



# L'optimisation Multiobjectif



# Introduction à l'Optimisation Multiobjectif

- Un exemple simple: La Poutre
- Les bases de l'optimisation multiobjectif
- Les méthodes classiques
- Le couplage aux méthode d'optimisation
- Les métriques de performance
- Les problèmes test
- Quelques exemples Continus/Combinatoire

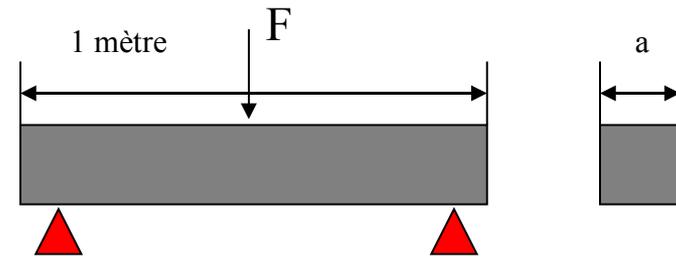


# Un exemple: La poutre pleine

$$S(a) = a^2$$

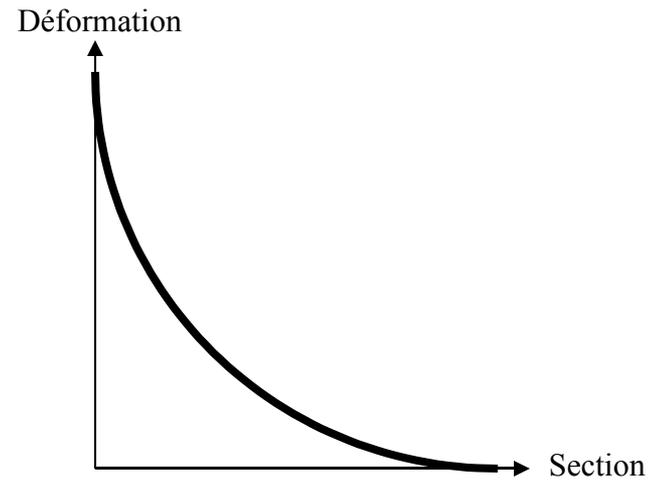
$$d(a) = 1000 + \frac{1 \cdot 10^{-2}}{192 + 2 \cdot 10^5 + \frac{a^4}{12}}$$

$$a \leq 0.1$$



On veut:

- Minimiser le poids de la poutre (sa section)
- Minimiser la déformation de la poutre





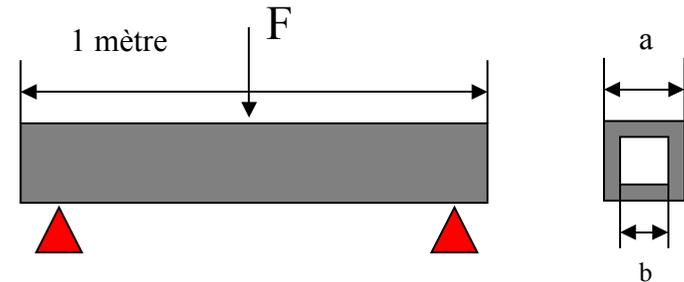
# Un exemple: La poutre creuse

$$S(a, b) = a^2 - b^2$$

$$d(a, b) = 1000 + \frac{1 \cdot 10^{-2}}{192 + 2 \cdot 10^5 + \frac{a^4 - b^4}{12}}$$

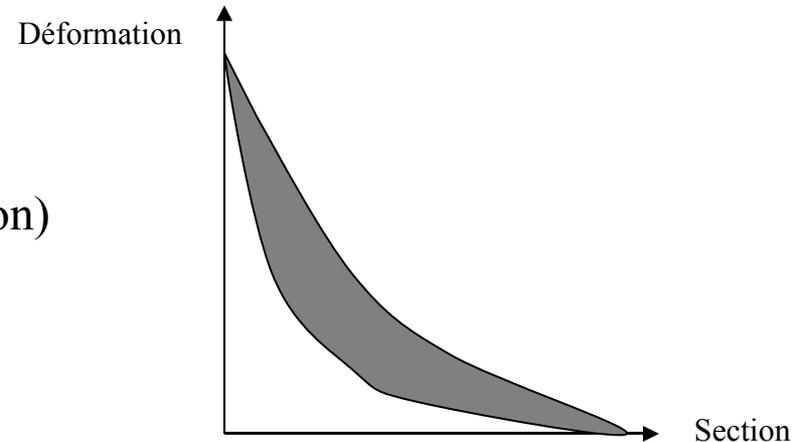
$$a \leq 0.1$$

$$b + 0.04 \leq a$$



On veut:

- Minimiser le poids de la poutre (sa section)
- Minimiser la déformation de la poutre





# Les bases de l'Optimisation Multiobjectif

Une légère différence dans la formulation du problème:

$$\min_{x \in D} f(x) \rightarrow \min_{x \in D} \vec{f}(x)$$

Une solution unique pour  
le problème mono-objectif



Une multitude de solutions pour  
le problème multiobjectif

Une définition « dure » de  
l'optimum en mono-objectif



Une ou plusieurs définition d'un  
optimum en optimisation mutiobjectif

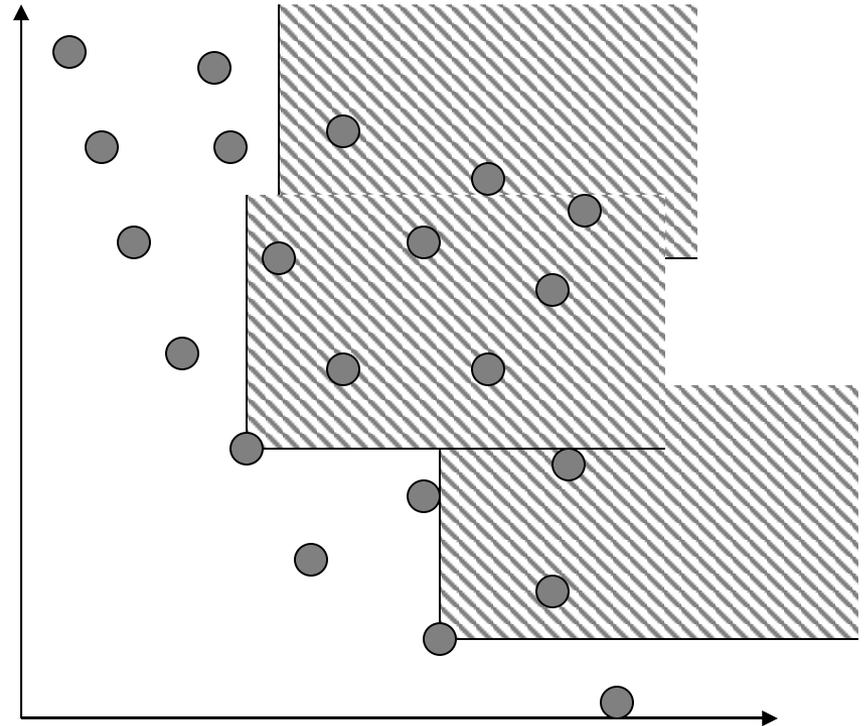


# Les bases de l'Optimisation Multiobjectif

L'Optimum de Pareto:

Un vecteur A est meilleur qu'un vecteur B, au sens de Pareto si:

- **Les** composantes de A sont **meilleures ou égales** que les composantes de B,
- **Au moins 1** composante de A est **meilleure** que la composante correspondante de B





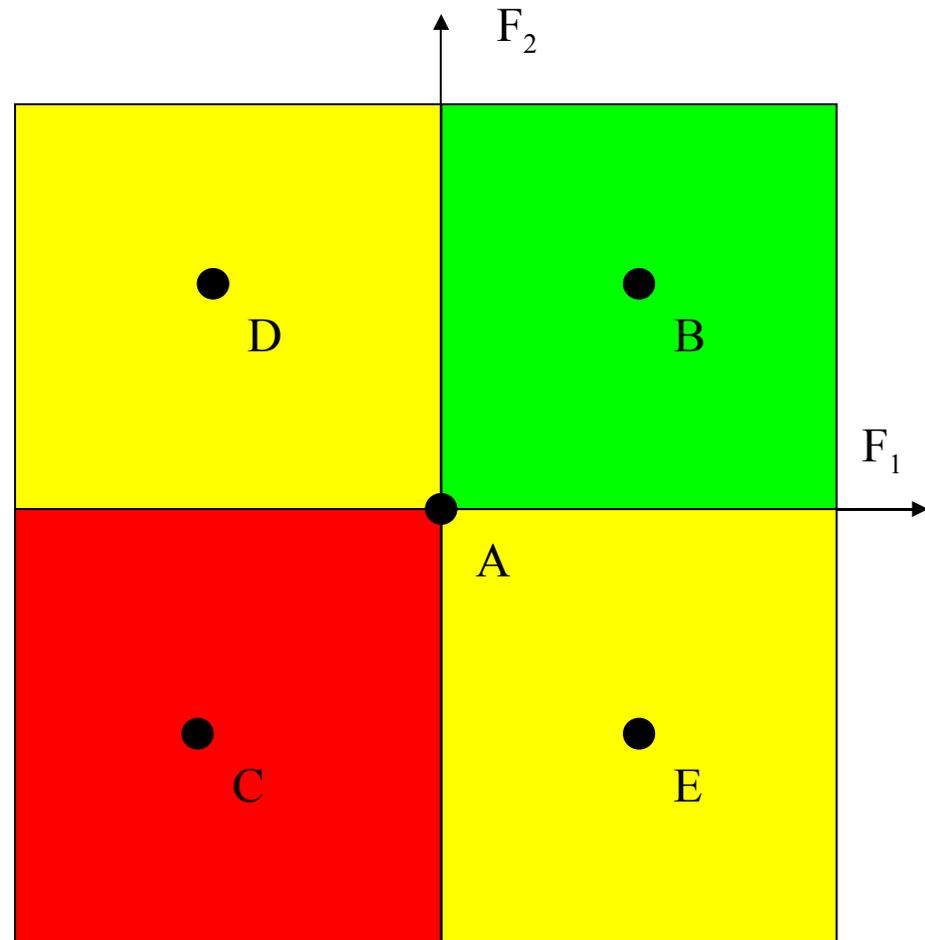
# Les bases de l'Optimisation Multiobjectif

A domine B  
(amélioration des 2 FObjs)

A est dominé par C  
(détérioration des 2 FObjs)

A est incomparable à E  
(amélioration de  $F_1$ , détérioration de  $F_2$ )

A est incomparable à D  
(détérioration de  $F_1$ , amélioration de  $F_2$ )

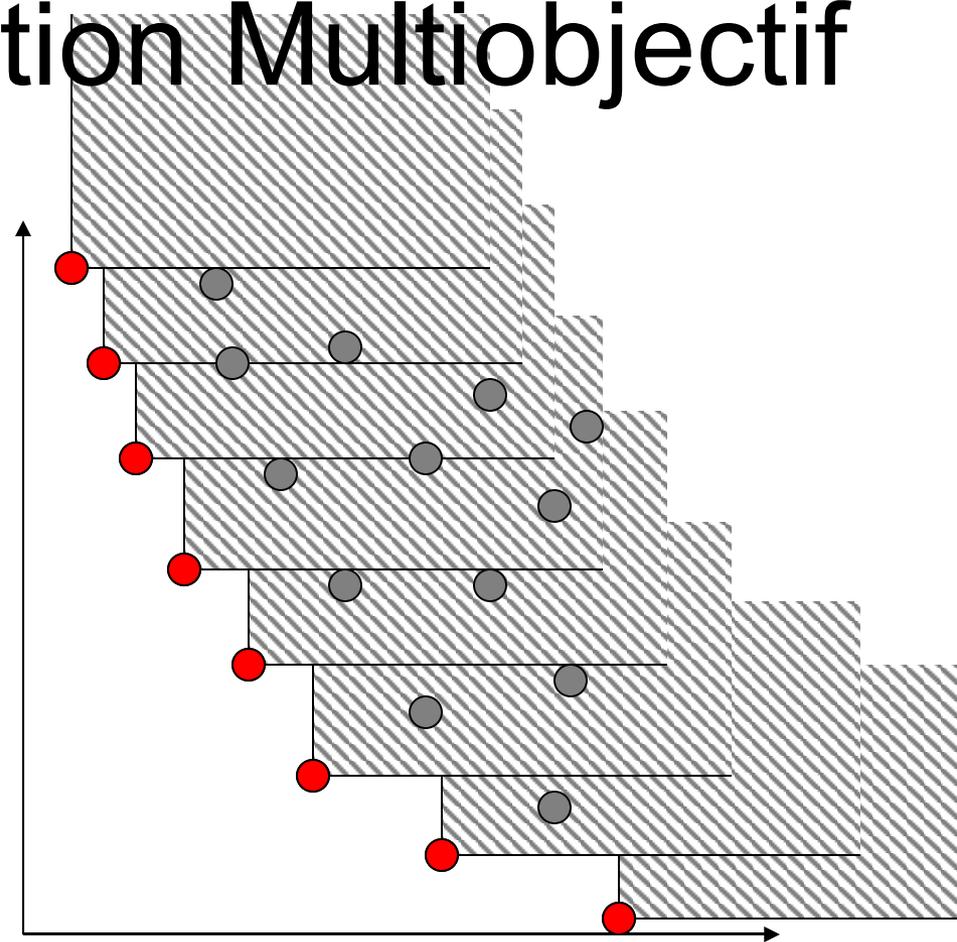




# Les bases de l'Optimisation Multiobjectif

L'ensemble des solutions optimales au sens de Pareto s'appelle:

- Surface de compromis
- Ensemble des solutions non-dominées
- Front de Pareto





# Les méthodes classiques

Il existe plusieurs manières de traiter un problème d'optimisation multiobjectif:

- En transformant le problème d'optimisation multiobjectif en problème d'optimisation mono-objectif sans contraintes:
  - Méthode de pondération des fonctions objectifs
  - Méthode de la distance à un but
- En transformant le problème d'optimisation multiobjectif en problème d'optimisation mono-objectif avec contraintes:
  - Méthode du but à atteindre
  - Méthode du compromis



# La pondération des fonctions objectifs

Le problème de départ:

$$\min \vec{f}(x)$$

$$x \in D \subseteq \mathbb{R}^n$$

Devient:

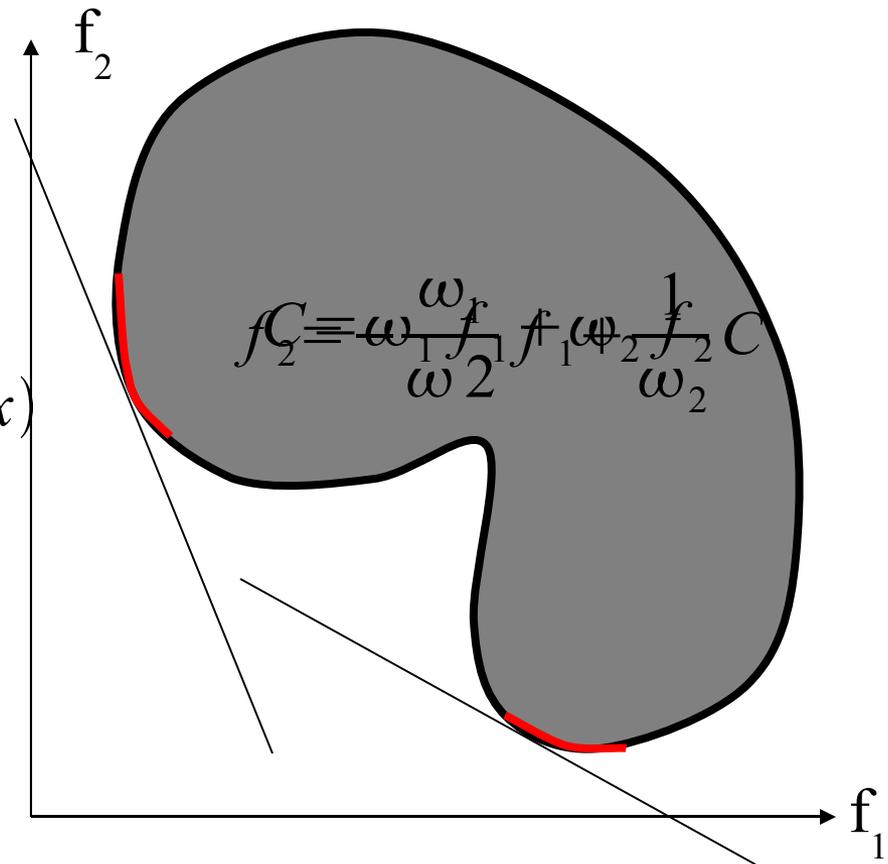
$$\min \omega_1 f_1(x) + \dots + \omega_p f_p(x)$$

$$x \in D \subseteq \mathbb{R}^n$$

Avec:

$$\omega_1 + \dots + \omega_p = 1$$

$$\omega_1, \dots, \omega_p \geq 0$$





# La méthode de Tchebychev

Le problème de départ:

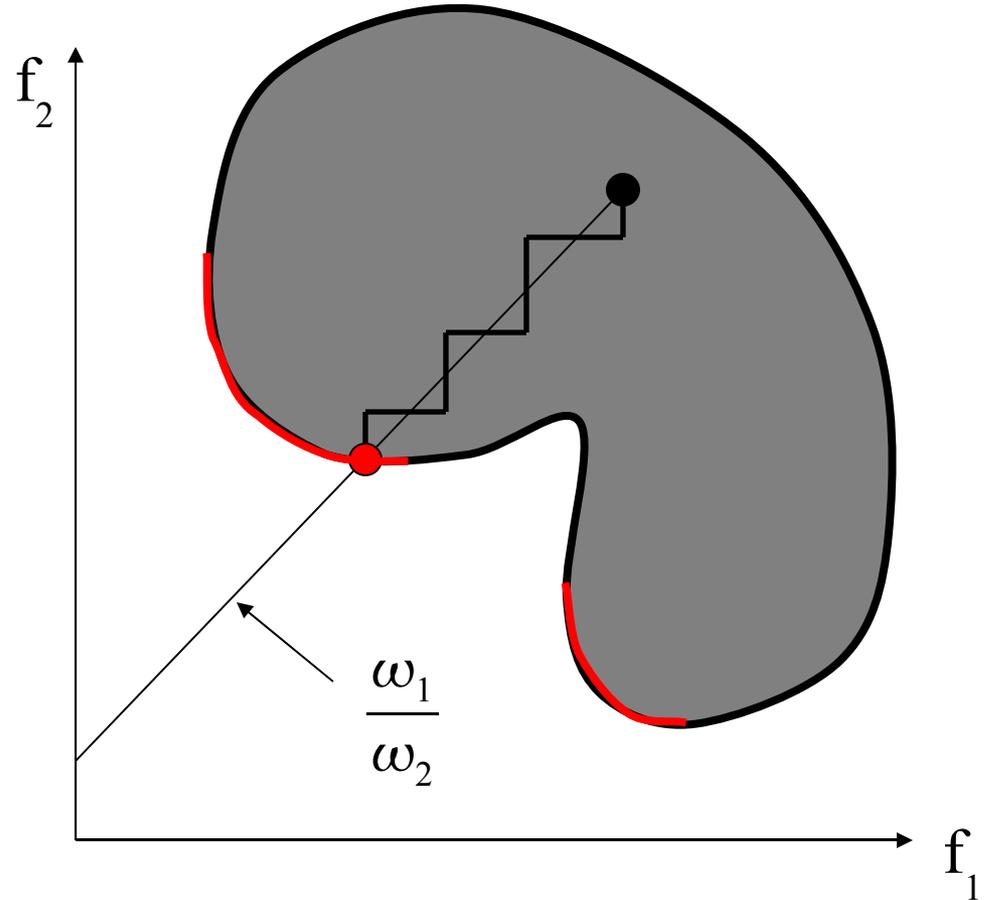
$$\min_{x \in D \subseteq \mathbb{R}^n} \vec{f}(x)$$

Devient:

$$\min_{x \in D \subseteq \mathbb{R}^n} \max_i \omega_i (f_i(x) - F_i)$$

Avec:

$$\omega_1 + \dots + \omega_p = 1$$
$$\omega_1, \dots, \omega_p \geq 0$$





# La méthode du but à atteindre

Le problème de départ:

$$\min \vec{f}(x)$$
$$x \in D \subseteq \mathbb{R}^n$$

Devient:

$$\min \lambda$$
$$f_1(x) - \omega_1 \lambda \leq F_1$$

...

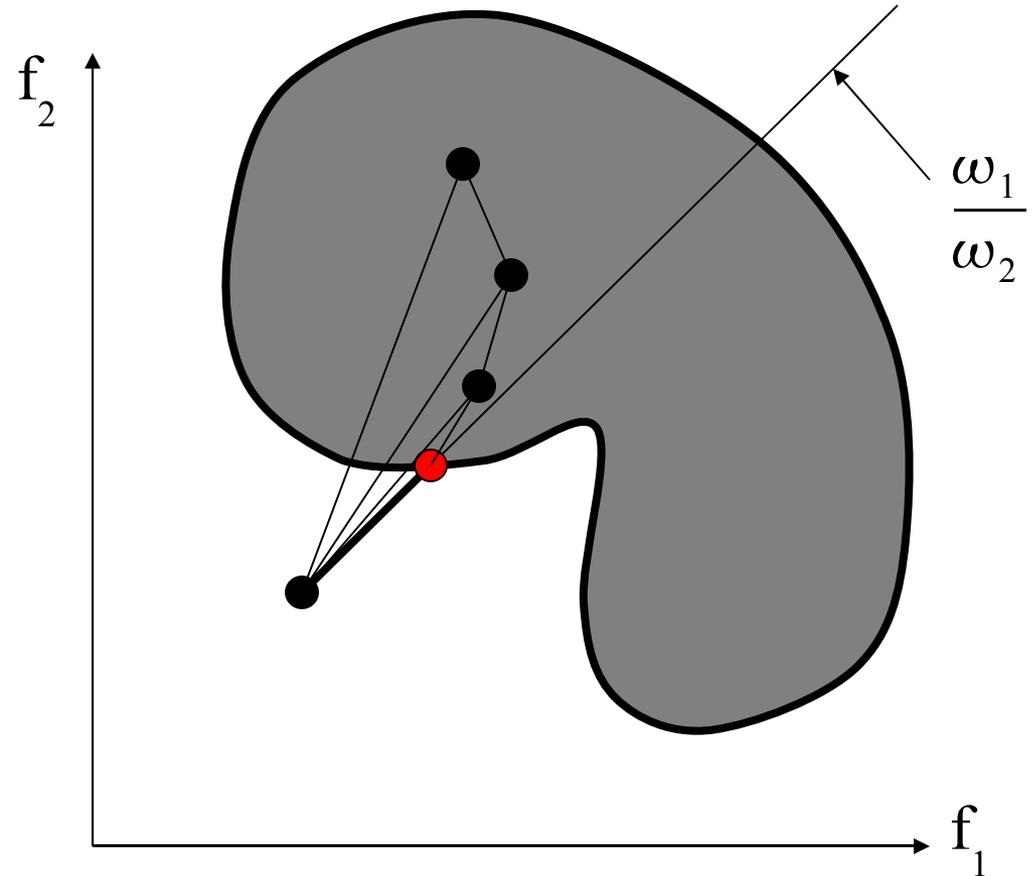
$$f_p(x) - \omega_p \lambda \leq F_p$$

$$x \in D \subseteq \mathbb{R}^n$$

Avec:

$$\omega_1 + \dots + \omega_p = 1$$

$$\omega_1, \dots, \omega_p \geq 0$$





# La méthode du compromis

Le problème de départ:

$$\min \vec{f}(x)$$
$$x \in D \subseteq \mathbb{R}^n$$

Devient:

$$\min f_1(x)$$

$$f_2(x) \leq \epsilon_2$$

...

