PSO
Optimización Basada en Cúmulo de Partículas

Dra. Andrea Villagra – Mg. Daniel Pandolfi
UNPA - UACO

Maestría en Informática y Sistemas

Agenda

1. Introducción a PSO
2. Componentes y Características
3. Aplicaciones

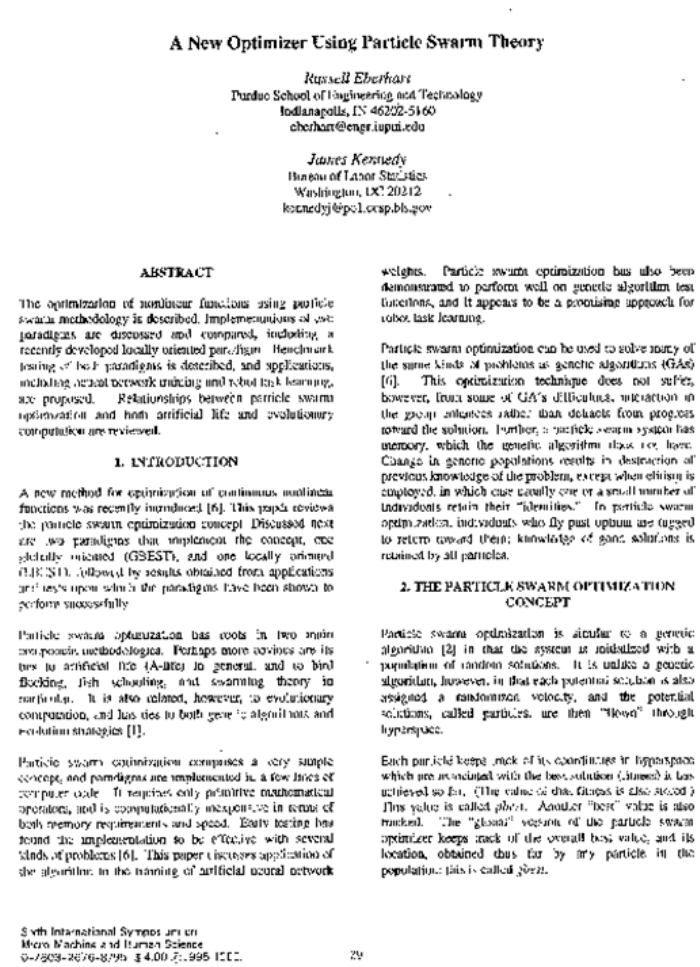
Conceptos básicos de PSO

- Inspirado en el comportamiento social del movimiento del vuelo de las aves o de los bancos de peces.
- Algoritmo de optimización inteligente. Pertenece a la clase de algoritmos de optimización llamados Metaheurísticas Poblacionales.
- Simple, poderoso y utilizado exitosamente en diversas aplicaciones en campos de ciencia, ingeniería, aprendizaje automático, minería de datos, investigación operativa, entre otros.

Conceptos básicos de PSO

- Propuesto inicialmente por James Kennedy y Russel Eberhart en 1995.

Russel Eberhart and James Kennedy



A New Optimizer Using Particle Swarm Theory

Russell Eberhart
Purdue School of Engineering and Technology
Indianapolis, IN 46202-5160
eberhart@engr.iupui.edu

James Kennedy
Bureau of Labor Statistics
Washington, DC 20212
kennedyj@pol.ocsp.bls.gov

ABSTRACT

The optimization of nonlinear functions using particle swarm methodology is described. Implementations of two paradigms are discussed and compared, including a recently developed locally oriented paradigm. Benchmark testing of both paradigms is described, and applications, including neural network training and robot task learning, are proposed. Relationships between particle swarm optimization and both artificial life and evolutionary computation are reviewed.

1. INTRODUCTION

A new method for optimization of continuous nonlinear functions was recently introduced [6]. This paper reviews the particle swarm optimization concept. Discussed next are two paradigms that implement the concept, one globally oriented (GBEST), and one locally oriented (LBEST), followed by results obtained from applications and tests upon which the paradigms have been shown to perform successfully.

