANTONI [(c)]

[(a ,b, c)] University of Corsica Pasquale Paoli
UMR CNRS 6134 Sciences for the environment
Technology information and communication project

[(a)] bpoggi@univ-corse.fr, [(b)]santucci@univ-corse.fr, [(c)]antoine-santoni@univ-corse.fr

ABSTRACT

The deployement of a Wireless Sensors Network appears as a strategical aspect in the environmental monitoring. A random deployement is efficient if the entities are mobile and able to take account of the positon of each others. If the WSN is static the deployment must be adapted at the monitoring area and must respect the intrinsic parameters of the WSN. To predict the best positionment of the WSN entities the simulation appears as an efficient way. This paper describes a framework based on the DEVS (Discrete Event System) formalism and GA (Genetic algorithms). This decision support tool for the deployment focuses on the improvement of two main characteristics of the network: the sensor coverage, and the connectivity between nodes.

Keywords: WSN, deployment, DEVS, Optimization, Simulation, decision support system

1. INTRODUCTION

Advances in Micro-Electro-Mechanical-Systems (MEMS) based sensor devices and miniaturization of processor and radio as sensor package have led to the emergence of Wireless Sensor Networks WSN) since the last decade. Nowadays the WSN appear like an efficent way for the environemental monitoring and the increasing of the different hardware platforms have contributed at several testebed in many research areas (Beutel and Römer 2009). It seems to be interesting to view the succes experiences in real world with WSN: animal habitat monitoring (Szewczyk and Mainwaring 2004), agriculture (Langendoen and Baggio 2006), volcano activities (Werner-Allen and Lorincz 2006), oceanography (Tateson and Roadknight 2005), health (Paek and Chintalapudi 2005), and wildifre (Antoine-Santoni, Santucci 2009). However, we can observe that these different papers don't integrate a realistic approach of the network deployement.

Indeed the deployment of Wireless Sensor Network can become one of the most important strategic aspects in the Quality of Service (QoS). The WSN can be identified by several characteristics: mobility or not of the nodes, routing or mac protocol, life of the node , connectivity, coverage, time of arrival of message, real time application. These characteristics are the intrinsic parameters of a WSN and it is important to determine their possible impacts on a deployment and inversely. In a static WSN, different of a mobile network, the entities of the network have the possibility to know the position of the neighboring nodes but they don't have the possibility to move to enhance the link connectivity with the neighborhood.

It seems to be evident to say that the deployment of WSN is dependant of the area of deployement (AoD): an efficient WSN must to be different if the goal is to monitor an animal habitat monitoring or a wildfire monitoring The simulation appears like the best way to test and develop several strategies of WSN deployment to enhance the connectivity, the coverage and reduce the cost of a deployment with a diminution of the density of the netieis in the area. We can find different works on this research area (Younis and Akkaya 2008; Dhillon and Chakrabarty 2003; Efrat and Har-Peled 2005; Cheng Du 2005; Grandham and Dawande 2003; Antoine-Santoni and Santucci 2008) with a pertinent analysis by Otero and Kostanic (2009)

However all these works focus on a particular problem or specific parameters and they don't propose a generic approach for the analysis of WSN deployment based on independent reflexion of domain and sometimes this limits the capacity to resolve some complex problems. Moreover, these approaches don't define the way of development of decision support tool for the WSN deployement.

In this paper we intoruduce a generic open framework to simulate the WSN sensor deployement and optimize it by using Articificial Intelligence: specifically the evolutionary computation algorithms. The paper is organized as follows: in section 2 we present the bases of the framework, e.g. the DEVS formalism and the GAs approach. Section 3 deals with the description of our approach with the different models.

The results obtained when using the integration of genetic algorithms as optimization technique into a DEVSimPy modeling scheme is carefully described in section 4. Finally conclusions and perspectives are given in the last section.

2. DEVS FORMALISM AND GA

When people want to improve the performance of a studied system the modeling and simulation phases can be integrated with an optimization technique. This is what we call optimization via simulation (OvS). Optimization via Simulation is a structured approach to determine optimal settings for input parameters (i.e., the best system design), where optimality is measured by a function of output variables associated with a simulation model (Swisher and Hyden 2004; Garcia and Patek 2007). OvS problems can often be formulated as finding the minimum (or maximum) of a function (called in the OvS terminology objective function) having the vector x as parameters. Such a vector represents the set of decision variables according to the OvS terminology. To evaluate the objective function, we can only run simulation experiments at a particular value of x. Our framework is defined using an automatic integration of genetic algorithms optimization techniques into a discrete event modeling using the DEVS (Discrete EVent system Specification)

2.1. DEVS formlism

The DEVS formalism DEVS (Discrete Event System Specification) introduced by Professor Zeigler (1976) based on the definition of two types of components: atomic components which are the basic elements of the model behavioral and coupling components which correspond to a grouping of elements atomic behavior (description hierarchy).

The role of atomic components is to provide a local description of the dynamic behavior of the system studied. This type of component (atomic model) has state variables and ports of entry / exit, through which all interactions with the external environment occur. There are two types of external interaction: interaction of the component input (external events) that can be managed by external transition functions (named delta_ext) and output interactions managed by output functions (called lambda). The evolution of state variables of the component is dictated by the external transition function when interacting with the outside or by an internal transition function (named delta_int) when the component needs to evolve independently of its environment outside. Each state variable has a lifetime managed in a function of progress of time (called ta). The behavior of an atomic component is then obtained by the following algorithm:

1. The system is in a state s (initial state)
2. If no external event occurs, the component does not change state during a period determined by e = ta (s)
3. After this period, the component generates an output through lambda (s) and executes its internal transition function delta_int
4. If an external event occurs, the component changes state by executing delta_ext based on input values, the current state and the lifetime of this state.

