

*The 15th International Conference on Flexible Automation and
Intelligent Manufacturing, July 18-20, 2005, University of
Deusto, Bilbao*

A particle swarm optimization algorithm for machine grouping problems

Carlos Andrés Romano¹, Sebastián Lozano Segura²

¹candres@omp.upv.es, CIGIP Research Center
Valencia University of Technology

²slozano@us.es, Industrial Management Department
University of Seville

WWW.CIGIP.ORG

Presentation outline

- Introduction
- Particle swarm optimization algorithms
- The machine-part grouping problem
- Proposed algorithm
- Computational experiences
- Conclusions and further research

WWW.CIGIP.ORG

Introduction

- Particle Swarm Optimization (PSO) algorithm is a population metaheuristic method used in different continuous domain problems.
- Applications to discrete domain functions are scarce.
- Our aim is to develop a PSO algorithm to solve a combinatorial optimization problem in the design of cellular manufacturing systems.

WWW.CCIP.ORG

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 a 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.CCIP.ORG

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.

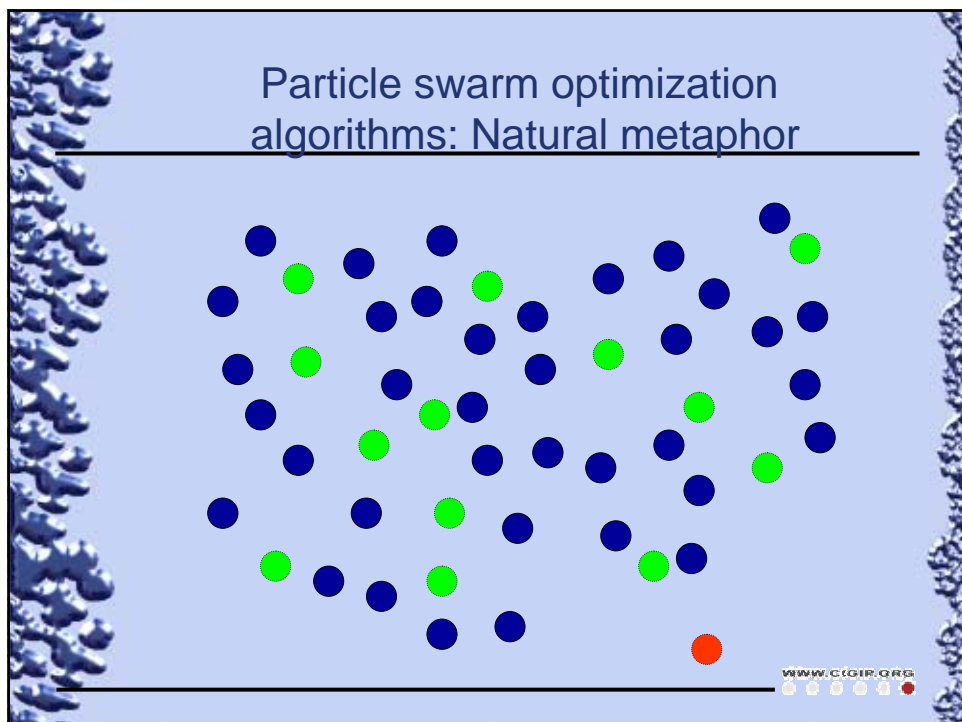
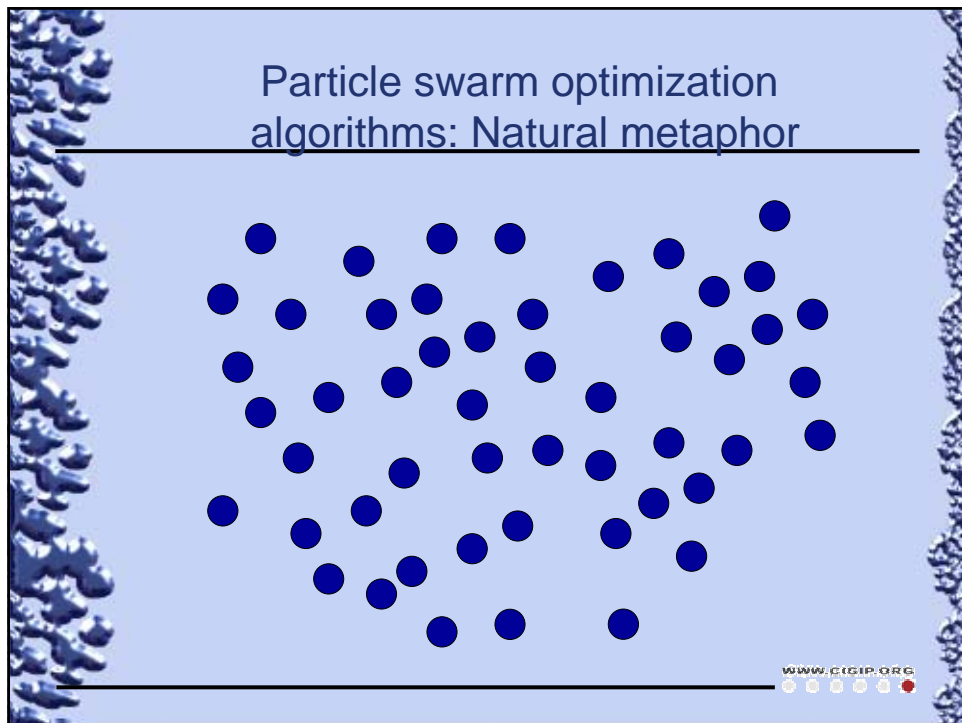
- In this aspect, PSO algorithms present some similarities with algorithms based in ant colonies (ACO). The main difference between ACO and PSO algorithms is the method used to memorize solutions previously visited and the procedure to generate new solutions (constructive in ACO versus path relinking 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 may use path relinking, it is not a population based method, so it does not get the benefits derived from the information interchange.