Particle swarm optimization has roots in two main component methodologies. Perhaps more obvious are its ties to artificial life (A-life) in general, and to bird flocking, fish schooling, and swarming theory in particular. It is also related, however, to evolutionary computation, and has ties to both genetic algorithms and evolution strategies [1].

Particle swarm optimization comprises a very simple concept, and paradigms are implemented in a few lines of computer code. It requires only primitive mathematical operators, and is computationally inexpensive in terms of both memory requirements and speed. Early testing has found the implementation to be effective with several kinds of problems [6]. This paper discusses application of the algorithm to the training of artificial neural network weights. Particle swarm optimization has also been demonstrated to perform well on genetic algorithm test functions, and it appears to be a promising approach for robot task learning.

Particle swarm optimization can be used to solve many of the same kinds of problems as genetic algorithms (GAs) [6]. This optimization technique does not suffer, however, from some of GA's difficulties; interaction in the group enhances rather than detracts from progress toward the solution. Further, a particle swarm system has memory, which the genetic algorithm does not have. Change in genetic populations results in destruction of previous knowledge of the problem, except when elitism is employed, in which case usually one or a small number of individuals retain their "identities." In particle swarm optimization, individuals who fly past optima are tugged to return toward them; knowledge of good solutions is retained by all particles.

2. THE PARTICLE SWARM OPTIMIZATION CONCEPT

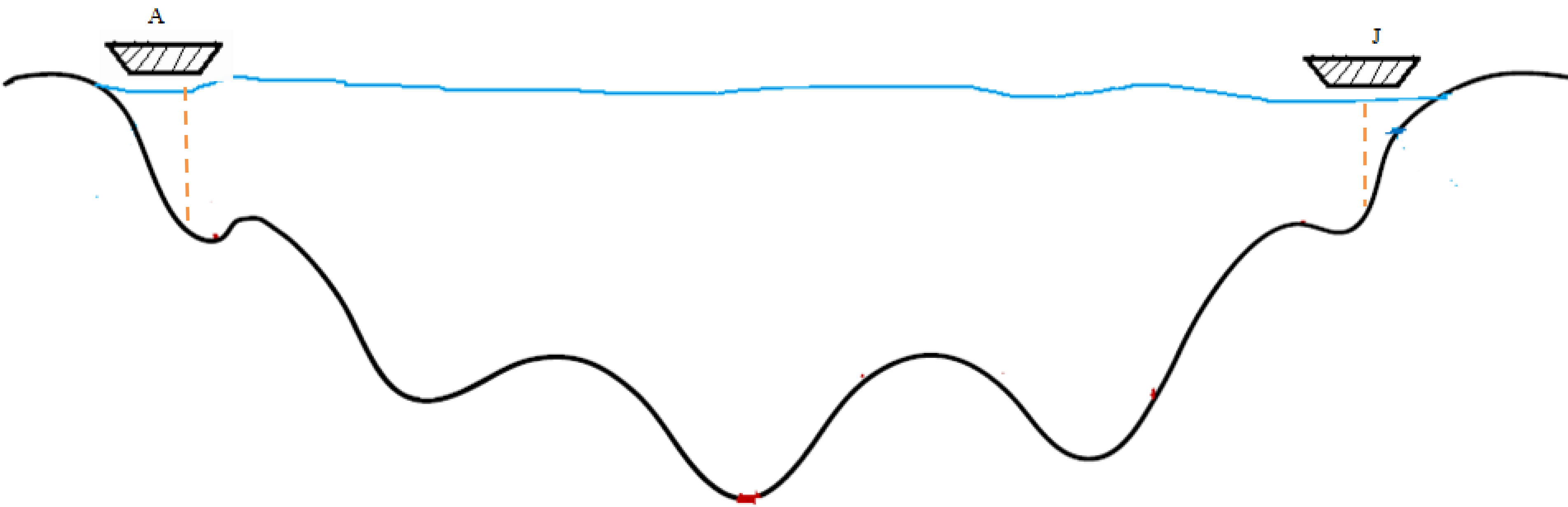
Particle swarm optimization is similar to a genetic algorithm [2] in that the system is initialized with a population of random solutions. It is unlike a genetic algorithm, however, in that each potential solution is also assigned a randomized velocity, and the potential solutions, called particles, are then "flown" through hyperspace.

