Málaga, junio de 2010

Informe Ejecutivo

TÍTULO: PSO-3.0-2010: PSO for the Intelligent Design of Communication Networks

RESUMEN: En este informe se presentan detalles sobre el diseño y la implementación del algoritmo de Cúmulo de Partículas (Particle Swarm Optimization - PSO), en su versión del standard 2007, y su empleo en la resolución de los problemas de telecomunicaciones abordados en el marco del proyecto DIRICOM. Como resultados anejos se incorporará el software generado en bibliotecas actuales (MALLBA) y disponibles en sitios web, además de incorporar en estos sitios manuales de uso e instalación.

OBJETIVOS:

1. Describir el diseño del algoritmo PSO Standard 2007.
2. Presentar los mecanismos utilizados para la resolución de los problemas.

CONCLUSIONES:

1. PSO Standard 2007 ha sido empleado con éxito en varios trabajos de comunicaciones: control de movilidad en redes GSM, configuración óptima de protocolos de transferencia de ficheros y de encaminamiento en VANETs.
2. PSO es eficiente y fácil de implementar. El standard 2007 requiere un reducido conjunto de parámetros bien especificados y analizados en la literatura.

RELACIÓN CON ENTREGABLES:

PRE: PSO-2.0-2008, VANET-1.3-2009, VANET-1.4-2009 (anterior o necesario de leer)

CO: PSO-1.0-2008 (de lectura recomendable)

Málaga, June, 2010

Executive Summary

TITLE: PSO-3.0-2010: PSO for the Intelligent Design of Communication Networks

ABSTRACT: This report summarizes the main design and implementation features of the Particle Swarm Optimization (PSO) algorithm. Specifically, it describes the Standard PSO 2007, and it use in the resolution of the telecommunication problems tackled in the scope of the DIRICOM project. Additionally, the software implementation of PSO was generated within the skeleton structure of the MALLBA library. This implementation is available (public) in the dedicated web site together with installation and using manuals.

GOALS:

1. Describing the design and implementation of the Standard PSO 2007.
2. Presenting the main mechanisms used for solving the tackled problems.

CONCLUSIONS:

1. Standard PSO 2007 has been successfully used in several works applied to telecommunications: mobility management, optimal configuration of file transfer, and routing protocols for VANETs.
2. PSO shows an efficient performance and is easy to implement. It requieres few configuration parameters well analyzed in the literature.

RELATION WITH DELIVERABLES:

PRE: PSO-2.0-2008, VANET-1.3-2009, VANET-1.4-2009 (mandatory reading)

CO: PSO-1.0-2008 (advisable reading)

PSO for the Intelligent Design of Communication Networks

DIRICOM

June 2010

1. Introduction

Particle Swarm Optimization [8] is a Swarm Intelligence technique initially developed for continuous optimization problems. However, many practical engineering problems in the field of the telecommunications are formulated as combinatorial optimization problems and specifically as binary decisions. In this report we describe how PSO is applied to the resolution of several optimization problems in telecommunications: mobility management in GSM mobile networks, parameter tuning in file transference protocols in VANETs, and parameter tuning of routing protocols in VANETs. These problems were studied in the scope of the DIRICOM project, and they require binary and continuous optimization.

The remaining of the report is organized as follows: Section 2 describe the Standard PSO 2007 algorithm and their main mechanisms for managing the solution representation. The three telecommunication problems are presented in section 3. Finally, conclusions are drawn in Section 4.

2. PSO Algorithm

In PSO, each potential solution to the problem is called particle *position* and the population of particles is called the *swarm*. We have followed the specification of the Standard PSO 2007 [6]. In this algorithm, each particle position x^i is updated each iteration g by means of the Equation 1.

$$x^i_{g+1} = x^i_g + v^i_{g+1} \tag{1}$$

where term v^i_{g+1} is the velocity of the particle, given by the following equation:

$$v^i_{g+1} = w \cdot v^i_g + \varphi_1 \cdot N[0,c] \cdot (p^i_g - x^i_g) + \varphi_2 \cdot N[0,c] \cdot (b^n_g - x^i_g) \tag{2}$$

In this formula, p^i_g is the best solution that the particle i has seen so far, b^n_g is the best particle of a neighborhood of n other particles (also known as the *social best*) randomly (uniform) selected from the swarm, and w is the inertia weight of the particle (it controls the trade-off between exploration and exploitation). Finally, φ_1 and φ_2 are the acceleration coefficients that control the relative effect of the personal and social best particles, while $N[0,c]$ is a gaussian random value in $[0,c]$ in which is sampled anew for each component of the velocity vector, and for every particle and iteration.