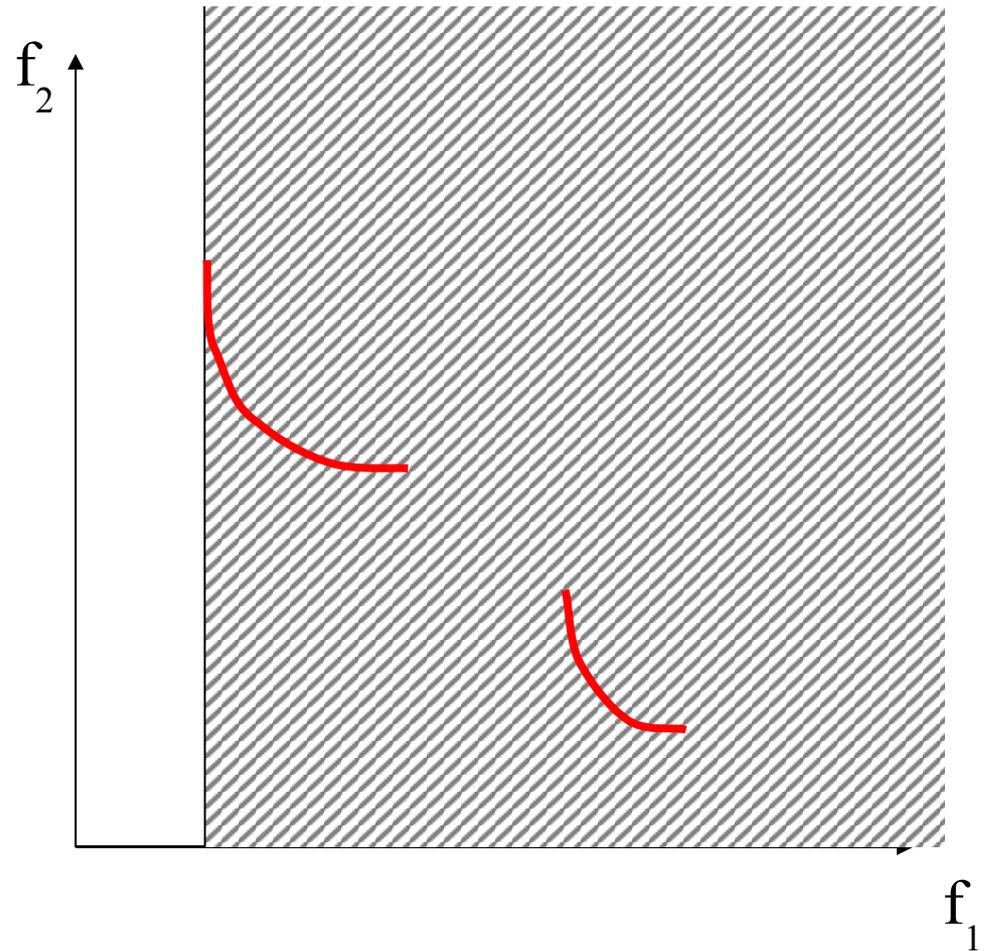
$$f_p(x) \leq \epsilon_p$$

$$x \in D \subseteq \mathbb{R}^n$$

Avec:

$$\omega_1 + \dots + \omega_p = 1$$

$$\omega_1, \dots, \omega_p \geq 0$$





# Le couplage aux méthodes d'optimisation

Pour la plupart des méthodes classiques d'optimisation, n'importe quel type de méthode d'optimisation peut servir pour effectuer une recherche de point non-dominé.

- On se ramène souvent à une fonction mono-objectif sans contraintes
- On se ramène parfois à une fonction mono-objectif avec contraintes

Certaines méthodes d'optimisation sont spécifiques à l'optimisation multiobjectif:

- MOGA, VEGA, NSGA (à base d'algorithme génétique)
- MOSA (à base de recuit simulé)



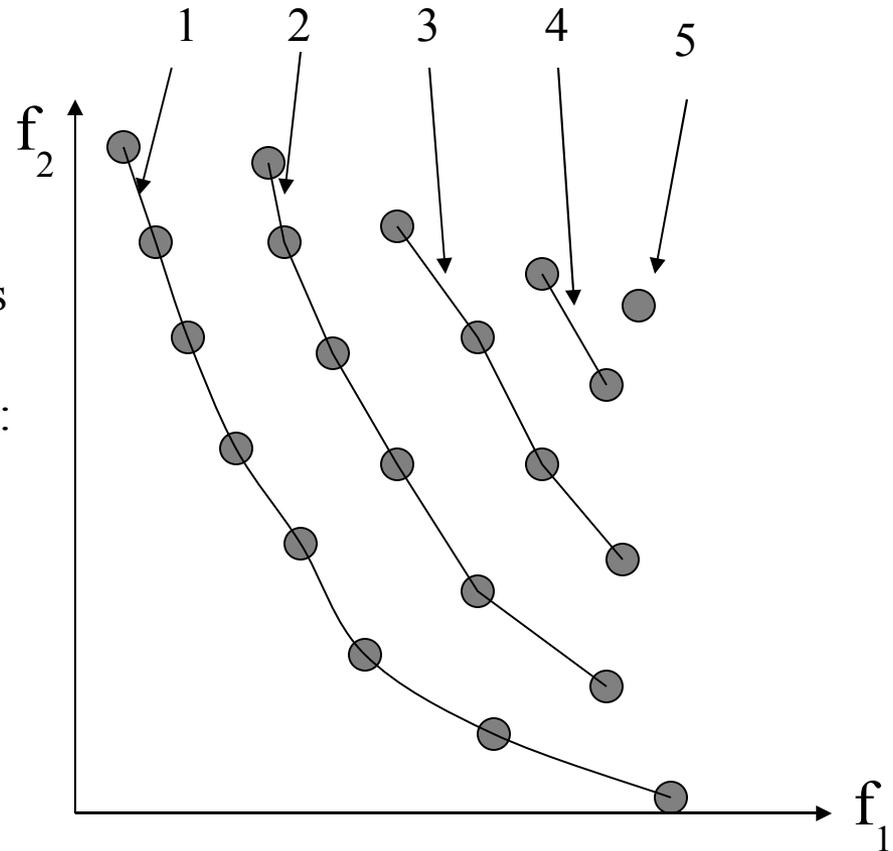
# Des méthodes spécifiques: MOGA

Ici, aucune forme d'agrégation des objectifs: On travaille sur les solutions directement

- Le rang de la surface de compromis est utilisé pour l'efficacité

Méthode **simple** et **efficace**. Cependant:

- Mauvaise **répartition** des solutions sur la surface de compromis
- Mauvais niveau de **convergence**





# Des méthodes spécifiques: NSGA

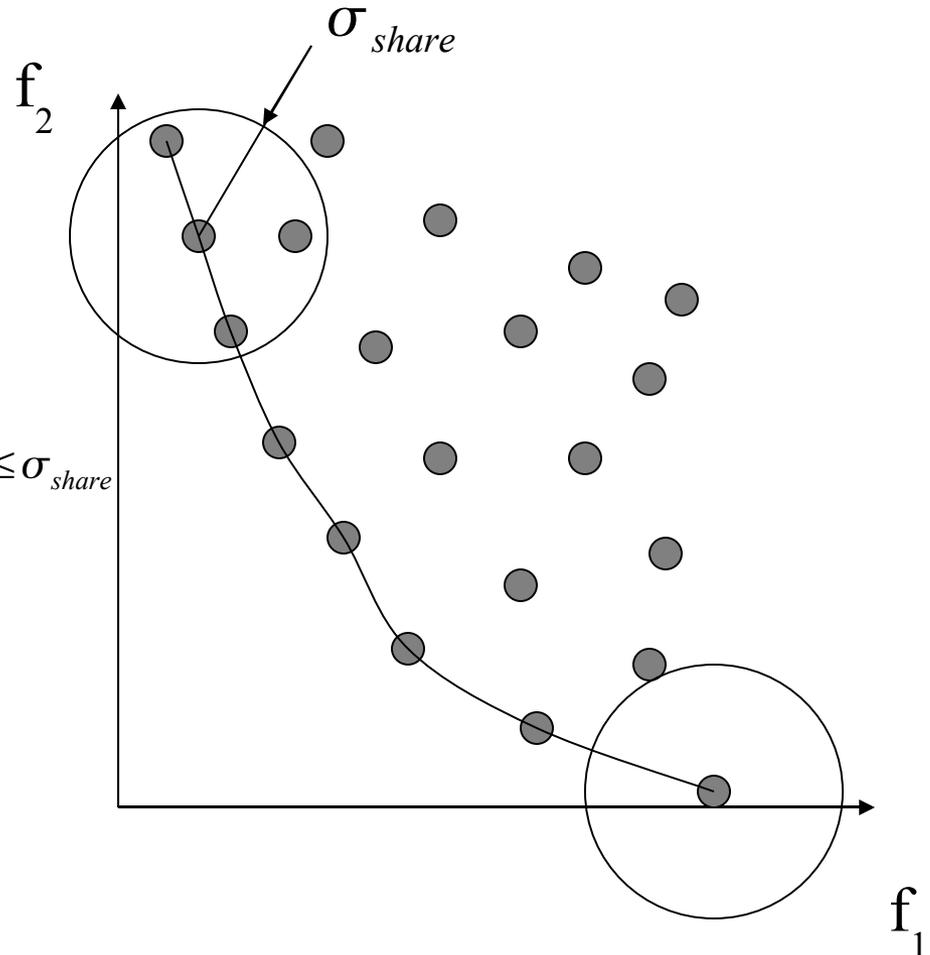
Comme MOGA + méthode pour améliorer la répartition:

$$m_i = \sum_{j=1}^K Sh(d(i, j))$$

$$Sh(d(i, j)) = 1 - \left( \frac{d(i, j)}{\sigma_{share}} \right) \quad \text{si } d(i, j) \leq \sigma_{share}$$

$$Sh(d(i, j)) = 0 \quad \text{sinon}$$

$$f_i = \frac{F}{m_i}$$





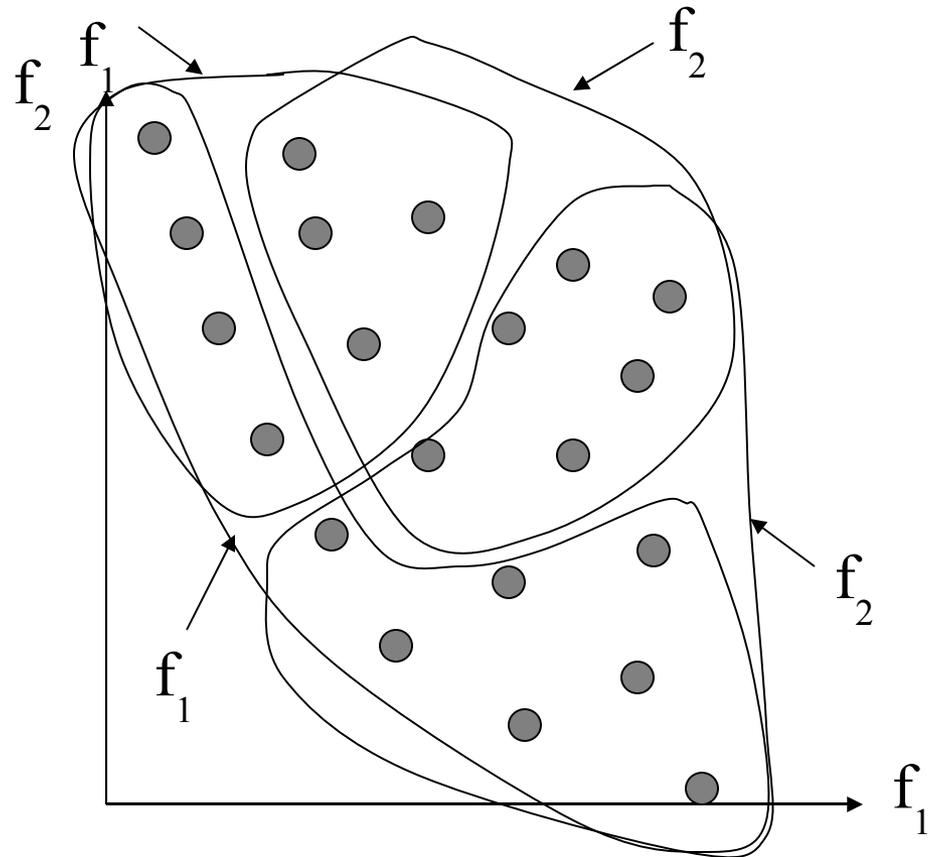
# Des méthodes spécifiques: VEGA

La première méthode d'optimisation multiobjectif à base de GA.

- Simple
- Efficace (bonne convergence)
- S'adapte facilement sur un GA mono-objectif

Mais

- Tend à privilégier les individus extrêmes





# Les différentes branches

Plusieurs familles dans l'optimisation multiobjectif:

- MCDM (Multi Criteria Decision Making)
- MCDA (Multi Criteria Decision Aid)
- MOO (Multi Objective Optimization)



# Les différentes branches: MCDM

Famille certainement à l'origine du multicritère.

Domaine d'intervention:

- Management
- Sélection de projet
- Économie

Une référence incontournable:

R.L. Keeney, L. Raiffa, « Decisions with multiple objectives: preferences and value tradeoff », Ed. Cambridge University Press, 1993



# Les différentes branches: MCDA

Une autre approche de la théorie de la décision.

Un maître à penser: B. Roy.

Cette branche a les mêmes domaines d'intervention que MCDM.

Une référence bibliographique de ce domaine:

B. Roy, « Méthodologie multicritère d'aide à la décision », Ed. Economica, 1985



# Les différentes branches: MOO

Domaine de prédilection: les problèmes d'ingénierie (mécanique, etc.)

Au début: quelques méthode simples.

Aujourd'hui, le nombre de contributions à l'optimisation multiobjectif explose.

- Des conférences dédiées à l'optimisation multiobjectif (EMOO 01)
- Des sessions spéciales au sein de certaines conférences renommées (MCDM)
- Des numéros spéciaux de journaux (Journal of Heuristics)

Les méthodes les plus utilisées: les algorithmes génétiques.

Un lien intéressant:

<http://www.lania.mx/~ccoello/EMOO>



# Les métriques de performance

On juge les surfaces de compromis sur plusieurs critères:

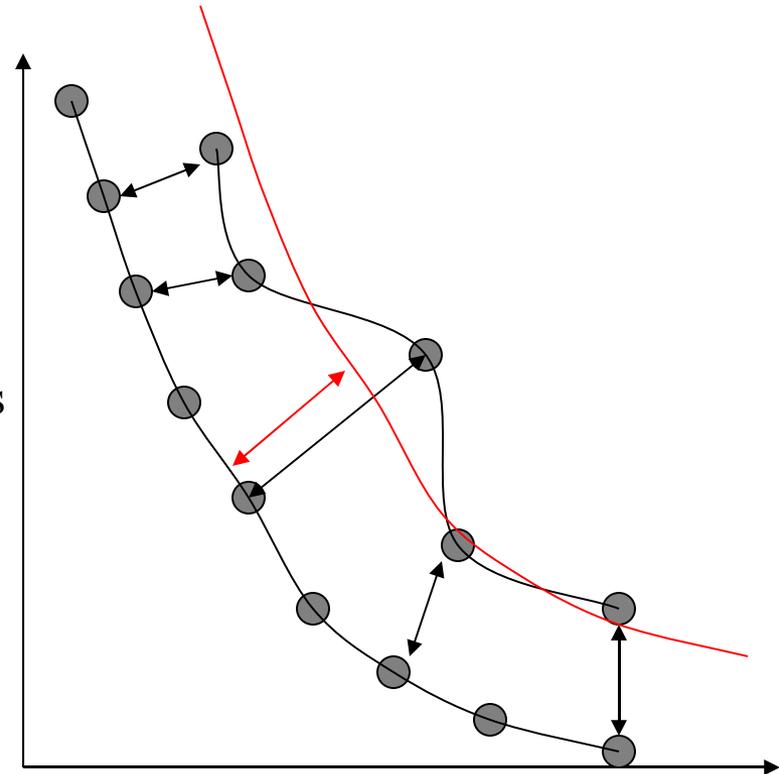
- Esthétiques:
  - Espacement des solutions sur la surface de compromis
  - Déformation de la surface de compromis
  - Distribution des points sur la surface de compromis
- Convergences:
  - Proximité de la surface de compromis par rapport à la surface de compromis idéale
  - Vitesse de convergence



# Les métriques de performance: Proximité

La métrique se calcule de la manière suivante:

- On extrait la surface de compromis
- On calcule la distance la plus proche entre le point de la surface de compromis calculée et la surface de compromis théorique
- On fait la moyenne de ces distances



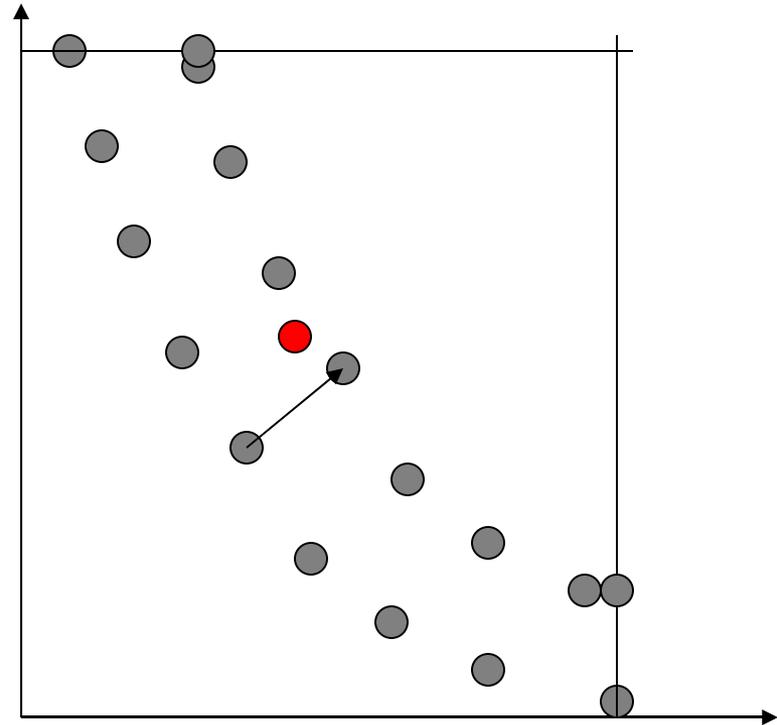


# Les métriques de performance: Vitesse

On procède de la manière suivante:

On crée une surface de compromis seuil

- On prend une surface de compromis théorique
- On la déplace d'un certain vecteur
- On ferme la surface seuil
- On arrête l'optimisation dès que le meilleur point passe sous la surface seuil



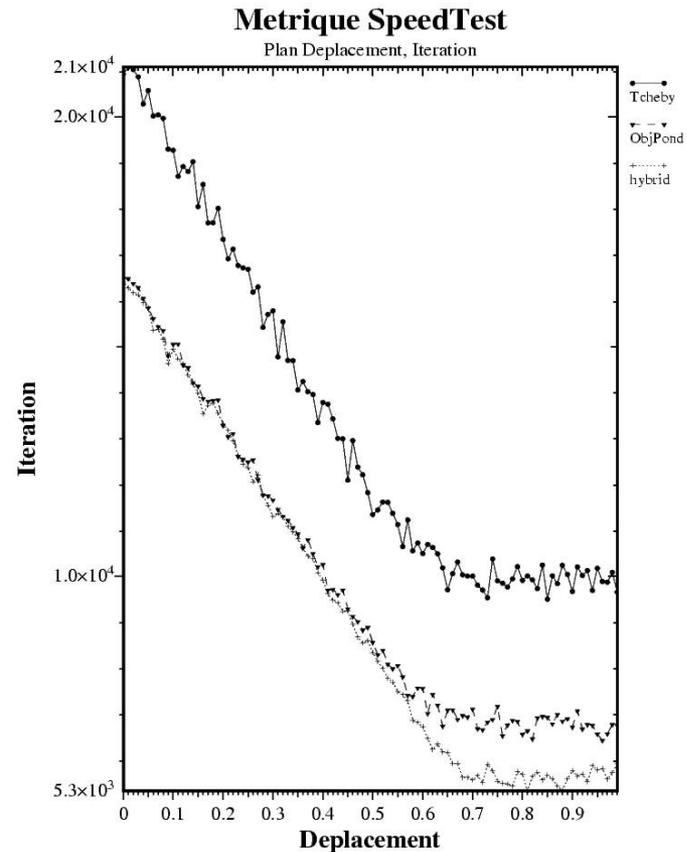


# Les métriques de performance: Vitesse

Tue Oct 16 15:54:19 2001

Cette métrique à été utilisée pour comparer les performances entre trois méthode multiobjectif:

- La pondération des fonctions objectifs
- La métrique de Tchebychev
- Une méthode hybride (Pondération+Tchebychev)