An atomic component can not by itself account for the entire behavior of a complex system. Also, in order to have a phased approach to behavioral description, it is necessary to define another behavioral element: the coupling component. A coupling component (coupled model) to describe how many are interconnected elements (called sub-components) to form a new component. The specificity of coupling components is that they can be considered as basic elements in a coupling component of the highest level (Zeigler 1987). From the DEVS formalism briefly described above, an object-oriented simulation methodology has been implemented. The main idea is to automatically generate, from a given model, the corresponding simulation algorithms, allowing exploitation of the model. For this, it is necessary to associate each model a specific control structure. This control structure, called the simulator, a role for the management dynamics of the model on which it depends. The model simulation aims to generate output events, from input events from a given system. The simulation phase is then decomposed into two steps: the creation of the simulator and then use it in order to generate output events (outcomes). A simulator is represented by a graph type structure, called shaft simulation. This structure is described by a set of classes corresponding to different types of nodes constituting the tree simulation. For each model studied, the tree associated simulation is obtained by instantiation. This tree simulation has over operation of the model which it is attached. It is said that a simulator pilot the model associated with it. Each model is managed by a specific simulator, since any component of a model has an image and a single tree in the corresponding simulation. Communication between a model and its simulator is established through an exchange of messages. These messages are the events processed during the simulation process.

2.2. DEVSimPy

DEVSimPy software is a GUI for Python M & S (Capocchi and Santucci 2011) Automatic discrete event models described in the DEVS formalism. It was originally built to allow manipulation of graphical models PyDEVS to facilitate coupling. Indeed, the models are PyDEVS Python files using an API developed by researchers Bolduc and Vangheluwe (2001) from McGill University (Montreal) that creates DEVS models. These files contain a description of models and specification of couplings (links) between them. The only drawback, which does not depend PyDEVS, is the large number of errors caused by poor coupling due to a lack of attention during the connection ports between models. More models are, the more mistakes increase and pollute the thinking of developer around the model's behavior. DEVSimPy therefore avoids these errors by using a graphical

interface for creating and manipulating visual couplings.
The software DEVSimPy then evolved into a collaboration tool construction, simulation, storage and sharing libraries DEVS models from a design of the API PyDEVS.
DEVSimPy uses the graphics library and the wxPython API PyDEVS modeling and simulation. Of course, the core language which is the cornerstone tools DEVSimPy is giving birth to the Python language.

2.3. Library Genetic Algorithm

As explained in the introduction, the WSN deployment is a complex optimization problem. This one is described by several interrelated decision variables. Moreover WSN deployment depends on two aspects: the AoD characterisitics and the goals of the deployement. Artificial Intelligence is an efficient way to resolve some complex problems and we choose to integrate Genetic Algorithms in order to define an original approach.
Genetic Algorithms (GAs) are a bio-inspyred approach (Goldberg 1989; Holland 1992; Koza 1992; Mitchell 1996) based on an observation of adaptation in natural systems as it was described by Darwin (1836) on the origin of species. In GAs each population's individual represents a possible solution to the problem. The genetic code of an individual is represented by a list of bits. Over an iteration process as show in Figure 1, the population evolves toward improving solutions.

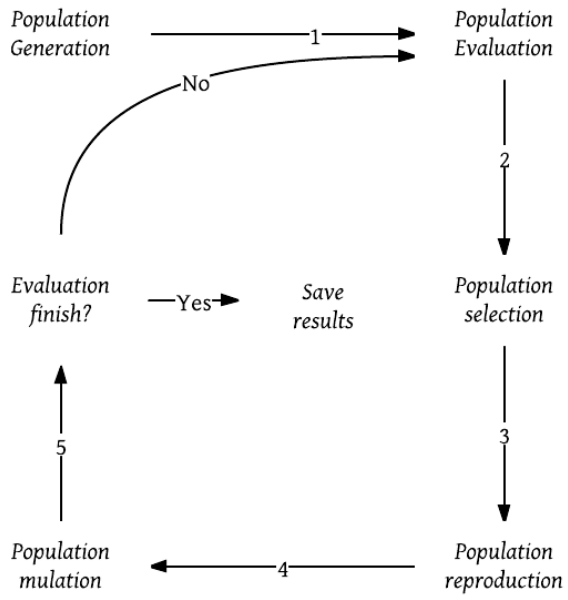

Figure 1 Genetic algorithms basic concept

The goal in the using of GAs is to provide a simulation and an optimization of WSN deployment in the same time. To reach this objective we choose to implement the different GAs functions as DEVS atomic models to allow building an interface between evaluation function and existing DEVS models as show in Figure 2.

When simulation begins the "population generator" model generates random solutions. Each created individual contains a list of node coordinates that represent the network topology. When all individuals are created they are send on the outpout port toward the "eval population" model.

