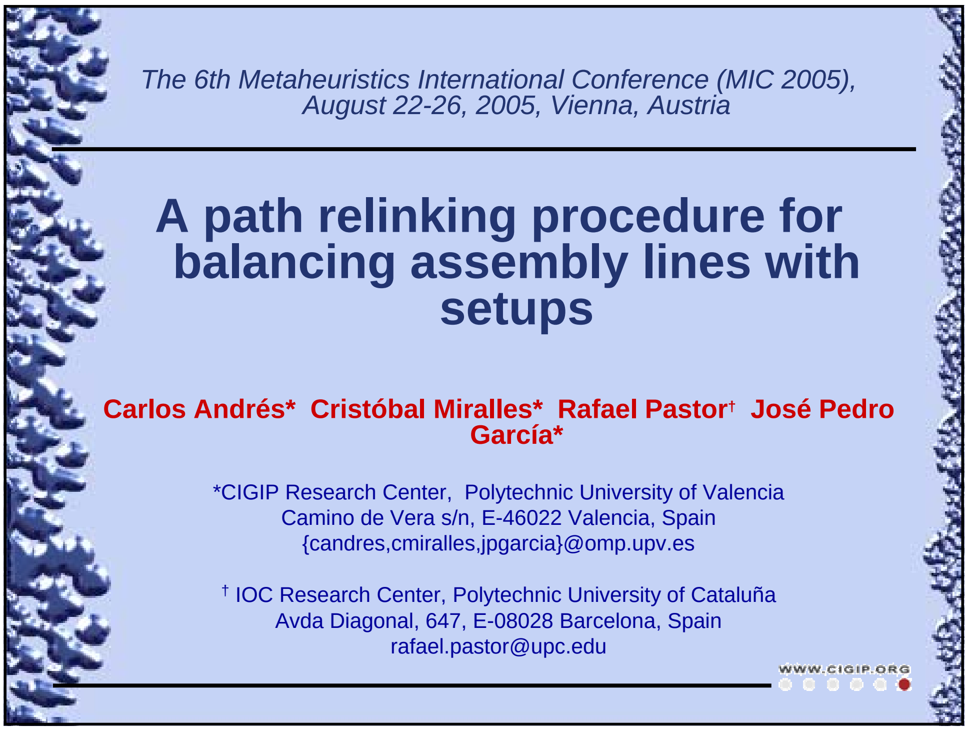

The 6th Metaheuristics International Conference (MIC 2005), August 22-26, 2005, Vienna, Austria

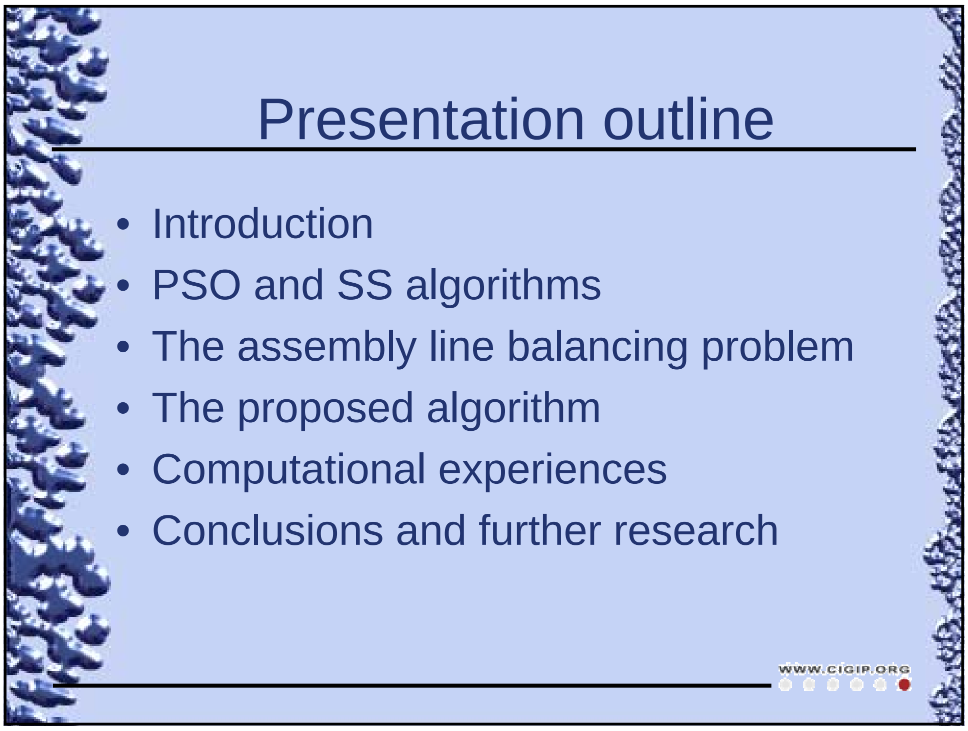
A path relinking procedure for balancing assembly lines with setups

Carlos Andrés* Cristóbal Miralles* Rafael Pastor† José Pedro García*

*CIGIP Research Center, Polytechnic University of Valencia
Camino de Vera s/n, E-46022 Valencia, Spain
{candres,cmiralles,jpgarcia}@omp.upv.es

† IOC Research Center, Polytechnic University of Cataluña
Avda Diagonal, 647, E-08028 Barcelona, Spain
rafael.pastor@upc.edu

WWW.CIGIP.ORG


Presentation outline

- Introduction
- PSO and SS algorithms
- The assembly line balancing problem
- The proposed algorithm
- Computational experiences
- Conclusions and further research

WWW.CIGIP.ORG

Introduction

- Particle Swarm Optimization (PSO) and Scatter Search (SS) algorithms are population metaheuristic methods that have been used to find the minimum of an objective function in different continuous domain problem.
- Since the original PSO algorithms were developed for optimization on a continuous domain, applications to discrete domain functions are scarce.
- In SS there applications to continuous and discrete domain are more frequent.
- Our aim is to develop an algorithm inspired in PSO with some features from SS procedures to solve some combinatorial problems (the assembly line balancing problem with setups and the machine part grouping problem).

WWW.CIGIP.ORG

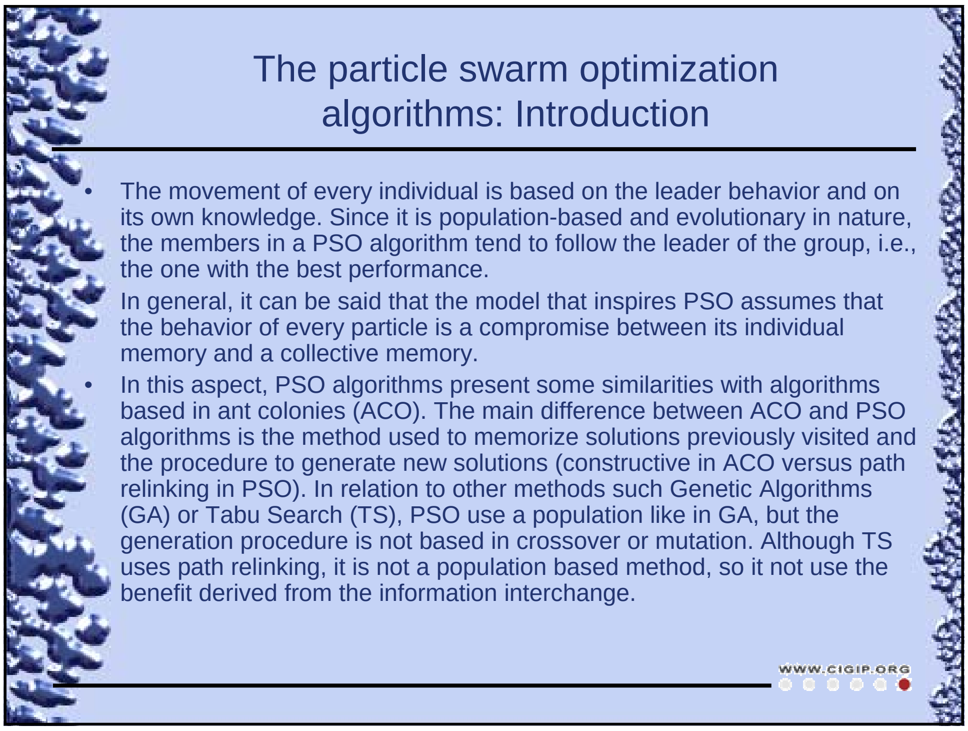
The particle swarm optimization algorithms: Introduction

PSO algorithms were proposed in the middle nineties* and they are one of the latest evolutionary optimization techniques.

Their biological inspiration is based on the metaphor of social interaction and communication in the flock of birds or school of fishes. In these groups there is a leader who guides the movement of the whole swarm.

•*Kennedy, J. y Eberhart, R.C., (1995) Particle Swarm Optimization, IEEE International Conference on Neural Networks, Australia.