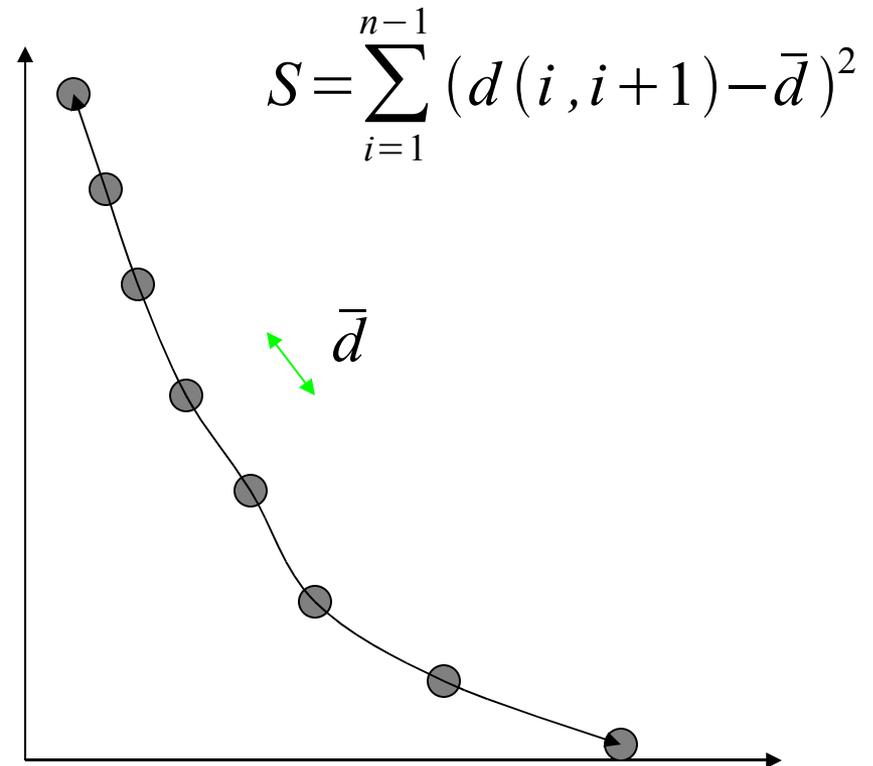




# Les métriques de performance: Espacement

On travaille sur la surface de compromis

- 1 – On mesure la longueur totale de la surface de compromis
- 2 – On calcule l'espace idéal entre deux points
- 3 – On calcule la somme des différences entre l'espace idéal et l'espace entre deux points





# Les métriques de performance: Déformation

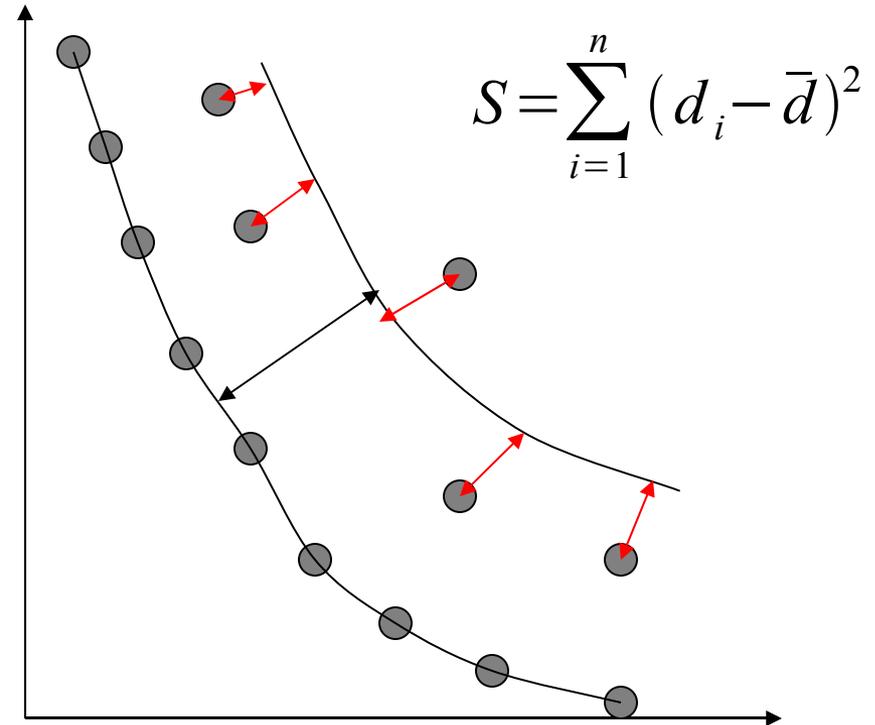
On travaille avec deux surfaces de compromis:

- La surface idéale
- La surface calculée

1 – On calcule la distance moyenne entre les deux surfaces de compromis

2 – On calcule les distances entre les points de la surface de compromis et les points les plus proches de la surface idéale

3 – On fait la somme des différences entre la distance moyenne et la distance calculée



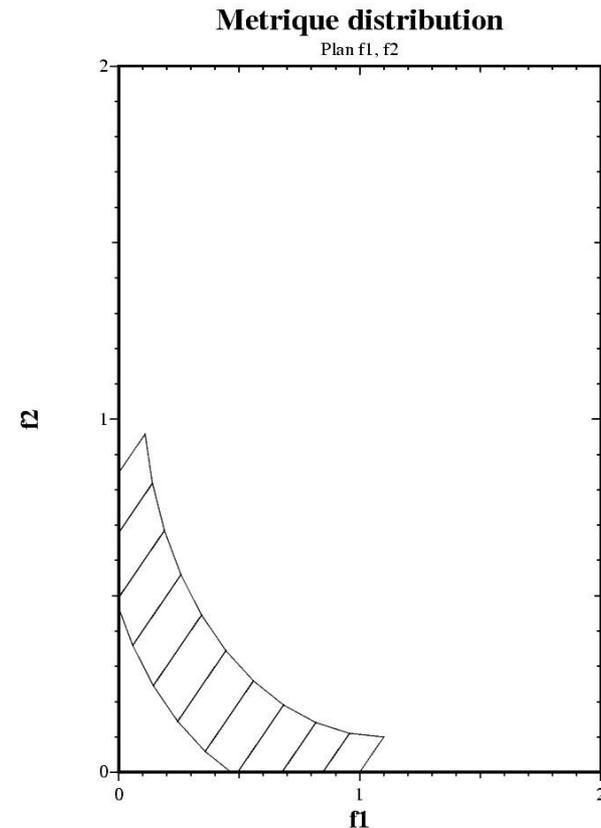


# Les métriques de performance: Distribution

Tue Oct 16 16:08:58 2001

Cette métrique permet de déterminer la distribution des points sur la surface de compromis.

Utilisée dans une étude sur l'influence du point de départ sur la distribution des points sur la surface de compromis.





# Les problèmes test

Pour évaluer les performances des méthodes d'optimisation multiobjectif:

**Il faut des problèmes test**

Ces problèmes permettent de:

- Connaître a priori la surface de compromis idéale
- Tester une méthode avec une surface de compromis ayant une certaine forme
  - Convexe
  - Non-Convexe
  - Discontinue
- Tester une méthode sur un ensemble avec des difficultés particulières
  - Surface de compromis « fantôme »
  - Non-Uniformité de la répartition des points dans l'ensemble solution



# Les problèmes test

Une première famille générique de problèmes test:

## Les problèmes test de Deb

- Ils sont basés sur une forme générique

$$\min f_1 = f(x_1, \dots, x_m)$$

$$\min f_2 = g(x_{m+1}, \dots, x_N) h(f, g)$$

f contrôle l'uniformité de la distribution de points

g contrôle la « multifrontalité »

h contrôle la forme de la surface de compromis

- Ils permettent de réaliser tous types de difficulté:

- Surface convexe
- Surface non-convexe
- Surface discontinue
- Répartition non-uniforme



# Les problèmes test

Un exemple: Le problème multifrontale de Deb

$$\min f_1(x) = f(x_1)$$

$$\min f_2(x) = g(x_2)h(f, g)$$

Avec:

$$f(x_1) = x_1$$

$$g(x_2) = 2 - \exp\left(-\left(\frac{x_2 - 0.2}{0.04}\right)^2\right) - 0.8 \exp\left(-\left(\frac{x_2 - 0.6}{0.4}\right)^2\right)$$

$$h(f, g) = \frac{1}{f}$$

La fonction DEB VII





# Quelques exemples Continus/Combinatoires

Une méthode simple pour obtenir des problèmes test combinatoire:

**La conversion continu/combinatoire**

La démarche à suivre est la suivante:

$$0 \leq x_1 \leq 1 \quad 0 \leq x_2 \leq 1$$

1 - On part d'un codage binaire

$$10000001 - 00001001$$

2 – On convertit ce codage en valeur numérique

$$129 \quad - \quad 9$$

3 – On normalise la valeur des variables pour qu'elles appartiennent au domaine réalisable

$$0.505 \quad - \quad 0.035$$



# Quelques exemples Continus/Combinatoires

Avantage:

- On peut régler la forme de la surface de compromis du problème test combinatoire
- On peut visualiser dans un espace 2D le trajet suivi par le point au cours de l'optimisation
- On peut étudier la forme du voisinage d'un point au cours de l'optimisation

Plan:

- Un problème test simple: le problème test convexe trigonométrique
  - Les résultats avec une méthode de pondération des fonctions objectifs
  - Les résultats avec une méthode de Tchebychev
- Un problème test simple: le problème test non-convexe trigonométrique
  - Les résultats avec une méthode de pondération des fonctions objectifs
  - Les résultats avec une méthode de Tchebychev



# Quelques exemples Continus/Combinatoires

Le problème test convexe trigonométrique:

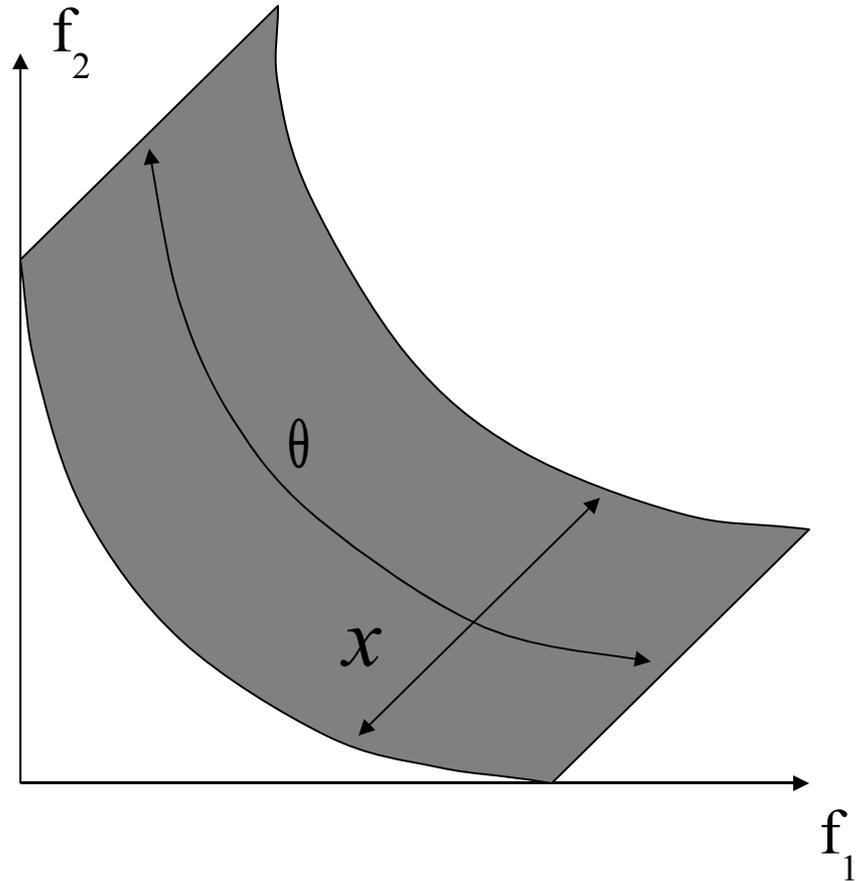
$$\min f_1(x, \theta) = 1 - \sin(\theta) + x$$

$$\min f_2(x, \theta) = 1 - \cos(\theta) + x$$

Avec :

$$0 \leq \theta \leq \frac{\pi}{2}$$

$$0 \leq x \leq 1$$





# Quelques exemples Continus/Combinatoires

Le problème test non-convexe  
trigonométrique:

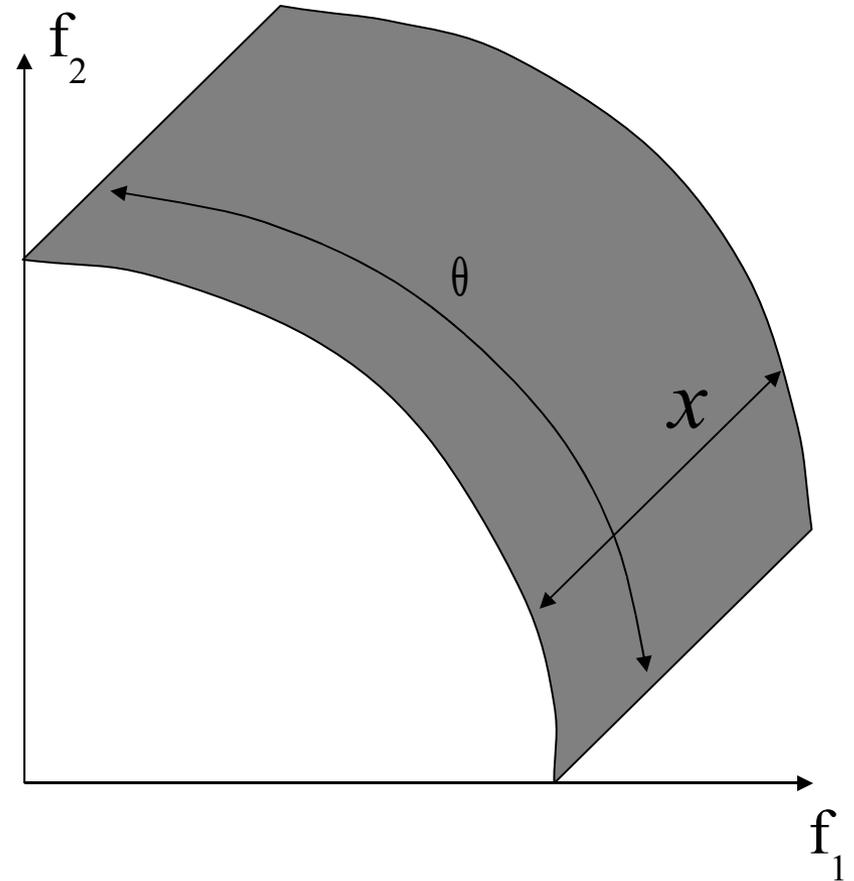
$$\min f_1(x, \theta) = \cos(\theta) + x$$

$$\min f_2(x, \theta) = \sin(\theta) + x$$

Avec :

$$0 \leq \theta \leq \frac{\pi}{2}$$

$$0 \leq x \leq 1$$



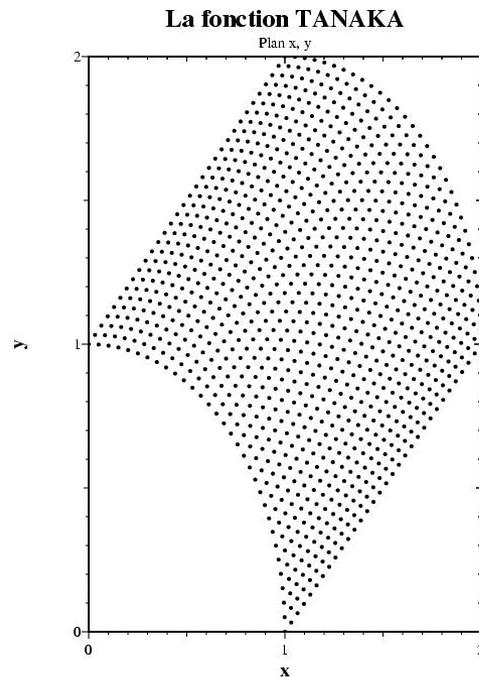


# Quelques exemples Continus

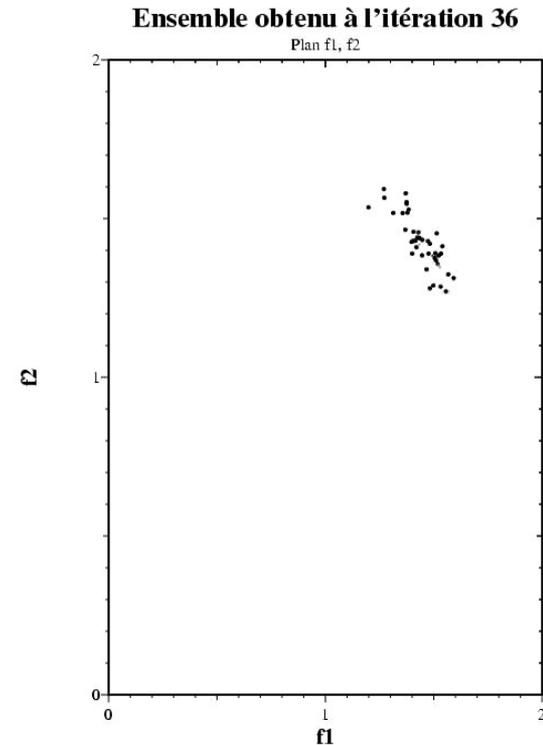
Pb Non-convex + Pondération des FObjs

On effectue un premier test sur un problème non-convexe continu.

Mon Sep 17 18:17:10 2001



Mon Sep 17 16:46:34 2001



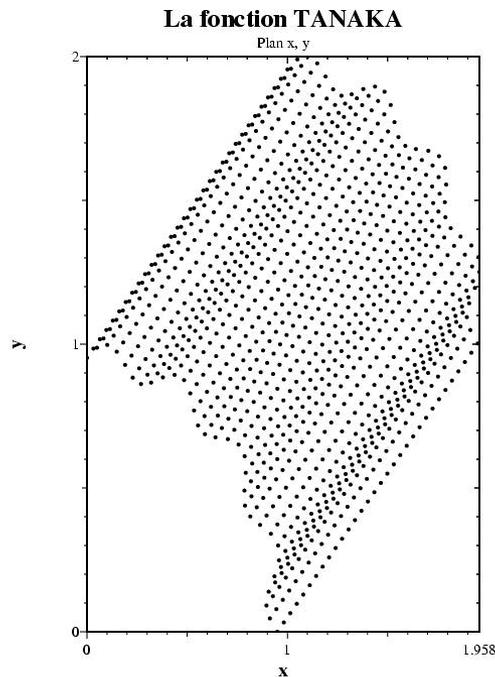


# Quelques exemples Continus

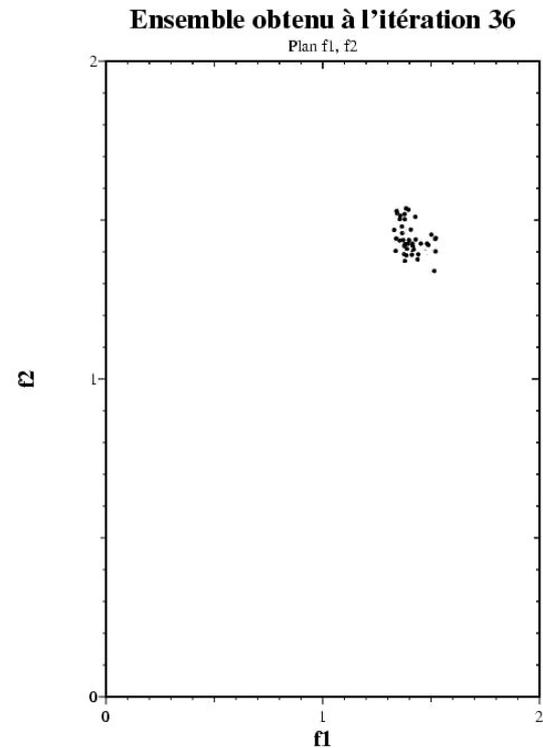
Pb Discontinu + Méthode de Tchebychev

On effectue un test sur un problème discontinu.

Mon Sep 17 18:17:12 2001



Mon Sep 17 17:01:48 2001



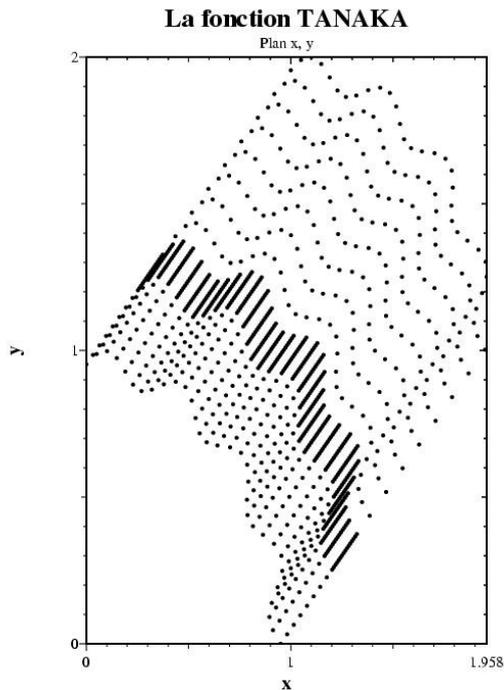


# Quelques exemples Continus

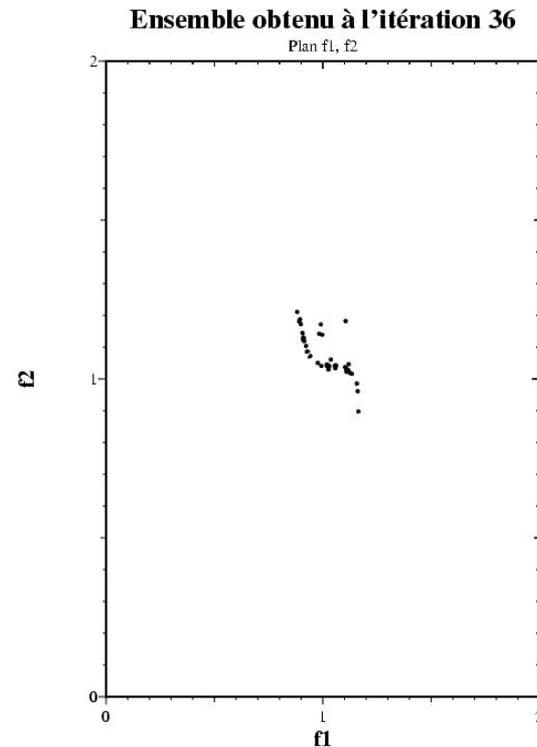
Pb Discontinu + Méthode de Tchebychev

On effectue un test sur un problème  
discontinu avec une variation de densité.