The Standard PSO 2007 provides a *quantisation* method [6] used in the case of requiring integer solution encoding. This quantisation is applied to each new generated particle (in Equation 1), and transforms the continuous values of particles to discrete ones. It consist on a Mid-Thread uniform quantiser method as specified in Equation 3. The quantum step is set to $\Delta = 1$.

$$Q(x) = \Delta \cdot \lfloor x/\Delta + 0{,}5 \rfloor \tag{3}$$

Similarly to the quantisation, the Binary PSO [9] uses a sigmoid function of the particle velocity as the probability distribution for the position, that is, the particle position in a dimension is randomly ($r \in U(0,1)$) generated using that distribution. The equation that updates the particle position becomes the following:

$$x^i_{g+1} = \begin{cases} 1 & if\ r < \frac{1}{e^{v^i_{g+1}}} \\ 0 & otherwise \end{cases} \tag{4}$$

Algoritmo 1 Pseudocode of Standard PSO 2007 for OCP

```
1: initializeSwarm()
2: while g < maxIterations do
3:    for each particle x^i_g do
4:       b^n_g=bestNeighbourSelection(x^i_g, n)
5:       v^i_{g+1}=updateVelocity (w, v^i_g, x^i_g, c, φ1, p_g, φ2, b^n_g) //Eq. 2
6:       x^i_{g+1}=Q(updatePosition(x^i_g, v^i_{g+1})) //Eqs. 1 and 3
7:       evaluate(x^i_{g+1})
8:       p^i_{g+1}=update(p^i_g)
9:    end for
10: end while
```

Algorithm 1 describes the pseudo-code of the Standard PSO 2007. The algorithm starts by initializing the swarm (Line 1). The corresponding elements of each particle (solutions) are initialized with random values representing the problem variables limits. Then, for a maximum number of iterations, each particle *flies* through the search space updating its velocity and position (Lines 4, 5, and 6), it is then evaluated (Line 7), and its personal best position p^i is also updated (Line 8). Finally, the best particle found so far is returned. Table 1 shows the parameters specified in the standard configuration of PSO 2007.

Cuadro 1: PSO parameters

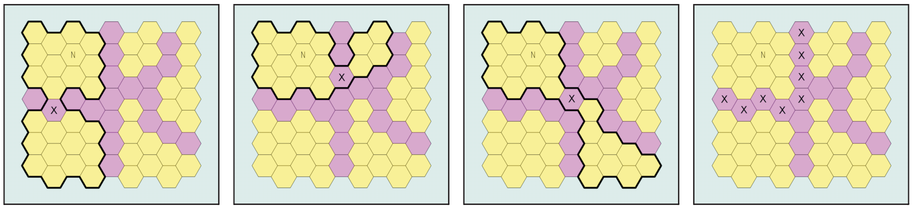
	Parameter	Value
PSO	Swarm Size	100
	Particle Size (N. Traffic Lights)	152
	Local and Social Coefficients ($\varphi_1 = \varphi_2$)	2.05
	Neighborhood size (n)	3
	Inertia Weight (w)	0.7213
	Maximum Random Limit (c)	1.19314

3. Solved Problems in Telecommunications

In this section, the problems tackled in the scope of the DIRICOM project are described. These problems are: mobility management in GSM networks, parameter tuning of file transfer protocol, and parameter tuning of routing protocols in VANETs.

3.1. Mobility Management in GSM Networks

The Mobility (location) Management [1] consists in reducing the total cost of managing a mobile cellular network. Two factors take part when calculating the total cost: the updating cost and the paging cost. The updating cost is



(a) (b) (c) (d)

Figura 1: Reporting Cells Scheme

the portion of the total cost due to location updates performed by roaming mobile terminals in the network. The paging cost is caused by the network during a location inquiry when the network tries to locate a user [1].

