METAHEURISTICS FOR THE OPTIMIZATION IN PLANNING AND SCHEDULING

PIERRE BORNE

p.borne@ec-lille.fr

METAHEURISTICS FOR THE OPTIMIZATION IN PLANNING AND SCHEDULING

- Introduction
- Tabu Search
- Simulated Annealing
- Genetic Algoritms
- Ant Colony Optimization
- Particle Swarm Optimization
- Tunneling Algorithms
- Pareto Optimality
- Conclusion

INTRODUCTION

- Importance of Planning and Scheduling
- NP Complex problems
- Exact approaches
    - Branch and Bound
    - Dynamic Programming
- Metaheuristics
    - Local searches
        - Tabu search
        - Hill climbing approach
    - Global searches
        - Simulated Annealing
        - Genetic Algorithms
        - Ant Colony Optimization
        - Particle Swarm Optimization
        - Tunneling Algorithms

NP HARD PROBLEMS

Problems in Planning and Scheduling are NP hard problems and the study of all the possibilities leads to combinatory explosion

For example if we want to check all the possibilities of order of N persons entering in a room, checking 1.000.000.000 possibilities per second

For N = 10 it needs 3.6 ms

For N = 20 it needs 77 years

For N = 30 it needs 8.4 $*10^{15}$ years

8.4 million of millions of millenniums

TABU SEARCH

PRINCIPLE

Iterative local search

Move to the best solution in the immediate neighbourhood

Interdiction to go back to the N previous solutions (Tabu list)

TABU SEARCH

EXAMPLE

12	13	10	7	8	5	9	6	7	10	3	4	3
11	16	14	17	16	18	16	14	13	8	5	2	7
8	13	12	13	10	12	17	16	13	6	3	6	8
7	17	13	11	10	16	13	16	14	13	8	9	7
9	6	16	14	6	18	16	15	13	11	7	10	9
13	11	14	13	16	12	14	16	17	15	14	17	10

SIMULATED ANNEALING

IDEA

- **Global optimization in a large search space**
- **Inspiration from annealing in metallurgy with heating and controlled cooling**

PRINCIPLE

- **At each step of calculation the current solution s is replaced by a nearly one chosen with a probability which depends of the variation of a fitness function e (Energy) via a parameter T (temperature) which gradually decrease during the process**
- **The solution change almost randomly for T large and tends globally to obtain the minimum of energy when T tends to zero**

SIMULATED ANNEALING

THE ALGORITHM

s: state, T: temperature, e: energy, k: time

Initialization

$s=s_0$, $e= E(s)$, $k=0$

while $k < k_m$ and $e > e_m$

s_{k+1}= neighbour (s_k)

$e_{k+1}= E(s_{k+1})$

if random $< P(e_k, e_{k+1}, T_k)$

Then $s_k:=s_{k+1}$, $e_k:= e_{k+1}$, $k:= k+1$

Return while

GENETIC ALGORITHMS

PRESENTATION

- Iterative algorithm to optimize a fitness function
- Based on biological evolution of a population
- A chromosome represents an individual (solution)
- Evolution by crossover, mutation and selection
- The individual with the best fitness have the greatest probability to be selected for the reproduction