Each particle keeps track of its coordinates in hyperspace which are associated with the best solution (fitness) it has achieved so far. (The value of that fitness is also stored.) This value is called *pbest*. Another "best" value is also tracked. The "global" version of the particle swarm optimizer keeps track of the overall best value, and its location, obtained thus far by any particle in the population; this is called *gbest*.

Sixth International Symposium on
Micro Machine and Human Science
0-7803-2676-8/95 $4.00 ©1995 IEEE

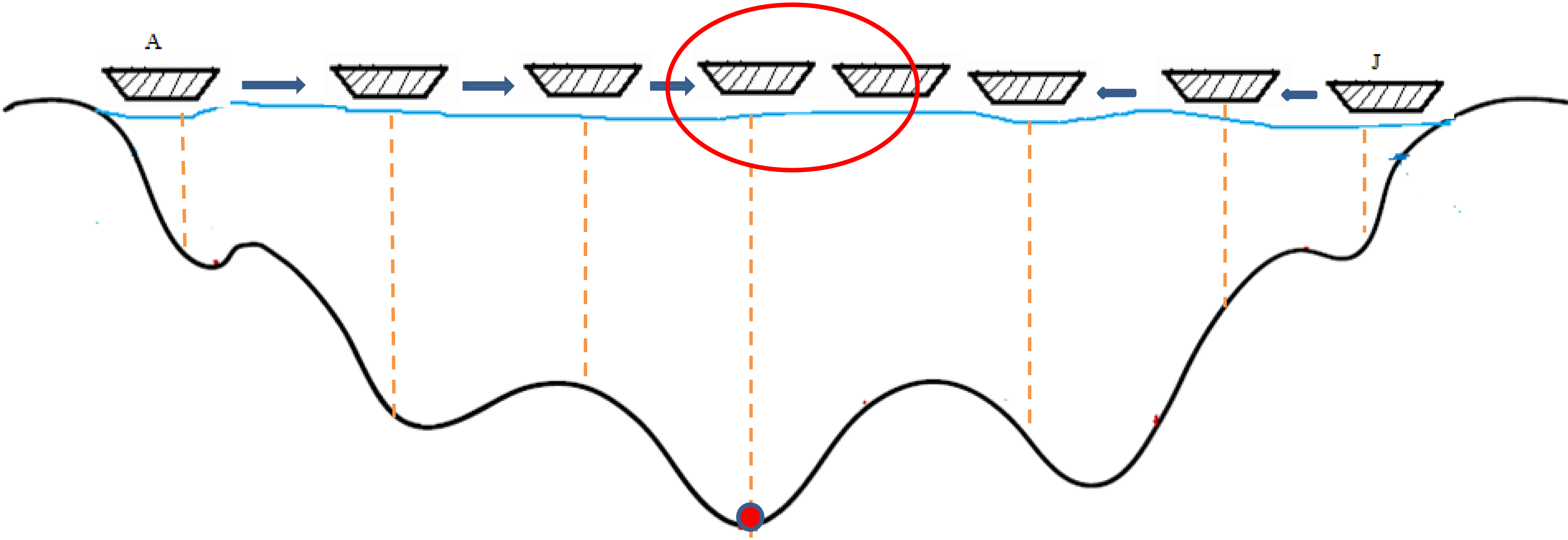
Ejemplo funcionamiento básico

- Ejemplo



Ejemplo funcionamiento básico

- Ejemplo



Mínimo global

Funcionamiento básico de PSO

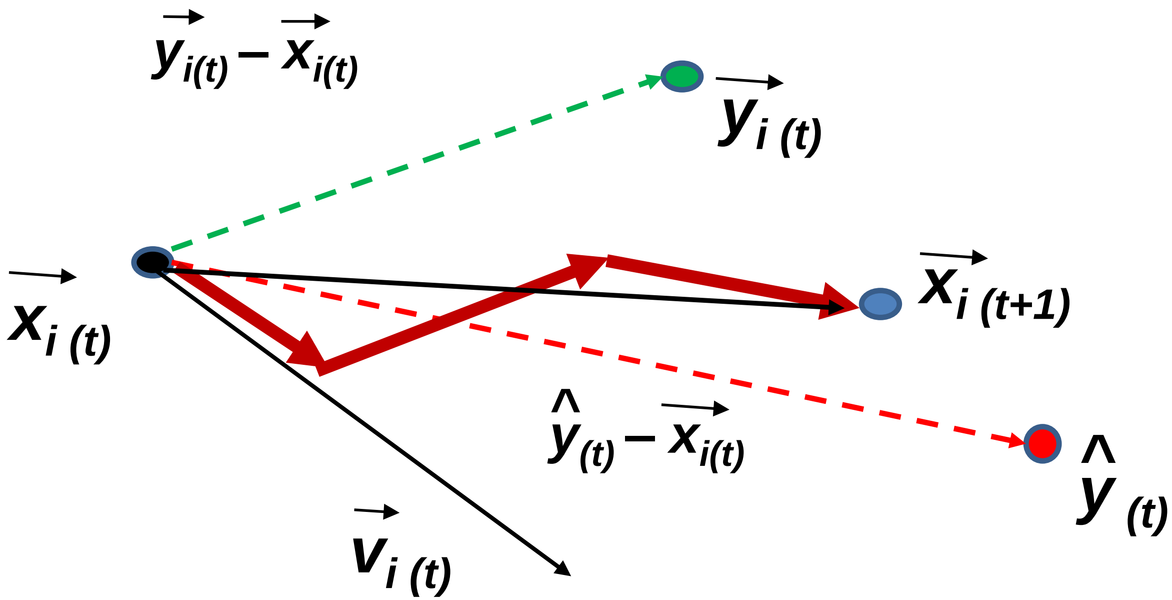
- Mecanismo de PSO similar al ejemplo anterior.
- Se utilizan dos principios:
  - Comunicación
  - Aprendizaje
- Cooperación para encontrar el óptimo, reglas básicas de PSO.

Funcionamiento básico de PSO

- PSO contiene una población (**cúmulo**) de soluciones candidatas llamadas **partículas**.
- Cada partícula es una posible solución al problema.
- Cada partícula tiene una posición en el espacio de búsqueda.
- El espacio de búsqueda es el conjunto de todas las posibles soluciones.

Movimiento de las Partículas

