Málaga, 22 de noviembre de 2008

Informe Ejecutivo

TÍTULO: PSO-1.0-2008: Implementación del Algoritmo de Cúmulo de Partículas Geométrico

RESUMEN: En este informe se presentan detalles sobre el diseño y la implementación del algoritmo de Cúmulo de Partículas Geométrico (GPSO, Geometric Particle Swarm Optimization). Se trata de una versión de PSO que utiliza un marco de operación basado en espacios geométricos y mediante el cual se obtienen versiones del algoritmo para trabajar con cualquier representación básica (continua, binaria y entera). 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 e implementación del algoritmo GPSO.
2. Implementación de la versión binaria de GPSO.

CONCLUSIONES:

1. GPSO ha sido empleado con éxito en varios trabajos de comunicaciones y genética.
2. GPSO es fácil de implementar y requiere un reducido conjunto de parámetros

RELACIÓN CON ENTREGABLES:

CO: PSO-2.0-2008 (simultáneo o aconsejable de leer)

Málaga, November 22nd, 2008

Executive Summary

TITLE: PSO-1.0-2008: Geometric Particle Swarm Optimization Implementation

ABSTRACT: This report summarizes the main design and implementation features of the Geometric Particle Swarm Optimization (GPSO) algorithm. It is a novel version of PSO which enables us to generalize this algorithm to virtually any solution representation in a natural and straightforward way, extending the search to richer spaces, such as combinatorial ones. Additionally, the software implementation of GPSO 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 both design and implementation of GPSO.
2. Implementation of the binary version of GPSO.

CONCLUSIONS:

1. GPSO has been successfully used in several works applied to telecommunications and genomics.
2. GPSO is easy to implement and requieres few configuration parameters.

RELATION WITH DELIVERABLES:

CO: PSO-2.0-2008 (advisable reading)

Geometric Particle Swarm Optimization Implementation

DIRICOM

November 2008

1. Introduction

Particle Swarm Optimization [6] is a Swarm Intelligence technique initially developed for continuous optimization problems. However, many practical engineering problems are formulated as combinatorial optimization problems and specifically as binary decisions (which is the case in feature selection). Several binary versions of PSO can be found in the literature [4, 7]. Nevertheless, all these versions are *ad hoc* adaptations from the original PSO and therefore their performance is usually improvable. A recent version called Geometric Particle Swarm Optimization (GPSO) [8] enables to generalize PSO to virtually any solution representation in a natural and straightforward way, extending the search to other search spaces, such as combinatorial ones. This property has been demonstrated for the cases of Euclidean, Manhattan and Hamming spaces in the cited work. In this report we describe the design and implementation of GPSO, and showing the main differences with regards to the canonical PSO.

The remaining of the report is organized as follows: Section 2 briefly describe the canonical PSO algorithm. The GPSO is presented in section 3. Finally, conclusions are drawn in Section 4.

2. Particle Swarm Optimization

Particle Swarm Optimization [6] is a population based metaheuristic inspired in the social behavior of birds within a flock, and initially designed for continuous optimization problems. In PSO, each potential solution to the problem is called *particle* and the population of particles is called *swarm*. In this algorithm, each particle position x^i is updated each generation g by means of the Equation 1.

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

where factor v^i_{g+1} is the velocity of the particle and is given by

$$v^i_{g+1} \leftarrow w \cdot v^i_g + \varphi_1 \cdot (p^i_g - x^i_g) + \varphi_2 \cdot (b_g - x^i_g) \tag{2}$$

In this formula, p^i_g is the best solution that the particle i has stored so far, b_g is the best particle (also known as the *leader*) that the entire swarm has ever created, and w is the inertia weight of the particle (it controls the trade-off between global and local experience). Finally, φ_1 and φ_2 are specific parameters which control the relative effect of the personal and global best particles ($\varphi_1 = \varphi_2 = 2 \cdot UN(0,1)$).