GENETIC ALGORITHMS

CROSSOVER

GENETIC ALGORITHMS

MUTATION

Example 1

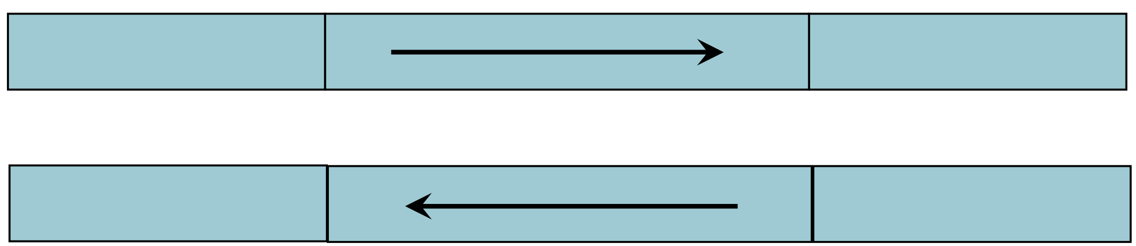
Before

After

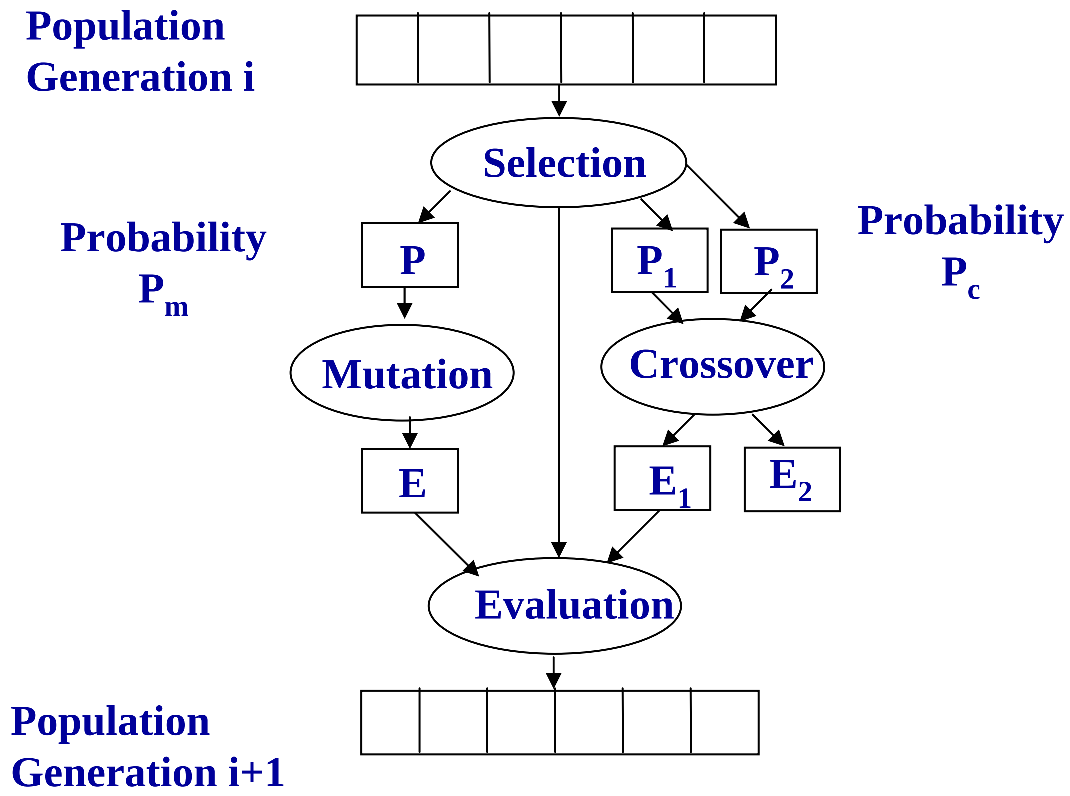
Example 2

Before

After



GENETIC ALGORITHMS



GENETIC ALGORITHMS

EXAMPLE OF CODING FOR PLANNING AND SCHEDULING

O_{ij} : operation i of job j

k_{ij} : machine use to achieve operation O_{ij}

t_{ij} : starting time of the operation O_{ij}

For example for three jobs, two with three operations and one with two operations

O_{11}, k_{11}, t_{11}	O_{21}, k_{21}, t_{21}	O_{31}, k_{31}, t_{31}
O_{12}, k_{12}, t_{12}	O_{22}, k_{22}, t_{22}	O_{32}, k_{32}, t_{32}
O_{13}, k_{13}, t_{13}	O_{23}, k_{23}, t_{23}	