- Para una partícula $\boldsymbol{i}$ la posición de la partícula se denota por el vector $\vec{\boldsymbol{x}}_i$



Modelo matemático de PSO

- Ecuación de actualización de la velocidad de la partícula:

$$v_{i(t+1)} = v_{i(t)} + c_1 (y_{i(t)} - x_{i(t)}) + c_2 (\hat{y}_{(t)} - x_{i(t)})$$

- Ecuación de actualización de la posición de la partícula

$$x_{i(t+1)} = x_{i(t)} + v_{i(t+1)}$$

Modelo matemático de PSO

- Algunas versiones del modelo matemático del PSO standard.
- Ecuación de actualización de la velocidad de la partícula:

$$v_{ij(t+1)} = w\, v_{ij(t)} + \boxed{c_1 r_1 (y_{ij(t)} - x_{ij(t)}) +} \boxed{c_2 r_2 (\hat{y}_{j(t)} - x_{ij(t)})}$$

Componente cognitivo

Componente social

- Ecuación de actualización de la posición de la partícula

$$x_{i(t+1)} = x_{i(t)} + v_{i(t+1)}$$

PSO Global y PSO Local

gbest PSO()
{
Inicializar S; /* S = Swarm */
do
{
for each particle $i \in \{1, \ldots, m = |S|\}$ {
/* Establecer la mejor posición personal */
if $f(S.\mathbf{x}_i) < f(S.\mathbf{y}_i)$
$S.\mathbf{y}_i = S.\mathbf{x}_i$;
/* Establecer la mejor posición global */
if $f(S.\mathbf{y}_i) < f(S.\hat{\mathbf{y}})$
$S.\hat{\mathbf{y}} = S.\mathbf{y}_i$;
}
for each particle $i \in \{1, \ldots, m\}$ {
Actualizar Velocidad (1);
Actualizar Posición (1);
}
} **while** (continuar);
}