Next the "population evaluator" model receives the population on it input port. The model can translate individual bits string representation if necessary. After this process the model broadcasts population over it output ports toward the decision support model(s) in order to produce simulation results. These results are stored until decision support simulation ends.The model ponderates the collected simulation results and computes a fitness for each solution. The fitness represents the response to the given problem like a score or an error value. When fitness computation ends, the population and individual fitness are send toward the selection model.

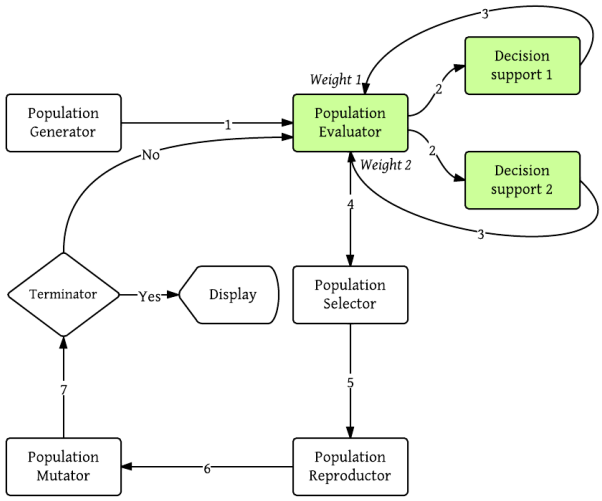

Figure 2 DEVS genetic algorithm concepts

The "population selector" model sorts the different individuals by their fitness and splits them into two groups:

1. The survivors
2. The condemned

These two groups are sent toward the "population reproductor" model.

By a crossover of genetic code the "population reproductor" will generate a new group of individuals from survivors. New children are created as visible on Figure 3

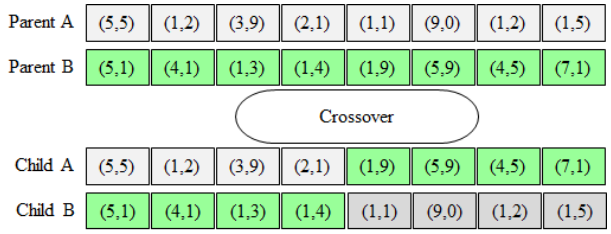

Figure 3 Genetic Algorithm crossover

The model sends it new population toward the "population mutator' model where the new solutions will be changed depending on mutation rate. This process avoids to stop into a local optimum.

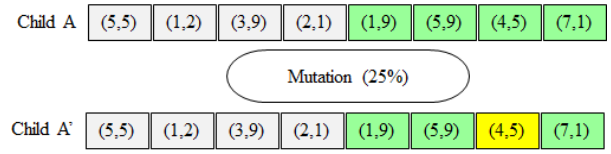

Figure 4 Genetic Algorithms mutation

During the optimization process user can vizualise the evolution into DEVSimPy using a specific pluggin. The user can drive the optimization process by defining a number of iterations to update the vizualisation interface.
The process loops until one or several fitness reach a user defined threshold or when a maximum number of iterations is exceeded.

3. WSN DEPLOYMENT MODELS

3.1. Connectivity model

WSN can be seen as a cooperative technology. Each sensor collects information and needs to relay them throught its neighbors towards the sink node and the gateway. The connecity between each other appears to be one of the main characteristic in the definition of Quality of Service (QoS) of a network. This QoS can be divided in two groups. The first group of values describes signal quality between each connected nodes named QoC (Quality Of Connectivity). The second group concerns numbers of neighbors for each node named QoN (Quantity of neighboor).

A DEVS model called "Connectivity" has been created to compute this type of simulation outputs.

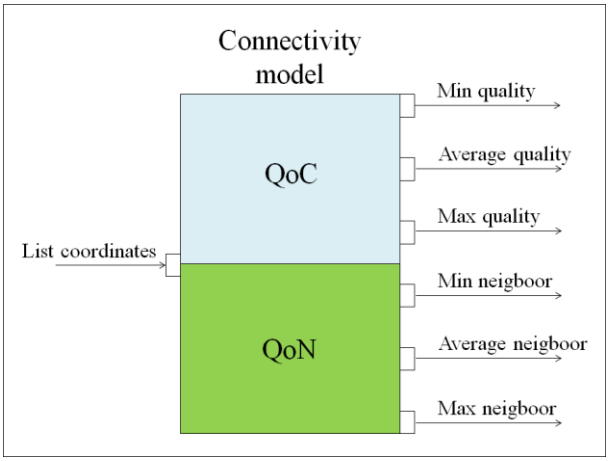

Figure 5 Connectivy model architecture

From a received list of coordinates during simulation on its single input port the connectivity model can generate different information using for the optimization of deployment process:

- Min quality : minimum quality signal
- Average quality : average quality signal
- Maximal quality : maximum quality signal
- Min neigboor : minimal number of neighbors
- Average neiboor : average number of neighbors
- Max neigboor : maximum of neighbors

QoC is estimated by using pysal (Python Spatial Analysis Library) and its module spatial weights. We define sensor communication range into the model. The Figure 6 represents for each link the estimatation model of the signal quality according to nodes distance. Different types of signal function can be used like: triangular, uniform, quadratic, epanechnikov, quartic, bisquare or gaussian.