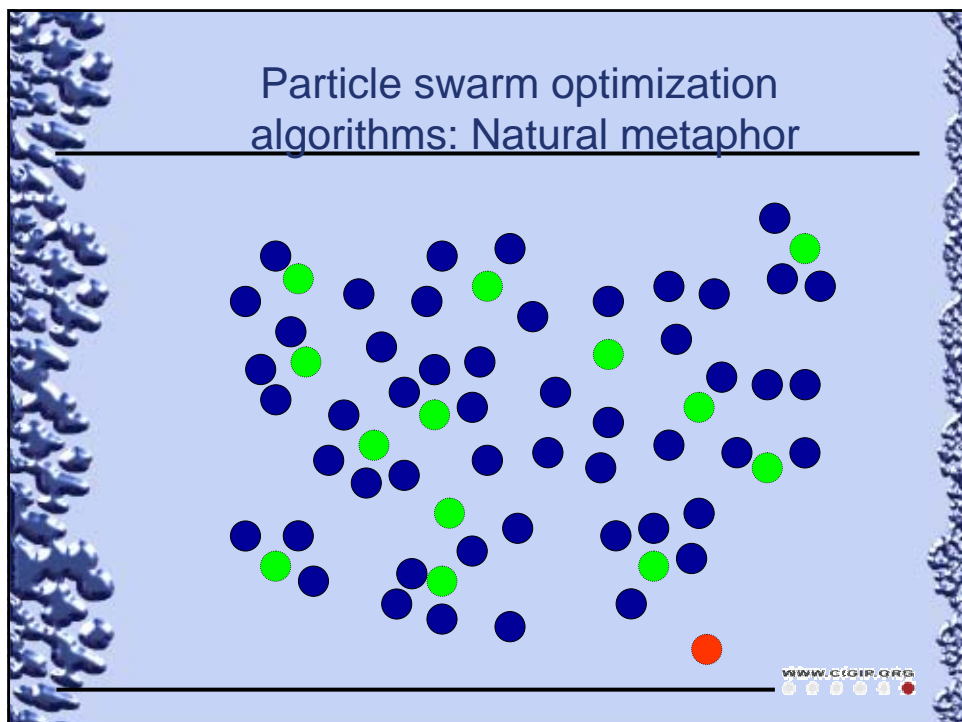
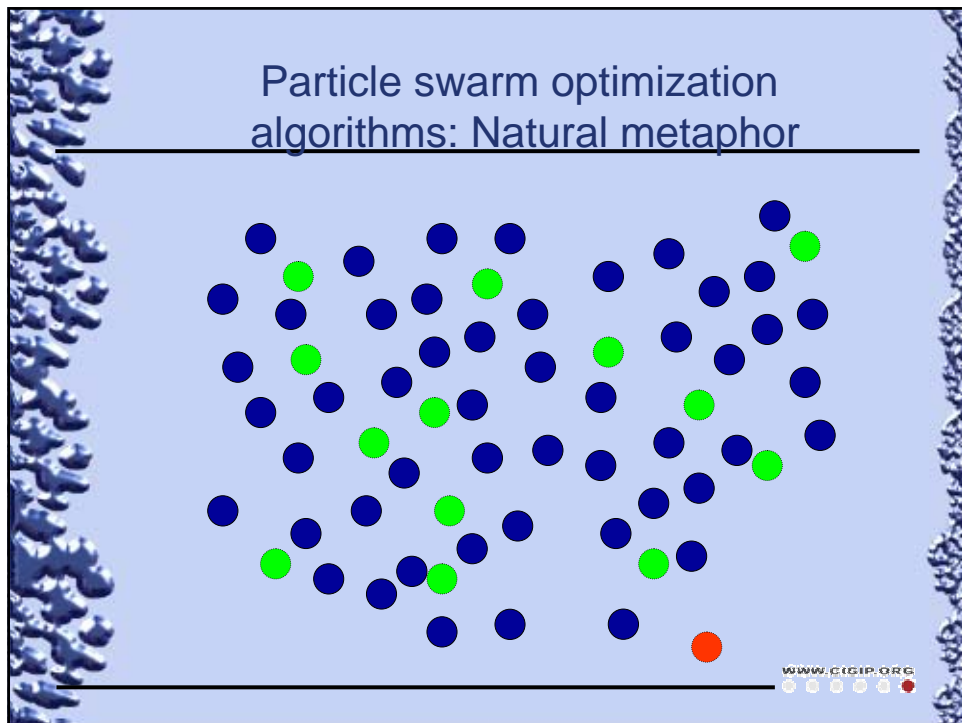
WWW.CCIP.ORG