Mon Sep 17 18:17:14 2001



Mon Sep 17 17:18:00 2001



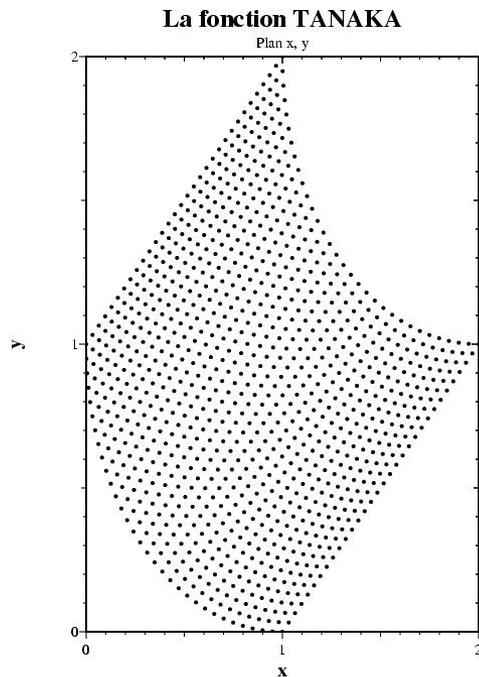


# Quelques exemples Combinatoires

Pb Convex + Pondération des FObjs

On effectue un test sur un problème  
convexe.

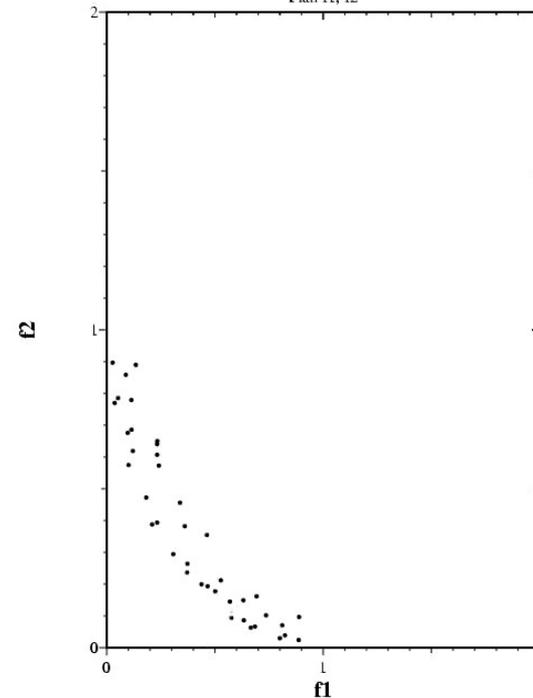
Mon Sep 17 18:17:08 2001



Thu Sep 13 16:08:20 2001

**Ensemble obtenu à l'itération 37**

Plan f1, f2



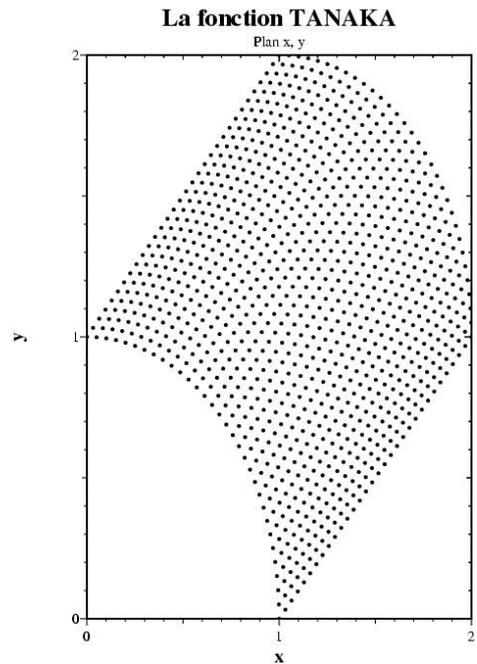


# Quelques exemples Combinatoires

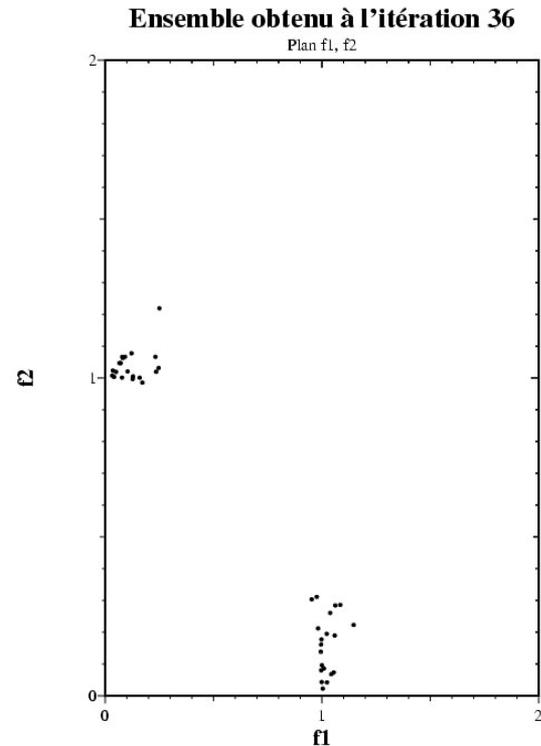
Pb Non convexe + Pondération des FObjs

On effectue un test sur un problème non-convexe.

Mon Sep 17 18:17:10 2001



Mon Sep 17 15:49:05 2001



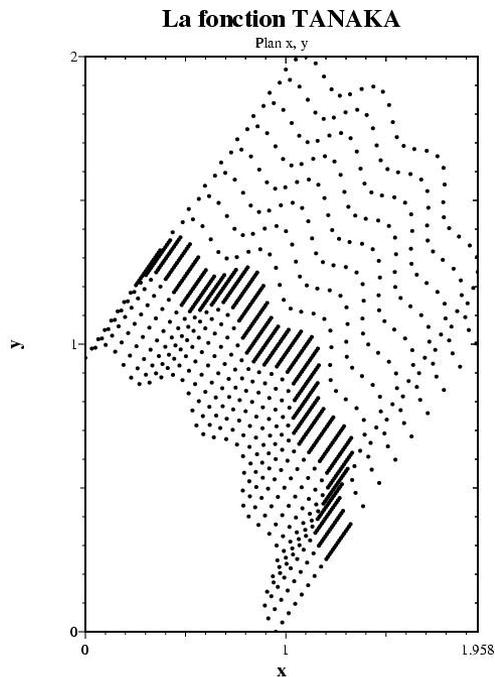


# Quelques exemples Combinatoires

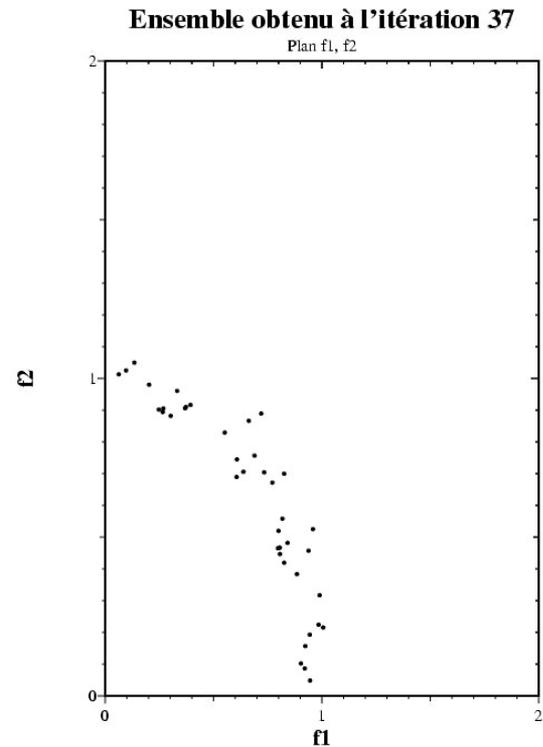
Pb Discontinu + Méthode de Tchebychev

On effectue un test sur un problème  
discontinu avec une variation de densité.

Mon Sep 17 18:17:14 2001



Mon Sep 17 17:51:06 2001



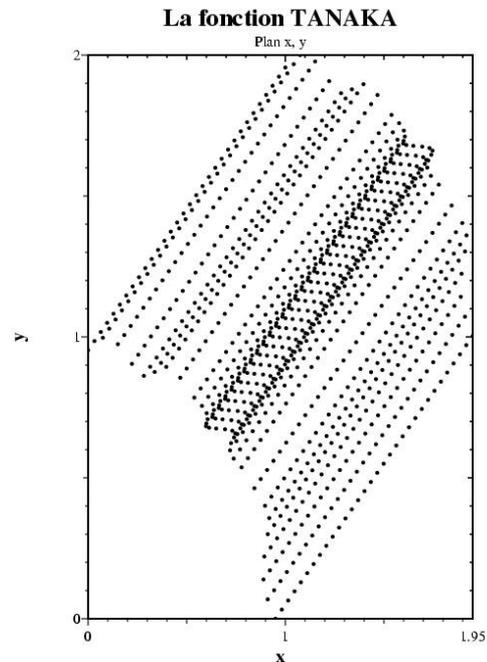


# Quelques exemples Combinatoires

Pb Discontinu + Méthode de Tchebychev

On effectue un test sur un problème  
discontinu avec une variation de densité.

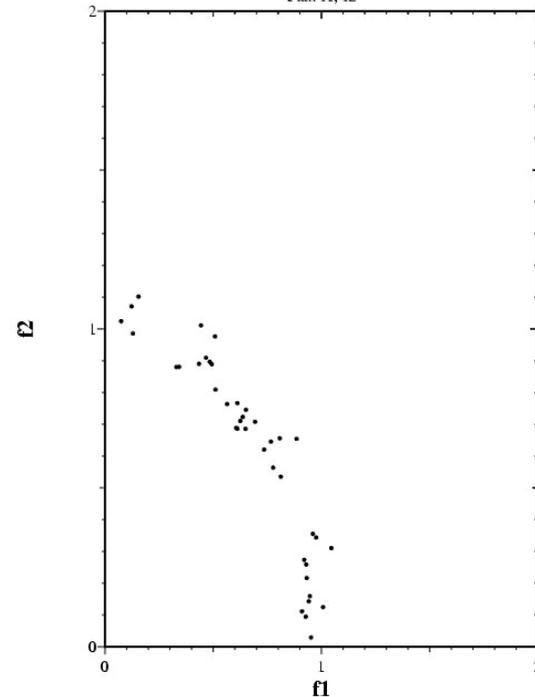
Mon Sep 17 18:17:16 2001



Mon Sep 17 18:02:47 2001

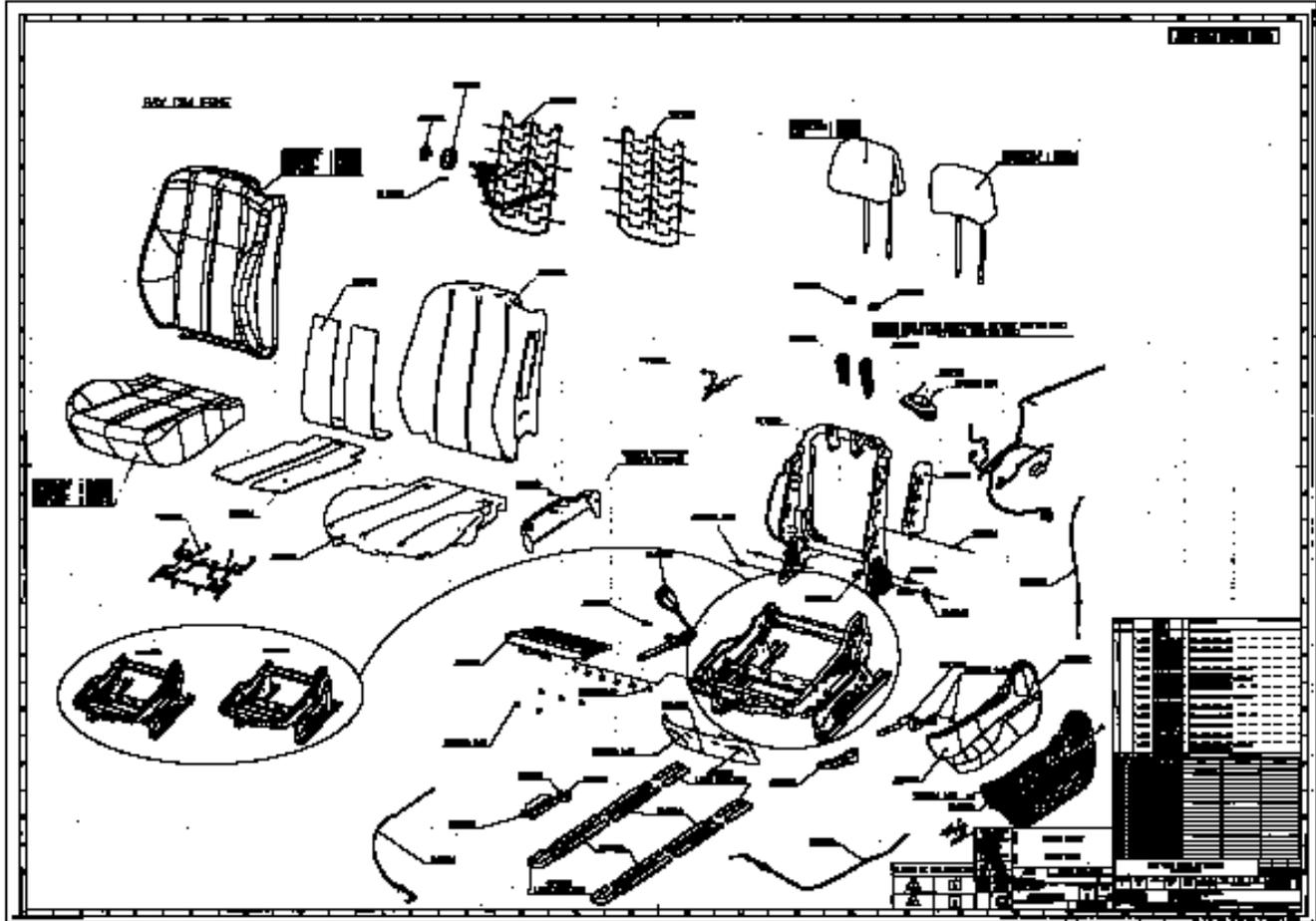
**Ensemble obtenu à l'itération 36**

Plan f1, f2





# Tolérance





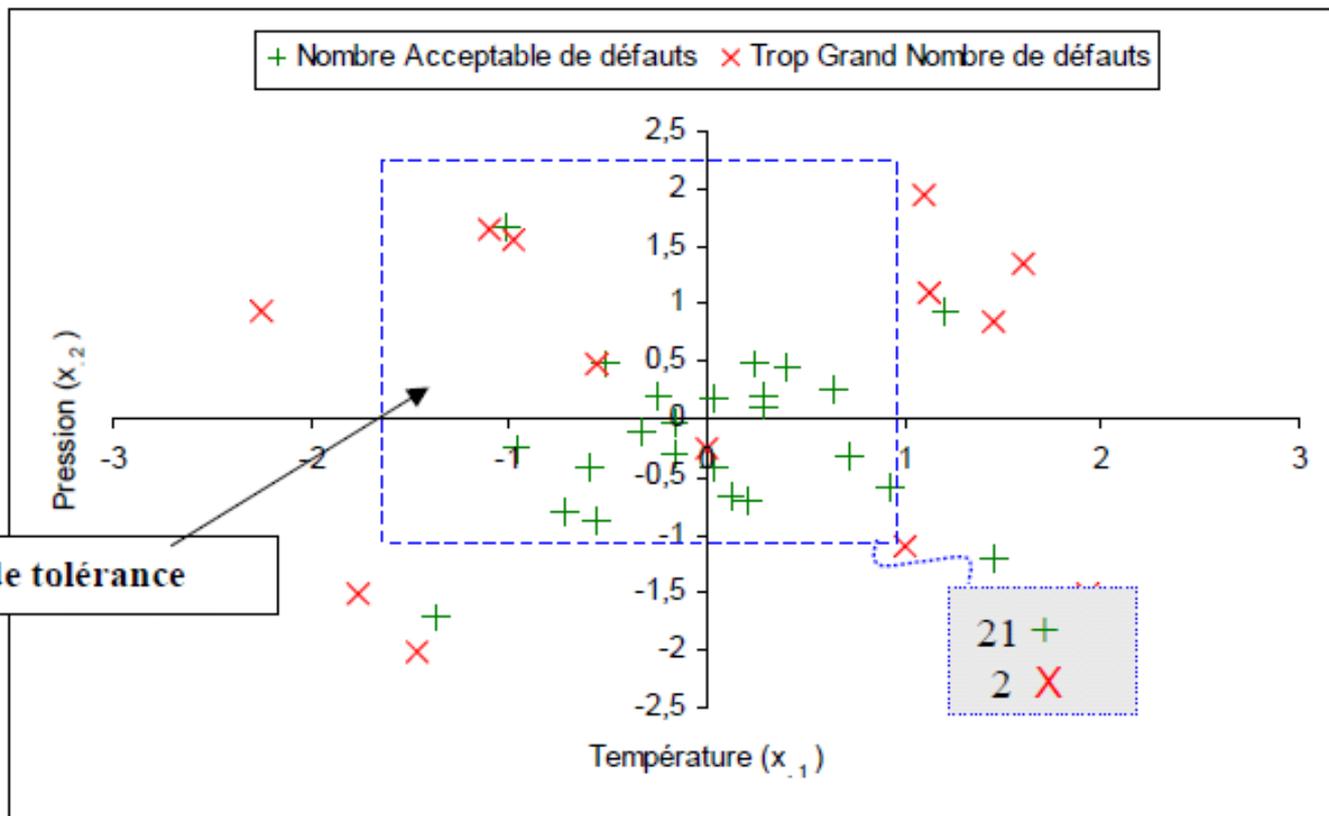
# Tolérance

## Présentation du processus

- Campagne d'essais sur un dispositif: des testeurs humains jugent la qualité du dispositif (bruit moteur, siège, etc ...)
- Détermination d'un critère objectif permettant de reproduire le jugement des essayeurs (phase d'objectivation)
  - Utilisation du Multidimensional Data Scaling: projection non linéaire d'un ensemble multidimensionnel vers un ensemble à dimension 2. On conserve la distance entre points (voir xgvis / xgobi)
- Définition de boîtes de tolérances sur différents paramètres techniques afin de rendre le dispositif attrayant au futur utilisateur



# Tolérance Objectif





# Tolérance

$b(A)$  nombre de bonnes réalisations dans la boîte A

$b$  nombre total de bonnes réalisations

$m(A)$  nombre de réalisations moyennes dans la boîte A

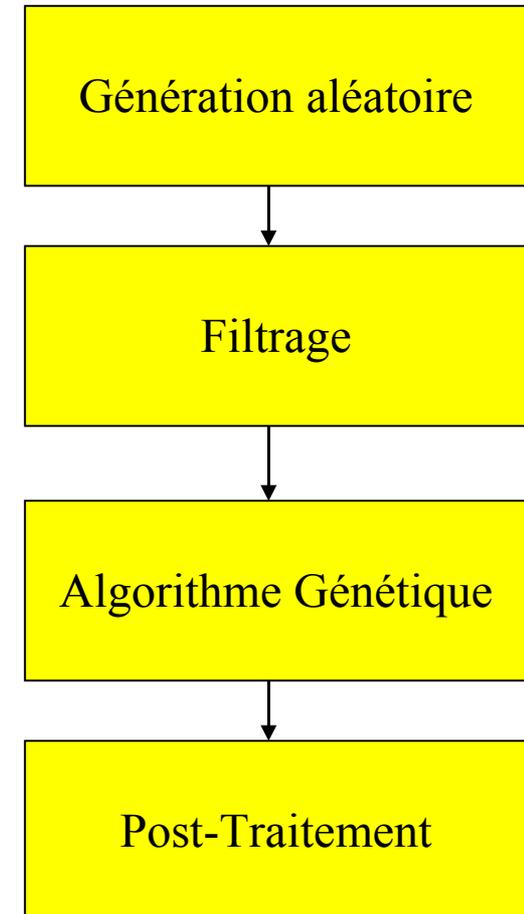
$m$  nombre total de réalisations moyennes

$\beta$  poids d'une réalisation moyenne

Application d'un algorithme génétique  
Multiobjectif (NSGA)

$$\text{Densité interne (boîte A)} = \frac{b(A) + \beta * m(A)}{\text{card}(A)}$$

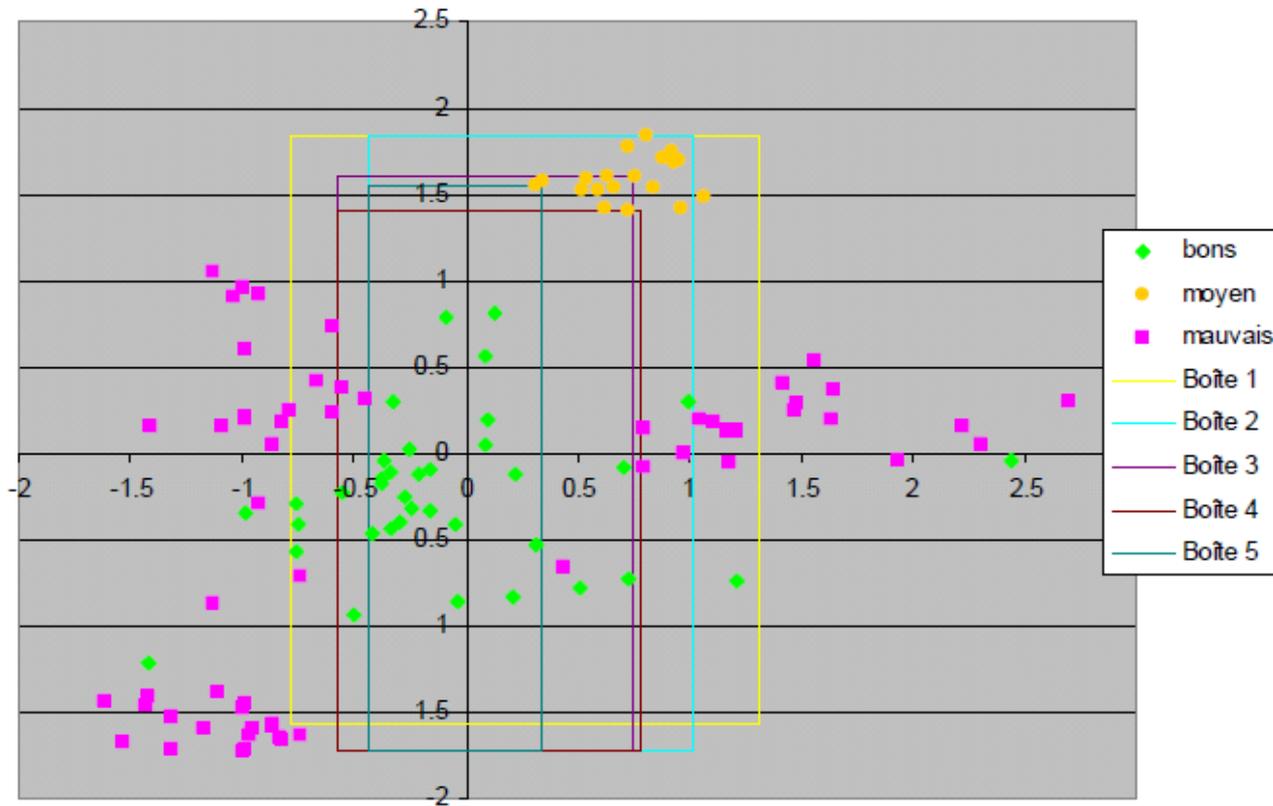
$$\text{Densité externe (boîte A)} = \frac{b(A) + \beta * m(A)}{b + \beta * m}$$