According to the reporting cells scheme, there are two types of cells: reporting cells (RC) and non-reporting cells (nRC). A neighborhood is assigned to each reporting cell, which consists of all nRC that must also page the user in case of an incoming call. For both RC and nRC, a *vicinity* factor is calculated representing the maximum number of reporting neighbors for each cell that must page the user (including the cell itself) in case of an incoming call. Obviously, the vicinity factor of each RC is the number of neighbors it has (see Fig. 1).

For nRC, the vicinity factor is calculated based on the fact that each nRC might be in the neighborhood of more than one RC, the maximum number of paging neighbors that contains such a cell is considered its vicinity factor. Therefore, to calculate the total cost of the network location management we use the following equation:

$$Cost = \beta \times \sum_{i \epsilon S} N_{LU}(i) + \sum_{i=0}^{N} N_P(i) \times V(i) \tag{5}$$

where, $N_{LU}(i)$ is the number of location updates for reporting cell number i, $N_P(i)$ is the number of arrived calls for cell i, $V(i)$ is the vicinity factor for cell i, S is the set of cells defined as reporting cells, and N is the total number of cells in the network. β is a constant representing the cost ratio of a location update to a paging transaction in the network (typically $\beta = 10$). This function is used either as *fitness function* by the PSO .

Since the PSO for Mobility Management was developed for binary representation, each particle i of the swarm consists of a binary vector $x_i = (x_{i1}, x_{i2}, ..., x_{in})$ representing a reporting cell configuration, where each element x_{ij} represents a cell of the network; x_{ij} can have a value of either “0”, representing a nRC, or “1”, representing a RC. For example, in an 6×6 network, the particle position will have a length (n) of 36.

3.2. Optimal File Transfer Configuration

Vehicular Ah hoc Networks (VANETs) deals with a set of vehicles and roadside units (traffic lights, signs, etc.) which are able to communicate with each other without requiring any underlying infrastructure. Such networks use IEEE 802.11 standards which implies that the nodes communicate within a limited range while moving, thus exhibiting a topology that may change quickly and in unpredictable ways. Therefore, it is crucial to provide the user with an efficient configuration of the communication protocols in order to offer the best quality of service (QoS) possible.

In the present deliverable, we solve the Optimal File Transfer Configuration (OFTC) problem in VANETs, which deals with the optimization of VDTP (*Vehicular Data Transport Protocol*) [2]. This problem have been defined with

[1] Other costs like the cost of database management to register user’s locations or the cost of the wired network (backbone) that connects the base stations to each other were not considered here, since these costs are assumed to be the same for all location management strategies and hence aren’t contemplated in comparisons.

the aim of optimizing the transmission time, the number of lost packets, and the amount of data transferred in realistic VANET scenarios.

The optimization framework used to solve this problem is composed by basically two main parts: an optimization algorithm and a simulation procedure. The optimization part is carried out by (independently) one metaheuristic algorithm, PSO in this case, adapted to find optimal (or cuasi-optimal) solutions in continuous search spaces (which is the case in this problem). The simulation process is a way of assigning a quantitative quality value to the factors regulating VDTP, thus leading to optimal configurations of this protocol tailored to a given scenario. This procedure is carried out by means of the *ns-2* [10] simulator in which we have implemented the VDTP protocol for sending files in VANETs.

The evaluation of each solution is then carried out by means of the simulation component. As Figure 2 illustrates, when PSO generates a new solution it is immediately used for configuring the VDTP. This configuration evaluates the quality of the solution by using the received *retransmission time*, *chunk size*, and *total number of attempts* [5]. Then, *ns-2* is started and maps a given VANET scenario instance, taking its time in evaluating the scenario with buildings, signal loss, obstacles, vehicles, speed, covered area, etc., under the circumstances defined by the three control parameters optimized by the algorithm. After the simulation, *ns-2* returns the global information about the *transmission time* required for sending the file, the *number of lost packets* generated during the simulation, and the *amount of data* exchanged between vehicles. This information is used to compute the *fitness* function.