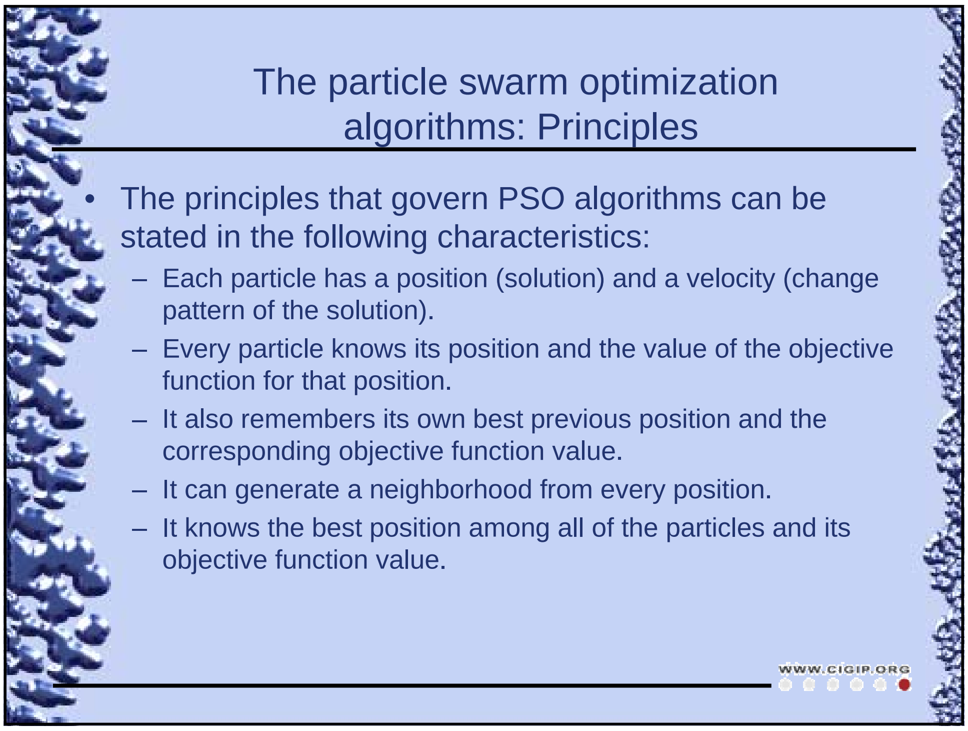
WWW.CIGIP.ORG



The particle swarm optimization algorithms: Introduction

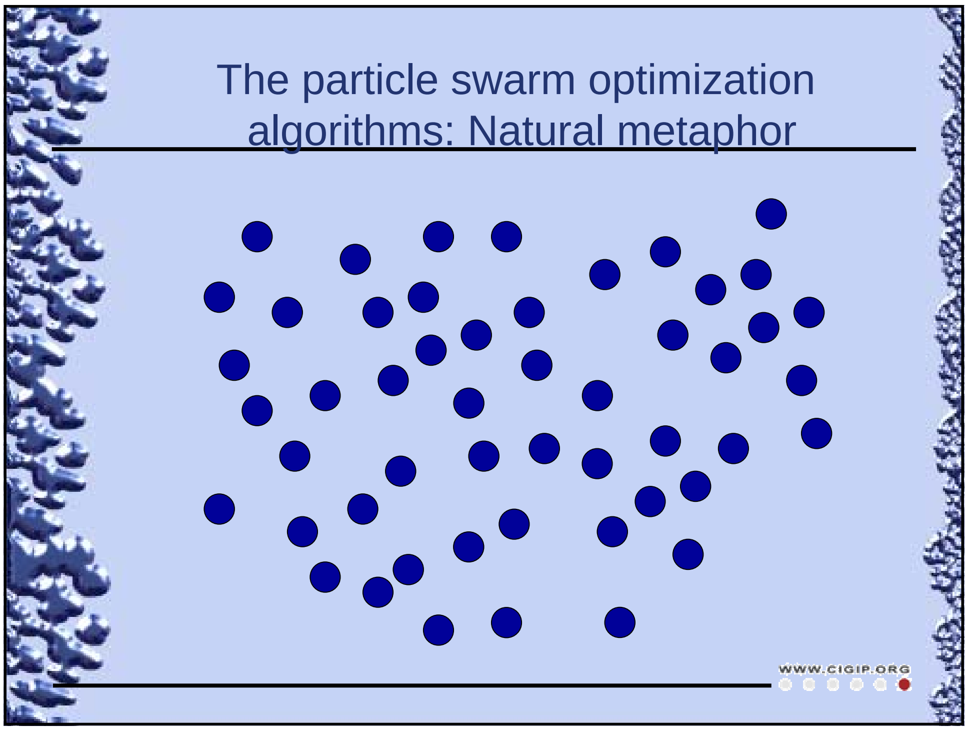
- The movement of every individual is based on the leader behavior and on its own knowledge. Since it is population-based and evolutionary in nature, the members in a PSO algorithm tend to follow the leader of the group, i.e., the one with the best performance.

  In general, it can be said that the model that inspires PSO assumes that the behavior of every particle is a compromise between its individual memory and a collective memory.
- In this aspect, PSO algorithms present some similarities with algorithms based in ant colonies (ACO). The main difference between ACO and PSO algorithms is the method used to memorize solutions previously visited and the procedure to generate new solutions (constructive in ACO versus path relinking in PSO). In relation to other methods such Genetic Algorithms (GA) or Tabu Search (TS), PSO use a population like in GA, but the generation procedure is not based in crossover or mutation. Although TS uses path relinking, it is not a population based method, so it not use the benefit derived from the information interchange.

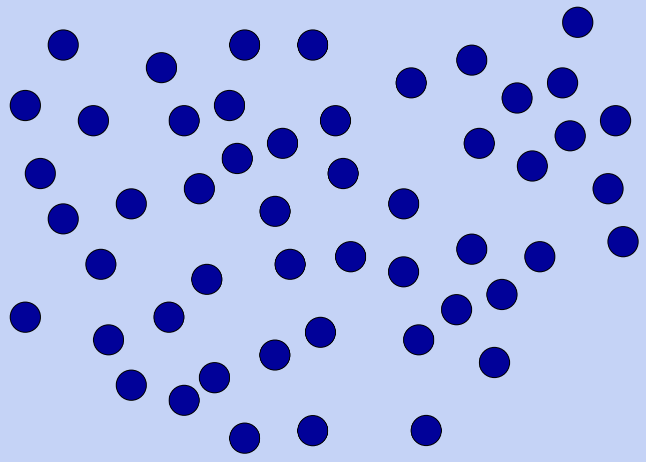


The particle swarm optimization algorithms: Principles

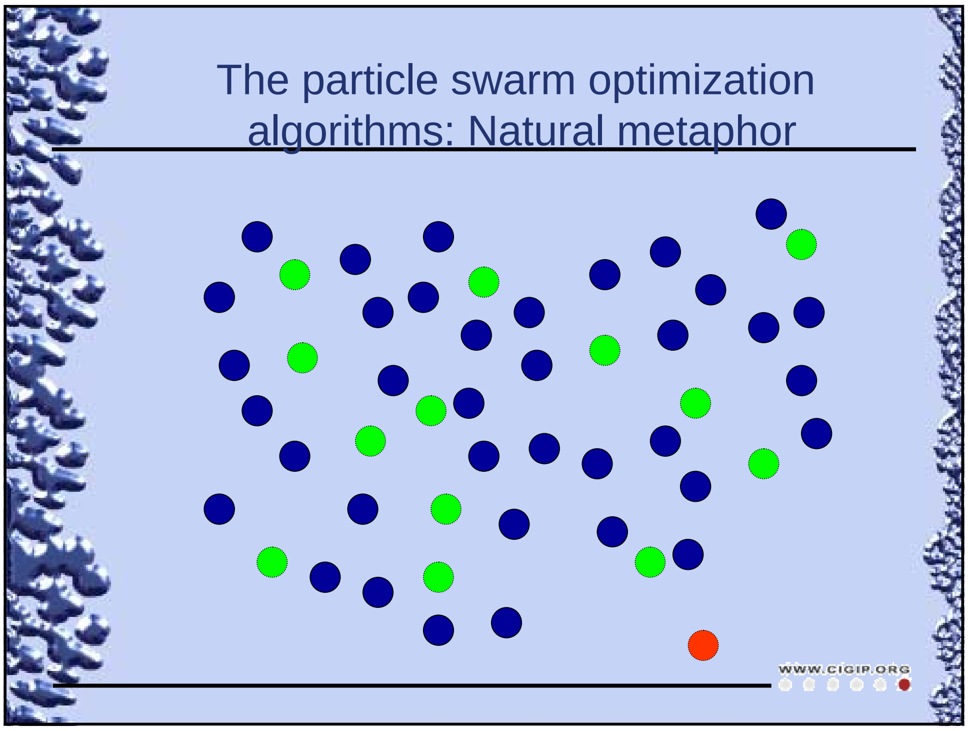
- The principles that govern PSO algorithms can be stated in the following characteristics:
  - Each particle has a position (solution) and a velocity (change pattern of the solution).
  - Every particle knows its position and the value of the objective function for that position.
  - It also remembers its own best previous position and the corresponding objective function value.
  - It can generate a neighborhood from every position.
  - It knows the best position among all of the particles and its objective function value.



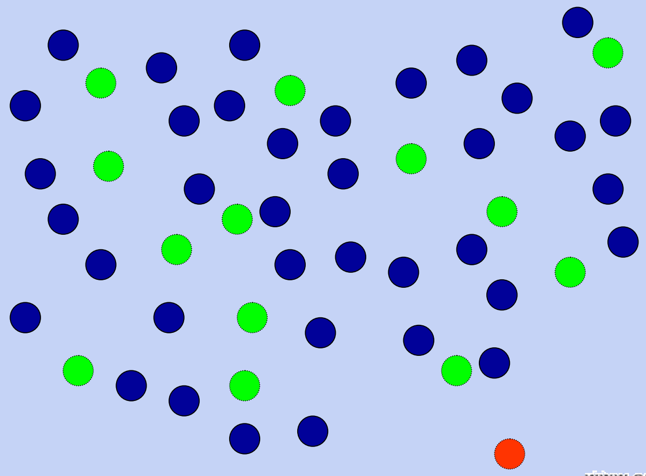
The particle swarm optimization algorithms: Natural metaphor