# Tolérance

Exemple de calcul de tolérances à trois modalités en deux dimensions





# Optimisation Robuste

Y. Collette



# Plan

- Introduction
- Le problème général
- La méthode FORM / SORM
- Exemple de la poutre
- La méthode de Monte-Carlo
- D'autres formulations du problème de départ
- Mode d'emploi
- Conclusion



# Un cas d'anthologie





# Objectifs

- Optimisation classique
  - On trouve une solution déterministe
- Réalité (risque **COP** – Conformity Of Production)
  - Les tolérances de fabrication: ça existe !
  - Les contraintes ne sont plus vérifiées à chaque fois
- Optimisation robuste
  - Prise en compte de ces incertitudes sur les paramètres dans le processus d'optimisation



# Fiabilité / Robustesse

- Fiabilité

C'est la probabilité qu'un système soit en défaut

- Robustesse

C'est la probabilité qu'un système fonctionne bien

$$R = 1 - P_f$$



# Conception robuste

## Sources de dispersions

### VIEILLISSEMENT

bouchage
matériaux (fatigue, chimie)
assemblage (usure)
fuites
corrosion
état de surface (usure)

### PROJET

changement matériaux
modification environnement technique
choix design
choix fournisseurs

### FABRICATION

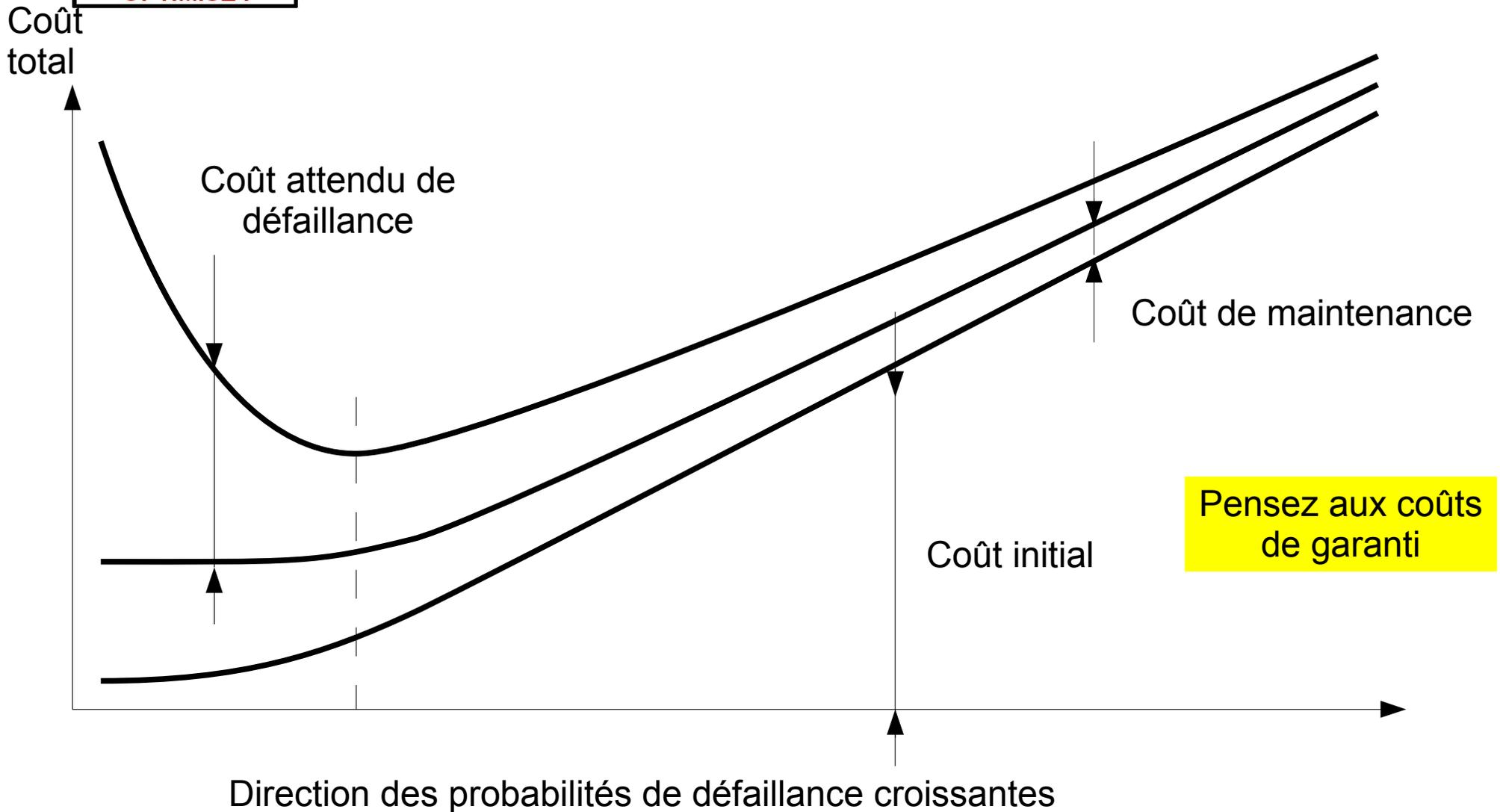
géométrie des pièces
caractéristiques matériaux
composants fournisseurs
fixation
réglage
montage
moyen, déroulement process
état de surface

### USAGE

température
gamme véhicules/moteurs
utilisation client
géométrie pièces voisines
perturbations EM
Conditions atmosph.
nature du choc
gonflage pneumatiques
vitesse



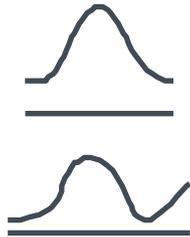
# Intérêt de l'optimisation fiabiliste / robuste



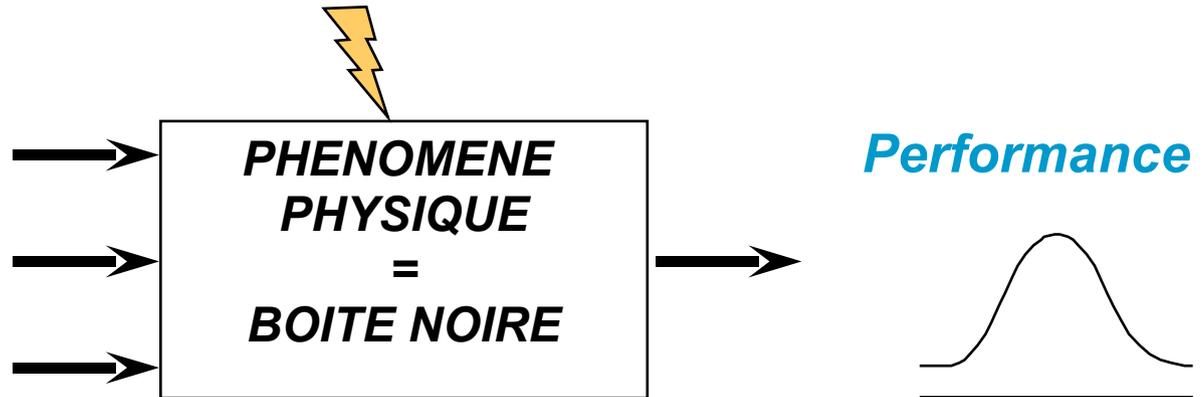


# Conception robuste

*Paramètres non contrôlés*



*Paramètres  
de  
conception*



## Analyse de robustesse d'une configuration a posteriori :

On simule la dispersion aléatoire des paramètres (conception et non contrôlés) et on mesure l'impact sur la performance.

→ **Méthode de Monte-Carlo**

Avant

## Mieux, l'optimisation robuste :

On prend en compte la dispersion aléatoire des paramètres (conception et non contrôlés) dès la conception !

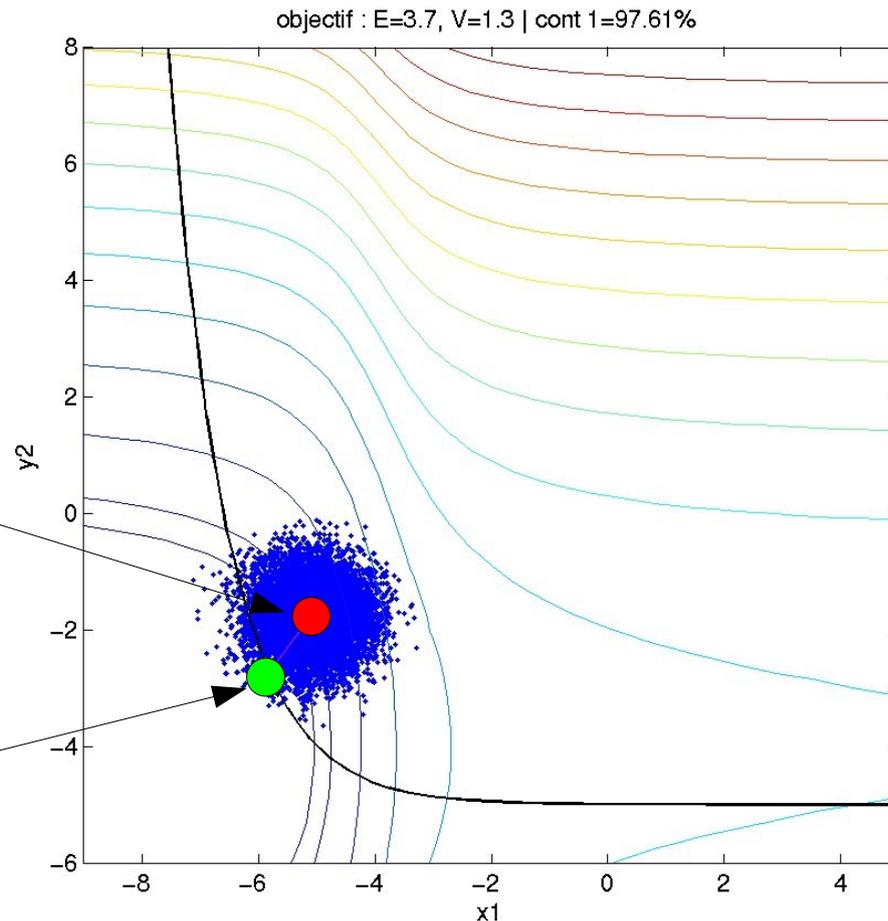
→ **Reformulation des contraintes et de l'objectif !**

Aujourd'hui



# Conception robuste

Illustration de l'impact de la dispersion dans les contraintes.



Solution optimale  
et robuste

Solution optimale

Optimisation  
robuste

Taux de défaillance  
2,39 %

Comment contrôler le taux de défaillance ?



# Le problème général

minimiser  $f(x)$   
avec  $g(x) < 0$

← Problème de départ

→ Problème général  
avec  $X$  variable aléatoire:  
 $X \sim N(\mu, \sigma)$

minimiser  $f(X)$   
avec  $P(g(X) < 0) > P_r$

Deux problématiques:

- Variabilité de la contrainte
- Variabilité de la fonction objectif



# Quelle formulation

- Formulation probabiliste

minimiser  $f(x)$

avec  $P(g(x) < 0) > \alpha$

FORM  
SORM  
Monte Carlo

- Formulation k-Sigma (méthode 6-Sigma)

minimiser  $f(x)$

avec  $\mu(g(x)) + k \cdot \sigma(g(x)) < 0$

Moment Matching Formulation  
Unscented Transform  
Monte Carlo

- Formulation pire cas

minimiser  $f(x)$

avec  $g(x + \delta x) < 0$

$g(x - \delta x) < 0$

Corner Space Evaluation  
Worst Case Analysis



# Optimisation Robuste

Formulation Probabiliste:  
FORM / SORM



# FORM / SORM

- FORM: First Order Reliability Method
- SORM: Second Order Reliability Method
- Principe:
  - On a une solution donnée (les  $x_i$  sont fixés)
  - On définit les variabilités de chaque paramètre
  - On se place dans un domaine Centré / Réduit
  - On évalue la probabilité de défaillance de la contrainte



# Le problème général

minimiser  $f(x)$   
avec  $g(x) < 0$  ← Problème de départ

Problème général  
avec  $X$  variable aléatoire:  
 $X \sim N(\mu, \sigma)$

minimiser  $f(X)$   
avec  $P(g(X) < 0) > P_r$

Deux problématiques:  
• Variabilité de la contrainte  
• Variabilité de la fonction objectif

$X \rightarrow$  Transformation iso-probabiliste (Rozenblatt)  $\rightarrow U$

Problème « simplifié »  
avec  $U$  variable aléatoire  
 $U \sim N(0, 1)$

minimiser  $f(U)$   
avec  $P(g'(U) < 0) > P_r$



# La transformation iso-probabiliste

Transformation qui conserve la distribution cumulée

$$F_{x_i}(x_i) = \Phi(u_i)$$

$$u_i = \Phi^{-1}(F_{x_i}(x_i))$$

Exemple: passage d'une gaussienne non-normalisée à une gaussienne normalisée

$$X \sim N(\mu, \sigma) \rightarrow U = \frac{X - \mu}{\sigma} \rightarrow X = \mu + \sigma U \rightarrow U \sim N(0,1)$$

Application de cette transformation au calcul de la probabilité de défaut

$$P_f = P(g(X) > 0) = \int_{g(X) > 0} f_x(x) dx = \int_{g'(U) > 0} \phi_U(u) du = P(g'(U) > 0)$$

$U$  Loi normale centrée réduite

$X$  Autre loi

$\Phi$  Distribution cumulée de  $U$

$F$  Distribution cumulée de  $X$

Etape 1: se ramener à des VA gaussiennes centrées réduites pour travailler sur un problème normalisé

D'autres transformations:  
Rosenblatt, Nataf



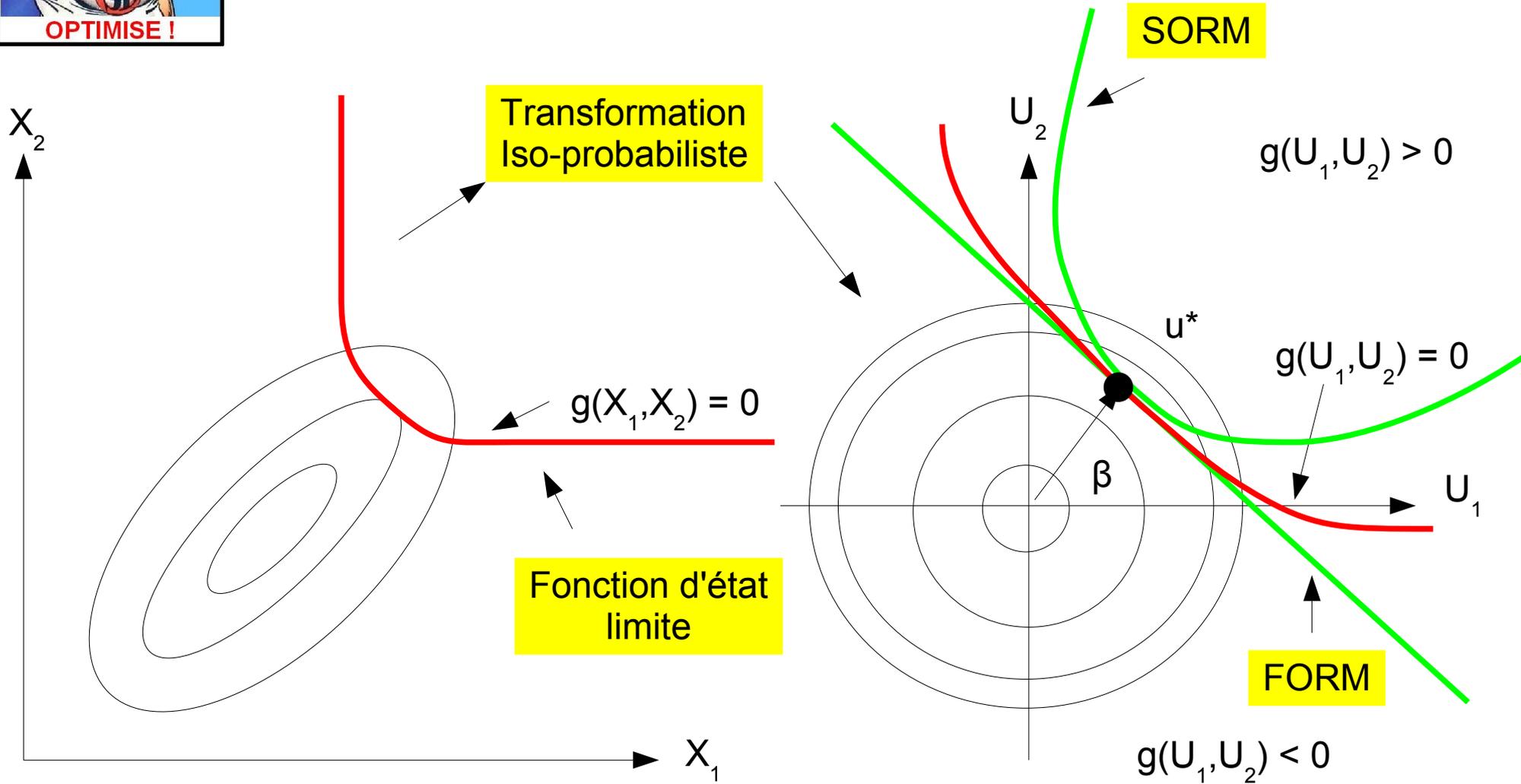
# Transformation à partir de ddp courantes

Loi	Paramètres	DDP	Transformation
Normale	$\mu, \sigma$	$p_1(x, \mu, \sigma) = \frac{1}{\sqrt{2\pi}\sigma} \exp\left(-\frac{(x-\mu)^2}{2\sigma^2}\right)$	$X = \mu + \sigma U$
Log-Normale	$\bar{\sigma}^2 = \ln(1 + (\sigma/\mu)^2)$ $\bar{\mu} = \ln(\mu) - 0.5\bar{\sigma}^2$	$p_2(x, \mu, \sigma) = p_1(x, \bar{\mu}, \bar{\sigma})$	$X = \exp(\mu + \sigma U)$
Weibull	$k > 0, \mu = v\Gamma(1+1/k)$ $\sigma^2 = v^2(\Gamma(1+2/k) - \Gamma^2(1+1/k))$	$p_3(x) = \frac{k}{v} \left(\frac{x}{v}\right)^{k-1} \exp\left(-\left(\frac{x}{v}\right)^k\right)$	$X = v \cdot (-\ln(\phi(-U)))^{(1/k)}$
Gumbel	$\mu = v + (0.577/\alpha)$ $\sigma = \pi/\sqrt{\sigma\alpha}$	$p_4(x) = \alpha \exp(-\alpha(x-v) - \exp(-\alpha(x-v)))$	$X = v - \ln(-\ln(\phi(U)))$
Uniforme	$\mu = (a+b)/2$ $\sigma = (b-a)/\sqrt{12}$	$p_5(x) = \frac{1}{b-a}$	$X = a + (b-a)\phi(U)$

$$\phi(u) = 1/\sqrt{2\pi} \exp(-u^2/2)$$



# Représentation graphique





# FORM

Comment s'exprime  $\Phi_U(u)$  ?

Après transformation iso-probabiliste, les VA sont indépendantes

$$\phi_U(u) = \prod_{i=1}^n \frac{1}{\sqrt{2\pi}} \exp(-1/2 u_i^2)$$

Etape 2: approximation de la frontière d'intégration autour d'un point  $u^*$

$$L(U) \simeq g'(u^*) + \nabla g(u^*)(U - u^*)^t$$

Question: Comment choisir  $u^*$  ?

Pour minimiser la perte de précision due à l'approximation, on linéarise  $g'(u)$  autour du point qui contribue le plus à l'intégration de la densité de probabilité (ddp):

- C'est le point qui a la ddp la plus grande sur la courbe  $g'(u)=0$ .
- On appelle ce point « Most Probable Point ».



