Mixed Integer Non Linear Porgamming problems with heuristic methods

Prof. François Marechal

IPESE
Industrial Process and Energy Systems Engineering

Non linear multi-objective optimisation

- Real problems are non linear
  - Process unit and Process flowsheet simulation
  - Part load operation
  - Process unit sizing
  - Process unit costing
- Real problem include integer decisions (Yes/No)

MINLP

Mixed Integer Non Linear Programming problems

Solving MINLP problems : decomposition method

- Master (M) - Slave (S) decomposition

$$\min_{X_M} \quad Obj(X_M, X_S(X_M), \pi) \qquad \Rightarrow \text{partition variable}$$

$$s.t. X_S(X_M) \quad \min_{X_S} Obj_S(X_S, X_M, \pi) \qquad \Rightarrow \text{Simple to solve}$$

$$s.t. \quad H(X_S, X_M, \pi) = 0$$

$$H(X_S, X_M, \pi) \geq 0$$

X_M Master Variables

X_S Slave Variables

π Parameters

Decomposition refers to Black Box strategy

$$\min_{X^*_{decision}} \quad TotalCost(X^*_{decision}, X(X^*_{decision}))$$

$$s.t. \quad G(X^*_{decision}, X(X^*_{decision})) \leq 0 \quad \text{inequality constraints}$$

$$where$$

$$X^*_{decision} = \{x_{decision}, y_{decision} \epsilon \{0,1\}\}$$

$X(X^*_{decision})$ Calculated by solving:

$F(X_{state}) = 0 \Rightarrow$ equipment model

$L(X_{state}) = 0 \Rightarrow$ linking equations

$T(X_{state}) = 0 \Rightarrow$ constitutive equations

$S(X_{state}) = 0 \Rightarrow$ Specification equations

$X_{decision} - X^*_{decision} = 0 \Rightarrow$ Specification of the value of decision variables

$$\min_{X_S} Obj_S(X_S, X_M, \pi)$$

$$H(X_S, X_M, \pi) = 0$$

$$H(X_S, X_M, \pi) \geq 0$$

$$where$$

$$X_{state} = \{x_{StateVariables}, x_{UnitParameters}, y_{decision} \epsilon \{0,1\}\}$$

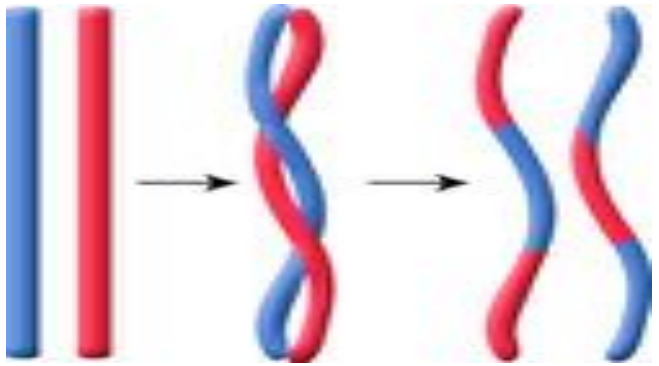
Heuristic methods to systematically generate optimal configurations

- Applies only on black box strategy
- Exploring the search domain
  - systematically
  - strategy based on some analogy
  - Use KPIs as fitness of individuals of a population
- Simulated annealing
  - based on the analogy with metallurgy
    - heating/cooling of metal to minimize the energy content
- Evolutionary algorithm
  - genetic algorithms
    - based on the analogy of the evolution
      - Best fitted individuals have a higher probability to survive and reproduce
      - Reproduction based on sharing gene info
- Particle swarm
  - initial speed + communication between agents
- Ants colony