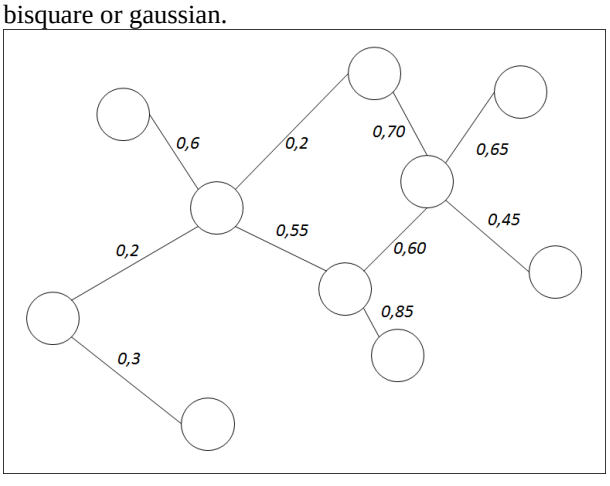

Figure 6 : Quality of links

The second group of characteristics concerns QoN. To limit nodes memory saturation and balance energy consumption over the network: the number of neighbor for each node must be the same. As shown in Figure 7 and Figure 8 the model computed the number of

neighbors of each node according to a neigboorood level. This one is in relation with WSN application and specificities of communication protocols.

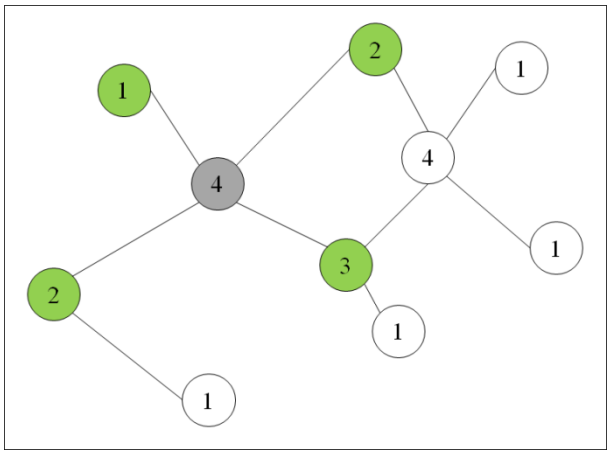

Figure 7 : Quantity of links with neighborhood level = 1

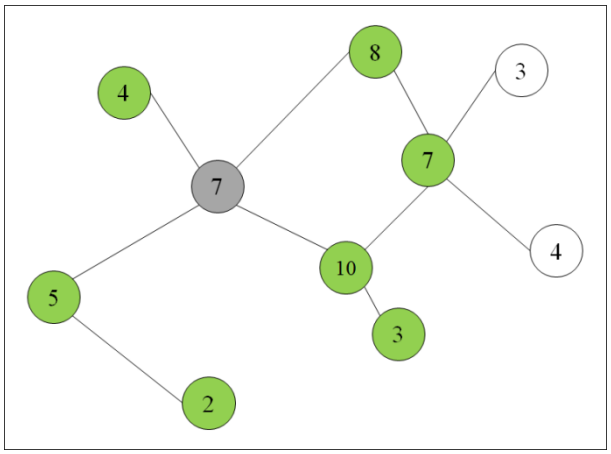

Figure 8 : Quantity of links with neighborhood level = 2

3.2. Coverage model

WSN goal is to collect data and information on a specific area. For example if the WSN application is to detect intrusion we need to cover all risks without exception.

In a WSN deployment each node localation influences other nodes locations.Therefore we need to put sensors at right places in order to maximize the area coverage and ensure a good distribution of equipements.

The coverage model provides two types of simulation output: converage informations and coverage repartition as shown in Figure 9.

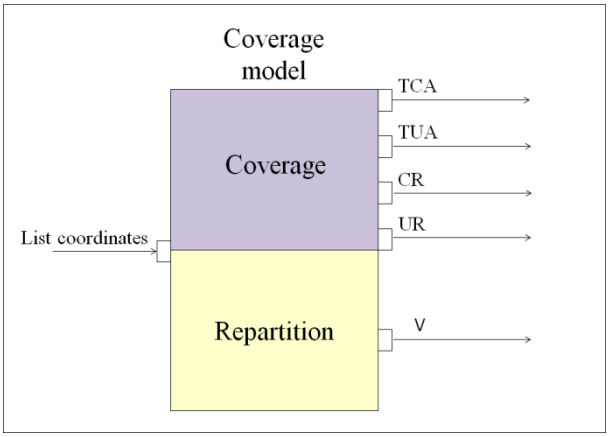

Figure 9 : Coverage model architecture

The coverage model provides different information computed with Shapely python library. The first information is TCA (Total Covered Area) givent by equation 1:

$$TCA \leftarrow \bigcup_{i=1}^{n} f(Si) \times g(Si) \cap TA \quad (1)$$

Where n is the number of sensor, f(Si) returns the sensor coverage range, g(Si) returns the sensor coverage attenuation depending on its location area or sub-area and TA referred total area. Then with this value the coverage model can easely compute the CR (Covered Rate) by equation 2:

$$CR \leftarrow TCA \times 100 \div TA \quad (2)$$

The second information is TUA (Total Uncover Area) given by equation 3:

$$TUA \leftarrow TA - TAC \quad (3)$$

With this value coverage model can compute the UR (Uncovered Rate) by equation 4:

$$UR \leftarrow TUA \times 100 \div TA \quad (4)$$

V is the variance of node number in each sub-area given by equation 5:

$$\sum_{i=1}^{n} (\alpha - \beta)^2 \quad (5)$$

Where n is the number of sub-area, α is the number of sensor into sub-area and β is the average number of sensor per sub-area.