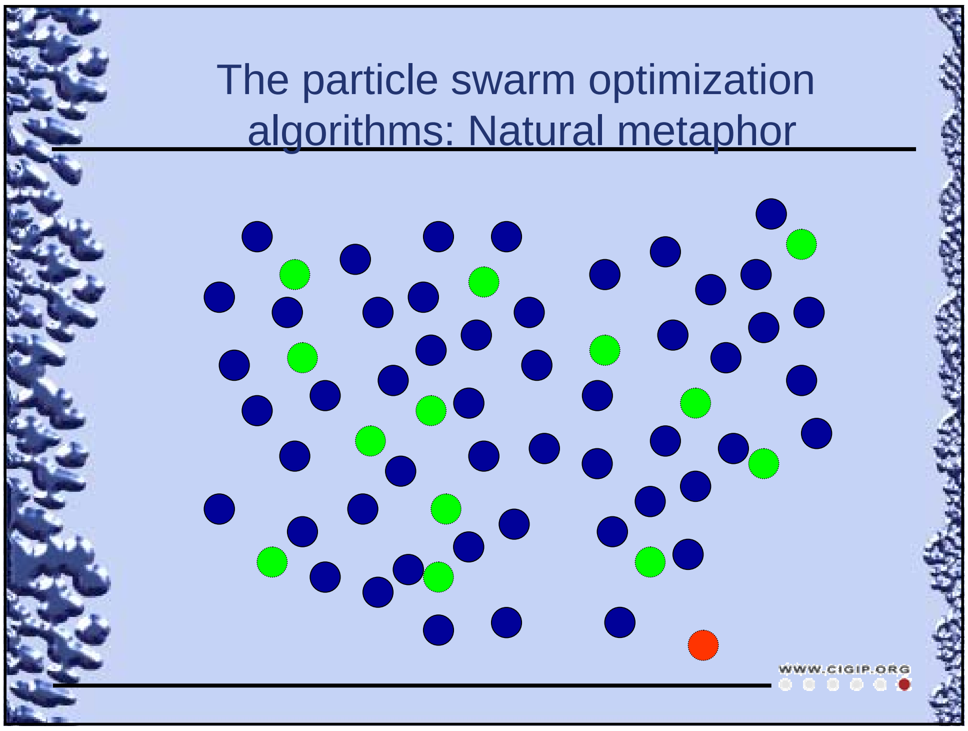
WWW.CIGIP.ORG



The particle swarm optimization algorithms: Natural metaphor

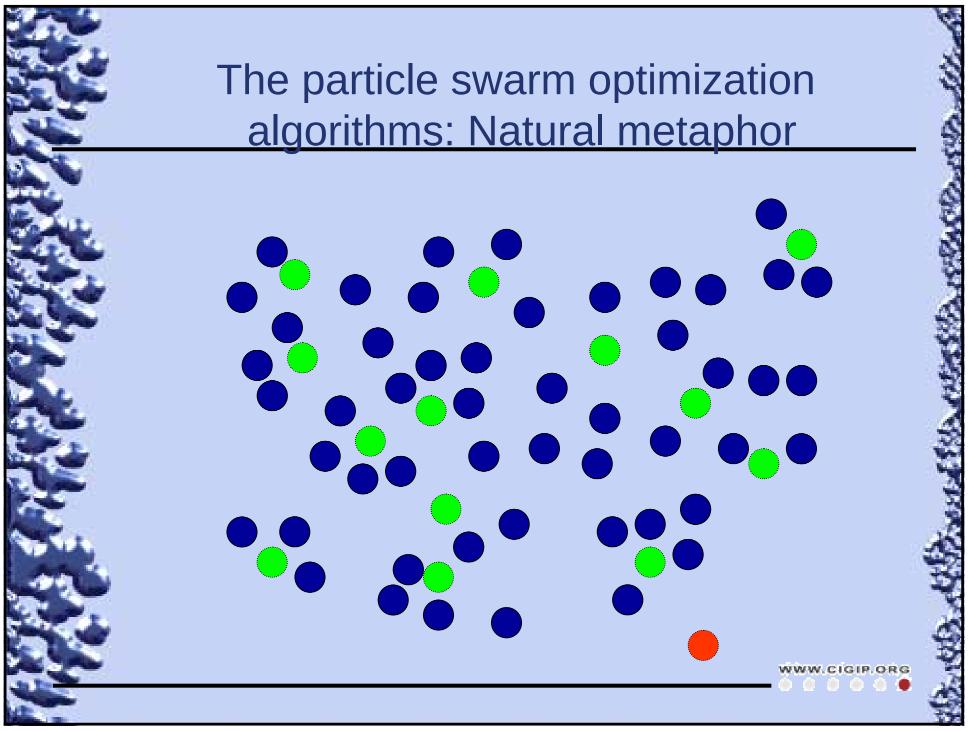


WWW.CIGIP.ORG



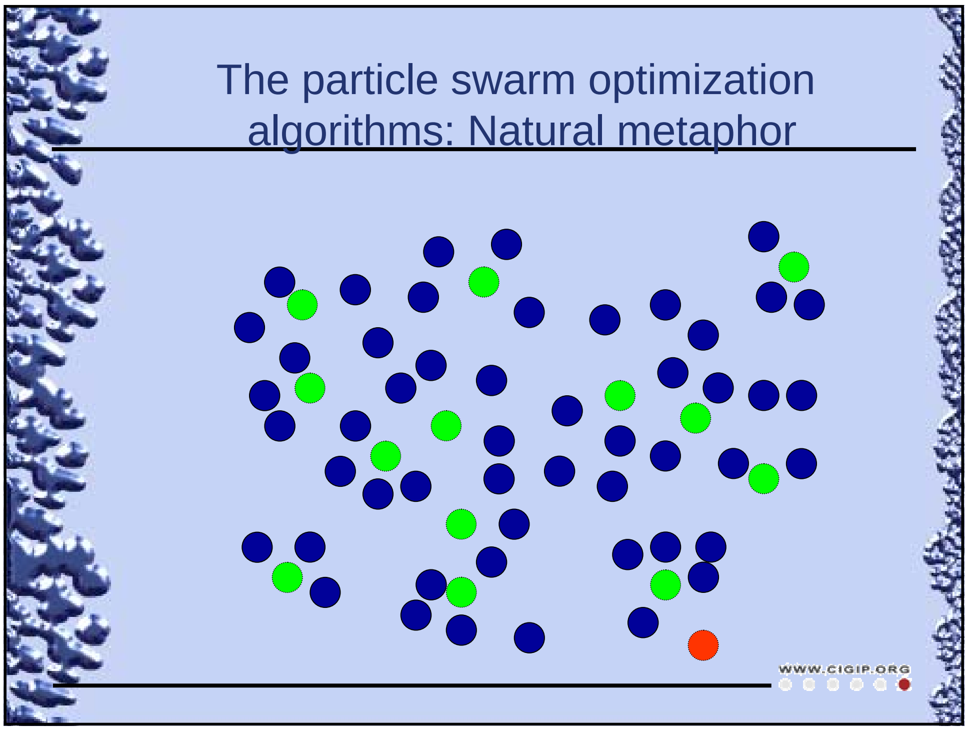
The particle swarm optimization algorithms: Natural metaphor

WWW.CIGIP.ORG



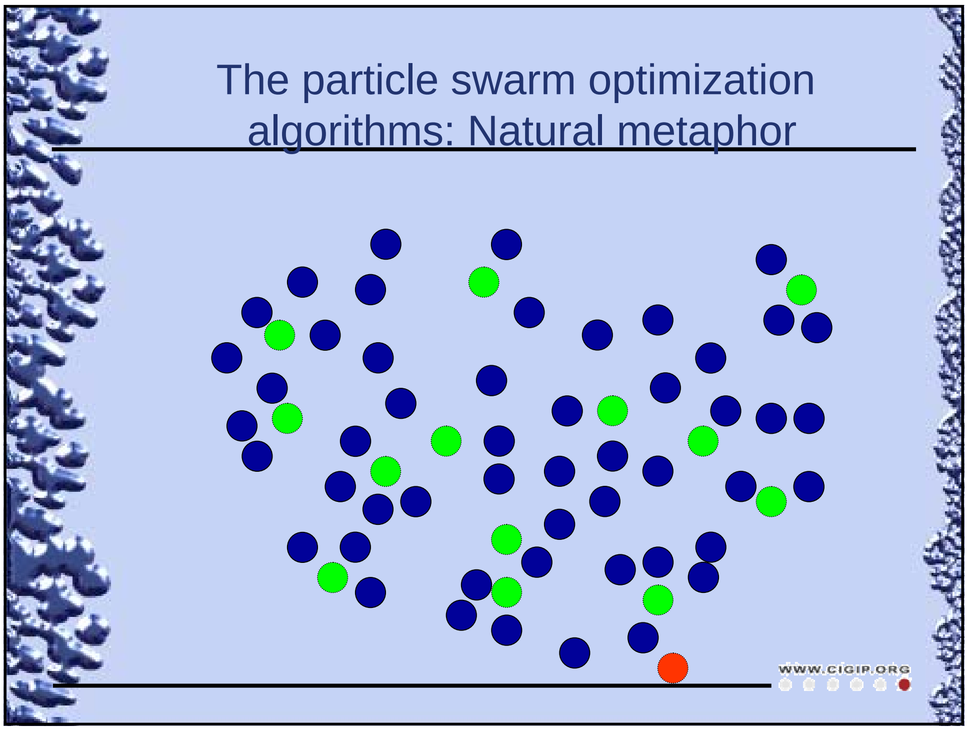
The particle swarm optimization algorithms: Natural metaphor

WWW.CIGIP.ORG



The particle swarm optimization algorithms: Natural metaphor

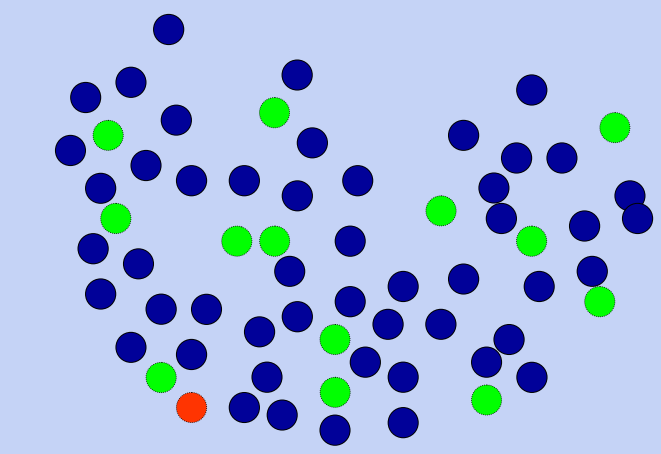
www.cigip.org



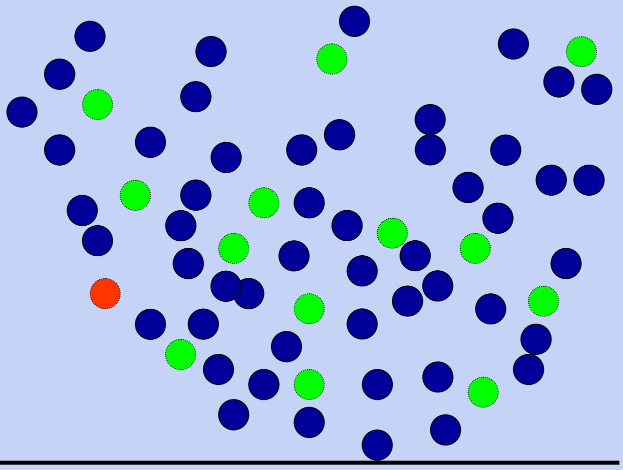
The particle swarm optimization algorithms: Natural metaphor