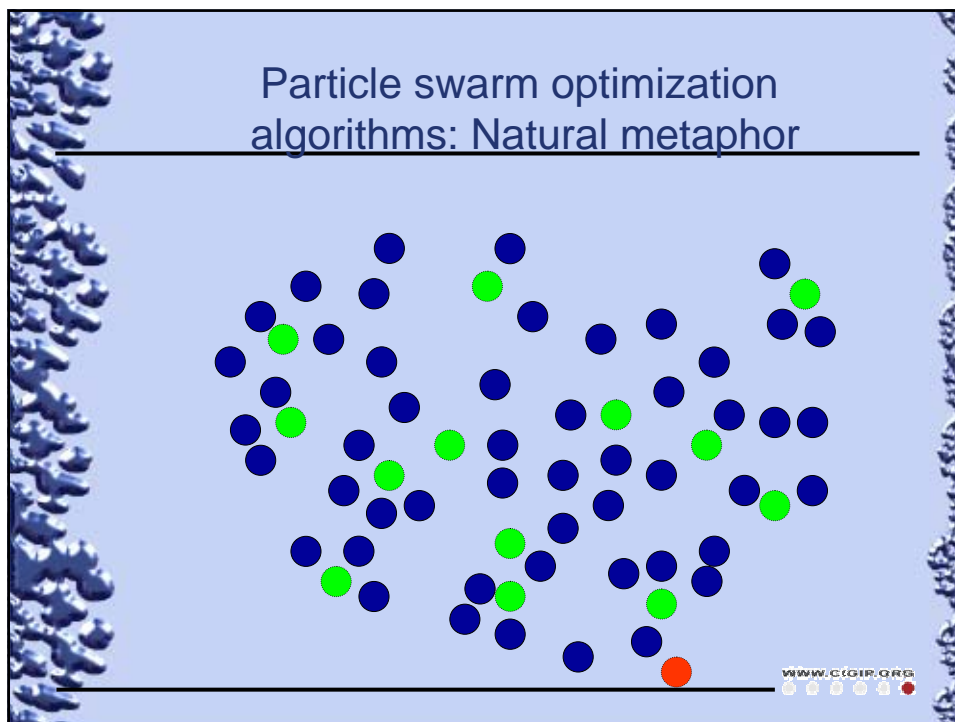
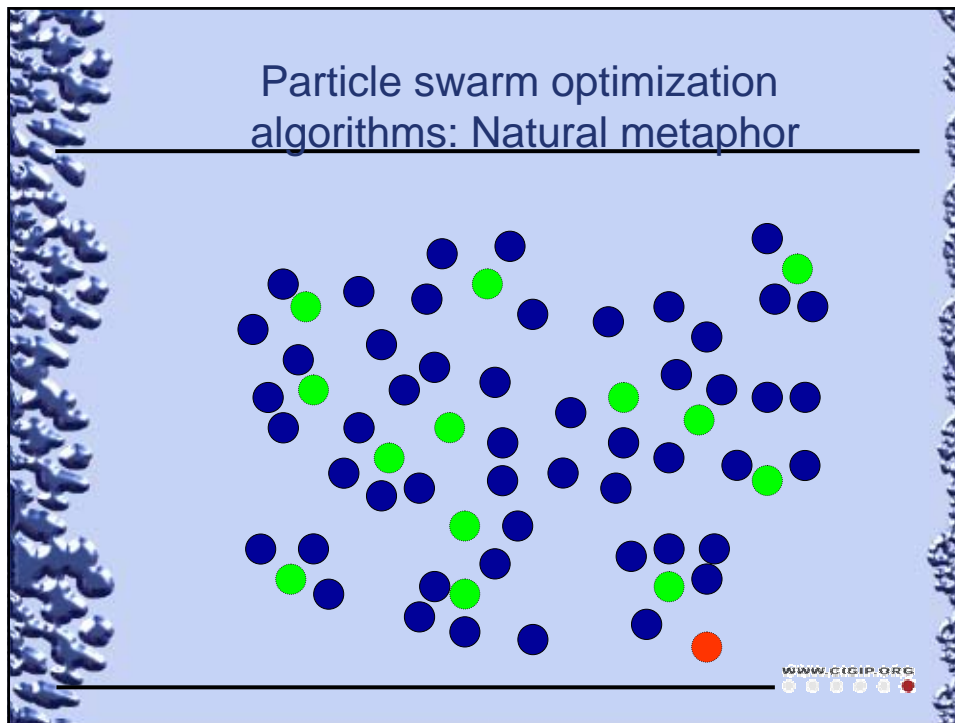
Particle swarm optimization algorithms: Principles