As represented in Figure 10 we define five types of coverage (we can extend this number) capacity associated with an attenuation value:

- Full coverage (coverage = coverage)

- Good coverage (coverage = coverage x 0.80)
- Middle coverage (coverage = coverage x 0.50)
- Bad coverage (coverage = coverage x 0.20)
- Null coverage (coverage = 0)

Figure 10 : Coverage Capacity

This model allows an easy representation of AoD. The AoD is exploding into several sub-areas with own values like signal attenuation as illustrated by Figure 11.

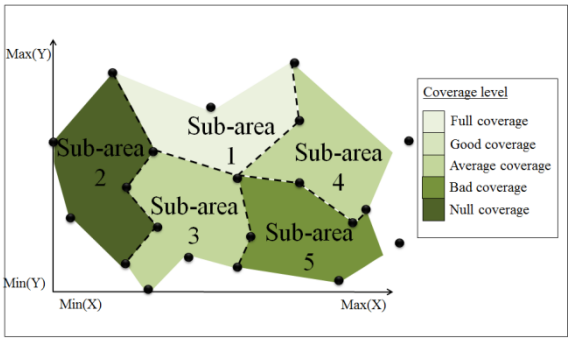

Figure 11 Area of Deployment

During simulation execution the model computes the different values using geometric operators such as intersection, union, difference as show in Figure 12.

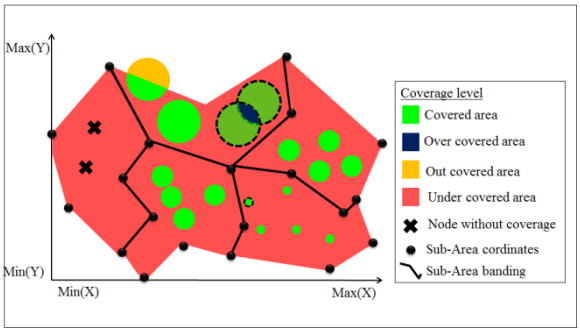

Figure 12 Covered area & uncovered area

4. RESULTS AND DISCUSSION

We have validated the previous DEVS models on three deployment optimization samples. In the first we only improve the connectivity. In the second we only improve the coverage and finally in the third we improve connectivity and coverage in a multicriteria application.

4.1. Connectivity

On this example we try to maximize the average number of neighbors for each deployed node. These nodes are deployed on AoD dimensioning 100 weights by 100 heights. The number of node is 30 with a transmission range of 10 meters.

If we observe Figure 13 we can see a random node distribution on the first iteration. Over the GAs evolution process we can see that deployment structure evolves toward height connectivity structure in witch all node are near. It is easely understandable: connectivity is inversely proportional to the distance between nodes.

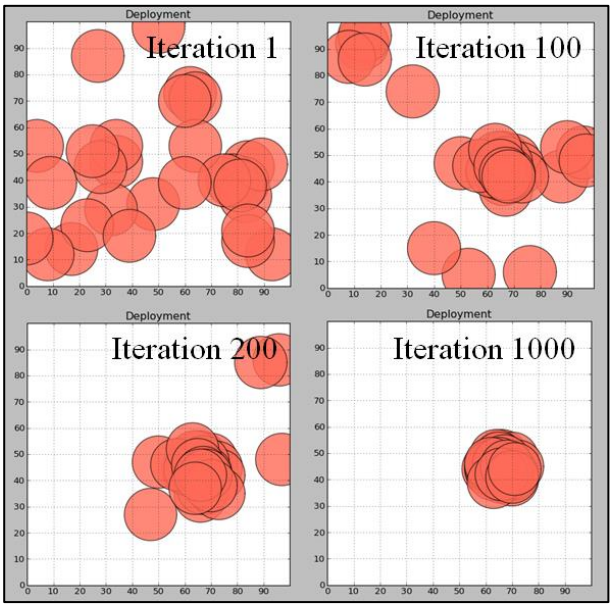

Figure 13 Connectivity optimization

4.2. Coverage

We try to maximize the deployment coverage illustrated by the Figure 15. The dimension of AoD is 100 meters weights by 100 meter height and sensor coverage range or detection range is 10 meters too. Our tool allows making distinction betweek sensing range and transmission range witch can be different.

The results show that coverage of 100 percent is really difficult to reach. This can be explaining by the difficulty to ajust cicrcle on space without blank areas. However we can note a improving of 23 percent of coverage area between the first random proposed deployment and the last optimized deployment. In the same time we can deduct that distance between nodes is proportional to the total coverage.