Algorithm 1 describes the pseudo-code of PSO. The algorithm starts by initializing the swarm (Line 1), which includes both the positions and velocities of the particles. The corresponding p^i of each particle is randomly initialized, as well as the leader g (Line 2). Then, during a maximum number of iterations, each particle *flies* through the search space updating its velocity and position (Lines 5 and 6), it is then evaluated (Line 7), and its p^i is also calculated (Lines 8). At the end of each iteration, the leader b is updated.

Algoritmo 1 Pseudocode of PSO

1: initializeSwarm()
2: locateLeader(b)
3: **while** $g <$ maxGenerations **do**
4:   **for** each particle x^i_g **do**
5:     updateVelocity(v^i_g) //Equation 2
6:     updatePosition(x^i_g)// Equations 1
7:     evaluate(x^i_g)
8:     update(p^i_g)
9:   **end for**
10:   updateLeader(b_g)
11: **end while**

3. Geometric Particle Swarm Optimization

The recent version of PSO called Geometric Particle Swarm Optimization (GPSO) [8] enables to generalize PSO to virtually any solution representation in a natural and straightforward way, extending the search to other search spaces, such as combinatorial ones. This property has been demonstrated for the cases of Euclidean, Manhattan and Hamming spaces in the cited work.

The key issue in this approach consists in using a multi-parental recombination of particles which leads to the generalization of a *mask-based crossover* operation, proving that it respects four requirements for being a *convex combination* in a certain space. A convex combination is an affine combination of vectors where all coefficients are non-negative. When vectors represent points in the space, the set of all convex combinations constitutes the convex hull (see [8] for a complete explanation). This way, the mask-based crossover operation substitutes the classical *movement* in PSO and *position update* operations, only adapted to continuous spaces. For Hamming spaces, which is the focus in our work, a *three-parent mask-based crossover* (3PMBCX) was defined in a straightforward way:

Given three parents a, b and c in $\{0,1\}^n$, generate randomly a crosso-ver mask of length n with symbols from the alphabet $\{a,b,c\}$. Build the offspring o by filling each position with the bit from the parent appearing in the crossover mask at the position.■

The weight values w_a, w_b and w_c indicate (for each element in the crossover mask) the probability of having values from the parents x_i, g_i or h_i, respectively. These values associated to each parent represent the *present* influence of the current position (w_a), the *social* influence of the global best position (w_b), and the *individual* influence of the historical best position found (w_c). A restriction of the geometric crossover forces w_a, w_b and w_c to be non-negative and add up to one. Figura 1 shows a typical operation scheme of the three parent mask-based crossover.

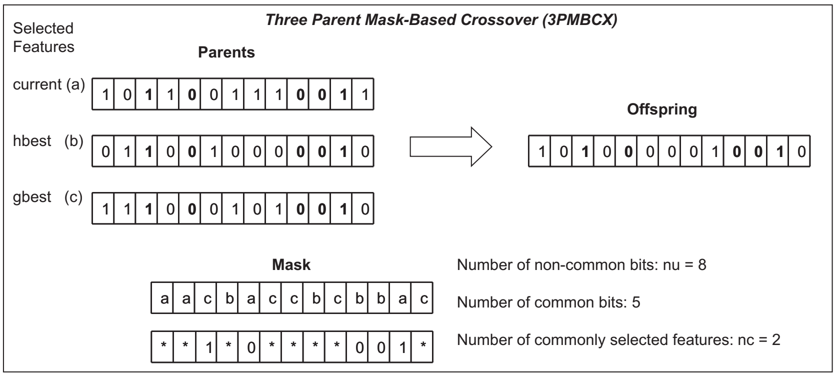

Figura 1: Three parent mask-based crossover operation

The pseudocode of the GPSO algorithm for Hamming spaces is illustrated in Algorithm 2. For a given particle i, three parents take part in the 3PMBCX operator (line 5): the current position x_g^i, the social best position b_g and the historical best position found p_g^i (of this particle). The weight values w_a, w_b and w_c indicate for each element in the crossover mask the probability of having values from the parents x_g^i, b_g or p_g^i, respectively. A constriction of the geometric crossover forces w_a, w_b and w_c to be non-negative and add up to one. These weight values associated to each parent represent the *inertia* value of the current position (w_a), the *social* influence of the global/local best position (w_b) and the *individual* influence of the historical best position found (w_c).