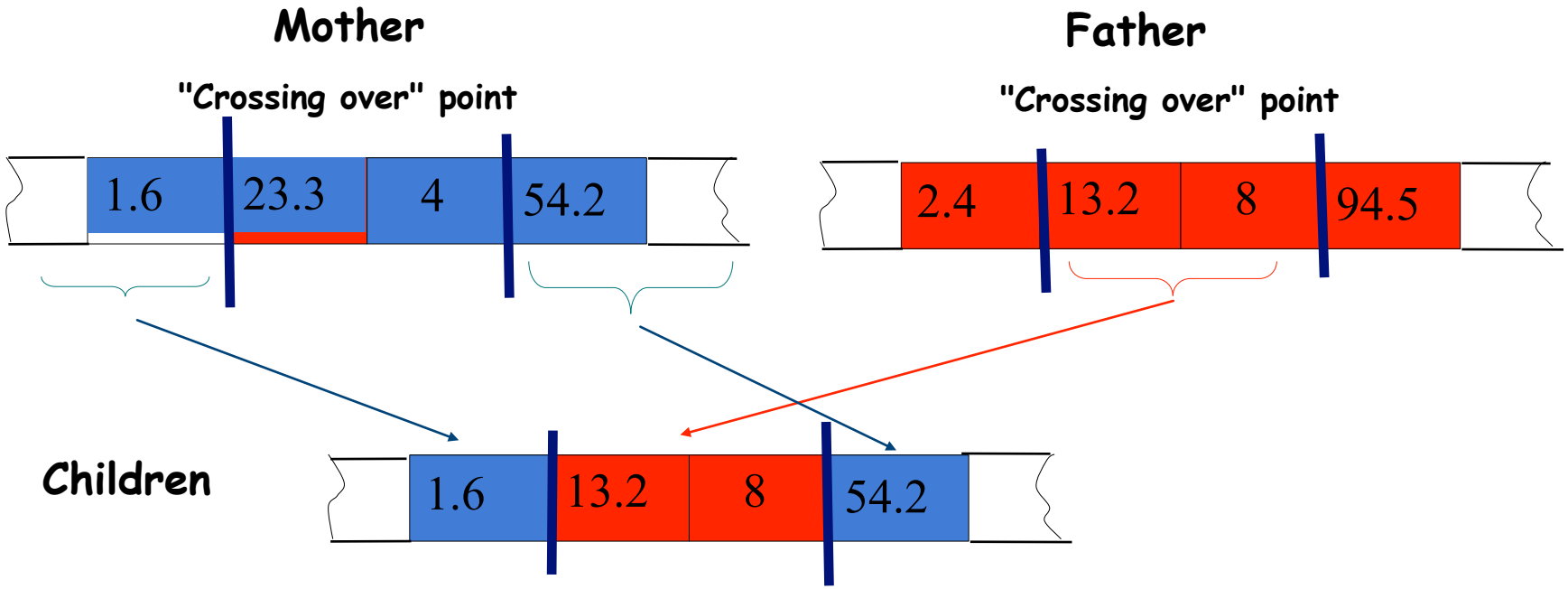
Evolutionary algorithms

- Characteristics : find the members of a population that best fits the goals (KPIs)
  - Population (X)
  - Objective Function(s) : System Performances Y=F(X)
  - No direction (No derivatives - No “iteration”)
  - Heavy duty : exploration => Computing time !
  - Problem definition is free
  - Random nature : explore the search space
  - Inequality constraints ?
- Principle
  - Initialization (random population generation (e.g. 100 initial values of X_M)
  - Reproduction => select parents that best fits & reproduce to generate new members
  - New member generation
    - Cross-over (random) : choose the shared gene in each parent
    - Mutation : allow for diversity
  - Update population (maintain population)
    - eliminate the worst individuals : the less fitting
    - re-group by types to preserve diversity

reproduction by "Crossing over"



Random selection of parents in the population
Random selection of the genes to share

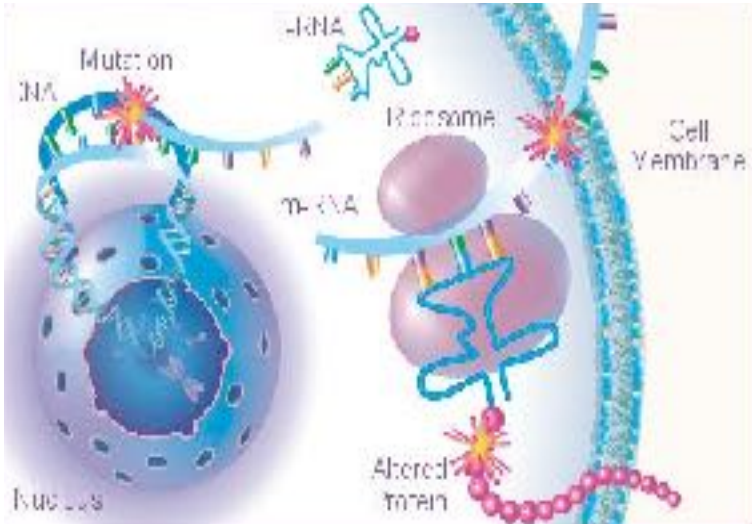


Cross-over operator (how to generate a child)

- Cross-over can use interpolation techniques
  - e.g. quadratic approximation based on a subset of the population
  - select randomly and/or take the bests
- Preserve the random nature !
  - e.g. random relaxation

Mutation : in order to preserve diversity

Mutation allows to ensure that the system will not be trapped in a local optimum and that the whole space will be observed