Since *ns-2* operates by simulating (and averaging) many potential variations scenario all fitting the actual vehicle system, there is a possibility of obtaining different fitness values even using the same VDTP configuration (solution). Therefore, in order to provide each solution with a fitness value as reliable as possible, a single evaluation of one solution requires $N = 10$ internal simulations, computing the global fitness (F) as the mean of all *ns-2* results (Equation 7).

$$F = \frac{1}{N} \sum_{i=1}^{N} \frac{transmission\_time_i + lost\_packets_i}{\log(data\_transferred_i + C)} \tag{6}$$

In this equation, $i \in [1 \cdots 10]$ is the number of simulations per solution evaluation. The factor $C = 2$ avoids division zero if there is no data transference, preventing a possible error in the fitness calculation. The data transferred is presented in logarithmic scale in order to make up for the difference in the range of values. This way, the algorithm looks for minimizing the global fitness[2].

3.3. Optimal Routing Protocol Configuration

Vehicular Ad Hoc Networks (VANETs) are self-configuring networks composed of a collection of vehicles and elements of roadside infrastructure connected with each other without requiring an underlying infrastructure. Currently, WiFi (IEEE 802.11 based) technologies are used to deploy such networks. The coverage limitations of WiFi technologies and the high mobility of VANET nodes generate frequent topology changes and network fragmentation. For these reasons, and without any central manager entity, the routing task is a challenging work. Thus, offering an efficient routing strategy is crucial to deploy VANETs in order to offer as high as possible QoS.

A way to obtain a suitable routing protocol for VANETs is to find an optimal configuration for an existent one. The huge number of possible configurations practically prevents obtaining an efficient protocol configuration without using automatic intelligent design tools. This motivates the use of metaheuristic techniques as well-suited tools to solve such kind of problems [7]. Unfortunately, only a few related approaches can be found in the literature.

OLSR (Optimized Link State Routing Protocol) [3] is a well-known routing protocol defined specifically for MANETs/VANETs with low bandwidth and high mobility. In the present report, we use PSO for the optimal

[2] A multi-objective evaluation [4] was not taken into account since objectives are not necessarily opposed in this work.

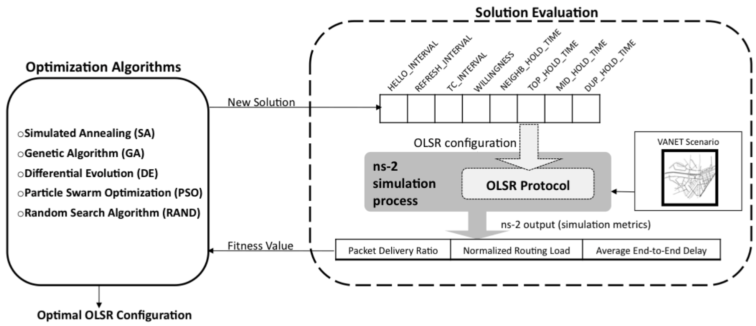


Figura 2: Optimization framework

Cuadro 2: Main OLSR Parameters. Default values following the RFC 3626 specification.

Parameter	Standard Configuration	Range
HELLO_INTERVAL	2.0 s	$1 \cdots 30$
REFRESH_INTERVAL	2.0 s	$1 \cdots 30$
TC_INTERVAL	5.0 s	$1 \cdots 30$
WILLINGNESS	3	{0, 1, 3, 6, 7}
NEIGHB_HOLD_TIME	$3 \times HELLO\_INTERVAL$	$3 \cdots 100$
TOP_HOLD_TIME	$3 \times TC\_INTERVAL$	$3 \cdots 100$
MID_HOLD_TIME	$3 \times TC\_INTERVAL$	$3 \cdots 100$
DUP_HOLD_TIME	30.0 s	$3 \cdots 100$

configuration of the OLSR protocol. Our goal is to automatically improve its performance overcoming if possible the decision of both, experts and standard (RFC 3626) [3] configurations.

The optimization strategy used to obtain a set of efficient OLSR parameters consists of coupling two different functional blocks (see Fig. 2): an optimization algorithm and a simulation procedure. The optimization task is carried out by PSO. The simulation procedure is used to assign a quality value (*fitness*) to the OLSR performance of the computed configurations during the evolutionary process. This procedure is carried out using the popular network simulator *ns-2* [10].