# FORM:

## Recherche du Most Probable Point

Le problème à résoudre est alors le suivant:

$$\begin{aligned} &\underset{u}{\text{maximiser}} && \prod_{i=1}^n \frac{1}{\sqrt{2\pi}} \exp(-1/2 u_i^2) \\ &\text{avec} && g'(u) = 0 \end{aligned}$$

Pour simplifier le problème, on utilise l'équivalence suivante:

$$\underset{u}{\text{maximiser}} \prod_{i=1}^n \frac{1}{\sqrt{2\pi}} \exp(-1/2 u_i^2) \equiv \underset{u}{\text{minimiser}} \sum_{i=1}^n u_i^2 \equiv \underset{u}{\text{minimiser}} \|u\|$$

Le problème devient alors:

$$\begin{aligned} &\underset{u}{\text{minimiser}} && \|u\| \\ &\text{avec} && g'(u) = 0 \end{aligned}$$

Problème à résoudre  
pour trouver le point le  
plus probable



# Différentes approches

RIA : Reliability Index Approach  
→ utilisation de l'analyse fiabiliste

minimiser  $\|u\|$   
avec  $G(u)=0$

PMA : Performance Measure Approach  
→ utilisation de l'analyse fiabiliste inverse

minimiser  $G(u)$   
avec  $\|u\|=\beta_t$

Trouver la probabilité de défaillance

Pour une probabilité de défaillance donnée, trouver la valeur de la contrainte

Mettre les formules pour les sensibilités dans le cas de plusieurs contraintes



# La probabilité de défaut

On effectue une linéarisation de l'équation d'état limite au point le plus probable

Au point  $u^*$ , on a  $g'(u^*)=0$ . On exprime la moyenne et la variance de la contrainte

$$L(U) = \sum_{i=1}^n \left. \frac{\partial g'(U)}{\partial u_i} \right|_{u_i^*} (u_i - u_i^*)$$

Moyenne et variance de  $L(U)$ :

$$\mu_L = - \sum_{i=1}^n \left. \frac{\partial g'(U)}{\partial u_i} \right|_{u_i^*} u_i^* \quad \sigma_L = \sqrt{\sum_{i=1}^n \left( \left. \frac{\partial g'(U)}{\partial u_i} \right|_{u_i^*} \right)^2}$$

Ces deux éléments seront utilisés par la suite pour le calcul de la probabilité de défaut



# La probabilité de défaut

Calcul de la probabilité de défaut:

$$P_f = P(L(U) < 0) = \Phi\left(\frac{-\mu_L}{\sigma_L}\right) = \Phi\left(\sum_{i=1}^n \alpha_i u_i^*\right)$$

Par définition

$$\alpha_i = \frac{\left. \frac{\partial g'(U)}{\partial u_i} \right|_{u_i^*}}{\sqrt{\sum_{i=1}^n \left( \left. \frac{\partial g'(U)}{\partial u_i} \right|_{u_i^*} \right)^2}} \rightarrow a = (\alpha_1, \dots, \alpha_n) = \frac{\nabla g'(u^*)}{\|\nabla g'(u^*)\|}$$

$$P_f = \Phi\left(\sum_{i=1}^n \alpha_i u_i\right) = \Phi(a u^{*t})$$

$$\frac{u^*}{\|u^*\|} = \frac{u^*}{\beta}$$

Par définition

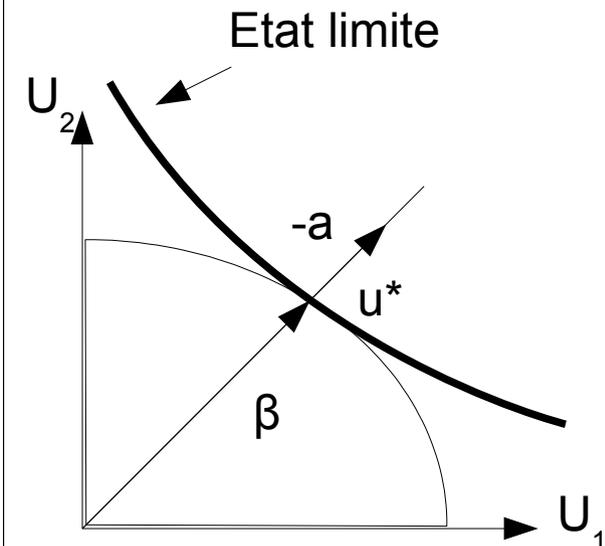
$$\text{On a } -a = \frac{u^*}{\beta} \rightarrow u^* = -\beta a$$

Au point le plus probable: égalité des vecteurs unitaires

Expression de la probabilité de défaut:

$$P_f = P(L(U) < 0) = \Phi(a u^{*t}) = \Phi(-\beta a a^t) = \Phi(-\beta)$$

$$R = 1 - P_f = 1 - \Phi(-\beta) = \Phi(\beta)$$





# FORM

## Comment trouver $\beta$ ?

Plusieurs méthodes sont applicables:

- La méthode Hasofer-Lind / Rackwitz-Fiessler (HL/RF)
- La méthode Improved HL/RF
- La méthode Abdo - Rackwitz

Sous scilab:

```
Pf = cdfnor("PQ", -1.7444, 0.0, 1.0)
```

Méthode HL/RF (Hasofer-Lind / Rackwitz-Fiessler):

$$u^{k+1} = \left[ \frac{-\nabla U^k}{\|\nabla U^k\|} \cdot U^k + \frac{g(U^k)}{\|\nabla U^k\|} \right] \cdot \frac{-\nabla U^k}{\|\nabla U^k\|}$$

On applique cette équation jusqu'à ce que:

$$\|U^{k+1} - U^k\| \leq \delta \text{ et } g(U^k) \leq \epsilon$$



# Calcul de $\beta$ dans un processus d'optimisation

Expression de la dérivée de la probabilité de défaillance en fonction des variables du problème d'optimisation

Corriger la probabilité de défaillance en fonction du coté où l'on se situe de l'état limite

$$\nabla_{\mu} P_{FORM} = \Phi(\beta(\mu)) \cdot \nabla_{\mu} \beta(\mu)$$

$$\nabla_{\mu} \beta(\mu) = J(x^*) \cdot \frac{\nabla_u \beta(u^*)}{\|\nabla_u \beta(u^*)\|}$$

si  $g(u^*) \leq 0$  alors  $P_f' = P_f$   
 sinon  $P_f' = 1 - P_f$

Gaussienne dans l'espace normé - centré

Jacobien transformation iso-probabiliste

Vecteur des paramètres d'entrée

$\mu$  dans l'espace d'origine



# Quand il y a plusieurs contraintes

	Borne Supérieure	Composantes indépendantes	Borne Inférieure
Série (et)	$P_f = \sum_{i=1}^n P_{f_i}$	$P_f = \prod_{i=1}^n P_{f_i}$	$P_f = \max_i P_{f_i}$
Parallèle (ou)	$P_f = \min_i P_{f_i}$	$P_f = 1 - \sum_{i=1}^n (1 - P_{f_i})$	$P_f = 0$

L'optimisation utilise plutôt des contraintes du type « et »

$$\begin{aligned} &\text{minimiser } f(x) \\ &\text{avec } g_1(x) \leq 0 \\ &\quad g_2(x) \leq 0 \\ &\quad L \leq w \leq U \end{aligned}$$



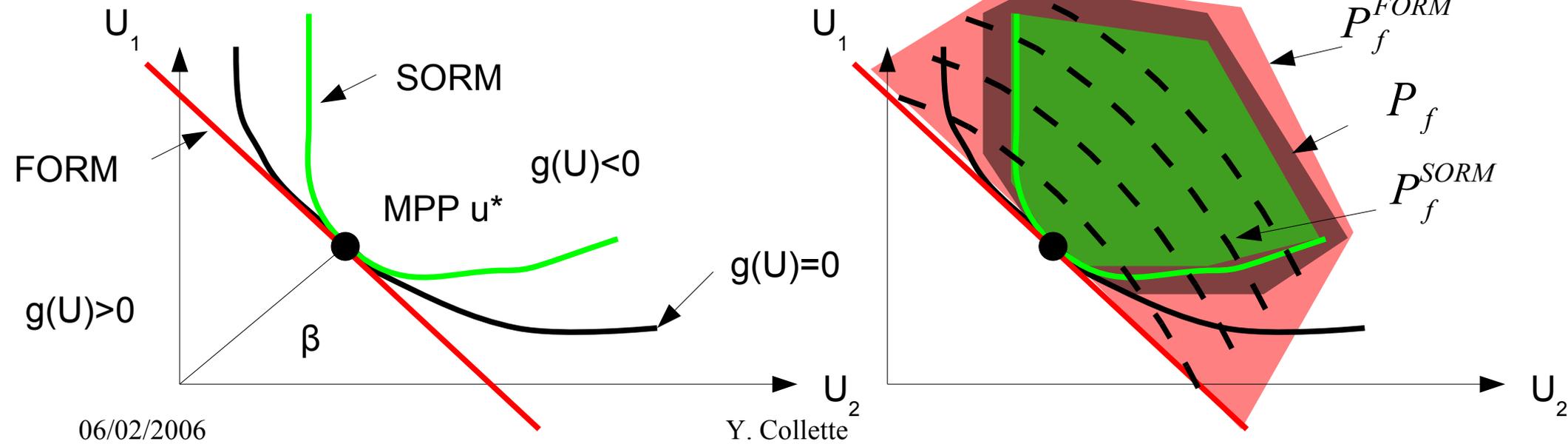
# FORM / SORM

FORM: First Order Reliability Method -> On approxime la fonction de performance par une fonction linéaire

$$g(U) \simeq l(u) = g(u^*) + \nabla g(u^*)(U - u^*)^t$$

SORM: Second Order Reliability Method -> On approxime la fonction de performance par une fonction quadratique

$$g(U) \simeq q(u) = g(u^*) + \nabla g(u^*)(U - u^*)^t + \frac{1}{2}(U - u^*)H(u^*)(U - u^*)^t$$





# SORM

FORM  $P_f = \Phi(-\beta)$

Hasofer – Lind (74)

SORM  $P_f = \Phi(-\beta) \prod_{i=1}^n (1 + \beta k_i)^{\frac{1}{2}}$

Breitung (84)

$$P_f = \Phi(-\beta) \prod_{i=1}^n \left( 1 + \frac{\phi(\beta)}{\Phi(-\beta)} k_i \right)$$

Hohenbircher – Rackwitz (88)

Courbature  $k_i$  - ce sont les valeurs propres de la matrice A :

$$A = \frac{B^t D B}{\|\nabla g_j(x^k)\|}$$

$\phi(y)$  ddp gaussienne

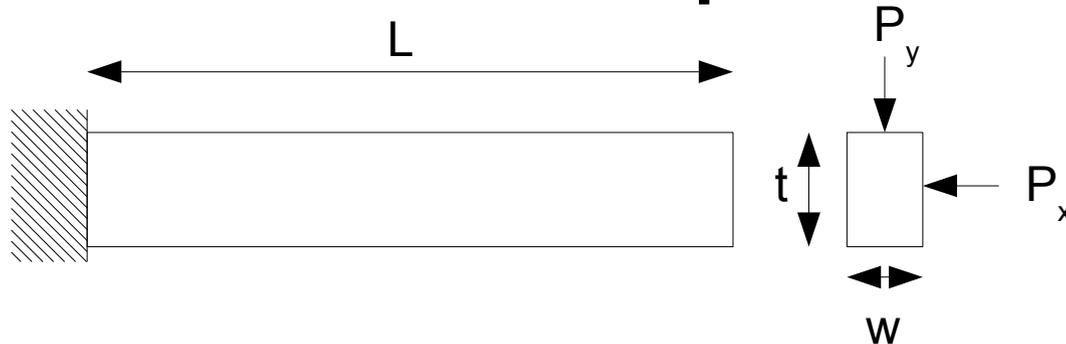
$D$  matrice Hessienne de  $g_j$  en  $x^k$

$B$  matrice orthogonale à la matrice  $B_0$

$$B_0 = \begin{pmatrix} 1 & 0 & \dots & 0 \\ 0 & 1 & \dots & 0 \\ \dots & \dots & \dots & \dots \\ \dots & \dots & 1 & 0 \\ \frac{\partial g_j}{\partial x_i} & \dots & \dots & \frac{\partial g_j}{\partial x_n} \end{pmatrix}$$



# Exemple : Fiabilité d'une barre - Pose du problème



Expression de la contrainte:

$$g(P_x, P_y) = D_0 - \frac{4L^3}{E \cdot w \cdot t} \sqrt{\left(\frac{P_y}{t}\right)^2 + \left(\frac{P_x}{w}\right)^2}$$

Expression de la probabilité de défaillance:

$$P_f = P(g(P_x, P_y) \leq 0)$$

Proba que le déplacement  
en bout de barre soit  
supérieur à  $D_0$

Paramètres (unités anglaise):

$L = 100$  (longueur)

$w = 2$  (largeur)

$t = 4$  (hauteur)

$P_x = N(500, 100)$  (loi normale)

$P_y = N(1000, 100)$  (loi normale)

$D_0 = 3$  (déplacement)

$E = 30 \cdot 10^6$  (module d'élasticité)

Manque de chance: contrainte non convexe – HL-RF ne fonctionne pas



# Exemple : Fiabilité d'une barre - Etape préliminaire

Transformation de  $P_x$  et  $P_y$  vers des lois normales centrées réduites  $U_x, U_y$ :

$$U = (U_x, U_y) = \left( \frac{P_x - \mu_{P_x}}{\sigma_{P_x}}, \frac{P_y - \mu_{P_y}}{\sigma_{P_y}} \right) \rightarrow X = (P_x, P_y) = (\mu_{P_x} + U_x \sigma_{P_x}, \mu_{P_y} + U_y \sigma_{P_y})$$

Expression de la fonction de performance:

$$g(U_x, U_y) = D_0 - \frac{4L^3}{E.w.t} \sqrt{\left( \frac{\mu_{P_x} - U_x \sigma_{P_x}}{w^2} \right)^2 + \left( \frac{\mu_{P_y} - U_y \sigma_{P_y}}{t^2} \right)^2}$$

Expression du gradient de la fonction de performance:

$$\nabla g(U) = \frac{4L^3}{E.w.t} \left( \frac{\sigma_{P_x} (\mu_{P_x} + U_x \sigma_{P_x})}{w^4 \sqrt{\left( \frac{\mu_{P_x} - U_x \sigma_{P_x}}{w^2} \right)^2 + \left( \frac{\mu_{P_y} - U_y \sigma_{P_y}}{t^2} \right)^2}}, \frac{\sigma_{P_y} (\mu_{P_y} + U_y \sigma_{P_y})}{t^4 \sqrt{\left( \frac{\mu_{P_x} - U_x \sigma_{P_x}}{w^2} \right)^2 + \left( \frac{\mu_{P_y} - U_y \sigma_{P_y}}{t^2} \right)^2}} \right)$$



# Exemple : HL-RF

## Problème test 1

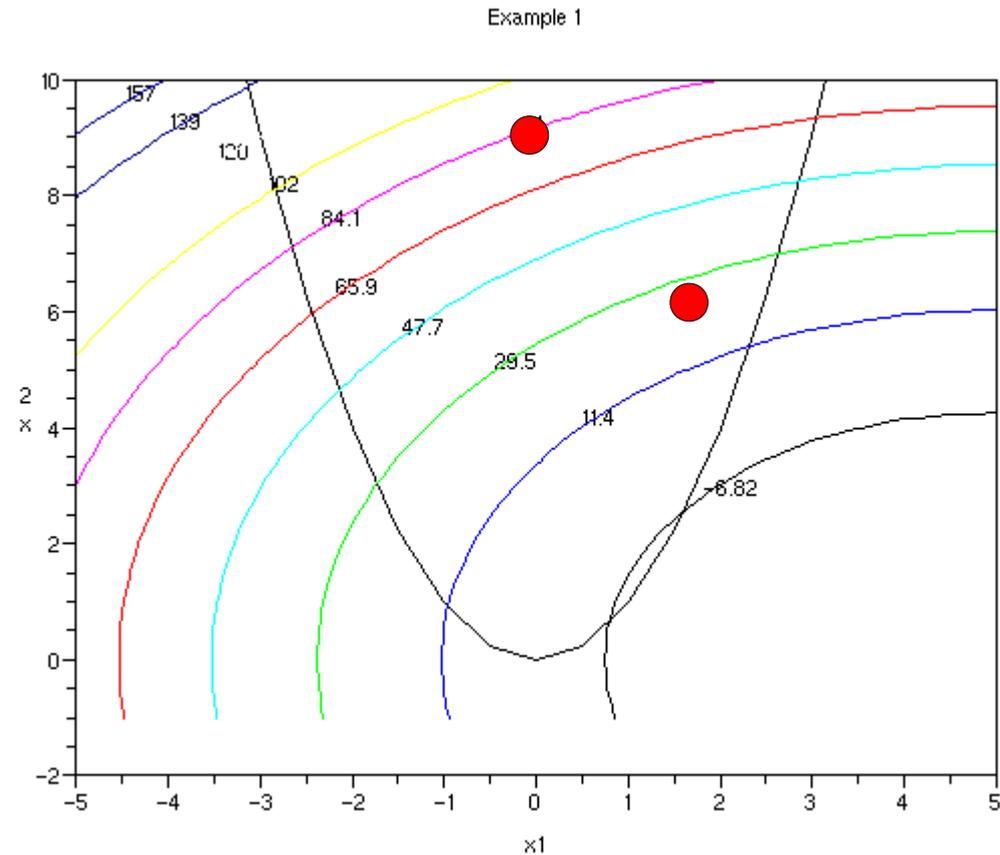
Problème test n°1: formulation déterministe

$$f(x) = (x_1 - 5)^2 + x_2^2 - 25$$

$$g(x) = x_1^2 - x_2$$

Expression de la contrainte probabiliste

$$P(g(x) \leq 0) > 0.9$$





# Exemple : HL-RF

## Problème test 2

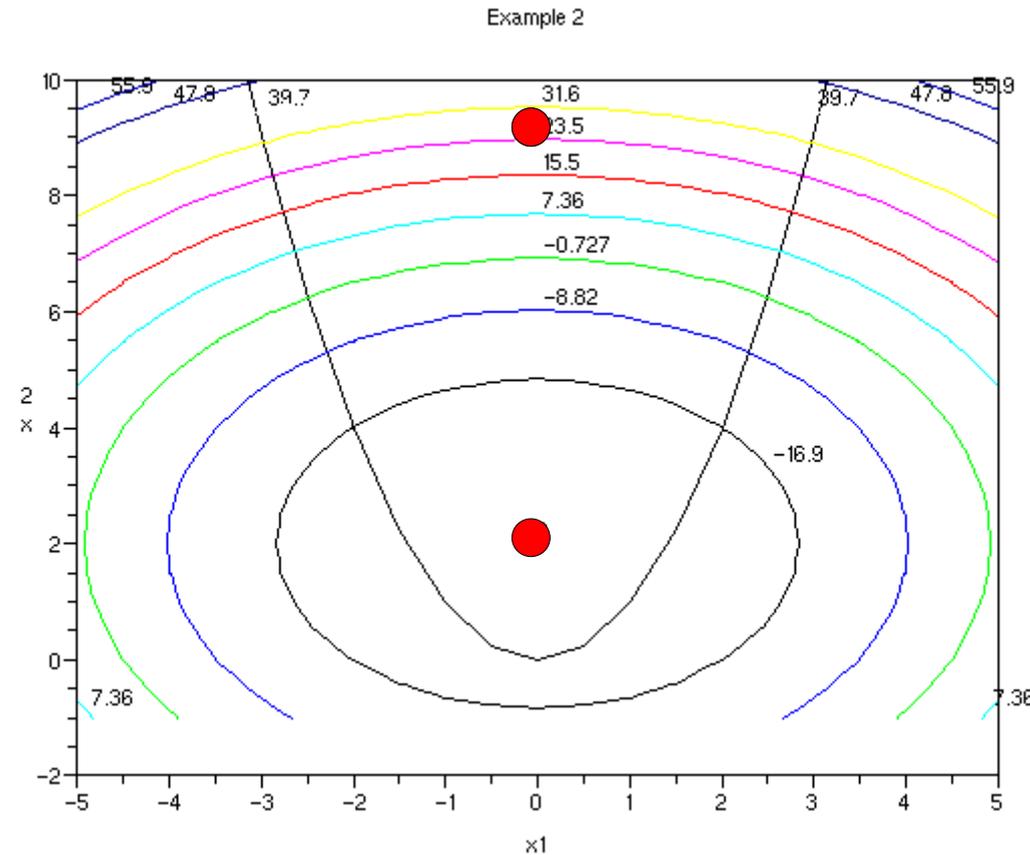
Problème test n°2: formulation déterministe

$$f(x) = x_1^2 + (x_2 - 2)^2 - 25$$

$$g(x) = x_1^2 - x_2$$

Expression de la contrainte probabiliste

$$P(g(x) \leq 0) > 0.9$$





# Exemple : HL-RF

## Problème test 3

Problème test n°3: formulation déterministe

$$f(x) = x_1 + x_2$$

$$g_1(x) = 20 - x_1^2 \cdot x_2$$

$$g_2(x) = 1 - (x_1 + x_2 - 5)^2 / 30 - (x_1 - x_2 - 12)^2 / 120$$

$$g_3(x) = x_1^2 + 8 \cdot x_2 - 75$$

Expression de la contrainte probabiliste

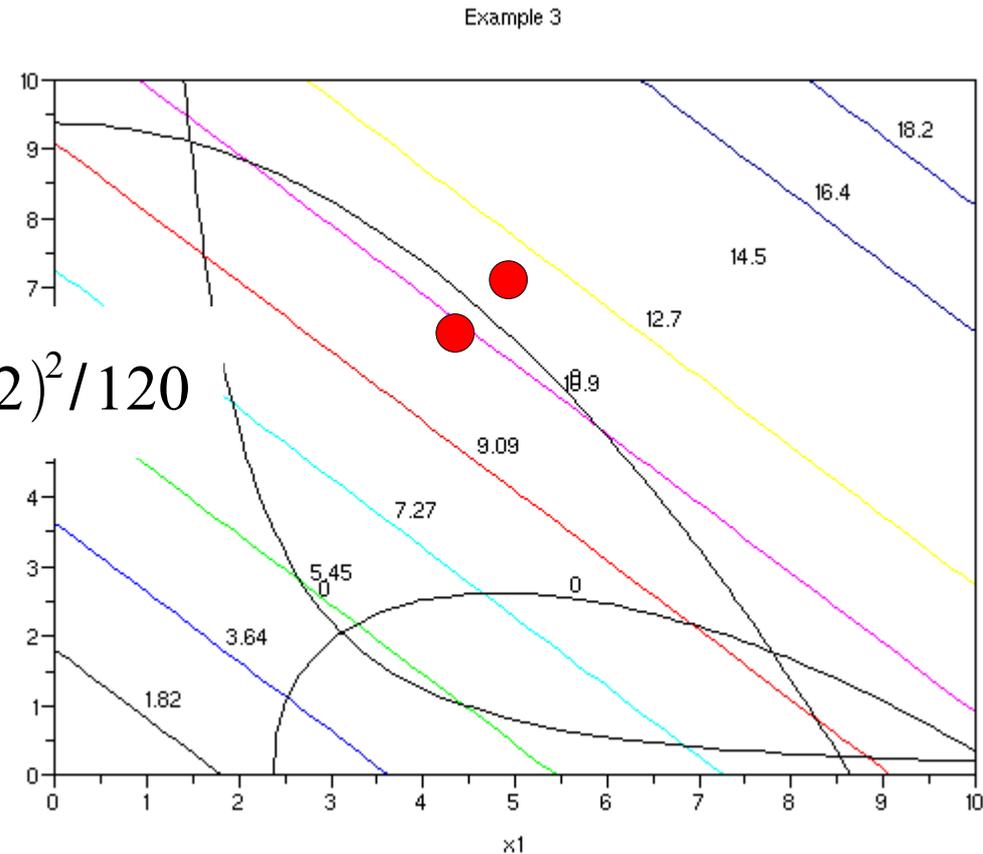
$$P(g_1(x) \leq 0) > 0.9$$

$$P(g_2(x) \leq 0) > 0.9$$

$$P(g_3(x) \leq 0) > 0.9$$

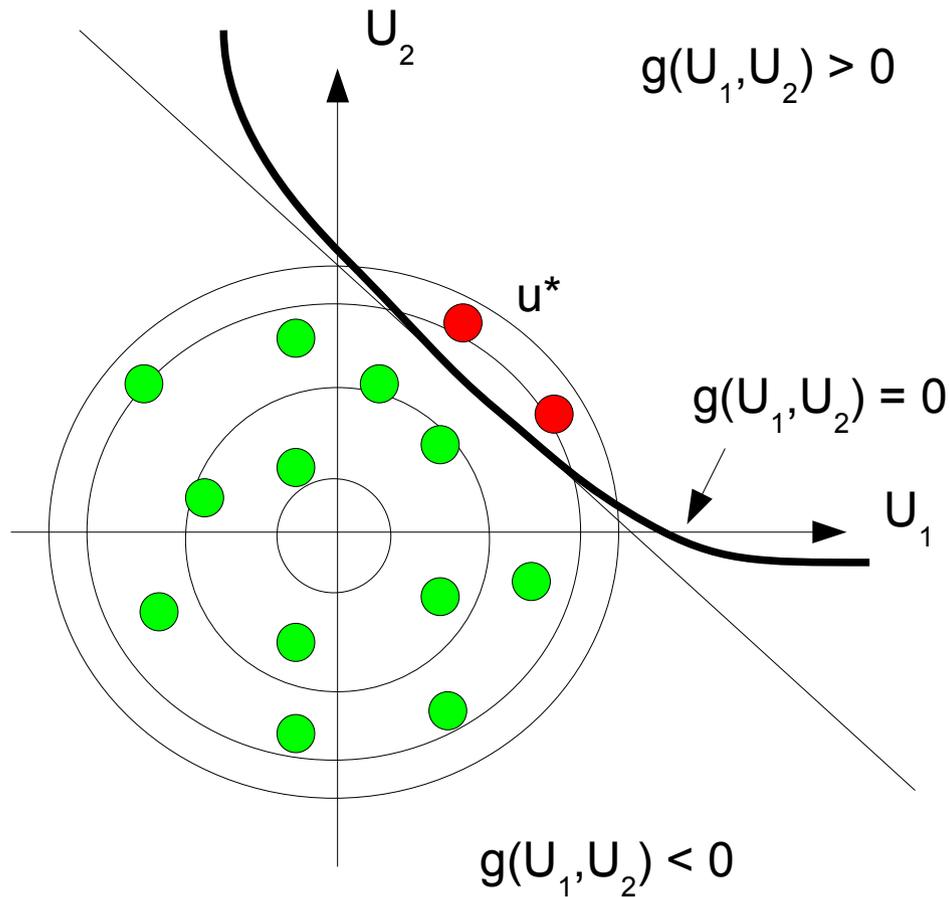
06/02/2006

Y. Collette





# Calcul de la probabilité de défaillance par Monte-Carlo



$$P_f = \int_{G(u) < 0} \Phi(u) du \simeq \frac{1}{N} \sum_{i=1}^N \Phi(u_i)$$

- On tire un grand nombre de points  $U^i$ .
- Si  $U^i$  est dans le plan  $g(U_1, U_2) > 0$  alors  
Somme +=  $\Phi(U^i)$   
N++
- $P_f = \text{Somme} / N$