Random Mutation

5

before mutation

	1.6	23.3	~~8~~	94.5	

after mutation

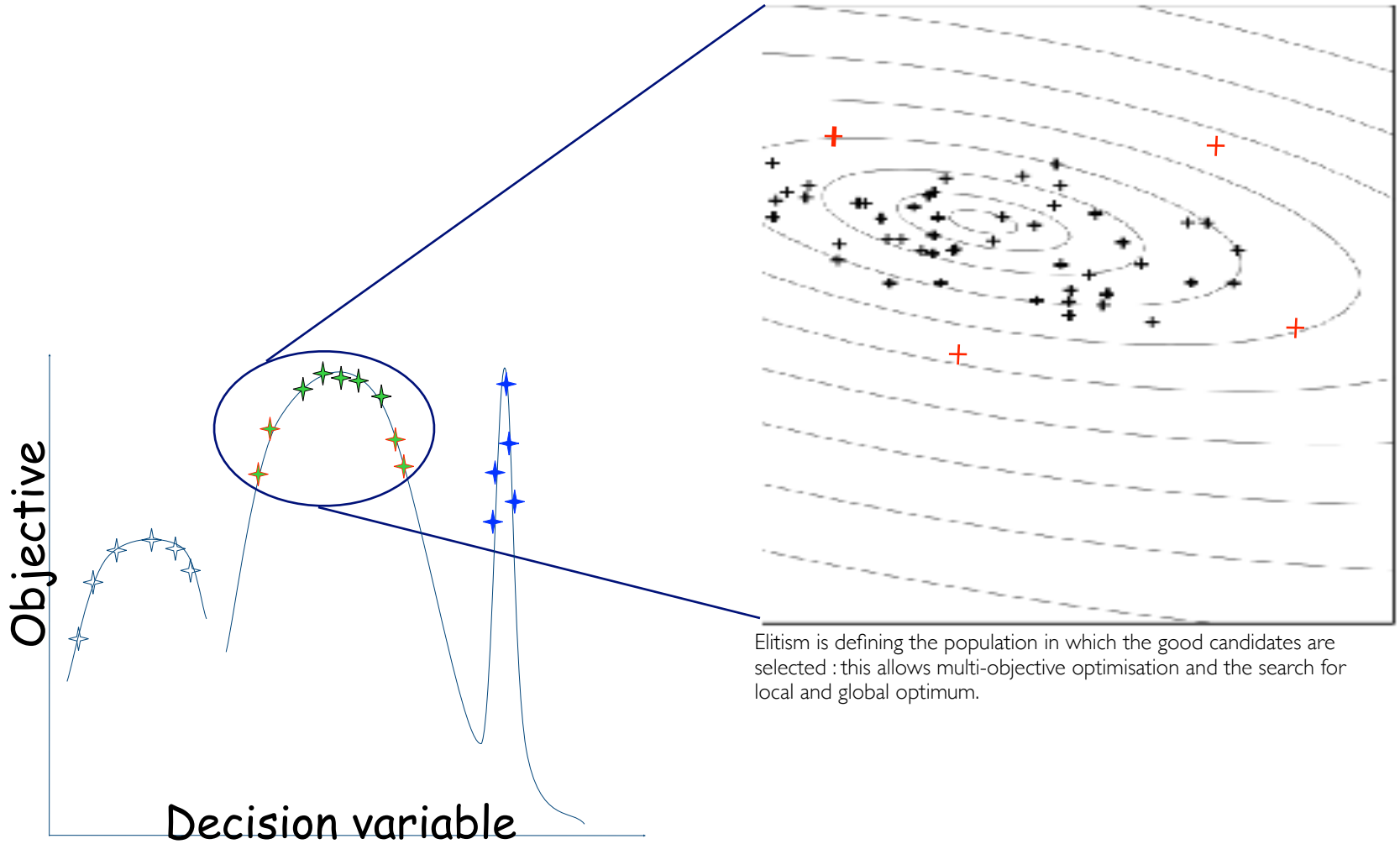
	1.6	23.3	5	94.5	

Selection : non dominated solutions

- Eliminates the worst fitted
  - be part of a Pareto for example
    - non dominated: be better than the others for at least on OBJ

Elimination of the worst fitted

Elitism : preserve the best candidates for reproduction



Elitism is defining the population in which the good candidates are selected : this allows multi-objective optimisation and the search for local and global optimum.