GENETIC ALGORITHMS

EXAMPLE OF CODING FOR PLANNING AND SCHEDULING

The various operations have to satisfy various constraints

- Precedence
- Resource disponibility
- Preemption possible or not
- Earliest starting time
- Due date
- Perisability

For optimizing various criteria

- Makespan
- Maximum workload
- Maximum delay penalty
- …

ARTIFICIAL IMMUNE SYSTEM

Main uses

- **Detection of intrusion in a network**
- **Optimization**

Optimization algorithm inspired of biological immune systems

AIS algorithm

- **Learning memory**
- **Pattern recognition**
- **Hypermutation**
- **Clonal selection**

Genetic algorithm type with a learning capacity (like artificial neural networks)

GRASP: GREEDY RANDOMIZED ADAPTIVE SEARCH PROCEDURE

Mixed approach: Multi-start metaheuristic for combinatorial problems

Two phases:

- Construction of a feasible solution

- Local optimization in the neighbourhood

The best overall solution is kept as the result

ANT COLONY OPTIMIZATION

PRINCIPLE

-Research of the shortest path to find the food

- Parallel search of solution

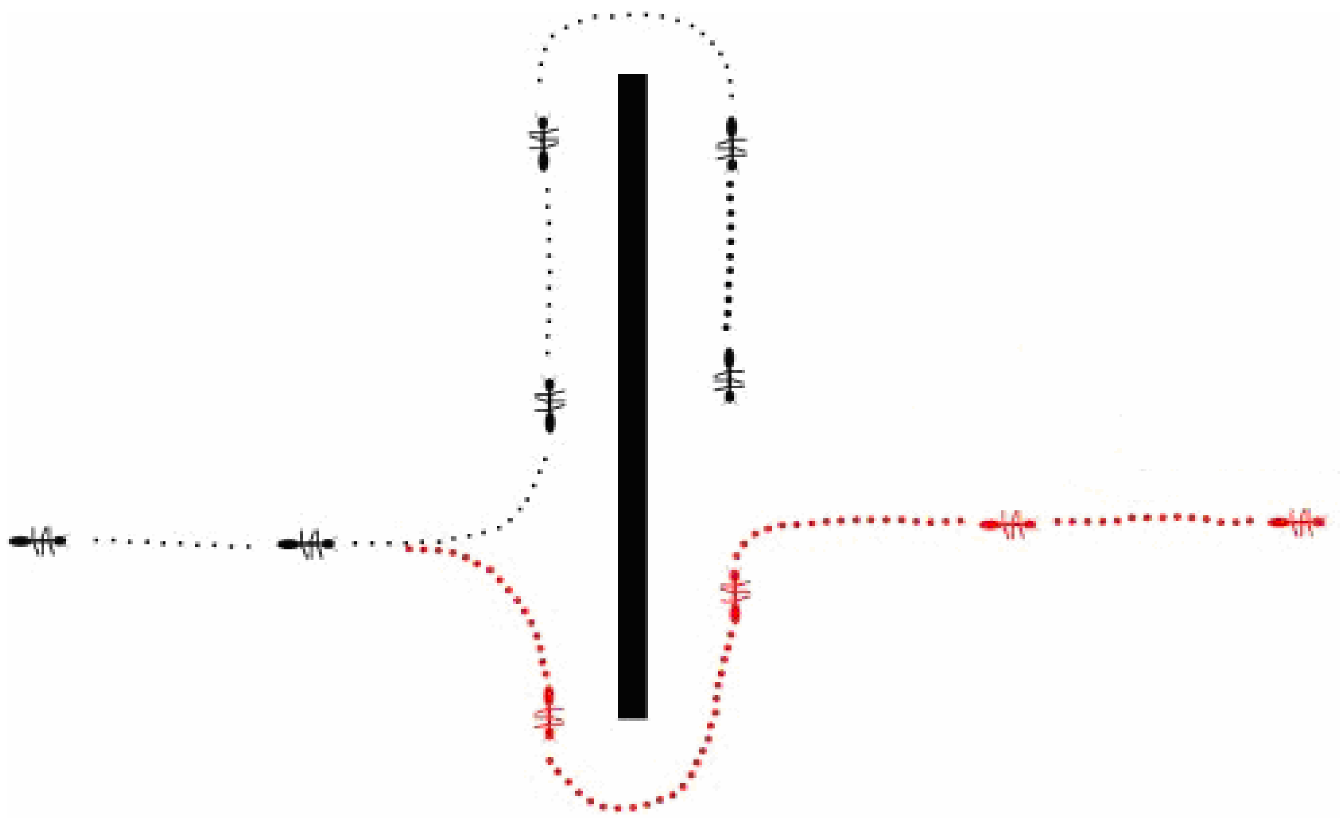
- Dynamic memory via the pheromone trail