Algoritmo 2 Pseudocode of Geometric PSO for Hamming spaces

1: initializeSwarm()
2: locateLeader(b)
3: **while** $g <$ maxGenerations **do**
4:     **for** each particle x_g^i **do**
5:         $x_{g+1}^i \leftarrow 3PMBCX((x_g^i, w_a), (b_g, w_b), (p_g^i, w_c))$
6:         mutate(x_{g+1}^i)
7:         evaluate(x_{g+1}^i)
8:         update(p_g^i)
9:     **end for**
10:     updateLeader(b_g)
11: **end while**

GPSO has been successfully used in the recent literature. For instance, it was used in [1] for gene selection in Cancer datasets. As evaluated in this work, the *three-parent mask-based crossover* makes the offspring inherit the shared selected features present in the three parents involved in the mating. In this case, as established in Definition 3, only one offspring is generated, which represents the new position of the current particle. Here, non-shared features are inherited by the offspring corresponding to the i^{th} parent ($\{a, b, c\}$) of the mask. This way, we can state (and experiments confirm this), that the 3PMBCX crossover operator for Hamming spaces of GPSO is specially suitable for the feature selection problem, showing thus an implicit property of the studied algorithm.

In [3, 5], GPSO was applied to the Mobile Location Management (MLM). It is an important and complex telecommunication problem found in mobile cellular GSM networks. Basically, this problem consists in optimizing the number and location of paging cells to find the lowest location management cost. In this work, GPSO was customized for the MLM problem by using the concept of Hamming spaces. The results were very encouraging for current applications, and show that GPSO outperformed existing methods in the literature.

We have implemented the proposed GPSO algorithm in C++ following the *skeleton* architecture of the MALLBA library [2]. This way, there are available both sequential and parallel (island based) versions of our algorithm. The software can be found in the following URL `http://neo.lcc.uma.es/software/mallba/index.php`.

4. Conclusions

This report summarizes the main design and implementation features of the Geometric Particle Swarm Optimization (GPSO) algorithm. It is a novel version of PSO which enables us to generalize this algorithm to virtually any solution representation in a natural and straightforward way, extending the search to richer spaces, such as combinatorial ones. Additionally, the software implementation of GPSO 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. GPSO has been successfully used in several works applied to telecommunications and genomics.

Referencias

[1] E. Alba, J. García-Nieto, L. Jourdan, and E.-G.Talbi. Gene Selection in Cancer Classification using PSO/SVM and GA/SVM Hybrid Algorithms. In *IEEE Congress on Evolutionary Computation CEC-07*, Singapore, Sep 2007.

[2] E. Alba, G. Luque, J. García-Nieto, G. Ordonez, and G. Leguizamón. Mallba: a software library to design efficient optimisation algorithms. *Int. J. of Innovative Computing and Applications*, 1(1):74–85, 2007.

[3] Enrique Alba, José García-Nieto, Javid Taheri, and Albert Zomaya. New research in nature inspired algorithms for mobility management in gsm networks. In *LNCS of the Fifth European Workshop on the Application of Nature-inspired Techniques to Telecommunication Networks and other Connected Systems (EvoCOMNET'08)*, pages 1–10, Napoly, Italy, 2008. Springer-Verlag.

[4] M. Clerc. Binary Particle Swarm Optimisers: Toolbox, Derivations, and Mathematical Insights, 2005.

[5] DIRICOM. Geometric PSO for Mobility Management in GSM Networks. Technical report, PSO-2.0-2008 - University of Malaga, Spain, 2008.

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

[7] J. Kennedy and R. Eberhart. A Discrete Binary Version of the Particle Swarm Algorithm. In *Proceedings of the IEEE International Conference on Systems, Man and Cybernetics*, volume 5, pages 4104–4109, 1997.

[8] A. Moraglio, C. Di Chio, and R. Poli. Geometric Particle Swarm Optimization. In *10th European conference on Genetic Programming (EuroGP 2007)*, Abril 2007.