In spite of being OLSR specifically designed for MANETs, the application of its parameter configuration in VANET scenarios usually shoes a limited level of QoS [11]. Taking into account the impact of the configuration parameters in the network behavior, we can offer an optimized OLSR configuration to deploy VANETs. Table 2 shows these OLSR parameters with their standard values as specified in RFC 3626. The range of values each parameter can take has been defined here by following OLSR restrictions, with the aim of avoiding pointless configurations. According to this table, we can use these OLSR parameters to define a real *vector solution* and a *search space*. This way, the vector solution can be fine-tuned automatically by means of the DE algorithm.

In order to evaluate the performance of the different OLSR configurations (DE solutions), we have measured the resulted QoS indicators of the network by means of three commonly used metrics in this area: *Packet Delivery Ratio (PDR)*, *Normalized Routing Load (NRL)*, and *Average End-to-End Delay (E2ED)* of a data packet. These metrics are returned by *ns-2* after simulating a VANET scenario. After each simulation, *ns-2* returns the three QoS indicators and this information is used to compute the *fitness* function:

$$fitness = w_1 \cdot (-PDR) + w_2 \cdot NRL + w_3 \cdot E2ED \cdot C \qquad (7)$$

The objective here consists in maximizing PDR, and minimizing both, NRL and E2ED. As expressed in Equation 7, we used an *aggregative minimizing function*, and for this reason PDR was formulated with a negative sign. Factors w_1, w_2, and w_3 (0,8, 0,1, and 0,1, respectively) were empirically assessed for weighing the influence of each metric on the fitness value. Constant $C = 0{,}01$ set the $E2ED$ with the same range to PDR and NRL factors.

4. Conclusions

This report summarizes the main design and implementation features of the Particle Swarm Optimization (PSO) algorithm. Specifically, it describes the Standard PSO 2007, and it use in the resolution of the telecommunication problems tackled in the scope of the DIRICOM project.

The Standard PSO 2007 has been successfully used in several works applied to telecommunications: mobility management, optimal configuration of file transfer, and routing protocols for VANETs. PSO shows an efficient performance and is easy to implement. It requieres few configuration parameters well analyzed in the literature.

Referencias

[1] Enrique Alba, José García-Nieto, Javid Taheri, and Albert Y. Zomaya. New research in nature inspired algorithms for mobility management in gsm networks. In *EvoWorkshops*, pages 1–10, 2008.

[2] CARLINK::UMA. D2006/10 - VDTP: A file transfer protocol for vehicular ad hoc networks. Technical report, University of Malaga, Spain, 2006.

[3] T. Clausen and P. Jacquet. Optimized Link State Routing Protocol (OLSR). IETF RFC 3626, 2003 [online] in URL http://www.ietf.org/rfc/rfc3626.txt.

[4] K. Deb. *Multi-Objective Optimization using Evolutionary Algorithms*. Wiley-Interscience Series in Systems and Optimization. John Wiley & Sons, Chichester, 2001.

[5] DIRICOM::UMA. VANET-2.0 Optimal File Transfer Configuration problem (definition). Technical report, University of Malaga, Spain, 2009.

[6] M. Clerc et al. Standard PSO 2011. Technical Report [online] http://www.particleswarm.info/, Particle Swarm Central, January 2011.

[7] J. García-Nieto, J. Toutouh, and E. Alba. Automatic tuning of communication protocols for vehicular ad hoc networks using metaheuristics. *Engineering Applications of AI*, 23(5):795 – 805, 2010.

[8] J. Kennedy and R. Eberhart. Particle Swarm Optimization. *IEEE International Conference on Neural Networks*, 4:1942–1948, Nov 1995.

[9] J. Kennedy and R. C. Eberhart. *Swarm Intelligence*. Morgan Kaufmann Publishers, San Francisco, California, 2001.

[10] Ns-2. The Network Simulator - ns-2. `http://www.isi.edu/nsnam/ns/`, 2006.

[11] J. Santa, M. Tsukada, T. Ernst, O. Mehani, S. Gómez, and F. Antonio. Assessment of VANET multi-hop routing over an experimental platform. *Int. J. Internet Protoc. Technol.*, 4(3):158–172, 2009.