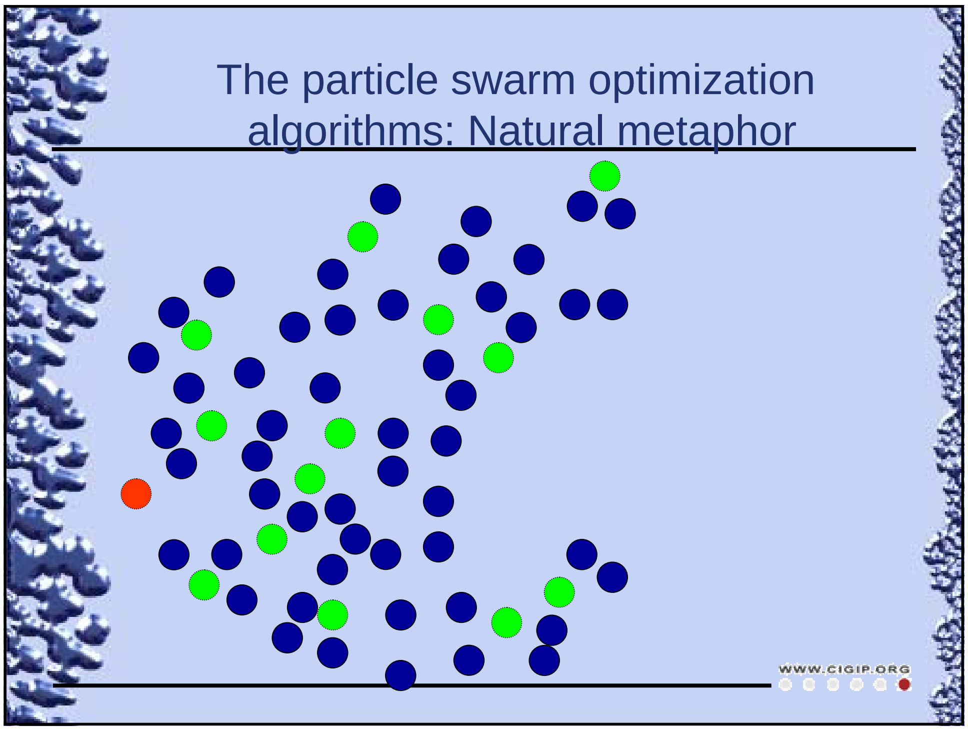
www.cigip.org

The particle swarm optimization algorithms: Natural metaphor

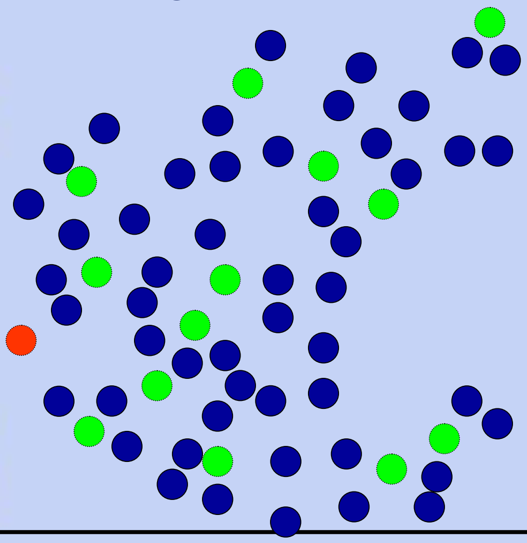


The particle swarm optimization algorithms: Natural metaphor

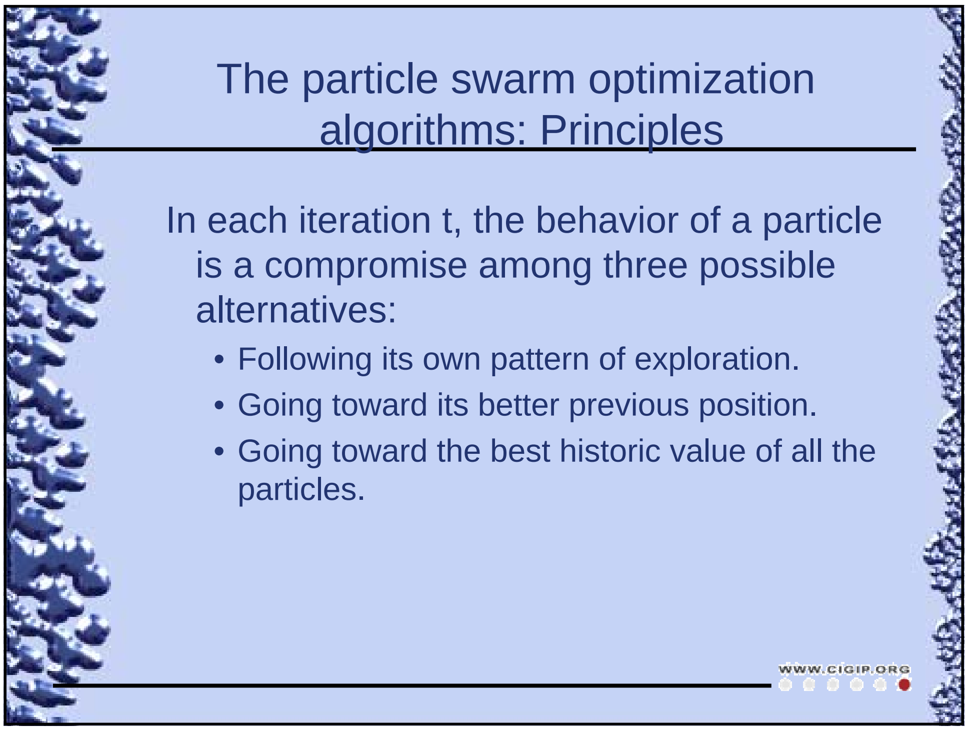




The particle swarm optimization algorithms: Natural metaphor



WWW.CIGIP.ORG

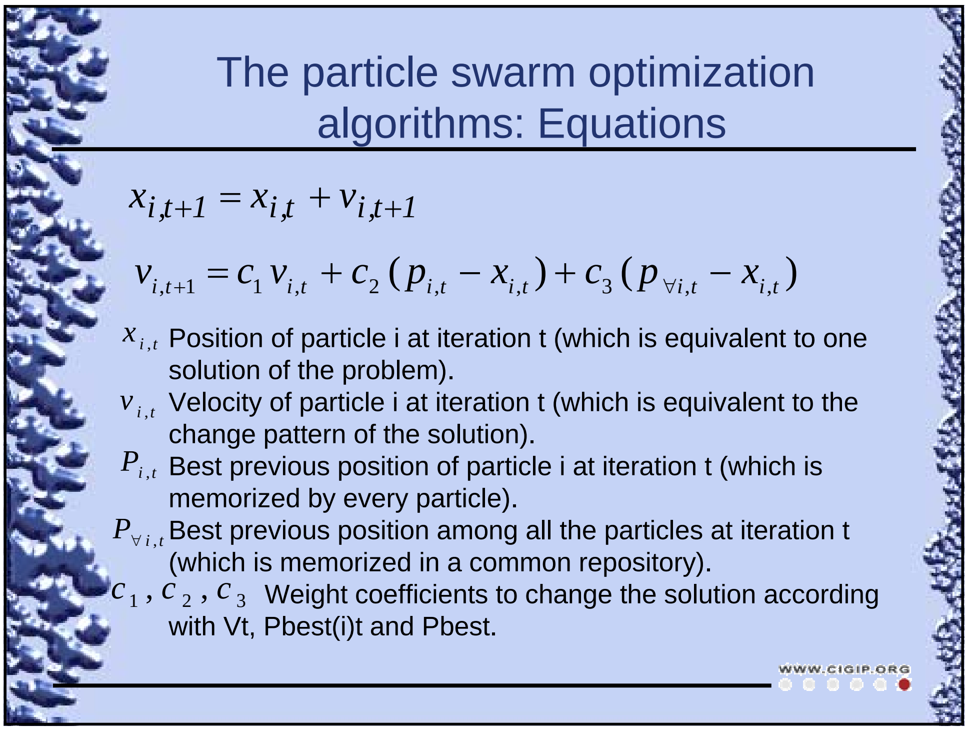


The particle swarm optimization algorithms: Principles

In each iteration t, the behavior of a particle is a compromise among three possible alternatives:

- Following its own pattern of exploration.
- Going toward its better previous position.
- Going toward the best historic value of all the particles.

WWW.CIGIP.ORG



The particle swarm optimization algorithms: Equations

$$x_{i,t+1} = x_{i,t} + v_{i,t+1}$$

$$v_{i,t+1} = c_1 v_{i,t} + c_2 (p_{i,t} - x_{i,t}) + c_3 (p_{\forall i,t} - x_{i,t})$$

$x_{i,t}$ Position of particle i at iteration t (which is equivalent to one solution of the problem).

$v_{i,t}$ Velocity of particle i at iteration t (which is equivalent to the change pattern of the solution).

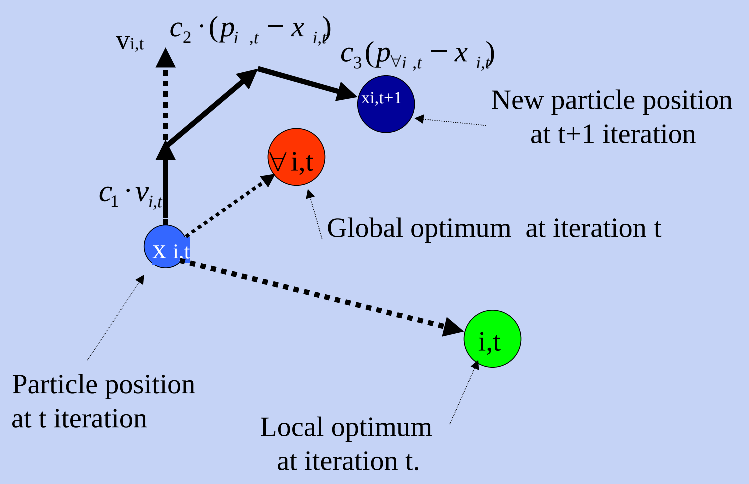
$P_{i,t}$ Best previous position of particle i at iteration t (which is memorized by every particle).

$P_{\forall i,t}$ Best previous position among all the particles at iteration t (which is memorized in a common repository).

c_1, c_2, c_3 Weight coefficients to change the solution according with Vt, Pbest(i)t and Pbest.

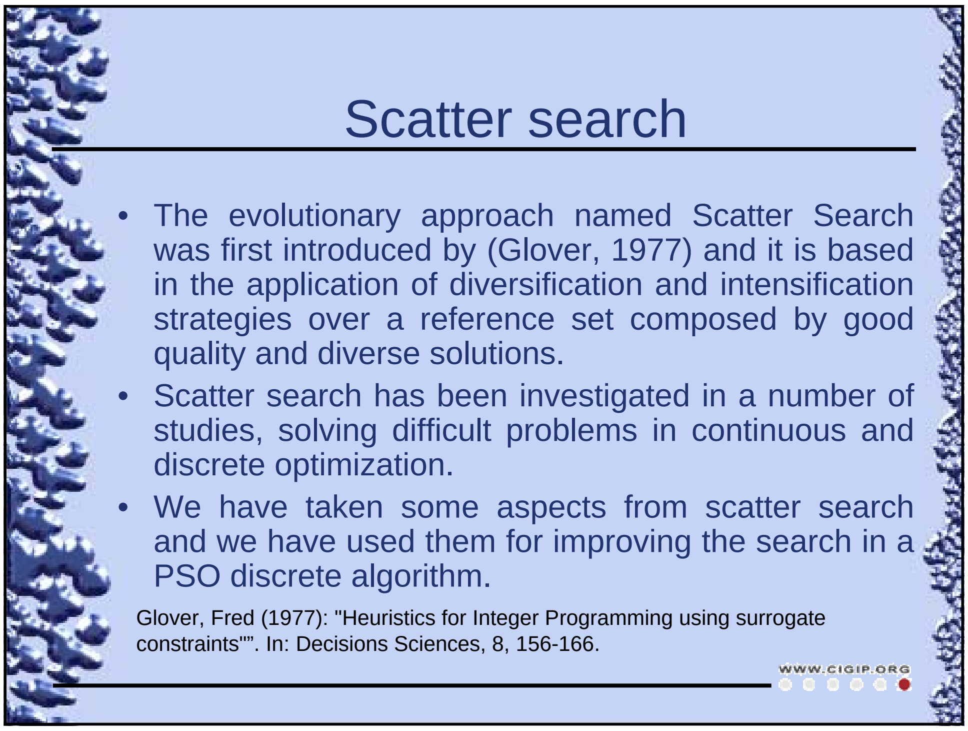
WWW.CIGIP.ORG

The particle swarm optimization algorithms: Equations



$v_{i,t}$
$c_2 \cdot (p_{i,t} - x_{i,t})$
$c_3 (p_{\forall i,t} - x_{i,t})$
xi,t+1
New particle position at t+1 iteration
$\forall$ i,t
$c_1 \cdot v_{i,t}$
x i,t
Global optimum at iteration t
i,t
Particle position at t iteration
Local optimum at iteration t.

WWW.CIGIP.ORG

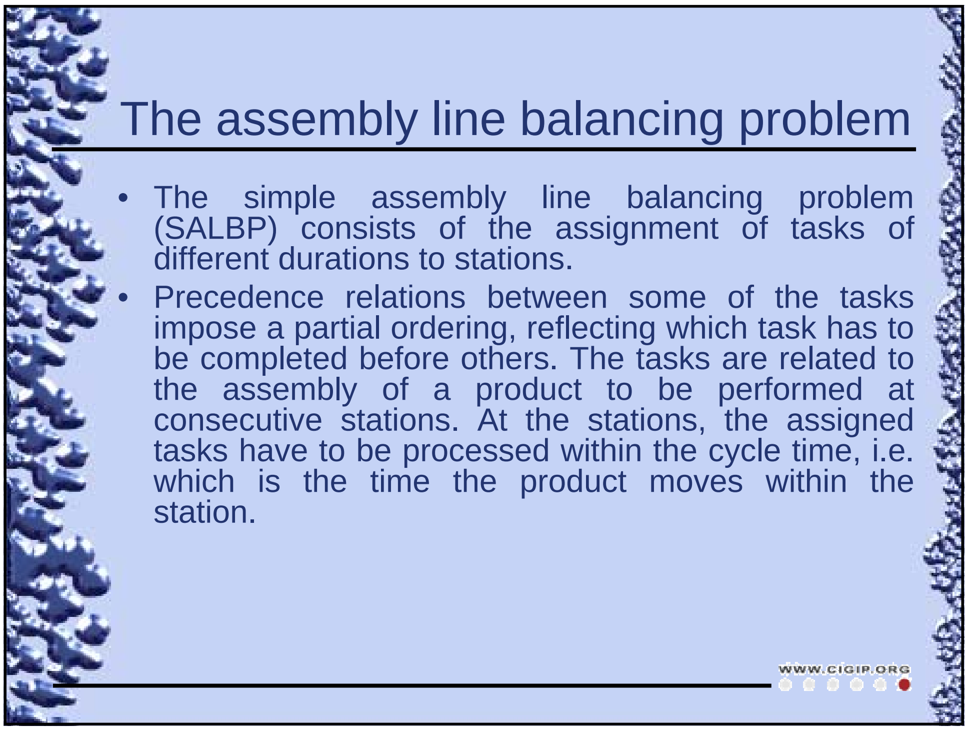


Scatter search

- The evolutionary approach named Scatter Search was first introduced by (Glover, 1977) and it is based in the application of diversification and intensification strategies over a reference set composed by good quality and diverse solutions.
- Scatter search has been investigated in a number of studies, solving difficult problems in continuous and discrete optimization.
- We have taken some aspects from scatter search and we have used them for improving the search in a PSO discrete algorithm.

Glover, Fred (1977): "Heuristics for Integer Programming using surrogate constraints"". In: Decisions Sciences, 8, 156-166.