- At the beginning : random evolution while laying down pheromone trails

- Strong probability for an ant to follow the path with the greatest pheromone trail reinforcing it

- Slow evaporation of the pheromone in order to avoid to stick on a local optimum

ANT COLONY OPTIMIZATION



ANT COLONY OPTIMIZATION

APPLICATION TO PLANNING AND SCHEDULING

P_{ijk}^{f} : **Probability for the ant *f* to assign the operation i of job j (O_{ij}) to machine k (O_{ijk})**

p_{ijk} : **Processing time of O_{ij} with the machine k**

τ_{ijk} : **Pheromone trail related to O_{ijk}**

D : **Set of non performed operations**

α, β *et* ρ : **Parameters of the algorithm**

ANT COLONY OPTIMIZATION

APPLICATION TO PLANNING AND SCHEDULING

$$\tau_{ijk}(t+1) = \rho.\tau_{ijk}(t) + \sum_{f} \Delta\tau_{ijk}^{f}(t+1)$$

$$\Delta\tau_{ijk}^{f}(t) = \frac{L_{\min}}{L_{ijk}^{f}(t)}$$

$$P_{ijk}^{f} = \frac{\left(\tau_{ijk}\right)^{\alpha} . \left(p_{ijk}\right)^{-\beta}}{\sum_{j \in D} \left(\tau_{ijk}\right)^{\alpha} . \left(p_{ijk}\right)^{-\beta}} \qquad if \quad j \in D$$

$$P_{ijk} = 0 \qquad otherwise$$

ANT COLONY OPTIMIZATION

APPLICATION TO PLANNING AND SCHEDULING

Remark

In order to avoid a premature convergence to a local optimum, a tabu search optimization is usually performed with the ant colony optimization algorithm

PARTICLE SWARM OPTIMIZATION

PRINCIPLE

- Stochastic optimization technique
- Cooperation between agents (the particles)
- Local exchange of information
- Characterization : position and velocity
- Memorization : position of the present performance, position of best own performance, position of the best known performance

PARTICLE SWARM OPTIMIZATION

ALGORITHM

- Initialization : random repartition of the particles
- For each particle : evaluation of the quality of the current position and memorization of the best position visited
- Determination of the best position reached by the swarm
- Evolution of each particle according to the available information