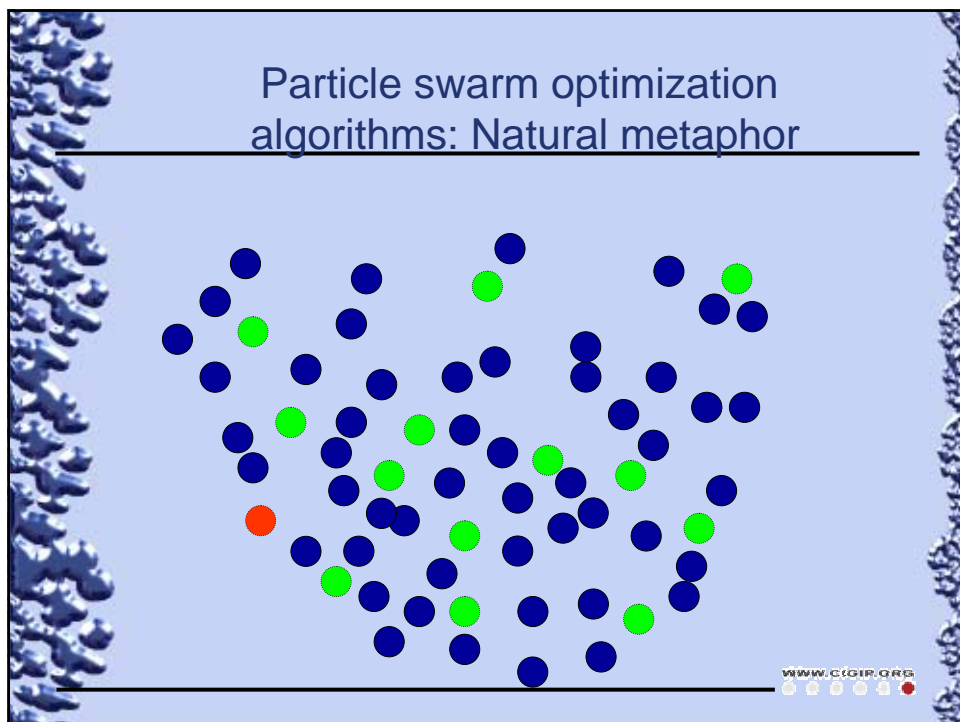
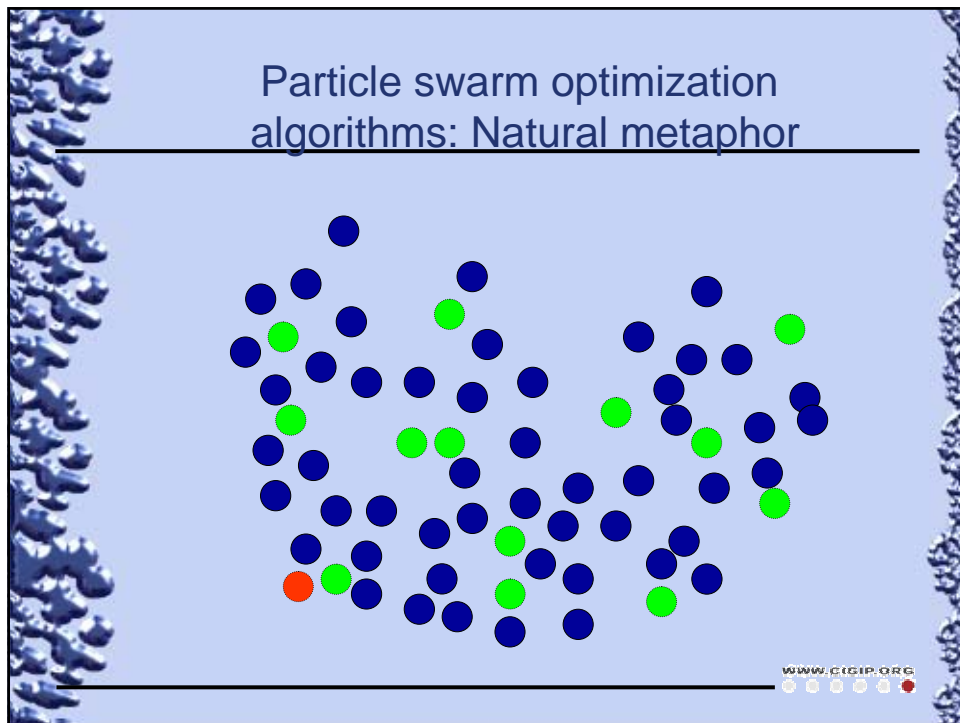
- The principles that govern PSO algorithms can be stated in the following characteristics:
 - Each particle has a position (solution) and a velocity (change pattern of the solution).
 - Every particle knows its position and the value of the objective function for that position.
 - It also remembers 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.