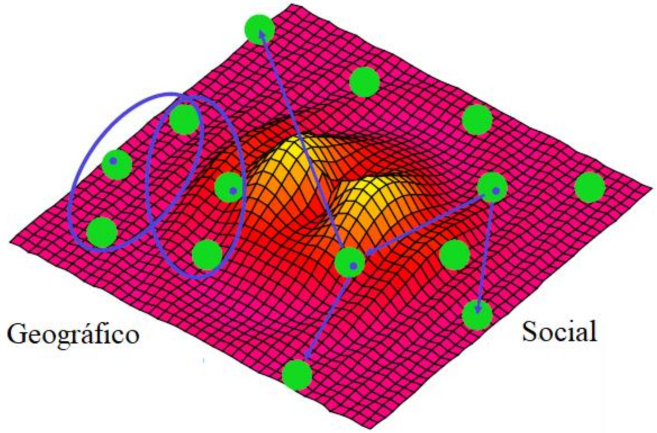
lbest PSO()
{
Inicializar S; /* S = Swarm */
do
{
for each particle $i \in \{1, \ldots, m = |S|\}$ {
/* Establece la mejor posición personal */
if $f(S.\mathbf{x}_i) < f(S.\mathbf{y}_i)$
$S.\mathbf{y}_i = S.\mathbf{x}_i$;
/* Establece la mejor posición del vecindario (local) */
if $f(S.\mathbf{y}_i) < f(S.\hat{\mathbf{y}})$
$S.\hat{\mathbf{y}} = S.\mathbf{y}_i$;
}
for each particle $i \in \{1, \ldots, m\}$ {
Actualizar Velocidad (2);
Actualizar Posición (2);
}
} **while** (continuar);
}

Valores de algunos parámetros

- Tamaño del cúmulo: Entre 20 y 40 partículas (problemas simples, 10; problemas muy complejos, 100-200).
- Los pesos de (c1 y c2): habitualmente c1=c2=2
- PSO Global vs. PSO Local: La versión global converge más rápido pero cae más fácilmente en óptimos locales y viceversa.

Topología del Cúmulo de partículas

- Las topologías definen el entorno de cada partícula individual.
- Los entornos pueden ser de dos tipos:
  - Geográficos: se calcula la distancia de la partícula actual al resto y se toman las más cercanas para componer su entorno.
  - Sociales: se define a priori una lista de vecinas para partícula, independientemente de su posición en el espacio.



- Los entornos sociales son los más empleados.

Aplicaciones de PSO

- Algunas publicaciones

Optimal Cycle Program of Traffic Lights With Particle Swarm Optimization

José García-Nieto, Ana Carolina Olivera, and Enrique Alba

***Abstract*—Optimal staging of traffic lights, and in particular optimal light cycle programs, is a crucial task in present day cities with potential benefits in terms of energy consumption, traffic flow management, pedestrian safety, and environmental issues. Nevertheless, very few publications in the current literature tackle this problem by means of automatic intelligent systems, and, when they do, they focus on limited areas with elementary traffic light schedules. In this paper, we propose an optimization approach in which a particle swarm optimizer (PSO) is able to find successful traffic light cycle programs. The solutions obtained are simulated with simulator of urban mobility, a well-known microscopic traffic simulator. For this study, we have tested two large and heterogeneous metropolitan areas with hundreds of traffic lights located in the cities of Bahía Blanca in Argentina (American style) and Málaga in Spain (European style). Our algorithm is shown to obtain efficient traffic light cycle programs for both kinds of cities. In comparison with expertly predefined cycle programs (close to real ones), our PSO achieved quantitative improvements for the two main objectives: 1) the number of vehicles that reach their destination and 2) the overall journey time.**

***Index Terms*—Particle swarm optimization, programming cycles of traffic lights, simulator of urban mobility (SUMO).**

In this sense, current research efforts in the field of automatic traffic control signals are directed to two main initiatives. On the one hand, automatic models of adaptation of signal control are designed [4]–[6] to change cycle program duration throughout the day as vehicles in queues demand these changes. The operation of these kinds of tools is directly related to the sensor system and real-time computation of the traffic flow. Although these tools successfully perform in several cities around the world [4], [7], the real management of the traffic network has a high operational cost and the real world generally tends to repeat traffic flow patterns (rush hour, holidays, etc.).