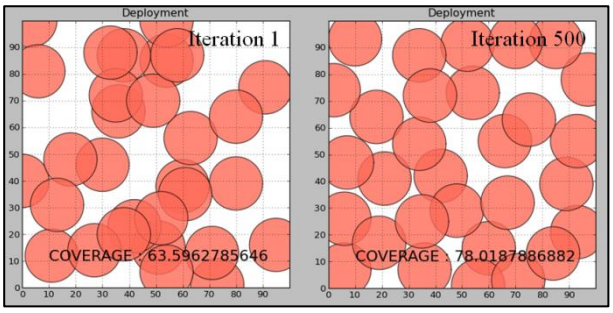

Figure 14 Coverage optimization

4.3. Qos Application

We apply these two concepts on a corsican area visible on Figure 15. After a manual and visual classification we isolate three differents sub-areas with specific coverage range atteunation as represented on Figure 16 Figure 16 and implemented into DEVSimPy as show in Figure 17:

1. Sub-area 1 : composed of rock presuming a good coverage
2. Sub-area 2 : composed of scrubland supposing a bad coverage
3. Sub-area 3 : composed of water imposing a Null coverage

Figure 15 Area of Deployment (raster view)

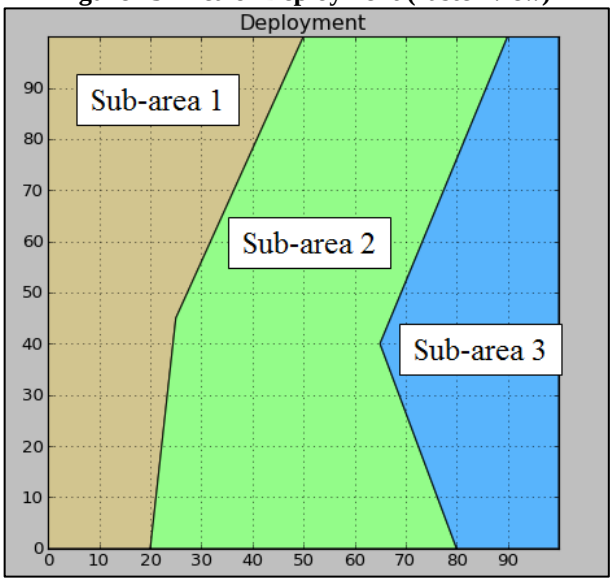

Figure 16 Area of Deplopyment (vector view)

We configure two groups of models: optimization models and decision support models through DEVSimPy interface. The solution's fitness are ponderate sums of coverage and connectivity outputs. The coefficients are 0.8 for coverage and 0.2 for connectivity. We consider this cofficients as criteria in our tool.

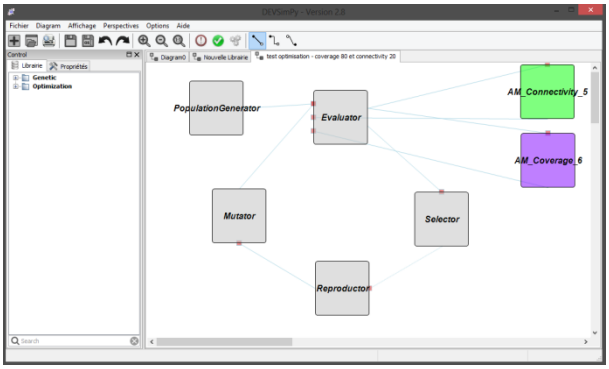

Figure 17 Deployment optimization

The Figure 18 represents the first deployment by out models. We can see a random and uniform node distribution between the different sub-areas. This configuration causes a poor AoD coverage. Moreover we can see a very low connectivity: the distance between each node is heigher than their coverage range. However two nodes are connected only and only if distance between them is lower than their range halved.

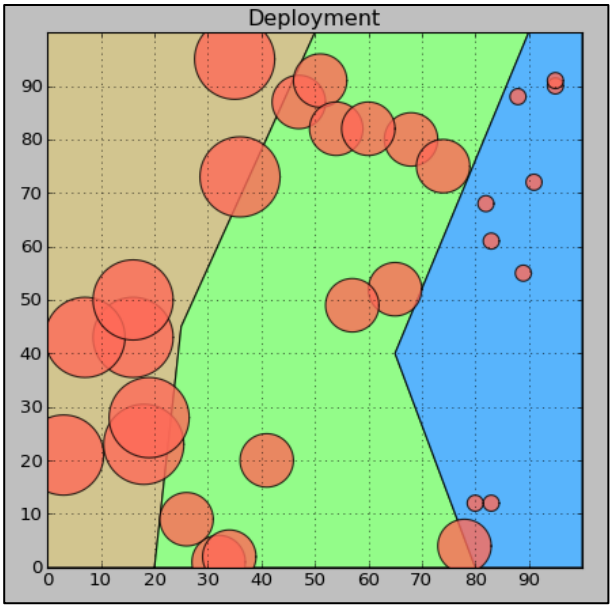

Figure 18 Generation 1

The Figure 20 represents the last deployment generation proposed by our tool. We can see the nodes move towards the sub-ara 1 where their coverage range is maximized as shown Figure 21. The results show the coverage maximization is satisfied. Moreover we can observe in the same time a height connectivy between the nodes result the connectivity criterion.

However some nodes stay in sub-area 2 where their coverage range is not maximized. This problem is characterstic for GAs. We are never sure to find the optimum.

As visible on the Figure 19, we can divide the curves evolutions into steps: between generations 0 and 60 the

fitness are increasing quickly, then we observe the fitness inscreasing much more slowly.