WWW.CCIP.ORG

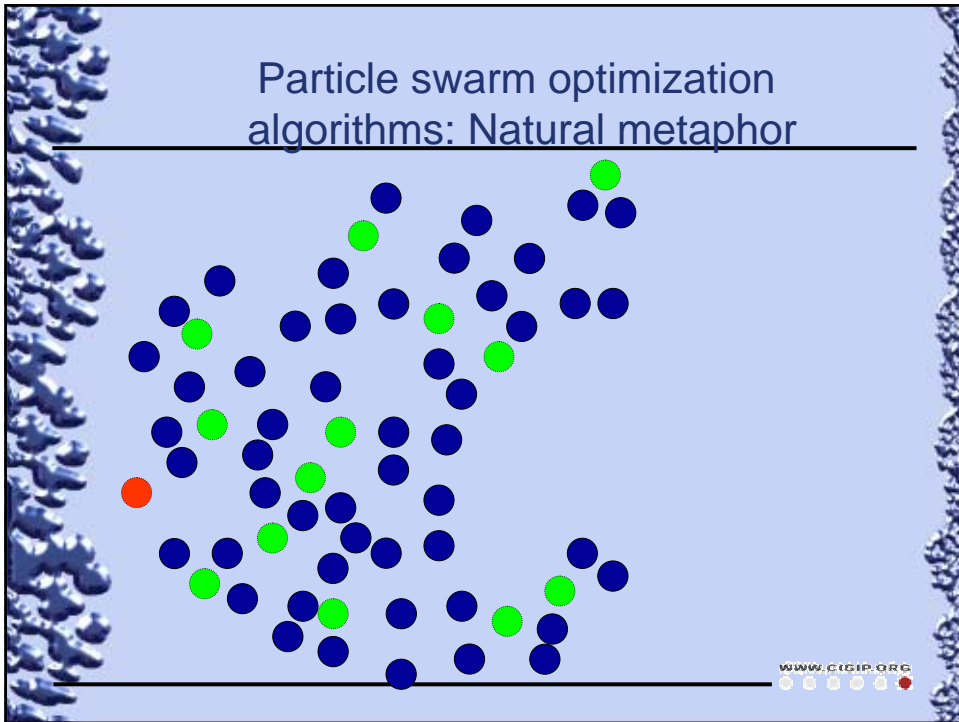








Particle swarm optimization algorithms: Natural metaphor



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.CCIP.ORG

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).