On the other hand, modern simulators [8]–[10] are very useful for helping in traffic management, since they provide researchers with an immediate and continuous source of information about traffic flow. In addition, economical issues are also taken into account in this kind of research, since the use of real traffic tests implies the necessity of additional staff and sensoring platforms. Many studies in

Aplicaciones de PSO

- Algunas publicaciones

Particle Swarm and Genetic Algorithm applied to mutation testing for test data generation: A comparative evaluation

Nishtha Jatana [a,*], Bharti Suri [b]

[a] USICT and Department of Computer Science and Engineering, MSIT, New Delhi, India
[b] USICT, GGSIPU, New Delhi, India

A R T I C L E I N F O

Article history:
Received 1 February 2019
Revised 23 April 2019
Accepted 15 May 2019
Available online 17 May 2019

A B S T R A C T

Search based test data generation has gained popularity in recent times. Mutation testing can assist in improving the effectiveness of the generated test data. We recently proposed PSO-MT (Particle Swarm Optimization along with Mutation Testing) for generation of test data. In this paper, we fortify our proposal by applying the proposed approach on larger programs from Software-artifact Infrastructure Repository (SIR). PSO exhibits similar working characteristics with those of Genetic Algorithm (GA) which has extensively been applied for evolution of test data with mutation testing. The results are evaluated

Aplicaciones de PSO

- Algunas publicaciones

smart cities

MDPI

Article

A Particle Swarm Optimization Algorithm for the Solution of the Transit Network Design Problem

Ernesto Cipriani [1], Gaetano Fusco [2], Sergio Maria Patella [3,*] and Marco Petrelli [1]

1 Department of Engineering, Roma Tre University, Via Vito Volterra 62, 00146 Rome, Italy; ernesto.cipriani@uniroma3.it (E.C.); marco.petrelli@uniroma3.it (M.P.)

2 Department of Civil, Construction and Environmental Engineering, Sapienza University of Rome, 00184 Rome, Italy; gaetano.fusco@uniroma1.it

Aplicaciones de PSO

- Algunas publicaciones

International Research Journal of Engineering and Technology (IRJET) e-ISSN: 2395-0056
Volume: 06 Issue: 01 | Jan 2019 www.irjet.net p-ISSN: 2395-0072

ADVANCE TECHNIQUE FOR IMAGE DATA CLASSIFICATION USING MAP-REDUCER BASED PSO FOR BIGDATA ANALYSIS

Ritu Patidar

***

Abstract:- Huge scale medical image processing is a serious problem in image classification. To discovery numerous parallel processing approach in the survey paper which addresses dissimilar techniques to handle huge measure dataset. In this research work compare few related works which are complete based on huge scale medical image data which classification. The survey

Reducer perfect is to major characterize the complete quantity of documents by a term-document partial point.to define and specifies the number of documents in the quantity and again define the number of dissimilar words happening through the quantity. Every matrix entry stores the quantity of times a precise word (column index) using technique incorporate the Bayesian classifier and Map-

Aplicaciones de PSO

- Algunas publicaciones

J Netw Syst Manage
DOI 10.1007/s10922-016-9385-9

CrossMark

A Survey of PSO-Based Scheduling Algorithms in Cloud Computing

Mohammad Masdari[1] · Farbod Salehi[1] · Marzie Jalali[1] · Moazam Bidaki[2]

Received: 17 April 2015 / Revised: 16 April 2016 / Accepted: 6 May 2016
© Springer Science+Business Media New York 2016

Abstract Cloud computing provides effective mechanisms for distributing the computing tasks to the virtual resources. To provide cost-effective executions and achieve objectives such as load balancing, availability and reliability in the cloud