On arrête le calcul quand:

$$\frac{\text{var}(P_f)}{\langle P_f \rangle} < K_{\text{limit}}$$

1- $\alpha$ : niv. de confiance

$q_\alpha$ : quantile associé

$\epsilon$ : précision souhaitée

$P_0$ : prob. de satisfaire la contrainte

$$N = \frac{q_\alpha^2 \cdot P_0}{\epsilon^2 \cdot (1 - P_0)}$$

Pour  $P_0=0.99$ ,  $\epsilon=0.1$ ,  $\alpha=0.05$  alors  $N = 38000$



# Optimisation Robuste

## La formulation k.Sigma



# Le problème général

minimiser  $f(x)$   
avec  $g(x) < 0$

← Problème de départ

Problème général  
avec  $X$  variable aléatoire:  
 $X \sim N(\mu, \sigma)$

→ minimiser  $f(X)$   
avec  $P(g(X) < 0) > P_r$

Deux problématiques:  
• Variabilité de la contrainte  
• Variabilité de la fonction objectif

↙ minimiser  $f(X)$   
avec  $E(g(X)) + k \cdot \text{Var}(G(x)) < 0$

On s'assure qu'une certaine variabilité de la  
contrainte n'entraîne pas de dépassement



# Variantes de reformulation

## Approche 6 Sigma

Cette approche repose sur l'hypothèse que  $g(x)$  suit une loi gaussienne  $N(\mu_g, \sigma_g)$

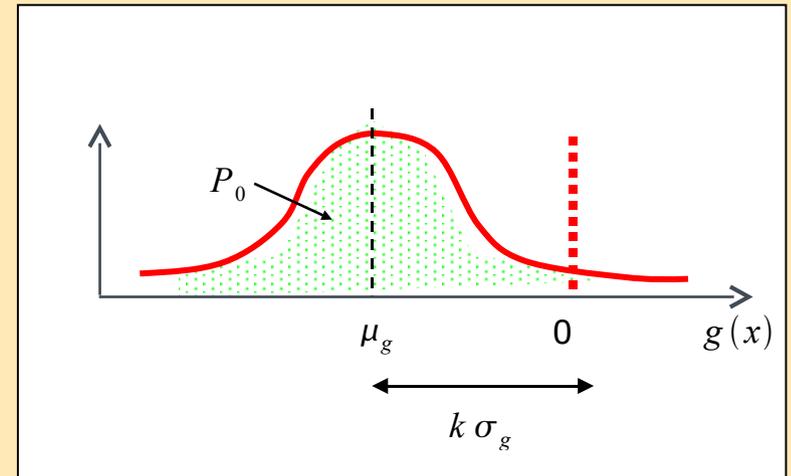
On reformule la contrainte  $g(x) \leq 0$

$$\mu_g + k \sigma_g \leq 0$$

avec

$$k = \Phi^{-1}(P_0)$$

→ DIFFICULTE : calcul de  $(\mu_g, \sigma_g)$



Comment déterminer k ?

$P_0 = 84,13 \%$	$k=1$
$P_0 = 97,73 \%$	$k=2$
$P_0 = 99,86 \%$	$k=3$
$P_0 = 99,9999999 \%$	$k=6$



# Unscented Transformation

L'idée : Approcher la distribution par un échantillon de points avec poids :  $\{g(x_i), W_i\}$

$$\text{Alors : } E[g] \simeq \sum_i W_i g(x_i), \sigma^2[g] \simeq \sum_i W_i (g(x_i) - E[g])^2$$

Le choix des points (et pondération) va dépendre des propriétés à vérifier.

Ex: l'échantillon dont la moyenne et covariance correspondent à  $(\mu_x, R_x)$  sera :

$$\begin{cases} x_0 = \mu & W_0 = k(n+k)^{-1} \\ x_i = \mu + (\sqrt{(n+k)R_x})_i & W_i = 0.5(n+k)^{-1} \\ x_{i+n} = \mu - (\sqrt{(n+k)R_x})_i & W_{i+n} = 0.5(n+k)^{-1} \end{cases}$$

où  $(\sqrt{(n+k)R_x})_i$  est la  $i$ ème ligne de la racine carrée de la matrice  $\sqrt{(n+k)R_x}$  et  $k$  un degré de liberté fixé à  $k=3-n$  quand  $x$  est un vecteur gaussien.

Dans le cas où  $x$  est un vecteur gaussien indépendant :  $x_i = \mu \pm (0, \dots, 0, \sqrt{3}\sigma_i, 0, \dots, 0)$

😊 + simplicité, différentiable, seulement  $(2n+1)$  points à évaluer

😞 - calcul de  $(n+k)R_x$



# Variantes de reformulation

Méthodes de calcul de  $(\mu_g, \sigma_g)$

- **Monte-Carlo** (Importance Sampling, LHS, Point Estimated Method, Unscented Transformation) :

$$E[g(x)] \simeq \frac{1}{N} \sum_{i=1}^N g(x_i)$$

😊 + *Précise*

😞 - *Coûteuse en temps de calcul*

$$\sigma^2[g(x)] \simeq \frac{1}{N} \sum_{i=1}^N (g(x_i) - \bar{g})^2$$

- **Moment Matching Formulation** : Approximation de Taylor

- **Au 1er ordre au voisinage de  $\mu$**  :

$$E[g(x)] \simeq g(\mu)$$

$$\sigma^2[g(x)] \simeq \sum_{i=1}^N (\nabla_{x_i} g(\mu) \sigma_i)^2$$

- **Au 2nd ordre au voisinage de  $\mu$**  :

$$E[g] \simeq g(\mu) + \frac{1}{2} \left( \sum_{i=1}^N \sigma_i^2 \frac{\partial^2 g(\mu)}{\partial x_i^2} \right)$$

$$\sigma^2[g] \simeq \sum_{i=1}^N \left( \frac{\partial g(\mu)}{\partial x_i} \right)^2 + \frac{1}{2} \sum_{i=1}^N \sum_{k=1}^N \left( \frac{\partial^2 g(\mu)}{\partial x_i \partial x_k} \right)^2$$

😊 + *Simplicité*

😊 + *Différentiable*

😞 - *Besoin de calculer le Hessien*



# Optimisation Robuste

Formulation Pire Cas



# Variantes de reformulation

## Reformulation MAX de la contrainte :

On se contente de respecter  $g(x) \leq 0$  pour une partie  $D(\mu)$  donnée de la population autour de  $\mu$

La contrainte s'écrit :  $\max_{x \in D(\mu)} [g(x)] \leq 0$   $x \rightarrow N(\mu, \Sigma^2)$

On évalue  $g(x)$  en 4 points

→ DIFFICULTE : choix de D et calcul du max

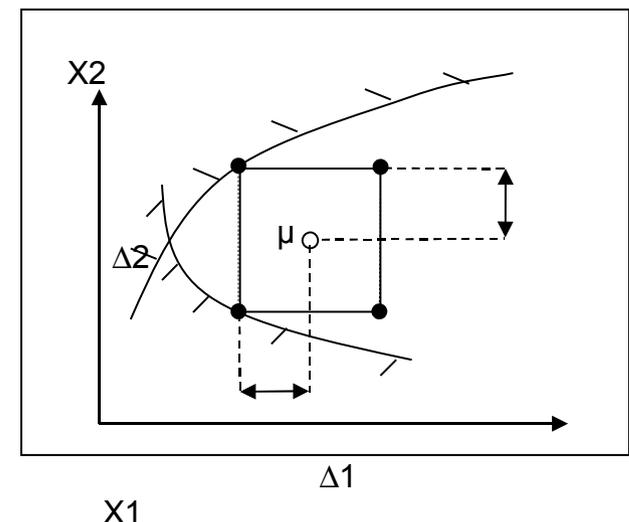
$$D(\mu) = \{ y \in \mathbb{R}^N ; |\mu_i - y_i| = \Delta_i, \forall i \}$$

À déterminer en fonction de la loi

## Méthodes numériques :

### → Corner Space Evaluations :

- + Simplicité et différentiabilité
- $2^N$  contraintes
- Convient plutôt à des lois uniformes
- Difficile de contrôler le % respect de la contrainte
- Le Max n'est pas nécessairement atteint dans D



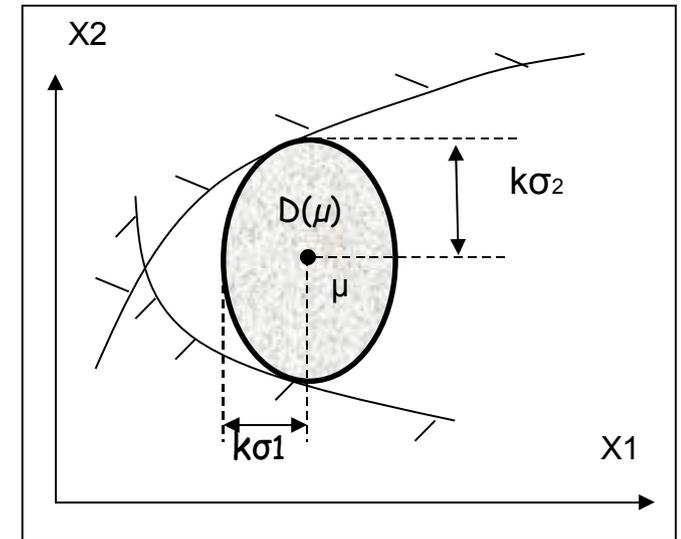


# Variantes de reformulation

## Variation Pattern Formulation :

$$D(\mu) = \left\{ y \in \mathbb{R}^N, \sum_{i=1}^N \left( \frac{y_i - \mu_i}{\sigma_i} \right)^2 \leq k^2 \right\}$$

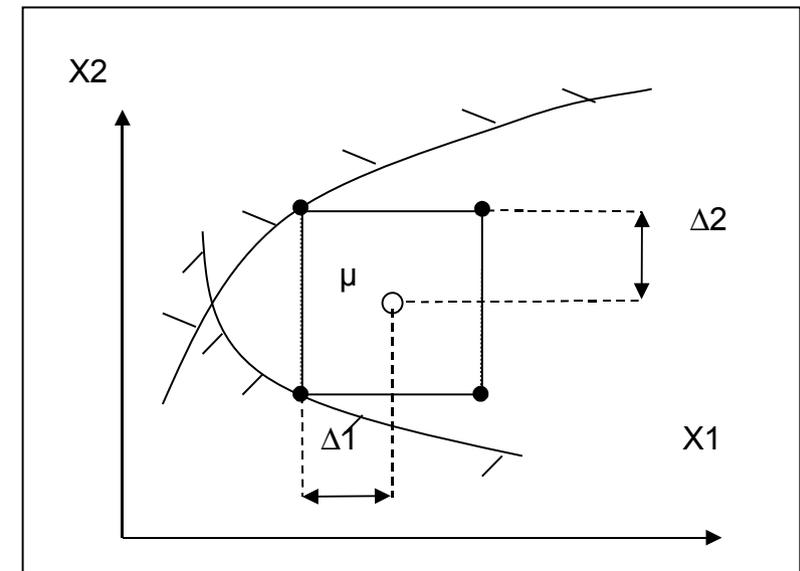
- 😊 + Contrôle du % de respect de la contrainte
- 😊 + Différentiabilité
- 😊 + 1 seule contrainte
- 😞 - Résolution d'un sous-pb
- 😞 - Convient seulement à des var gaussiennes



## Worst Case Analysis :

$$\max_{D(\mu)} g(x) \stackrel{hyp}{\approx} g(\mu) + \Delta g(\mu)$$

- $$\Delta g(\mu) = \sum_{i=1}^N |\partial_{x_i} g(\mu) \Delta_i|$$
- 😊 + Simplicité
  - 😊 + Différentiable presque partout
  - 😞 - Difficile de contrôler le % de respect de la contrainte





# Comparatif

Approches & Méthodes		contrôle fiabilité	différentiabilité	Type de variables	sous pb	$\mu$ et $\sigma$ accessibles	degré de non-linéarité des fonctions	Remarques
Fiabilité	MCS	très précis	N	Toutes	N	O	tout	nb très important d'évaluations de fonctions
	IS,LHS	précis	N		N	O		nb important d'évaluations de fonctions
	FORM	assez précis	O		O	N	faiblement	
	SORM	précis	O		O	N	moyennement	
MAX	WCA	peu précis	O	Toutes	N	N	faiblement	Nécessite les dérivées premières
	CSE	peu précis	O	uniformes	N	N	faiblement	nb important de contraintes
	VPF	peu précis	N	Gaussiennes	O	N	moyennement	effort important développement informatique
6 sigma	MC	assez précis	O	Gaussiennes	N	O	faiblement	nb très important d'évaluations de fonctions
	MMF (ordre 1)	peu précis	O		N			
	MMF (ordre 2)	assez précis	O		N			Nécessite les dérivées premières, secondes et troisièmes
	PEM	assez précis	O		N			nb de points d'évaluation important
	UT	assez précis	O		N			



# Optimisation Robuste

Traitement de la fonction objectif



# Reformulation de l'objectif

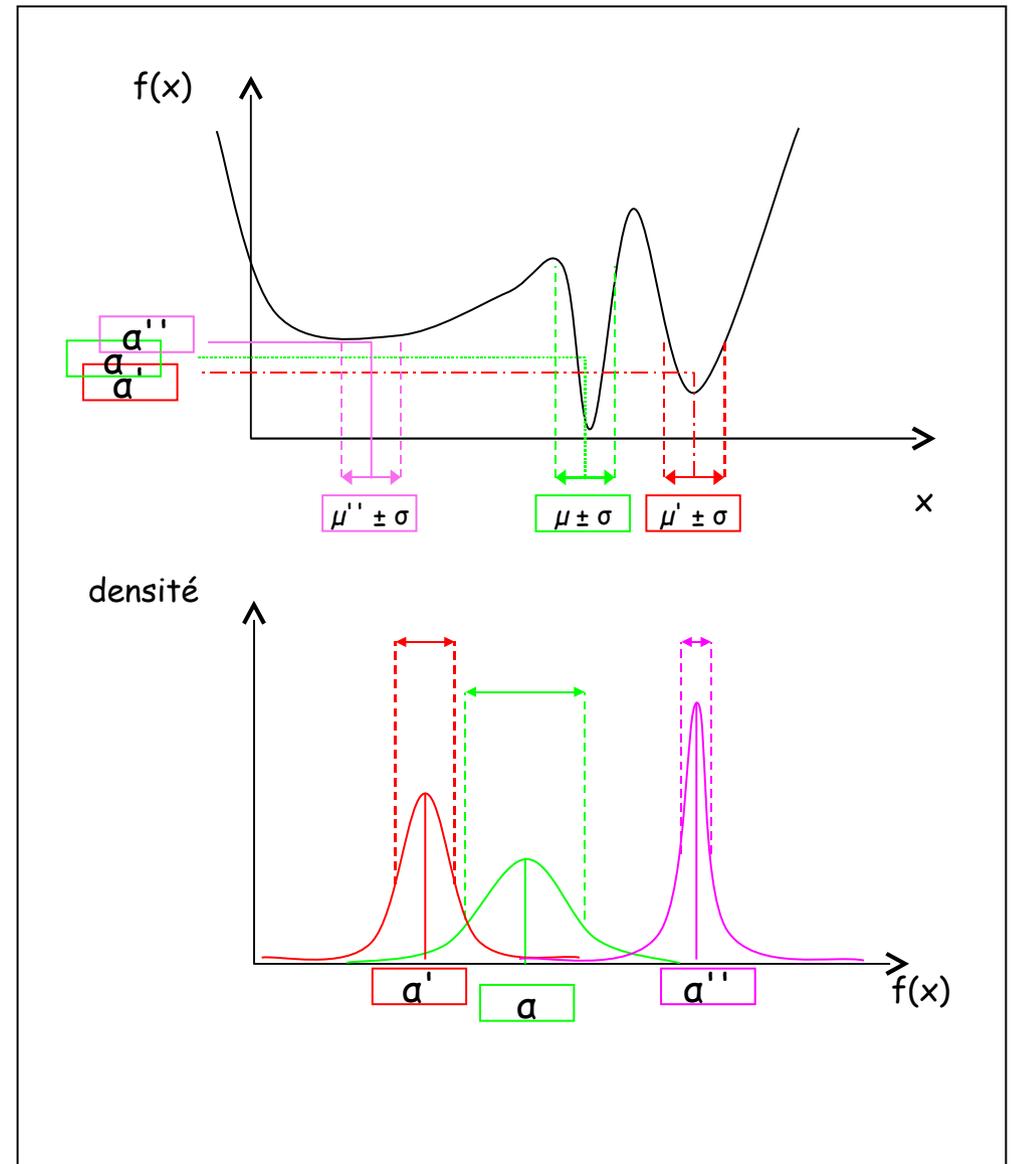
## Impact des dispersions sur l'objectif.

Que veut-on minimiser ?

- La valeur nominale de  $f$  ?
- la valeur moyenne de  $f$  ?
- l'écart-type de  $f$  ?

En général :

→ On recherche un compromis entre la moyenne et l'écart-type





# Reformulation de l'objectif

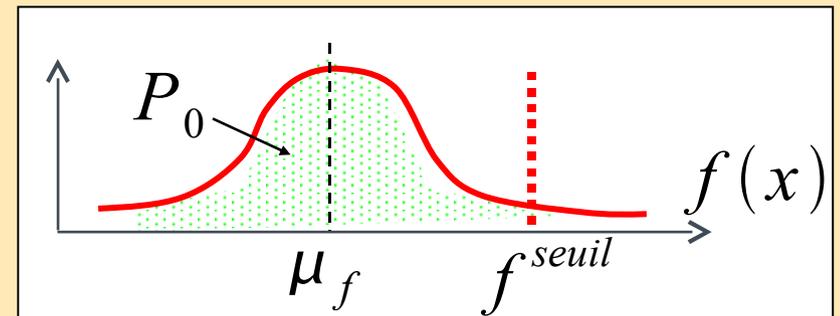
Une formulation multi-objectif : 
$$\min_x [E(f(x)), \sigma^2(f(x))]$$

Une formulation inspirée de la fiabilité (taux de défaillance  $1 - P_0$ ):

Minimiser le seuil  $f^{seuil}(\mu)$  en dessous duquel se trouve un pourcentage donné de la population  $P_0$  autour du point  $\mu$ .

$$\min_{\mu} f^{seuil}(\mu) \quad P[f(x) \leq f^{seuil}(\mu)] = P_0$$

$$x \rightarrow N(\mu, \sigma^2)$$



→ Même méthodes que pour les contraintes

(Monte Carlo, Moment Matching, Worst Case Analysis, FORM/SORM, 6 sigma ...)



# Mode d'emploi de l'optimisation robuste

## → Sur les variables :

1. Identifier les variables qui présentent un caractère aléatoire
2. Identifier la loi de distribution

## → Où mettre de la robustesse ?

1. Dans les contraintes ?
2. Dans l'objectif ?

## → Pour choisir sa formulation et la méthode numérique, prendre en compte :

1. les lois de distribution
2. le degré de contrôle voulu sur la fiabilité
3. Le besoin d'une formulation différentiable
4. factoriser les techniques entre fiabilité et robustesse

## → Post-traitement indispensable :

1. calcul de la robustesse et fiabilité a posteriori via un Monte-Carlo.



# Conclusion

La robustesse, c'est :

- Soit la contrainte;
- Soit la fonction objectif

Ne traite que des ddp classiques (normales, etc.)

Discipline en forte croissance

La décomposition en polynômes de chaos :

- Possibilité de traiter n'importe quelle ddp
- Possibilité de traiter n'importe quelle non-linéarité
- Pas encore appliqué à l'optimisation robuste

De nouvelles méthodes sont en cours de mise au point ...



# L'aide à la décision multicritère

Y. Collette



# Plan

- Introduction
- Définitions
  - Ordre / Préordre
  - Critère
- L'approche Française
- L'approche Américaine
- La méthode PROMETHEE
- La méthode ELECTRE



# Introduction

## Ecole Américaine

Le décideur cherche à maximiser une fonction de préférence implicite.

Cette fonction est stable dans le temps.

Le décideur est toujours capable de choisir entre deux solutions.

**Ordre total**

**Tout est comparable !**

## Ecole Française

Le décideur manque d'informations pour construire sa fonction de préférence.