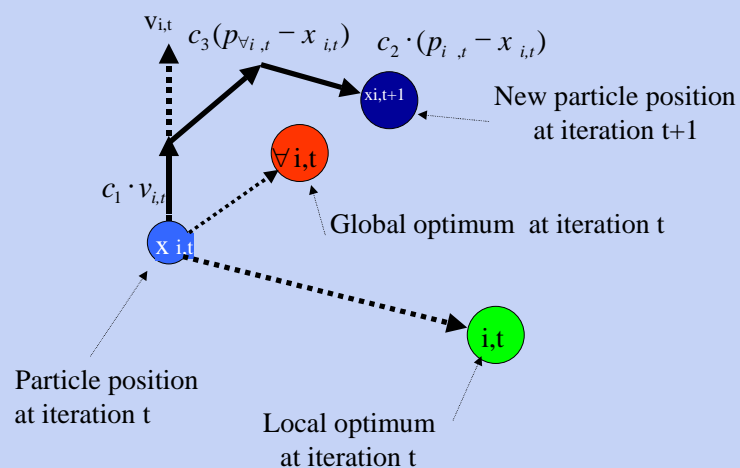
$p_{i,t}$ Best previous position of particle i at iteration t (which is memorized by every particle).

$p_{\forall i,t}$ 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 $v_{i,t}$, $p_{i,t}$ and $p_{\forall i,t}$

WWW.CCIP.ORG

Particle swarm optimization algorithms: Equations



WWW.CCIP.ORG

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.
- Most techniques use only the information available on the part-route sheets. Part-route sheets indicate the sequence of facilities or machines used to process each part. From it, we can construct a binary matrix A , called part-machine incidence matrix, where $a_{ij} = 1$ means that machine i is required to process part j and $a_{ij} = 0$ otherwise.

WWW.CEIP.ORG

The part-machine grouping problem

- 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.
- Cell coupling occurs due to parts requiring machines that belong to a different cell. We call any operation included in the part route sheet which has to be performed outside the assigned cell an "exceptional element".
- For a given cell configuration, the degree of interaction between cells can be measured by the number of exceptional elements.

WWW.CEIP.ORG

The part-machine grouping problem: Mathematical formulation.

- M Number of machines
- P Number of Parts
- C Number of Cells
- i Index of machines ($i=1,\dots,M$)
- j Index of parts ($j=1,\dots,P$)
- k Index of cells ($k=1,\dots,C$)
- $A=[a_{ij}]$ MxP binary machine-part incidence matrix
- M_{\max} Maximum number of machines per cell

WWW.CEIP.ORG

The part-machine grouping problem: Mathematical formulation.

$$y_{ik} = \begin{cases} 1 & \text{if machine } i \in \text{cell } k \\ 0 & \text{otherwise} \end{cases}$$

$$z_{jk} = \begin{cases} 1 & \text{if part } j \in \text{family } k \\ 0 & \text{otherwise} \end{cases}$$

- Minimize: $\sum_{k=1}^C \sum_{i=1}^M \sum_{j=1}^P a_{ij} z_{jk} (1 - y_{ik})$

$$\begin{aligned} \sum_{k=1}^C y_{ik} &= 1 & \forall i & & \sum_{i=1}^M y_{ik} &\leq M_{\max} & \forall k \\ \sum_{k=1}^C z_{jk} &= 1 & \forall j & \end{aligned}$$

WWW.CEIP.ORG

The part-machine grouping problem: Mathematical formulation.

- This integer program has a quadratic objective function and two sets of variables (y_{ik} and z_{jk}).
- To solve it, only one set of variables (for example y_{ik}) need to be directly searched since given the values of those variables, the optimal values of the other set can be easily computed.
- That is, given the machine cells partition the assignment of parts to each family is trivial: each part is assigned to the family/cell that contains the majority of the machines that part requires.
- If some of the machines required by a part type are not in that cell, they correspond to exceptional elements and are added to compute the objective function (to minimize).