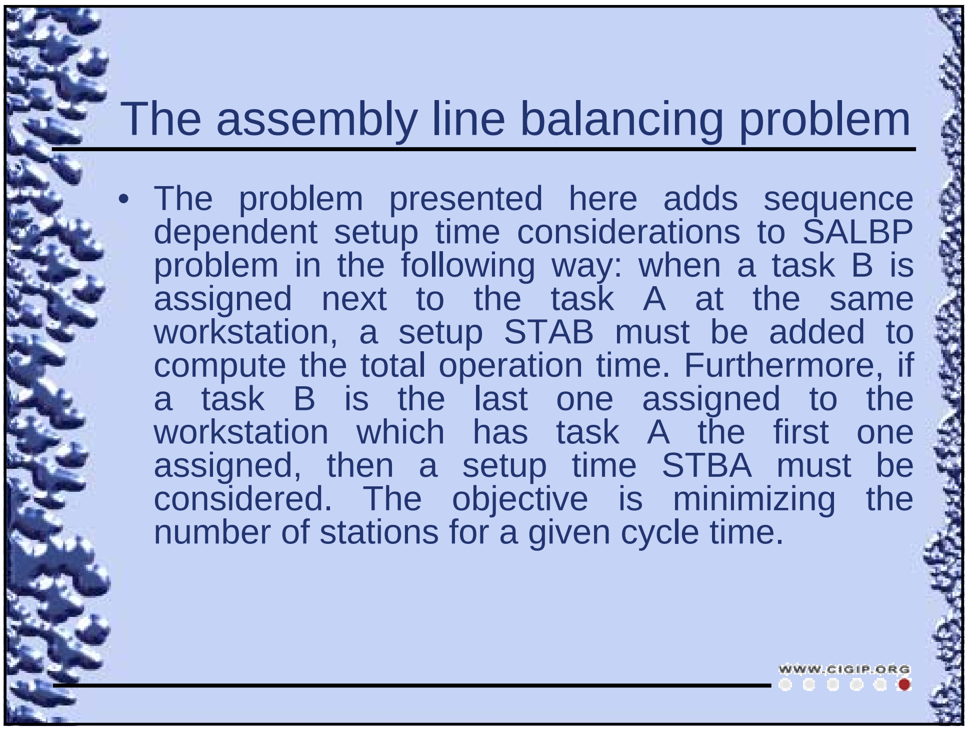
WWW.CIGIP.ORG



The assembly line balancing problem

- The simple assembly line balancing problem (SALBP) consists of the assignment of tasks of different durations to stations.
- Precedence relations between some of the tasks impose a partial ordering, reflecting which task has to be completed before others. The tasks are related to the assembly of a product to be performed at consecutive stations. At the stations, the assigned tasks have to be processed within the cycle time, i.e. which is the time the product moves within the station.

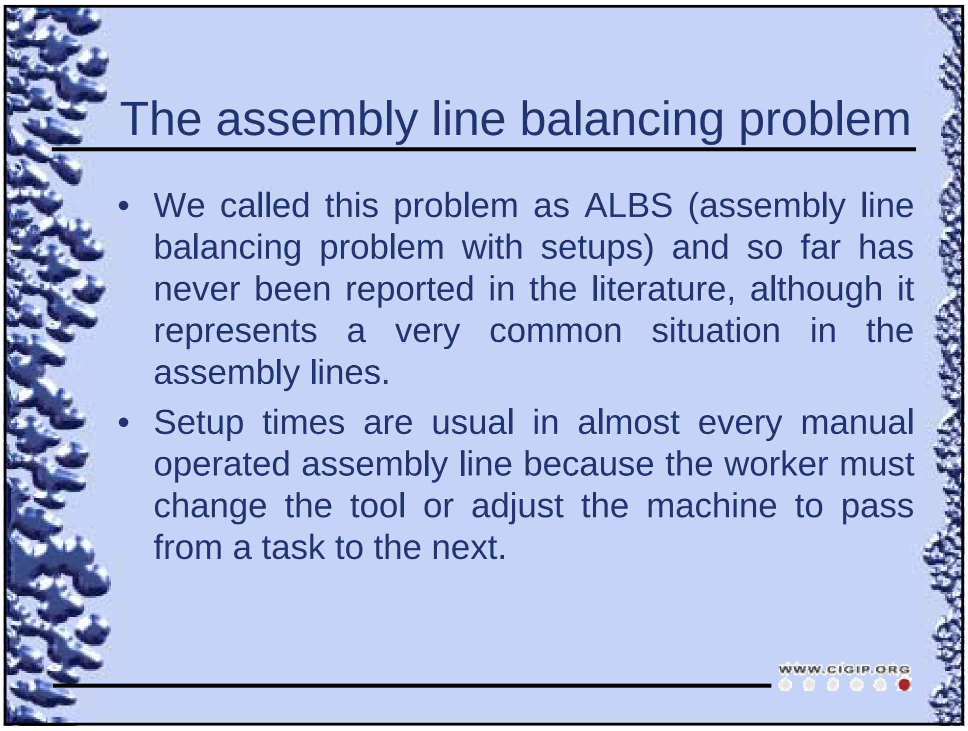
WWW.CIGIP.ORG



The assembly line balancing problem

- The problem presented here adds sequence dependent setup time considerations to SALBP problem in the following way: when a task B is assigned next to the task A at the same workstation, a setup STAB must be added to compute the total operation time. Furthermore, if a task B is the last one assigned to the workstation which has task A the first one assigned, then a setup time STBA must be considered. The objective is minimizing the number of stations for a given cycle time.

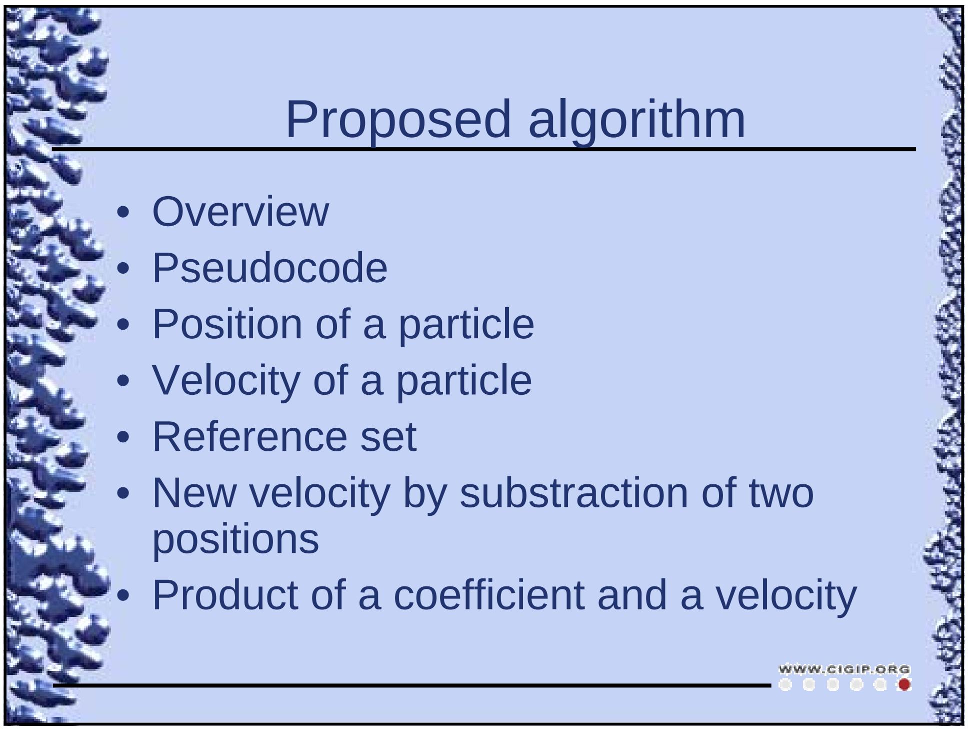
WWW.CIGIP.ORG



The assembly line balancing problem

- We called this problem as ALBS (assembly line balancing problem with setups) and so far has never been reported in the literature, although it represents a very common situation in the assembly lines.
- Setup times are usual in almost every manual operated assembly line because the worker must change the tool or adjust the machine to pass from a task to the next.

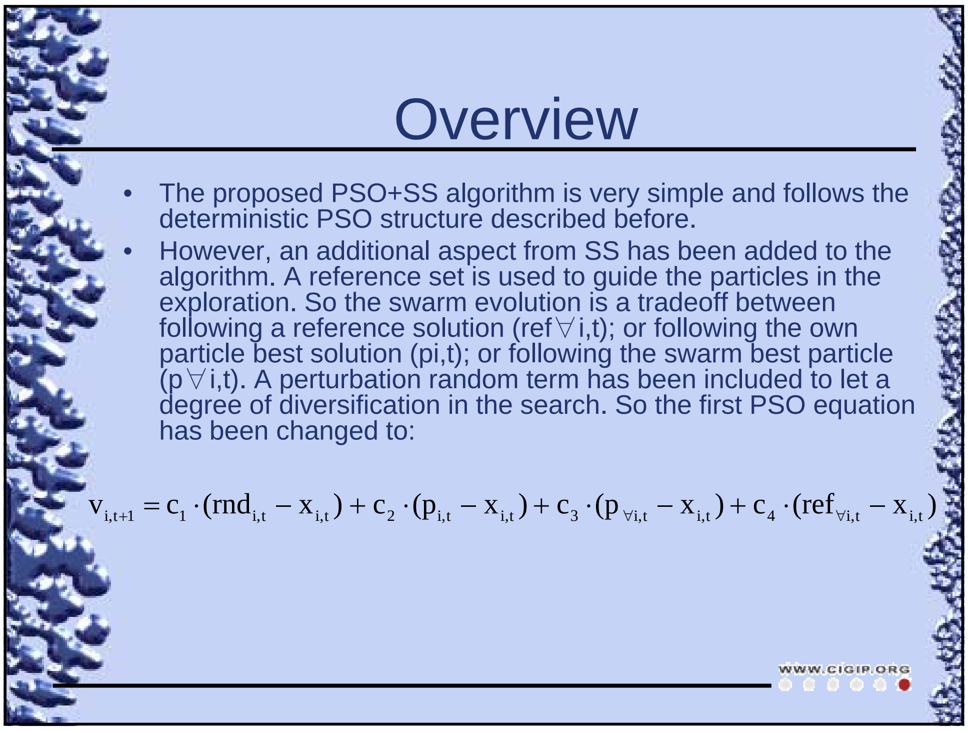
WWW.CIGIP.ORG



Proposed algorithm

- Overview
- Pseudocode
- Position of a particle
- Velocity of a particle
- Reference set
- New velocity by substraction of two positions
- Product of a coefficient and a velocity

WWW.CIGIP.ORG

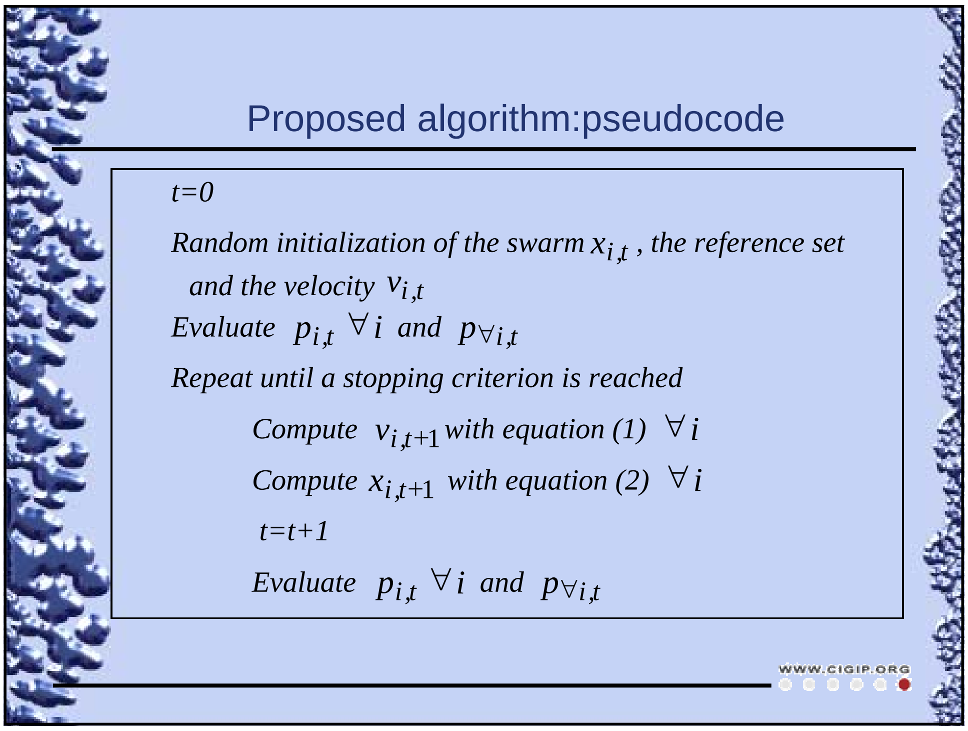


Overview

- The proposed PSO+SS algorithm is very simple and follows the deterministic PSO structure described before.
- However, an additional aspect from SS has been added to the algorithm. A reference set is used to guide the particles in the exploration. So the swarm evolution is a tradeoff between following a reference solution (ref∀i,t); or following the own particle best solution (pi,t); or following the swarm best particle (p∀i,t). A perturbation random term has been included to let a degree of diversification in the search. So the first PSO equation has been changed to:

$$v_{i,t+1} = c_1 \cdot (rnd_{i,t} - x_{i,t}) + c_2 \cdot (p_{i,t} - x_{i,t}) + c_3 \cdot (p_{\forall i,t} - x_{i,t}) + c_4 \cdot (ref_{\forall i,t} - x_{i,t})$$