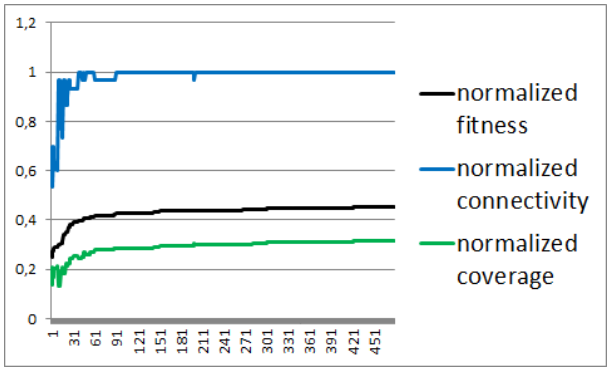

Figure 19 Normalized results

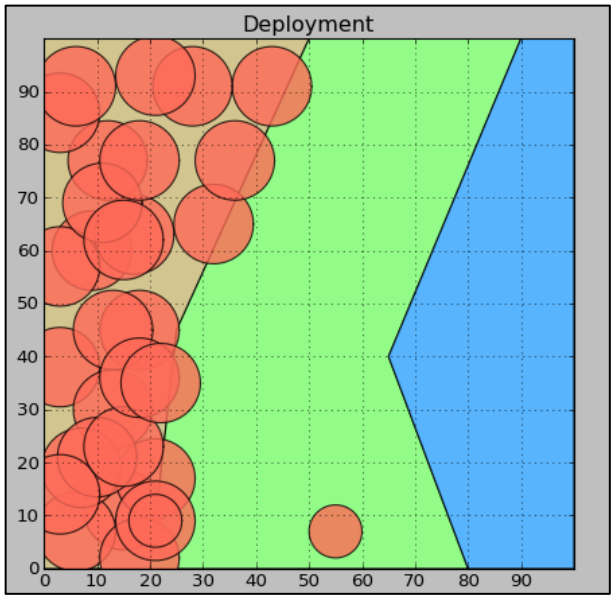

Figure 20 Generation 500

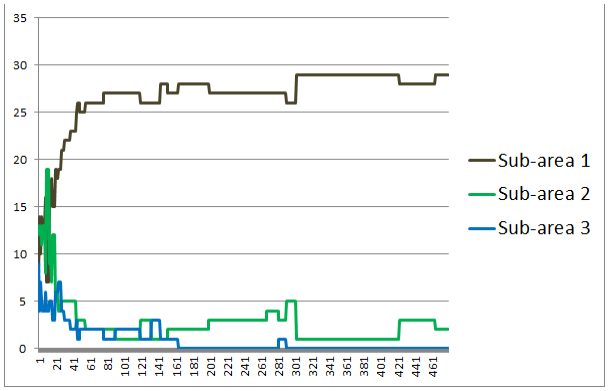

Figure 21 Node distribution

CONCLUSION AND FUTURE WORKS

In the paper we have introduced a generic DEVS framework using GAs to optimize WSN deployment according a development way targeting the definition of decision support tool.

Building on DEVS models, this framework allowed the simulation of WSN deployement according to two intrinsic parameters: connectivity and coverage. Our approach wants to be generic, not dependant of hardware platforms and limited by the domain. The first results are very interesting because they show the capacity to detect the best area for the deployment. The used example is our simulation is very simplistic however the objective is reached. The approach is able to propose the best deployement using an AI technical, AG. The GAs models implemented are generics and can be use for other optimization.

However this work is limited by the example and by the kind of AG. We propose to enhance this first approach by the three following axes:

- to test with a complex AoD, by example an insutrial site with more constrains
- to use DEVS parallel or Parallel Genetic Algorithms (PGAs) concepts to to reduce execution time because this aspect can become troublesome with the increasing of propulations
- to implement other metaheuristics algorithms like harmonic search or simulated annealing to compare the results and compute time.

REFERENCES

Antoine-Santoni, T., Santucci, J.F, De Gentili, E. Silvani, X., Morandini, F., 2009, Performance of a Protected Wireless Sensor Network in a Fire. *Analysis of Fire Spread and Data Transmission*, *Sensors*, 5878-5893

Antoine-Santoni, T., Santucci, J.F., De Gentili, E., Costa, B., 2008, Wildfire impact on deterministic deployment of a Wireless Sensor Network by a discrete event simulation, *Proceedings of the 14th IEEE Mediterranean Electrotechnical Conference (MELECON '08)*

Antoine-Santoni, T., Santucci, J.F., De Gentili, E., Costa, B., 2008, Wildfire impact on deterministic deployment of a Wireless Sensor Network by a discrete event simulation. *(MELECON '08)*

Beutel, J., Römer, K., Ringwald, M., Woehrle, M., 2009, Deployment Techniques for Sensor Networks Signals and Communication Technology, 219-248

Bolduc, J.S., Vangheluwe, H., 2001, the modeling and simulation package PythonDEVS for classical hierarchical DEVS, Technical report, MDSL-TR-2001-01, McGill University, Montreal, Canada

Capocchi, L., Santucci, J.F., Poggi, B., Nicolai, C., 2011, DEVSimPy: A collaborative Python software for Modeling and Simulation of DEVS systems, WETICE 2011, IEEE Computer Society

Cheng, X., Du, D.Z, Wang, L., Xu, B., 2007 Relay Sensor Placement in Wireless Sensor Networks, *IEEE Transactions on Computers*, 134-138.

Darwin, C., 1836, T*he origin of species*, New York, P.F. Collier

Dhillon, S.S., Chakrabarty, K., 2003 Sensor Placement for Effective Coverage and Surveillance in Distributed Sensor Networks, *Proceedings of IEEE Wireless Communications and Networking Conference*, New Orleans, LA, March.