WWW.CEIP.ORG

Proposed algorithm

- Pseudocode
- Position of a particle
- Velocity of a particle
- New velocity by subtraction of two positions
- Product of a coefficient and a velocity

WWW.CEIP.ORG

Proposed algorithm:pseudocode

$t=0$

Random initialization of the swarm $x_{i,t}$ 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 $v_{i,t+1} \forall i$

 Compute $x_{i,t+1} \forall i$

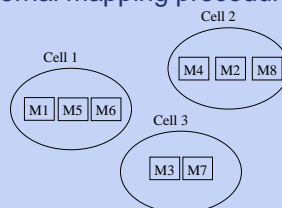
$t=t+1$

 Evaluate $p_{i,t} \forall i$ and $p_{\forall i,t}$

WWW.CCIP.ORG

Position of a particle

- The position of a particle represents an encoded solution of the problem. This encoding corresponds to a vector of M positions. Each component of that vector takes a value between 1 and C which represents the cell to which the machine is assigned.
- Note that due to the type of codification used, the same solution may correspond to different positions. Thus, the positions (1,2,3,2,1,1,3,2), (2,1,3,1,2,1,3,1), (1,3,2,3,1,1,2,3) represent all the same solution to the problem. To take this into account, an appropriate internal mapping procedure has been developed.



WWW.CCIP.ORG

Velocity of a particle

- The velocity of a particle must be understood in the context of a combinatorial optimization problem as an ordered set of transformations on a solution, i.e. a velocity is a succession of changes or movements.
- In the proposed approach the insertion movement has been selected because computational experiences show that it is more efficient than exchange movements. For example, a movement coded as (3,1) represents changing machine 3 to cell 1. Applied to a certain solution, not all movements are feasible since some of them may violate the maximum cell size.

$$(1,2,3,2,1,1,3,2) \xrightarrow{(3,1)} (1,2,1,2,1,1,3,2)$$

WWW.CCIP.ORG

NEW VELOCITY BY SUBTRACTION OF TWO POSITIONS

It represents the necessary movements to change from the position given by the second term to the position given by the first term (path relinking).

PRODUCT OF A COEFFICIENT AND A VELOCITY

- The coefficients c_n ($n=1,2,3$) correspond to the probability of selection of the movements of the corresponding velocities.

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

WWW.CCIP.ORG

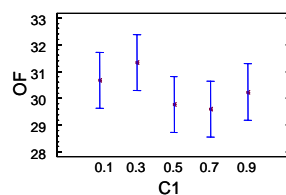
Computational experiences

- The swarm size has been fixed to 20 and the algorithm processing time has fixed to 2 seconds.
- An experimental design has been generated. Six possible values have been tested for every coefficient c_n : 0.10, 0.30, 0.50, 0.70 and 0.90.
- The experiments consisted in one execution of the algorithm for every instance and cell size. After running the experiments, the statistical package Statgraphics 10™ has been used to perform an analysis of the experiments. The figures show the medium value and 95% confidence intervals for the different levels of the three parameters.

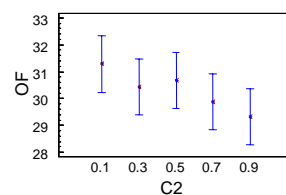
WWW.CCIP.ORG

Computational experiences

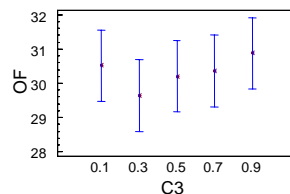
Means and 95 Percent Confidence Intervals



Means and 95 Percent Confidence Intervals



Means and 95 Percent Confidence Intervals



WWW.CCIP.ORG

Computational experiences

- 70 instances have been solved and PSO results have been compared with the known 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 obtains the optimum a large number of times (50 out of 70) with small differences in the rest of occasions.

WWW.CCIP.ORG

Extensions

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 particle best solution or following the swarm best particle. A perturbation random term has been included to let a degree of diversification in the search. So PSO equation changes to

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

WWW.CCIP.ORG

Computational experiences

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

- It can be seen that PSO obtains the optimum a large number of times (64 out of 70) with minimum differences in the rest of occasions.

WWW.CCIP.ORG

Conclusions

- A PSO metaheuristic has been proposed for solving the 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.

WWW.CCIP.ORG

Further research

- Other update equations for velocity and position
- Dynamic coefficients
- Multiobjective optimization

WWW.CCIP.ORG

Thank you

WWW.CCIP.ORG