Evolutionary algorithms : conditions

- **Black box approach**
  - Equality constraints are solved explicitly
  - Inequality constraints ideally transformed into decision variables bounds
- **Fast F(X) calculation**
  - Search space exploration => huge number of evaluations
- **Robust F(X) calculation**
  - search space response
- **No efficient mathematical programming methods**
  - Non differentiable problems
  - MINLP
- **Limited number of degrees of freedom**

Evolutionary algorithm : advantages

- **Global optimisation**
  - Exploration of the search space
- **Black Box**
  - Accepts different type of objective function
    - incl. observations
- **Non differentiable problems**
  - The objective function can have jumps or steps
- Easy to **parallelise**
- Freedom in the choice of decision variables
  - x1*x2*x3 is not a problem
- **Multi-objective problem**
  - Efficient use of the computing time
  - Dominancy criteria

Evolutionary algorithm : drawbacks

- **Speed of resolution of F(X)**
  - Requires a large number of F(X) evaluation
  - Use of surrogate models
- **Robust** and fast F(X) evaluation
- **Number of decision variables**
  - Convergence properties is a combinatorial function of the number of variables
- **Limited Feasible domain**
  - Probability of finding feasible F(X) is low
  - Choice of the decision variables
- **Constraints handling**
  - Equality or inequality

Evolutionary algorithm

- Handling inequality constraints

$$\min_{X} OBJ(X)$$
$$st.$$
$$F(X) = 0$$
$$G(X) \leq 0$$

Well chosen penalty function

$$\min_{X^d} OBJ(X^d, Y(X^d)) + P(X^d)$$
$$st.$$
$$Y(X^d) = F(X^d)$$
$$P(X^d) = \sum (max(G(X^d, Y(X^d)), 0))^2$$
$$X^d_{max} \leq X^d \leq X^d_{max}$$

Evolutionary algorithm

- Choosing the appropriate decision variables

$$\min_{x_1,x_2} f(x_1,x_2)$$

$$st.$$

$$x_1 \leq x_2$$

$$x_1^{min} \leq x_1 \leq x_1^{max}$$

$$x_2^{min} \leq x_2 \leq x_2^{max}$$

Becomes

$$\min_{x_1^*,x_2} f(x_1 = x_1^{min} + (min(x_2, x_1^{max}) - (min(x_2, x_1^{max}) - x_1^{min})) \cdot x_1^*, x_2)$$

$$st.$$

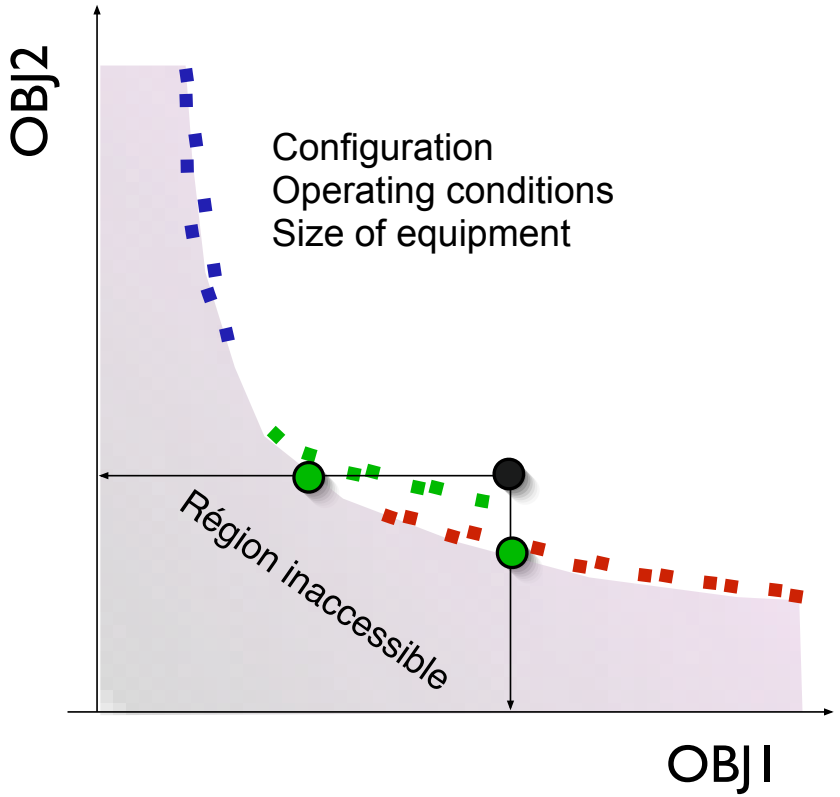
$$0 \leq x_1^* \leq 1$$

$$x_2^{min} \leq x_2 \leq x_2^{max}$$

Works well for Evolutionary algorithm
do not use for mathematical programming

Multi-objective optimisation and evolutionary algorithm

The population includes the Pareto set



Clustering techniques (big data)

- identify decision variable sub-spaces
- generate multiple Pareto curves

Dominancy

- Remains in the population if at least better than others for 1 objective
- Preserve sub-optimal population

The goal is indeed to take decisions : being informed about the collection of good solutions allows to have a better knowledge of what is building a solution and for which reason the filan solution will be selected

Hybrid methods

- Use Evolutionary algorithm to find initial point for mathematical programming
- Global optimization (find min of min)
- Limited number of NLP
- Do it in 2 directions
  - min obj1
  - min obj2