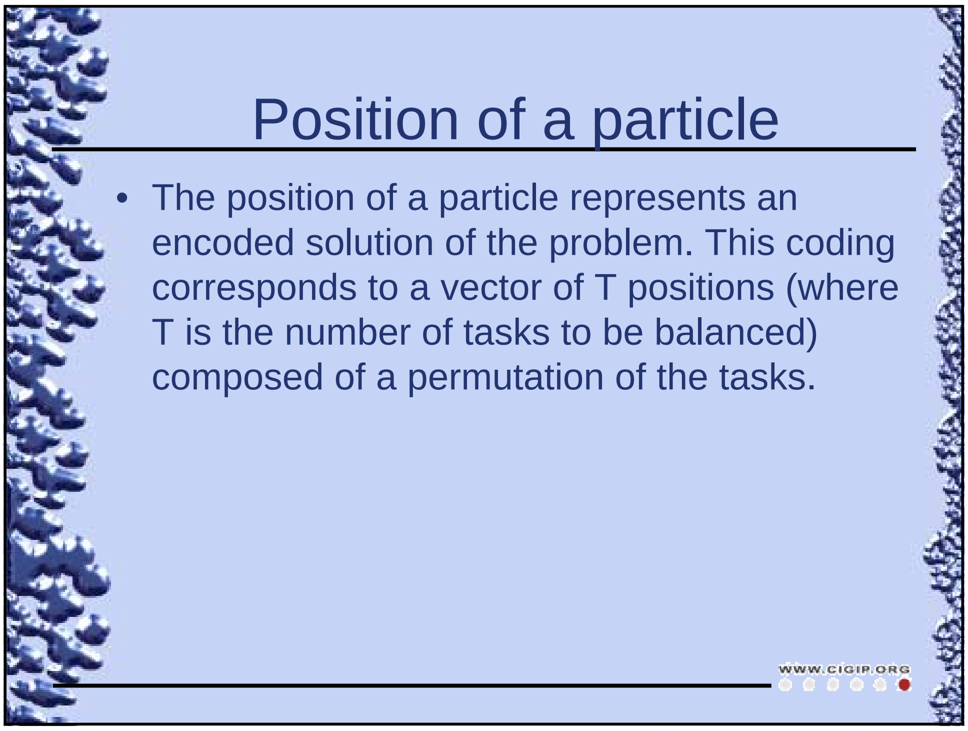
WWW.CIGIP.ORG



Proposed algorithm:pseudocode

```
t=0
Random initialization of the swarm $x_{i,t}$ , the reference set
  and the velocity $v_{i,t}$
Evaluate $p_{i,t}$ $\forall i$ and $p_{\forall i,t}$
Repeat until a stopping criterion is reached
    Compute $v_{i,t+1}$ with equation (1) $\forall i$
    Compute $x_{i,t+1}$ with equation (2) $\forall i$
    t=t+1
    Evaluate $p_{i,t}$ $\forall i$ and $p_{\forall i,t}$
```

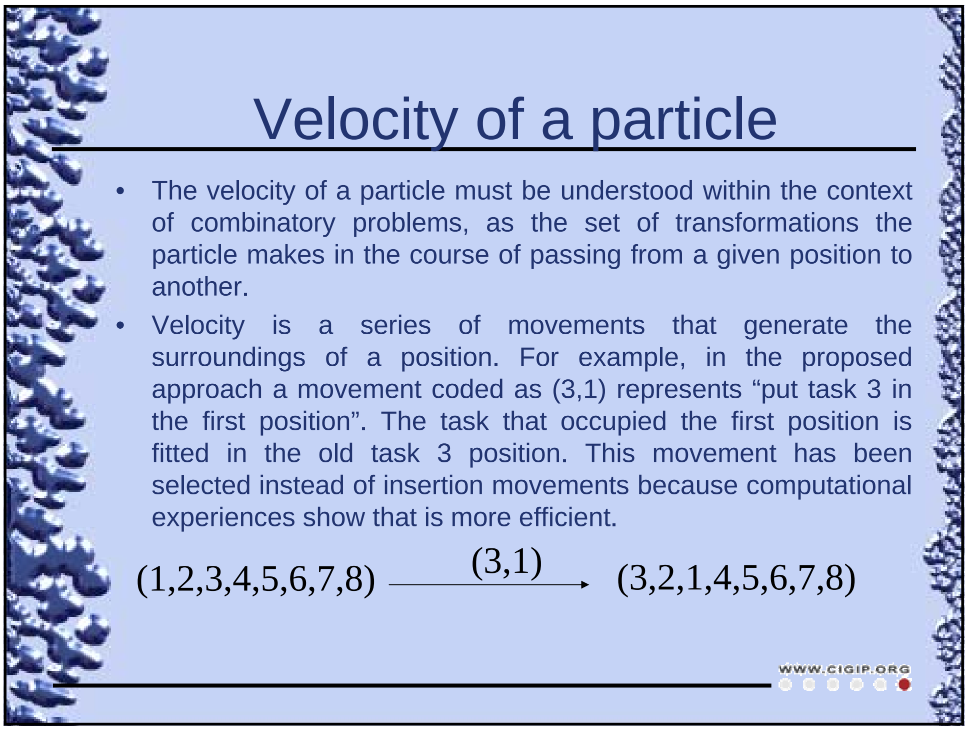
WWW.CIGIP.ORG



Position of a particle

- The position of a particle represents an encoded solution of the problem. This coding corresponds to a vector of T positions (where T is the number of tasks to be balanced) composed of a permutation of the tasks.

WWW.CIGIP.ORG

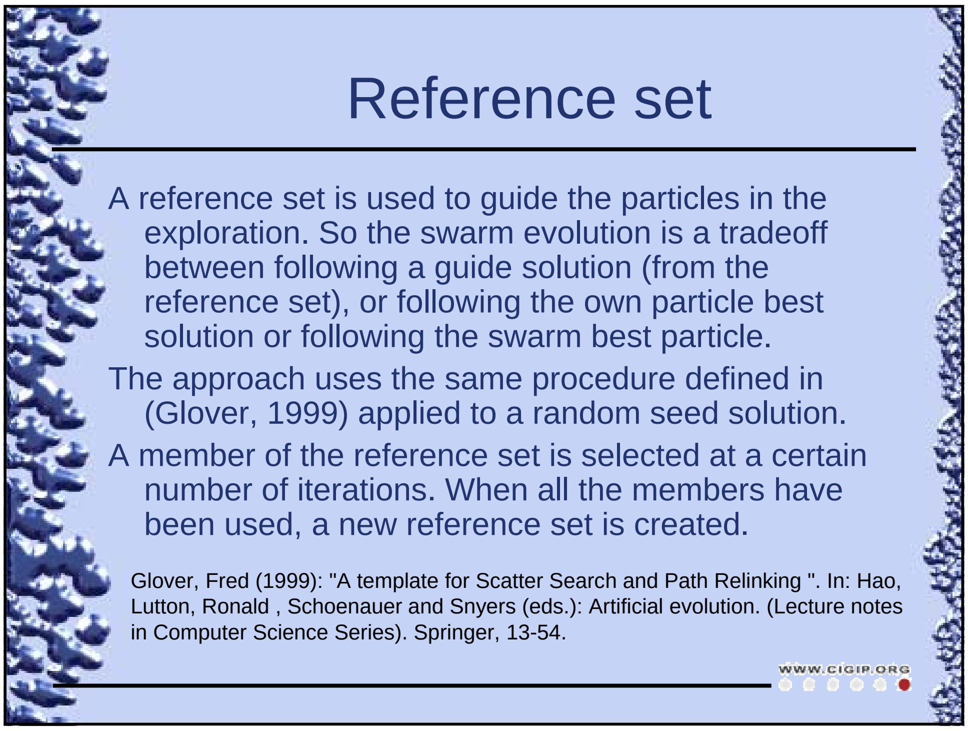


Velocity of a particle

- The velocity of a particle must be understood within the context of combinatory problems, as the set of transformations the particle makes in the course of passing from a given position to another.
- Velocity is a series of movements that generate the surroundings of a position. For example, in the proposed approach a movement coded as (3,1) represents “put task 3 in the first position”. The task that occupied the first position is fitted in the old task 3 position. This movement has been selected instead of insertion movements because computational experiences show that is more efficient.

$$(1,2,3,4,5,6,7,8) \xrightarrow{(3,1)} (3,2,1,4,5,6,7,8)$$

WWW.CIGIP.ORG



Reference set

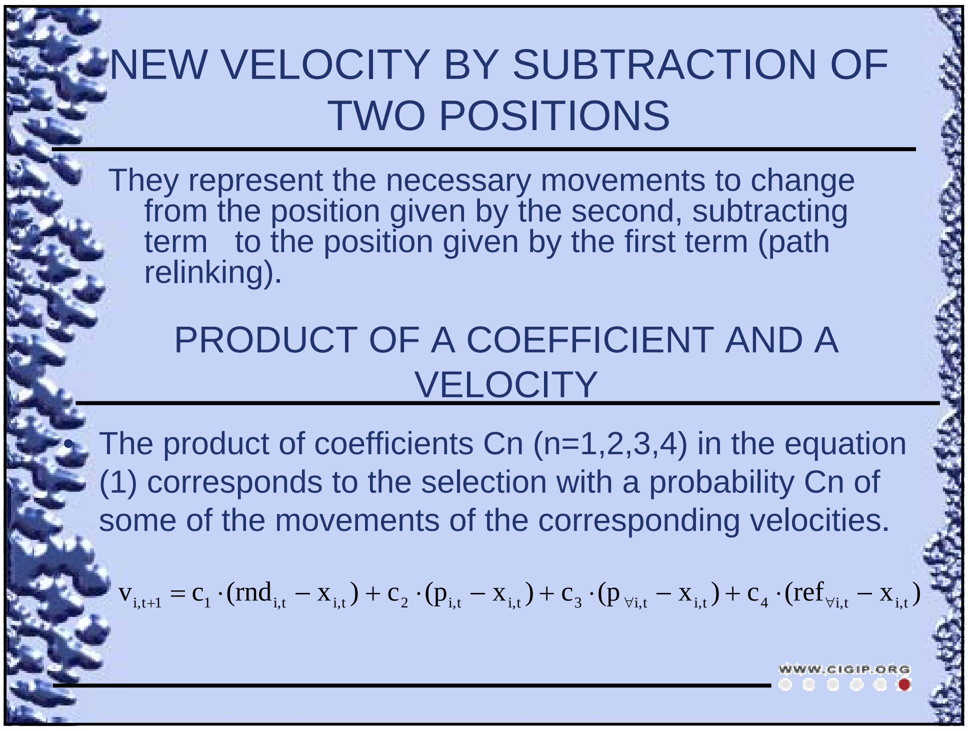
A reference set is used to guide the particles in the exploration. So the swarm evolution is a tradeoff between following a guide solution (from the reference set), or following the own particle best solution or following the swarm best particle.