PARTICLE SWARM OPTIMIZATION

ALGORITHM

$x_i(t)$: position of the particle i at time t

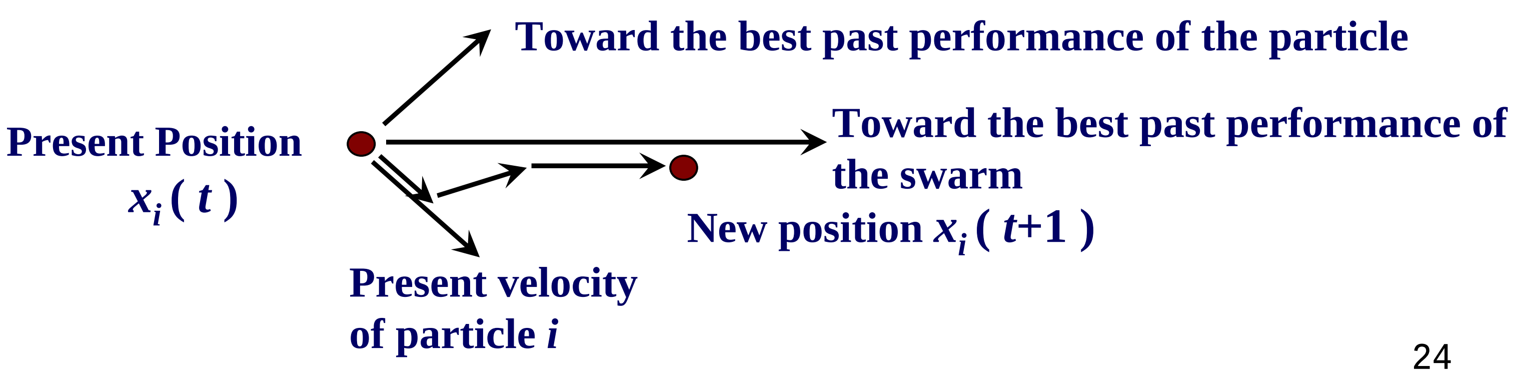
$v_i(t)$: velocity of the particle i at time t

$x_{im}(t)$: best known position reached by the particle i at time t

$x_M(t)$: best known position reached by the swarm at time t

$$x_i(t+1) = f(x_i(t), v_i(t), x_{im}(t), x_M(t))$$

Very often this relation is linear



PARTICLE SWARM OPTIMIZATION

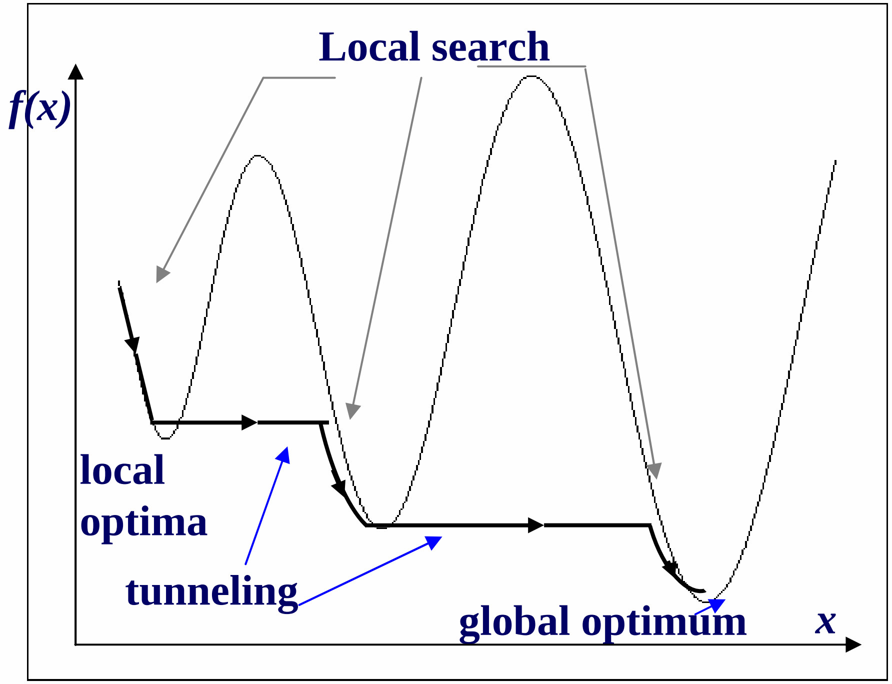
APPLICATION TO PLANNING AND SCHEDULING

$x_i(t)$ needs to represent a solution of the problem at time t

In job shop scheduling problem many variables are discrete, vector $x_i(t)$ must indicate the affectation of the various resources to the various operations with the planning of their action, and the evolution must take into account the various constraints which complexify the implementation of the method