Efrat, A., Har-Peled, S., Mitchell, J.S.B., 2005, Approximation Algorithm for Two Optimal Location Problems in Sensor Networks, *Proceedings of the 3rd International Conference on Broadband Communications, Networks and Systems*, Boston, Massachusetts.

Garcia, A., Patek, S. D., & Sinha, K. (2007). A decentralized approach to discrete optimization via simulation: Application to network flow. *Operations research*, *55*(4), 717-732.

Grandham, S.R., Dawande, M., Prakash, R., Venkatesan, S., 2003, Energy Efficient Schemes for Wireless Sensor Networks with Multiple Mobile Base Stations, *Proceedings of the IEEE Globecom*, San Francisco, CA.

Goldberg, D.E, 1989, *Genetic Algorithms in Search, Optimization and Machine Learning*, Boston, MA, USA, Addison-Wesley Longman Publishing Co., Inc

Holland, J., 1992, *Adaptation in Natural and Artificial Systems*, Cambridge, MA: MIT Press

Koza, J., 1992, Genetic Programming: On the Programming of Computers by Means of Natural Selection, Cambridge, MA: MIT Press

Langendoen, K., Baggio, A., Visser, O., 2006, Murphy loves potatoes: experiences from a pilot sensor network deployment in precision agriculture, *Parallel and Distributed Processing Symposium. (IPDPS 2006) 20th International*.

Mitchell, M., 1996, *An introduction to genetic algorithms*, Cambridge, MA, USA, MIT Press

Otero, C.E, Kostanic, I., Otero, L.D, 2009, *Development of a Simulator for Stochastic Deployment of Wireless Sensor Networks JOURNAL OF NETWORKS, VOL. 4, NO. 8, OCTOBER 2009*.

Paek, J., Chintalapudi, K., Govindan, R., Caffrey, J., Masri, S., 2005, A wireless sensor network for structural health monitoring: Performance and experience, *the Proceedings of the. 2nd IEEE Workshop on Embedded Networked Sensors* (EmNets '05)

Swisher, J. R., Hyden, P. D., Jacobson, S. H., & Schruben, L. W. 2004. A survey of recent advances in discrete input parameter discrete-event simulation optimization. *IIE Transactions*, *36*(6), 591-600.

Szewczyk, R., Mainwaring, A., Polastre, J., Anderson, J., Culler, D., 2004, An analysis of a large scale habitat monitoring application, *Proceedings of the 2nd international conference on Embedded networked sensor systems* (SenSys '04), 214-226, New York, NY, USA

Tateson, J., Roadknight, C., Gonzalez, A., Fitz, S., Boyd, N., Vincent, C., Marshall, I., 2005, Real world issues in deploying a wireless sensor network for oceanography, *Proceedings of Workshop on Real-World Wireless Sensor Networks* (REALWSN '05)

Werner-Allen, G., Lorincz, K., Johnson, J., Lees, J., Welsh, M., 2006, Fidelity and yield in a volcano monitoring sensor network, *Proeedings. of 7th Symposium Operating Systems Design and Implementation*, 381–396, New York, NY, USA

Younis, M., Akkaya, K., 2008, Strategies and Techniques for Node Placement in Wireless Sensor Networks: A Survey, Ad Hoc Networks 6 (2008) 621-655.

Zeigler, B.P., 1976, Theory of Modeling and Simulation, Wiley Interscience, New York.

Zeigler, B.P., 1987, Hierarchical, modular discrete-event modelling in an object-oriented environment, SIMULATION, vol. 49, 1987, p. 219–230

Zennaro, M., Bagula, A., Gascon, D., Bielsa Noveleta, A., 2010, Planning and deploying long distance wireless sensor networks: the integration of simulation and experimentation, *Proceedings of the 9th international conference on Ad-hoc, mobile and wireless networks*, 191-204, Edmonton, AB, Canada

AUTHORS BIOGRAPHY

Bastien Poggi was born in Ajaccio, France in 1988. In 2011, he received a MSc in Computer Science from University of Corsica, Corte, France. Currently, he is prepared a PhD in the “Sciences for environnement” laboratory at University of Corsica. His main research focuse on DEVS (Discret EVent system), GAs (Genetics Algorithms) applied to WSN (Wireless Sensor Network).

Jean-François Santucci is Full Professor in Computer Science at the University of Corsica since 1996. His main research interests are Modeling and Simulation of complex systems. He has been author or co-author of more than 150 papers published in international journals or conference proceedings. Furthermore he has been the advisor or co-advisor of more than 20 PhD students and since 1998 he has been involved in the organization of ten international conferences. He is conducting newly interdisciplinary researches involving computer science, archaeology and anthropology: since 2006 he is working in interdisciplinary research topics: in the one hand he is performing researches in the archaeoastronomy field (investigating various aspects of cultural astronomy throughout Corsica and Algeria) and on the other hand he is applying computer science approaches such as GIS (Geographic Information Systems) or DEVS (Discrete EVent system Specification) to anthropology.

Thierry Antoine-Santoni is an associate professor in Computer Science at the University of Corisca since 2010. He maintained his doctoral thesis in 2007. His main topics of research are divided in two axes: modeling and simulation of complex systems and wireless sensor network. He developed different aspects and relations between using DEVS simulation and WSN testbeds for the monitoring of environmental and industrial phenomena.