The approach uses the same procedure defined in (Glover, 1999) applied to a random seed solution.

A member of the reference set is selected at a certain number of iterations. When all the members have been used, a new reference set is created.

Glover, Fred (1999): "A template for Scatter Search and Path Relinking ". In: Hao, Lutton, Ronald , Schoenauer and Snyers (eds.): Artificial evolution. (Lecture notes in Computer Science Series). Springer, 13-54.

WWW.CIGIP.ORG



NEW VELOCITY BY SUBTRACTION OF TWO POSITIONS

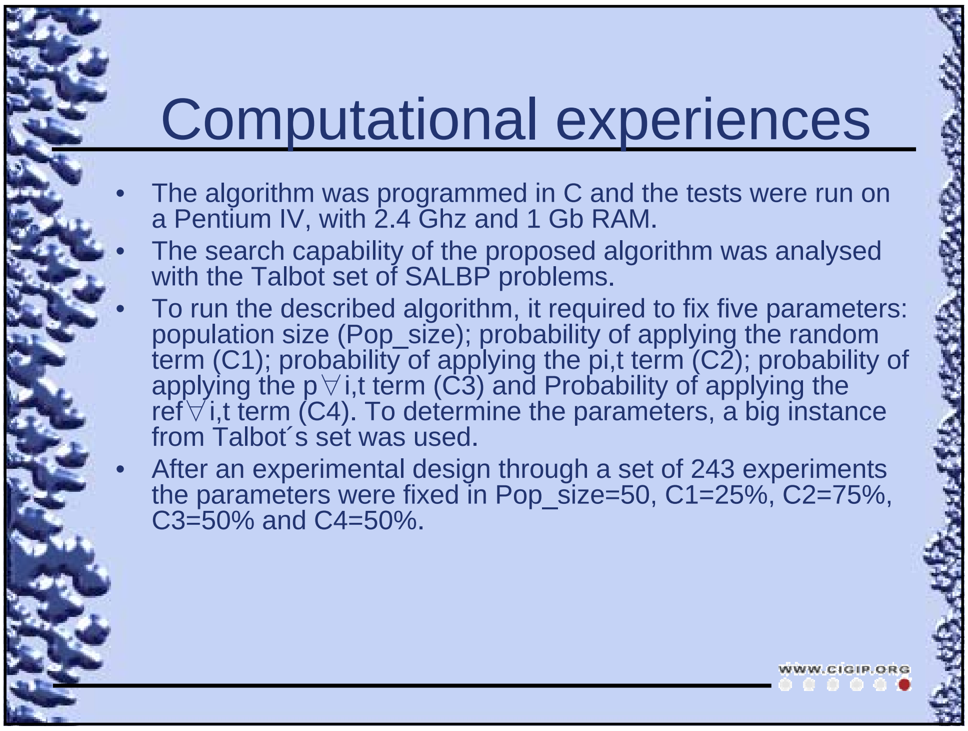
They represent the necessary movements to change from the position given by the second, subtracting term to the position given by the first term (path relinking).

PRODUCT OF A COEFFICIENT AND A VELOCITY

- The product of coefficients Cn (n=1,2,3,4) in the equation (1) corresponds to the selection with a probability Cn of some of the movements of the corresponding velocities.

$$v_{i,t+1} = c_1 \cdot (rnd_{i,t} - x_{i,t}) + c_2 \cdot (p_{i,t} - x_{i,t}) + c_3 \cdot (p_{\forall i,t} - x_{i,t}) + c_4 \cdot (ref_{\forall i,t} - x_{i,t})$$

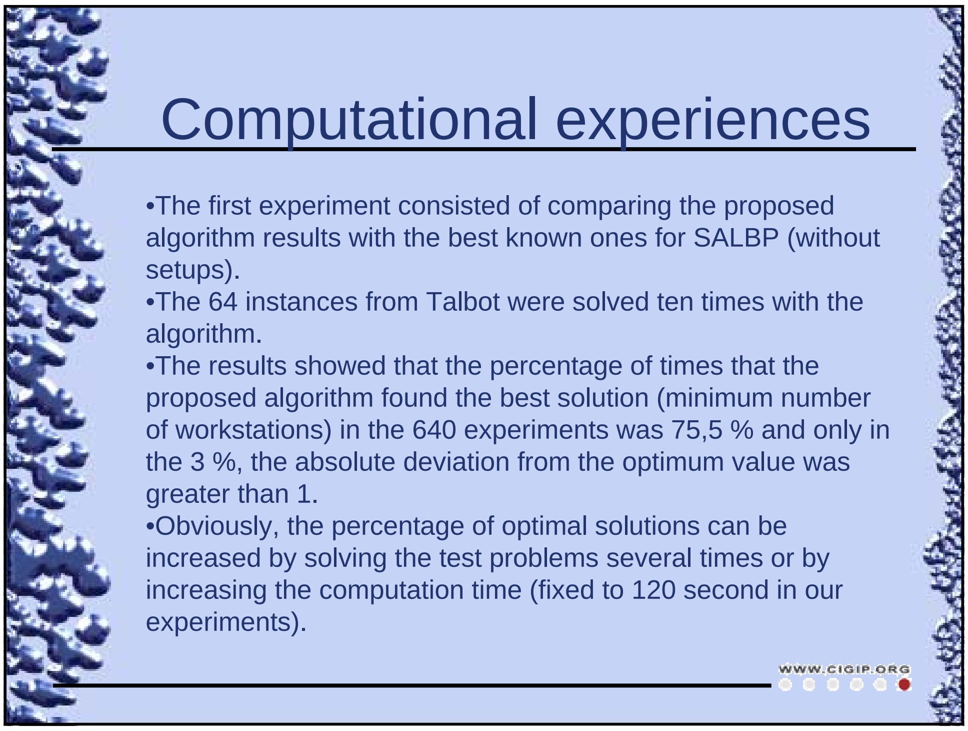
WWW.CIGIP.ORG



Computational experiences

- The algorithm was programmed in C and the tests were run on a Pentium IV, with 2.4 Ghz and 1 Gb RAM.
- The search capability of the proposed algorithm was analysed with the Talbot set of SALBP problems.
- To run the described algorithm, it required to fix five parameters: population size (Pop_size); probability of applying the random term (C1); probability of applying the pi,t term (C2); probability of applying the p∀i,t term (C3) and Probability of applying the ref∀i,t term (C4). To determine the parameters, a big instance from Talbot´s set was used.
- After an experimental design through a set of 243 experiments the parameters were fixed in Pop_size=50, C1=25%, C2=75%, C3=50% and C4=50%.

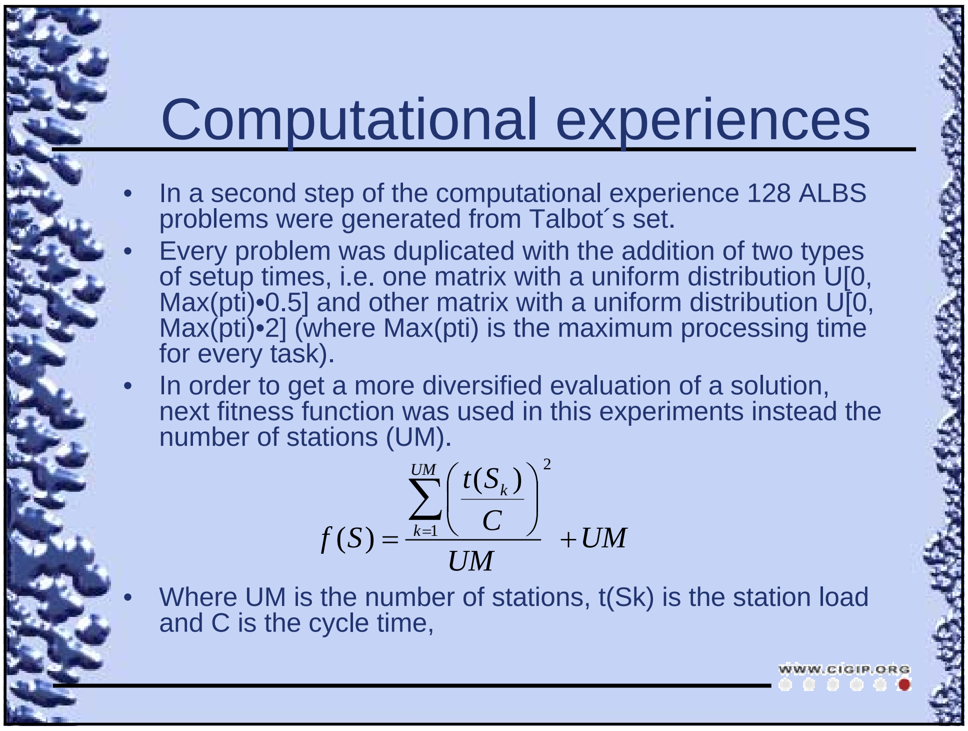
WWW.CIGIP.ORG



Computational experiences

•The first experiment consisted of comparing the proposed algorithm results with the best known ones for SALBP (without setups).
•The 64 instances from Talbot were solved ten times with the algorithm.
•The results showed that the percentage of times that the proposed algorithm found the best solution (minimum number of workstations) in the 640 experiments was 75,5 % and only in the 3 %, the absolute deviation from the optimum value was greater than 1.
•Obviously, the percentage of optimal solutions can be increased by solving the test problems several times or by increasing the computation time (fixed to 120 second in our experiments).

WWW.CIGIP.ORG



Computational experiences

- In a second step of the computational experience 128 ALBS problems were generated from Talbot´s set.
- Every problem was duplicated with the addition of two types of setup times, i.e. one matrix with a uniform distribution U[0, Max(pti)•0.5] and other matrix with a uniform distribution U[0, Max(pti)•2] (where Max(pti) is the maximum processing time for every task).
- In order to get a more diversified evaluation of a solution, next fitness function was used in this experiments instead the number of stations (UM).

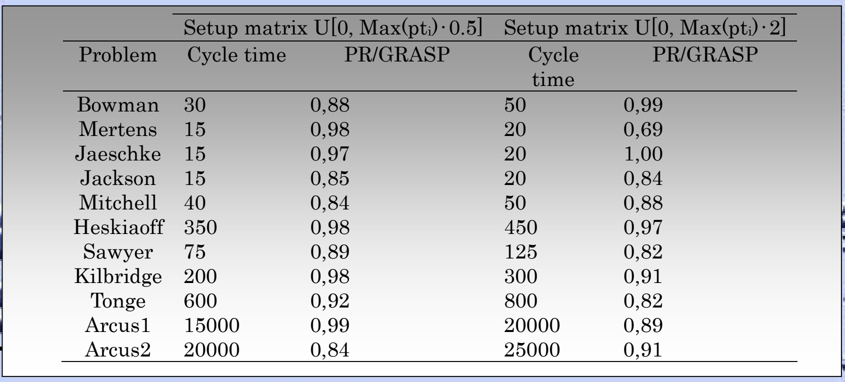
$$f(S) = \frac{\sum_{k=1}^{UM} \left(\frac{t(S_k)}{C} \right)^2}{UM} + UM$$

- Where UM is the number of stations, t(Sk) is the station load and C is the cycle time,

WWW.CIGIP.ORG

Computational experiences

- Every instance was solved with the proposed algorithm and a GRASP algorithm. For every problem instance, the algorithms were applied 10 times.
- Next table shows the ratio between the average of the values obtained by the path relinking procedure and the GRASP algorithm:

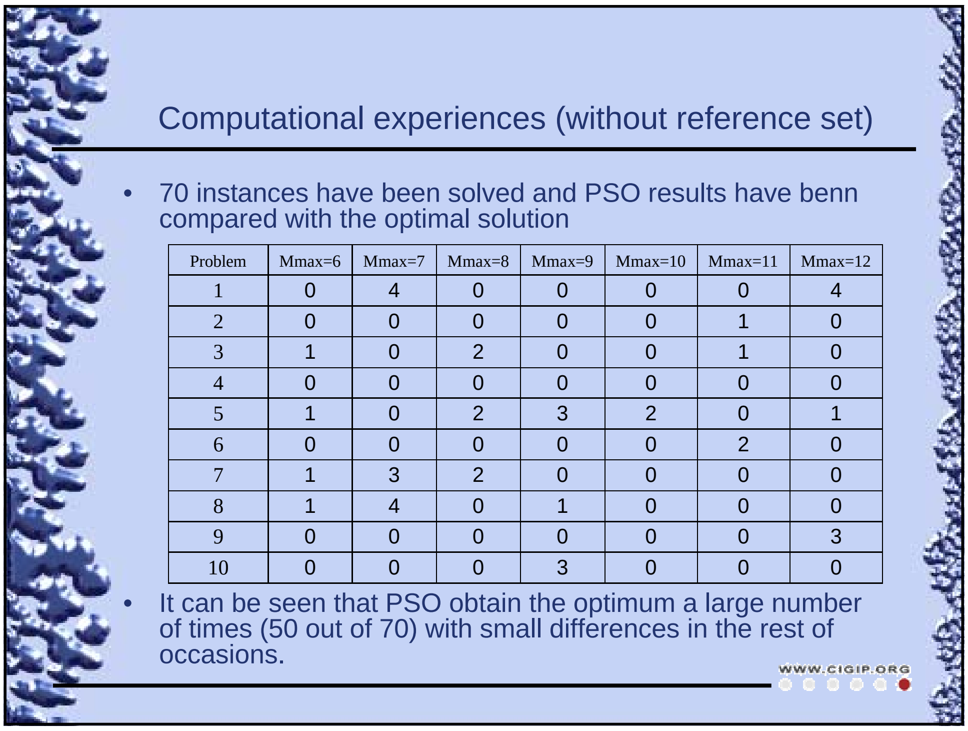


	Setup matrix $U[0, Max(pt_i)\cdot 0.5]$		Setup matrix $U[0, Max(pt_i)\cdot 2]$	
Problem	Cycle time	PR/GRASP	Cycle time	PR/GRASP
Bowman	30	0,88	50	0,99
Mertens	15	0,98	20	0,69
Jaeschke	15	0,97	20	1,00
Jackson	15	0,85	20	0,84
Mitchell	40	0,84	50	0,88
Heskiaoff	350	0,98	450	0,97
Sawyer	75	0,89	125	0,82
Kilbridge	200	0,98	300	0,91
Tonge	600	0,92	800	0,82
Arcus1	15000	0,99	20000	0,89
Arcus2	20000	0,84	25000	0,91

Other experiences: the part machine grouping problem

- The cell formation problem involves grouping similar parts into part-families and machines into clusters of manufacturing resources called manufacturing cells.
- Each family is processed by one and only one of the manufacturing cells.
- The machine-part cell formation problem can be stated as follows: given a set of machines, a set of parts, and a part-machine incidence matrix; assign parts and machines to a fixed number of part families and associated manufacturing cells, so that the cell-coupling (measured by the number of out-of-cell operations or intercellular movements) is minimized and each cell does not contain more than a specified maximum number of machines.

www.cigip.org



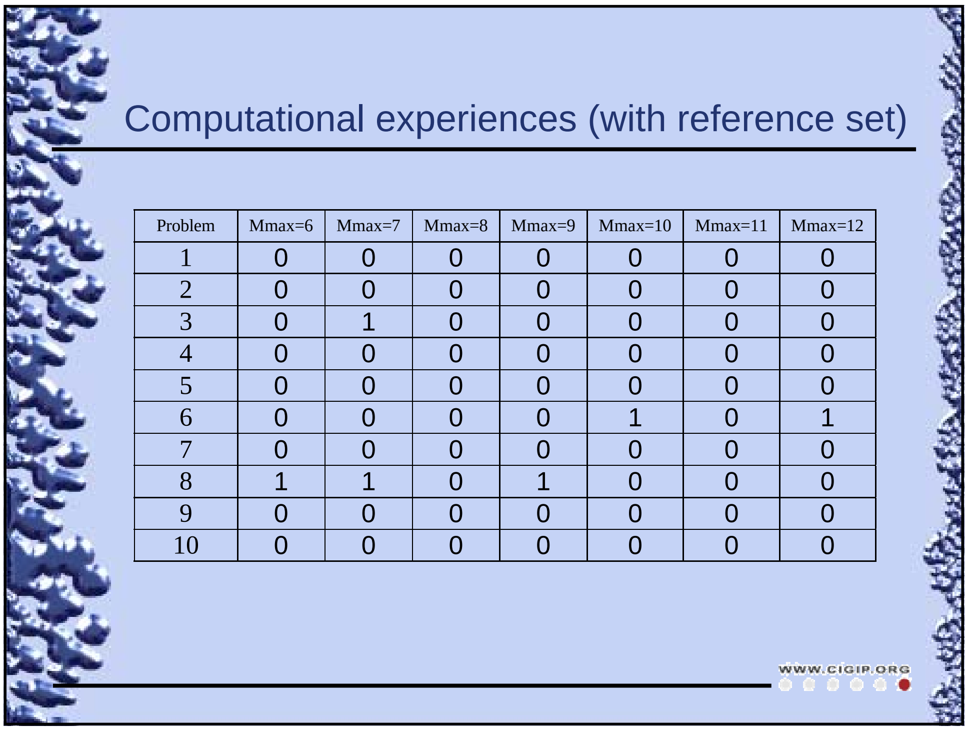
Computational experiences (without reference set)

- 70 instances have been solved and PSO results have benn compared with the optimal solution

Problem	Mmax=6	Mmax=7	Mmax=8	Mmax=9	Mmax=10	Mmax=11	Mmax=12
1	0	4	0	0	0	0	4
2	0	0	0	0	0	1	0
3	1	0	2	0	0	1	0
4	0	0	0	0	0	0	0
5	1	0	2	3	2	0	1
6	0	0	0	0	0	2	0
7	1	3	2	0	0	0	0
8	1	4	0	1	0	0	0
9	0	0	0	0	0	0	3
10	0	0	0	3	0	0	0

- It can be seen that PSO obtain the optimum a large number of times (50 out of 70) with small differences in the rest of occasions.

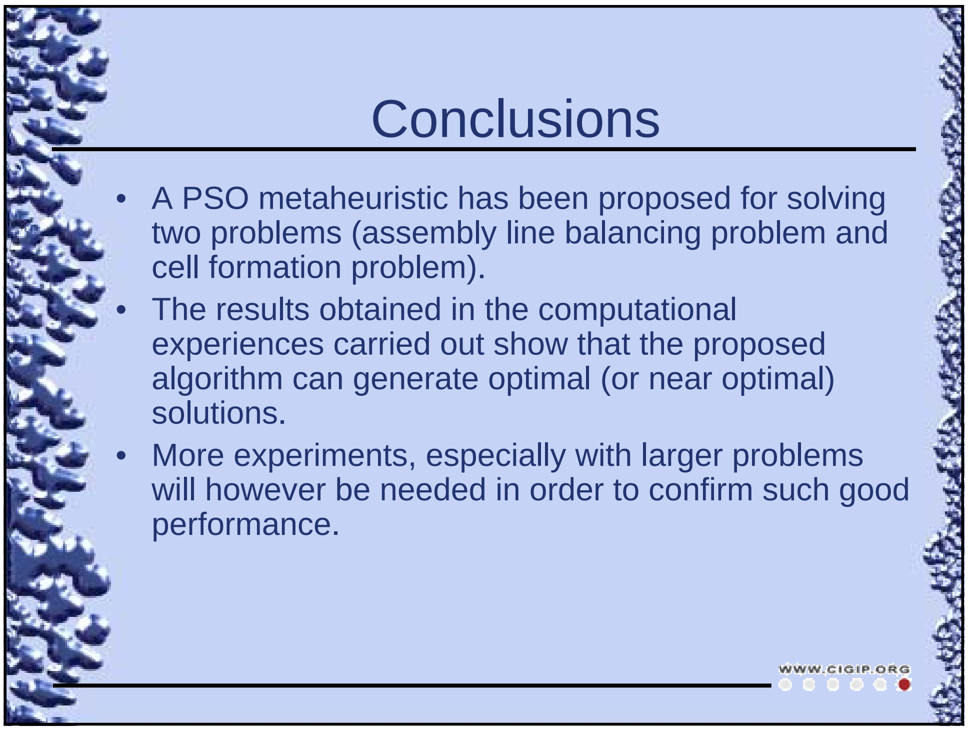
WWW.CIGIP.ORG



Computational experiences (with reference set)

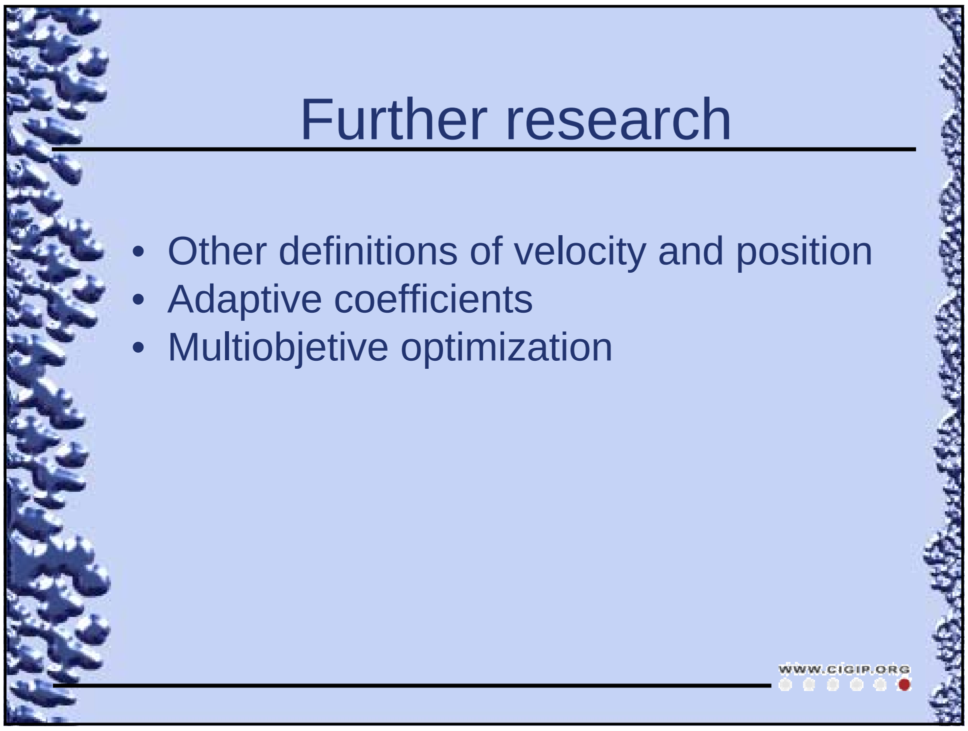
Problem	Mmax=6	Mmax=7	Mmax=8	Mmax=9	Mmax=10	Mmax=11	Mmax=12
1	0	0	0	0	0	0	0
2	0	0	0	0	0	0	0
3	0	1	0	0	0	0	0
4	0	0	0	0	0	0	0
5	0	0	0	0	0	0	0
6	0	0	0	0	1	0	1
7	0	0	0	0	0	0	0
8	1	1	0	1	0	0	0
9	0	0	0	0	0	0	0
10	0	0	0	0	0	0	0

WWW.CIGIP.ORG



Conclusions

- A PSO metaheuristic has been proposed for solving two problems (assembly line balancing problem and cell formation problem).
- The results obtained in the computational experiences carried out show that the proposed algorithm can generate optimal (or near optimal) solutions.
- More experiments, especially with larger problems will however be needed in order to confirm such good performance.



Further research

- Other definitions of velocity and position
- Adaptive coefficients
- Multiobjetive optimization

Thanks for your attention

WWW.CIGIP.ORG