TUNNELING ALGORITHMS

- Enable to escape to local optima
- Two main approaches
    - Stochastic tunneling
    - Tunneling with penalty functions



TUNNELING ALGORITHMS

- Stochastic Tunneling:

Application of a non linear transformation to the objective function

- Tunneling with penalty :

Modification of the value of local optima by adding penalty values in order to escape to these local optima

MULTI OBJECTIVE OPTIMIZATION

Several (n) criteria are considered at the same time

- **Ordered Weighted Averaging (OWA) operator**

If the score of criterion $c_i(x)$ is u_i

$$c_{OWA}(x) = \sum_{i=1}^{n} w_i u_i$$

with $w_i \in [01], \sum_{i=1}^{n} w_i = 1$

- **Choquet Integral**

The weights w_i are calculated according to the interaction between various criteria

- **Pareto optimality**

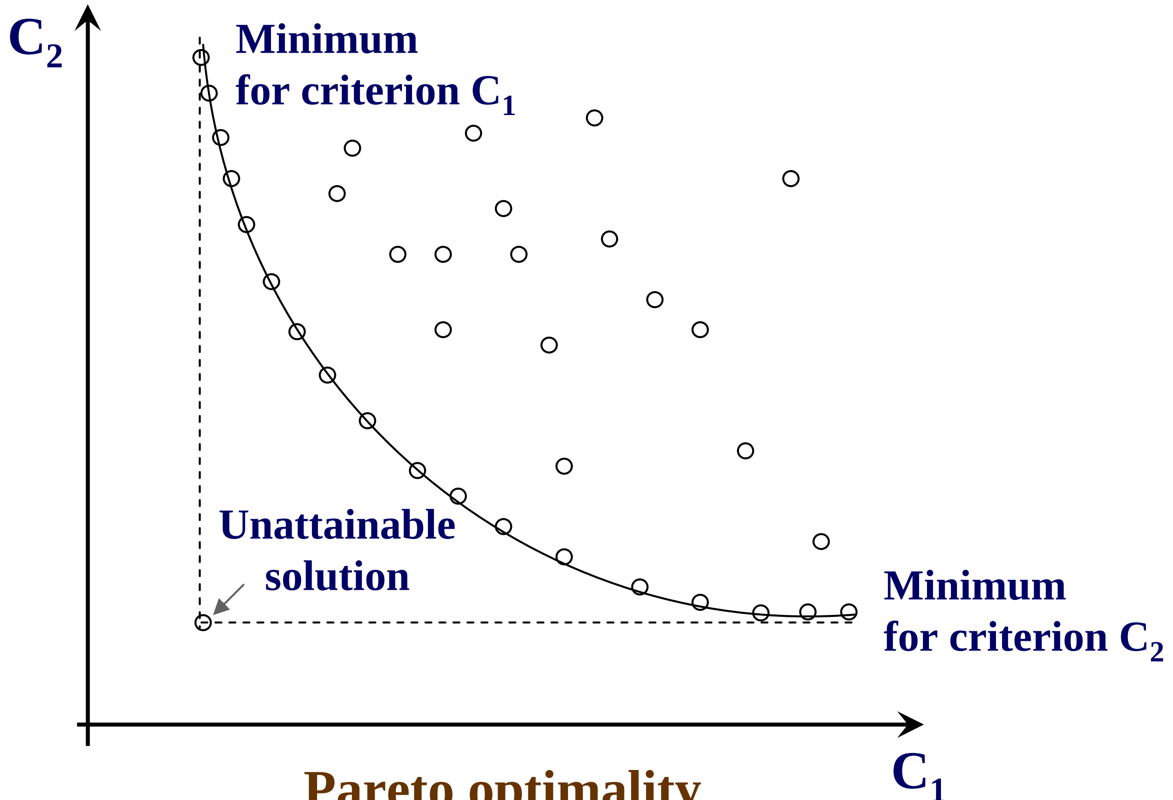
PARETO OPTIMALITY

Measure of efficiency in multicriteria optimization

⇨ **Non dominated solution : there is no other solution that performs at least as well on every criteria and which is strictly better on at least one of the criteria**

⇨ **For a Pareto-optimal solution, a criterion cannot be improved without damaging at least one of the other criteria**

PARETO OPTIMALITY



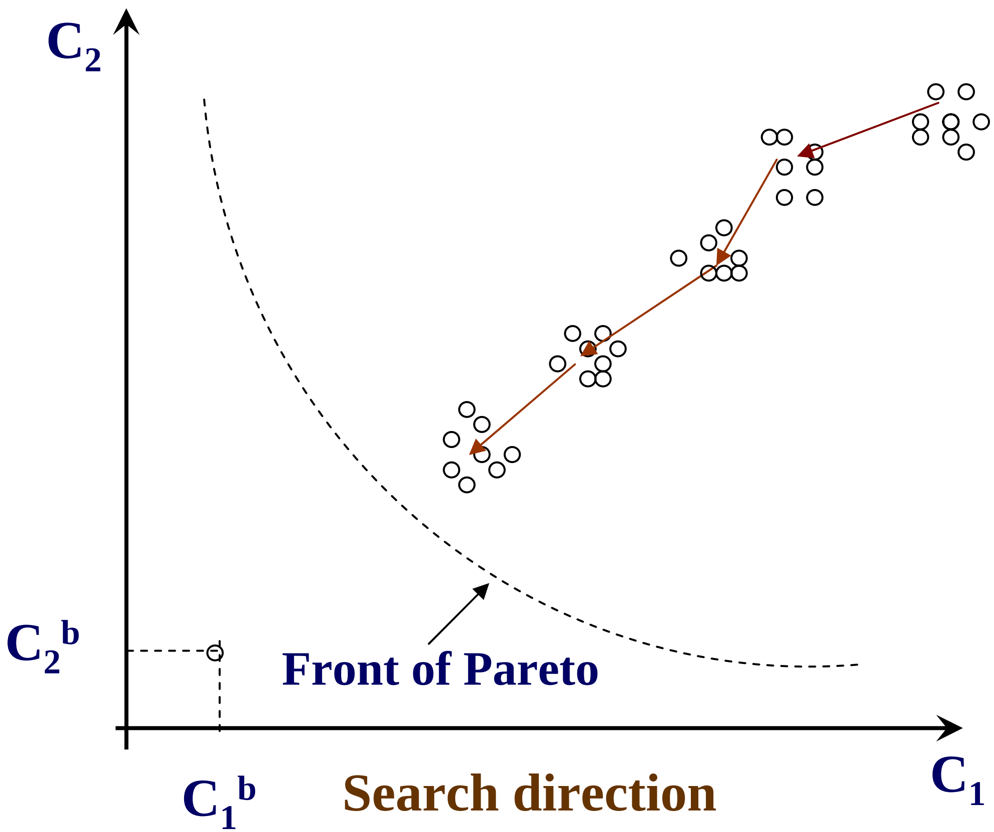
Pareto optimality

PARETO OPTIMALITY

Association with OWA approach

- **Normalization of the criteria**
- **Determination of lower bounds of the criteria**
- **Aggregation of the various criteria with adaptive weights**
- **Dynamic search direction in order to evolve in the direction of the lower bounds point**

PARETO OPTIMALITY



CONCLUSION

- Various metaheuristics
- Implementation in planning and scheduling problems
- Difficulties of implementation due to the discrete nature of some variables
- Interest of hybrid approaches