Il peut être incapable de choisir entre deux solutions.

**Ordre partiel**

**Incomparabilité**



# Applications typiques

- Sélection de zones de développement industriel
- Approche de décision multicritère pour la priorisation de projets industriels
- Tri de dossier
- Classement de candidats
- Sélection de zones d'exploration pétrolière
- Sélection de l'implantation d'un nouveau campus
- Sélection pour un système d'inspection automatique
- Sélection pour un système de gestion de procédure d'incidents dans un centre informatique

**Le but: reproduire le processus de sélection / classement d'un expert humain**



# Applications typiques

- Aide à la décision pour l'acquisition d'équipements (véhicules, hélicoptères, ordinateurs,...)
- Sélection et classement de personnel
- Affectation de personnel à une tâche
- Sélection et classement de plans d'investissement
- Classement de demande de prêts bancaire
- .....



# Les relations binaires

On définit trois relations principales :

La relation de préférence notée  $aPb$  si  $a$  est préférée à  $b$

La relation d'indifférence notée  $aIb$  s'il y a indifférence entre  $a$  et  $b$

La relation d'incomparabilité ou de refus notée  $aRb$  s'il y a incomparabilité ou refus de  $a$  sur  $b$

Ces relations possèdent un certain nombre de propriétés:

$aPb \rightarrow \sim bPa$	P est asymétrique
$aIa$	I est réflexive
$aIb \rightarrow bIa$	I est symétrique
$aRa$	R est irréflexive
$aRb \rightarrow bRa$	R est symétrique



# Comment construire une relation de préférence

Construisons une relation de préférence  $S$  simple:

$aSb$  si  $aPb$  ou  $aIb$

Passons au cas concret:

$$\begin{cases} aPb \rightarrow g(a) > g(b) \\ aIb \rightarrow g(a) = g(b) \end{cases}$$

Si la relation binaire engendre un classement avec ex-aequo:  
On a un pré-ordre total.  
S'il n'y a pas d'ex-aequo:  
On a un ordre total.



# Les relations binaires

## Exemples

Action	C1	C2
a	100	100
b	30	20



$a P b$

Relation de préférence

Action	C1	C2
a	100	20
b	30	100



$a R b$

Relation d'incomparabilité  
Refus



# Les relations binaires

## Exemples

Action	C1	C2
a	100	99
b	20	100



$a P b$

Action	C1	C2
a	100	99
b	99	100



Relation d'indifférence

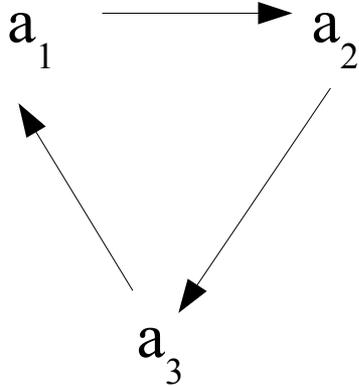
$a I b$

Relation d'indifférence  
à seuil



# Paradoxe de Condorcet

Non transitivité de la relation d'indifférence



Exemple du café sucré:

on ne fait pas de différence entre le café  $a_i$  et le café  $a_{i+1}$  qui ont quelques mg de sucre de différence.

Par contre, on fait une différence très nette entre le café  $a_0$  et le café  $a_{100}$  qui ont quelques g de sucre de différence

Même principe avec Pain + jambon



# Les relations binaires

**Préordre:** une relation réflexive et transitive

On note  $E$  l'ensemble dans lequel le préordre est établi

Réflexivité :  $x R x \forall x \in E$

Transitivité :  $x R y$  et  $y R z$  implique  $x R z \forall x, y, z \in E$

**Préordre total :** quand tous les éléments sont comparables

**Préordre partiel :** présence possible d'ex-aequo

**Ordre :** un préordre anti-symétrique

Anti-symétrie :  $x R y$  implique  $y R x (x \neq y) \forall x, y \in E$

Dans un ordre, il n'y a pas d'ex-aequo



# Les relations binaires

## Quelques exemples

### Quasi-ordre : modèle à seuil

$$\begin{cases} a P b \rightarrow g(a) > g(b) + q \\ a I b \rightarrow |g(a) - g(b)| \leq q \end{cases}$$

**Définition:** une structure de préférence est une structure de **quasi-ordre** si elle est représentable par le **modèle à seuil**

### Pseudo-ordre : modèle à deux seuils

$$\begin{cases} a P b \rightarrow g(a) > g(b) + p(g(b)) \\ a Q b \rightarrow g(b) + p(g(b)) \geq g(a) > g(b) + q(g(b)) \\ a I b \rightarrow \begin{cases} g(b) + q(g(b)) \geq g(a) \\ g(a) + q(g(a)) \geq g(b) \end{cases} \end{cases}$$

**Définition:** une structure de préférence est une structure de **pseudo-ordre** si elle est représentable par le **modèle à deux seuils**



# Relations de préférences

## Exemples

Relation de préférence avec seuil d'indifférence :

$$\begin{cases} aPb \rightarrow g(a) > g(b) + q \\ aIb \rightarrow |g(a) - g(b)| \leq q \end{cases}$$

Relation de préférence avec seuil d'indifférence variable :

$$\begin{cases} aPb \rightarrow g(a) > g(b) + q(g(b)) \\ aIb \rightarrow \begin{cases} g(a) \leq g(b) + q(g(b)) \\ g(b) \leq g(a) + q(g(a)) \end{cases} \end{cases}$$

Relation de préférence sur des intervalles :

$$\begin{cases} aPb \rightarrow x_a > y_b \\ aIb \rightarrow [x_a, y_a] \cap [x_b, y_b] \neq \emptyset \end{cases}$$



# Relations de préférences

## Exemple

Relation de préférence sur des intervalles :

$$\begin{cases} aPb \rightarrow x_a > x_b \text{ et } y_a > y_b \\ aIb \rightarrow [x_a, y_a] \subset [x_b, y_b] \text{ ou } [x_b, y_b] \subset [x_a, y_a] \end{cases}$$

Relation avec seuil sur l'indifférence et la préférence :

$$\begin{cases} aPb \rightarrow g(a) > g(b) + p(g(b)) \\ aQb \rightarrow g(b) + p(g(b)) \geq g(a) > g(b) + q(g(b)) \\ aIb \rightarrow \begin{cases} g(b) + q(g(b)) \geq g(a) \\ g(a) + q(g(a)) \geq g(b) \end{cases} \end{cases}$$



# Les critères

## Famille cohérente de critères

Famille de critères qui satisfait 3 exigences:

**Exhaustivité** : il ne faut pas qu'il y ait trop peu de critères

**Cohérence** : Cohérence entre préférences locales de chaque critère et les préférences globales. Si l'évaluation de  $a_i$  est égale à l'évaluation de  $a_k$  sur tous les critères sauf 1, et qu'elle est meilleure sur 1 critère alors l'affirmation «  $a_i$  est préférée à  $a_k$  » peut être émise.

**Non redondance** : il ne faut pas qu'il y ait des critères qui se dupliquent. Il faut qu'il y ait un nombre de critères tel que la suppression d'un critère laisse une famille de critères qui ne satisfasse plus au moins une des deux exigences précédentes.



# Théorème d'Arrow

L'information contenue dans  $n$  préordres totaux n'est pas suffisante pour déterminer univoquement un préordre total qui les agrège.

Face à cette affirmation: deux réactions sont possibles

- Ajouter de l'information
- Se contenter d'une structure globale plus pauvre qu'un préordre total global



# Théorème d'Arrow

## Notations

$S_{/a,b}$  désigne la restriction de  $S$  à l'ensemble  $\{a, b\}$

$$(S_1, \dots, S_n)_{/a,b} = (S_{1/a,b}, \dots, S_{n/a,b})$$

Procédure d'agrégation :

$$f : S_E^n \rightarrow S_E$$

$E$  un ensemble fini d'actions

$S_E$  l'ensemble des préordres totaux sur  $E$

$S_E^n$  l'ensemble des n-uples de préordre totaux sur  $E$

Etant donné un préordre total  $S_i$  sur  $E$ , on désigne par  $P_i$  la relation définie par  $a P_i b$  si  $a S_i b$  et  $b S_i a$



# Théorème d'Arrow

## Enoncé du théorème

$E$  un ensemble tel que  $|E| \geq 3$

Il n'existe pas de procédure d'agrégation qui satisfasse les 3 conditions suivantes:

### Règle d'unanimité :

$\forall (S_1, \dots, S_n) \in S_E^n, \forall a, b \in E$  tq  $a P_i b \forall i, a S b$  et  $b S a$  où  $S = f(S_1, \dots, S_n)$

### Règle de non-dictature :

$\exists i \in \{1, \dots, n\}$  tq  $\forall (S_1, \dots, S_n) \in S_E^n, \forall a, b \in E$  :  
 $a P_i b$  implique  $a S b$  et  $b S a$  où  $S = f(S_1, \dots, S_n)$

### Règle d'indépendance par rapport aux actions non-concernées :

$\forall (S_1, \dots, S_n), (S_1', \dots, S_n') \in S_E^n, \forall a, b \in E$  :

si  $(S_1, \dots, S_n)_{/a,b} = (S_1', \dots, S_n')_{/a,b}$ ,

si  $S = f(S_1, \dots, S_n)$  et si  $S' = f(S_1', \dots, S_n')$  alors  $S_{/a,b} = S'_{/a,b}$

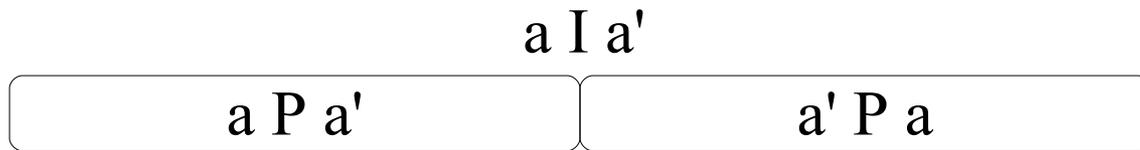


# Les critères

Type de critère

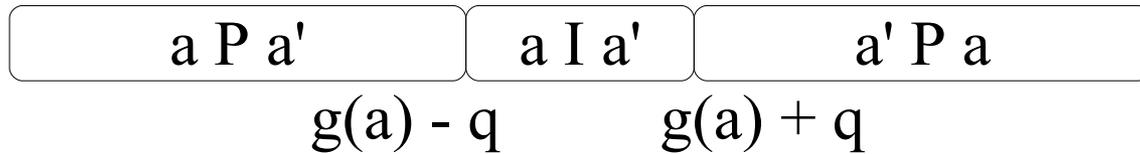
Ordre engendré

vrai-critère



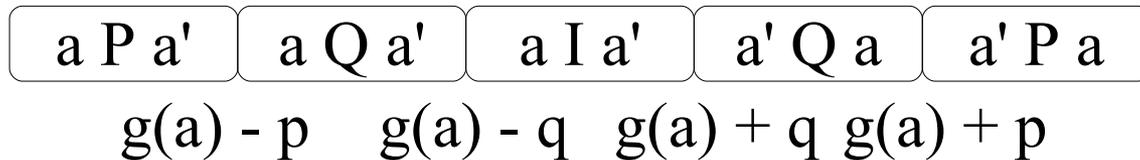
Préordre complet

quasi-critère



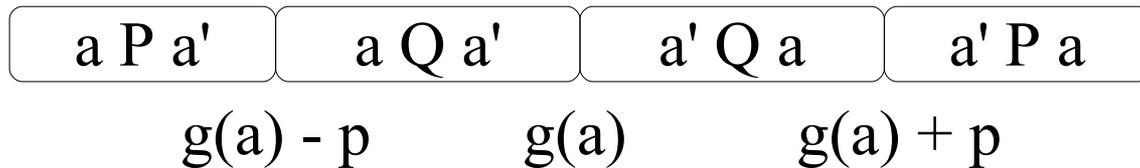
quasi-ordre

pseudo-critère



pseudo-ordre

pré-critère



??

S – Surclassement  
P – Préférence

I – Indifférence  
Q – Préférence faible



# L'aide à la décision

## L'approche Américaine



# Les fonctions d'utilité

$$U(a) = \sum_{i=1}^n U_i(g_i(a))$$

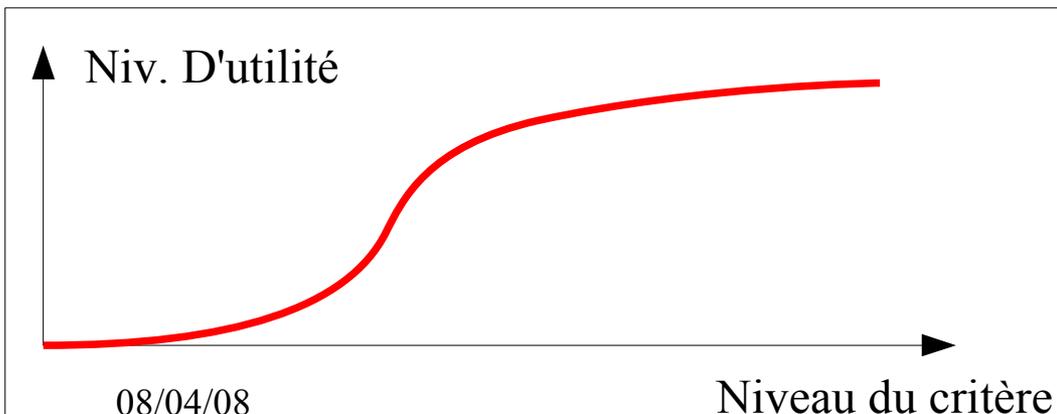
$U_i$  : fonction d'utilité associée au critère  $g_i$

Comment construire la fonction  $U_i$  :

## Méthode I :

Demander au décideur de déterminer l'état  $z_j$  qu'il considère entre  $x_j$  et  $y_j$ .

L'incomparabilité n'existe pas



$$\left\{ \begin{array}{l} U(z_j) = \frac{1}{2} [U(x_j) + U(y_j)] \\ U(v_j) = \frac{1}{2} [U(x_j) + U(z_j)] \\ U(w_j) = \frac{1}{2} [U(z_j) + U(y_j)] \\ \dots \end{array} \right.$$



# Les fonctions d'utilité

## Méthode II :

On demande au décideur de déterminer l'état  $z_j$  tel qu'il considère comme équivalent:

- D'obtenir  $z_j$
- D'obtenir  $x_j$  avec une probabilité  $1/2$  et  $y_j$  avec une probabilité  $1/2$

On en déduit

$$U_j(z_j) = 1/2 \cdot U_j(x_j) + 1/2 \cdot U_j(y_j)$$

On continue ensuite avec  $(x_j, z_j)$ , avec  $(z_j, y_j)$ , ...



# Les fonctions d'utilité

## Méthode III :

On demande au décideur de déterminer l'état  $z_j$  tel qu'il considère comme équivalent:

- D'obtenir  $z_j(p)$
- D'obtenir  $x_j$  avec une probabilité  $p$  et  $y_j$  avec une probabilité  $1-p$  et ce pour différentes valeurs de  $p$ .

On en déduit

$$U_j(z_j) = p \cdot U_j(x_j) + (1-p) \cdot U_j(y_j), \forall p$$

On continue ensuite avec  $(x_j, z_j)$ , avec  $(z_j, y_j)$ , ...



# L'aide à la décision

## L'approche Française



# Quelques approches simples

## Combiner des critères

### Présentation d'un exemple simple

	$M_1$	$M_2$	$M_3$	$M_4$
A	18	10	13	11
B	16	17	12	15
C	17	15	11	13
D	12	18	10	12
E	15	16	8	8

Agrégation par pondération

$$F(x) = \frac{\sum_{i=1}^n \omega \cdot M_i(x)}{\sum_{i=1}^n \omega_i}$$



# Quelques approches simples

## Combiner des critères

- Inconvénients de cette méthode :
  - Comment choisir les poids ?
  - Le problème de la compensation

### Exemple pour 2 points et 3 critères

	Productivité $p=0.65$	Intelligence $p=0.35$	Moyenne arithmétique
A	0.45	0.90	0.6075
B	0.60	0.55	0.5825



# Quelques approches simples Des difficultés ?

- Le problème de l'incomparabilité de deux actions
- Le problème de l'indifférence
- Le problème de la transitivité de la relation de préférence
- Le problème de la stabilité dans le temps des préférences



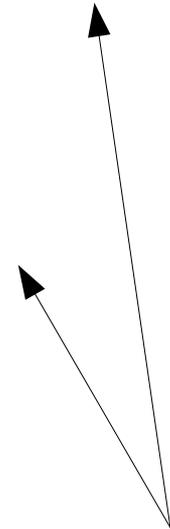
# Cahier des charges d'une méthode d'aide à la décision

- Prise en compte de la plage de variation
- Elimination des l'effets d'échelle
- Une comparaison par pairs doit mener à un ordre partiel (s'il y a des incomparabilités) ou complet
- La méthode doit être simple
- Les paramètres techniques doivent être interprétables par l'utilisateur
- Les poids associés aux critères doivent être facile à interpréter



# Quelques méthodes

- **Prométhée** : traite des données numériques
- **Oreste** : traite des données ordinales
- **Electre** : Comparaison par pairs – surclassement avec incomparabilité
- **AHP** : Comparaison par pairs – pas d'incomparabilité
- ....



Méthodes qui vont être présentées par la suite

Analytical  
Hierarchy  
Process



# Aide à la décision

## La méthode Prométhée



# Prométhée: Définitions

Un ensemble d'Actions qui doivent être rangées par ordre de préférences:

L'ensemble  $[a_1, \dots, a_k]$  noté  $A$

Un ensemble de critères de préférence ou de sélection:

L'ensemble  $[C_1, \dots, C_m]$

Des poids affectés à chaque critère:

L'ensemble  $[\omega_1, \dots, \omega_m]$  noté  $\Omega$

Tel que  $\sum \omega_i = 1$

Un ensemble de fonctions de préférences qui expriment comment on préfère une action plutôt qu'une autre

L'ensemble  $[P_1, \dots, P_m]$  noté  $P$



# Prométhée: Exemple

Il y a 6 appels d'offre:

$$a_1, \dots, a_6$$

6 critères de sélection sont retenus:

$C_1$ : temps moyen de maintenance par jour (en minutes)

$C_2$ : valeur technique de l'équipement (sur 100)

$C_3$ : coût (en million d'euros)

$C_4$ : coût estimé de maintenance pendant la durée de vie de l'équipement (en millions d'euros)

$C_5$ : nombre de pannes estimées par an

$C_6$ : niveau de sécurité des équipements offerts



# Prométhée Bases

Les trois étapes de la méthode:

1. Sélection des critères généralisés
2. Détermination de la relation de surclassement
3. Evaluation des préférences



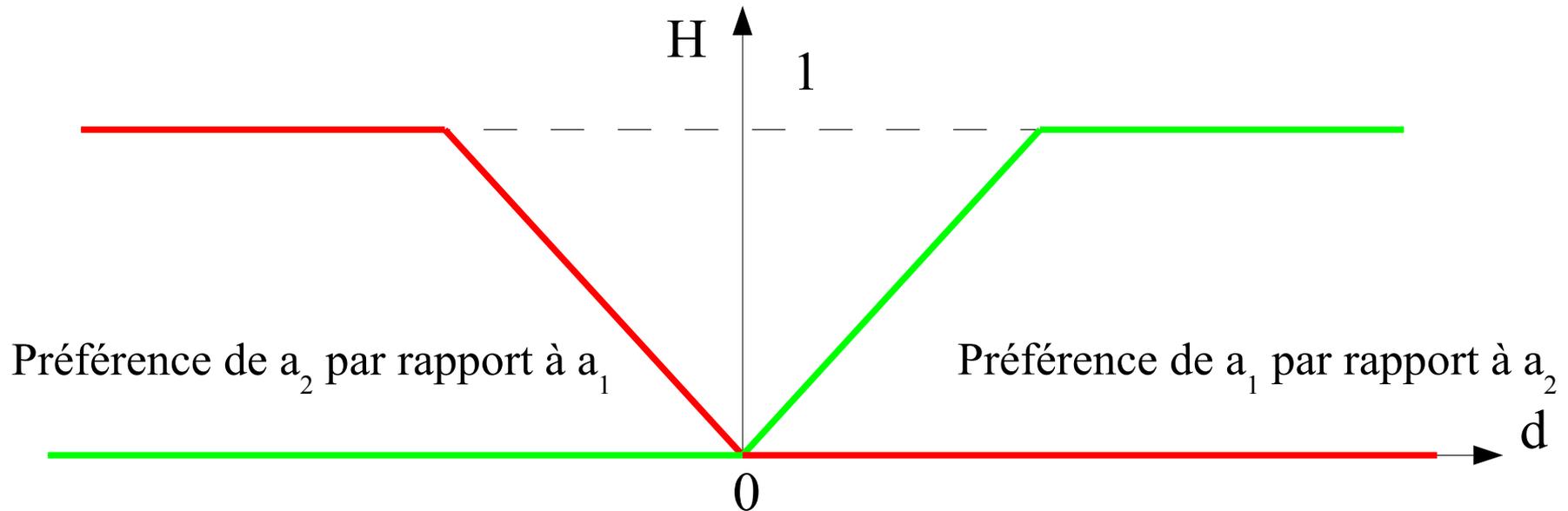
# Prométhée

## Concept de critère généralisé

$C_i(a)$  est un critère à optimiser

On considère la fonction de préférence

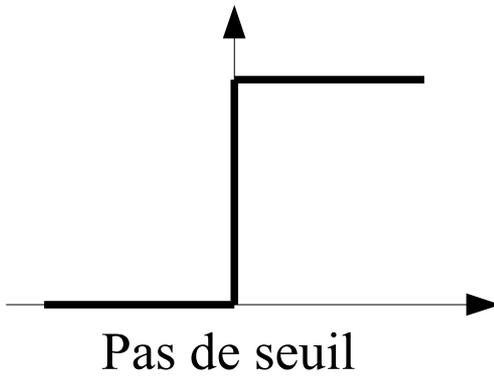
$$d = C_i(a_1) - C_i(a_2)$$



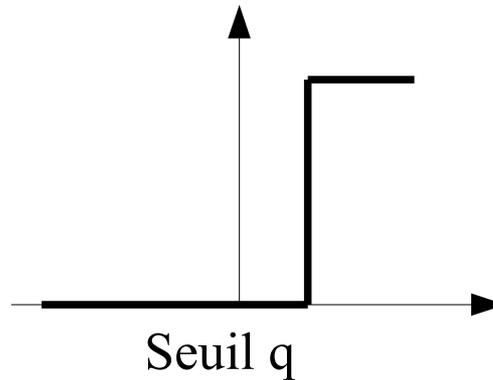


# Les critères usuels

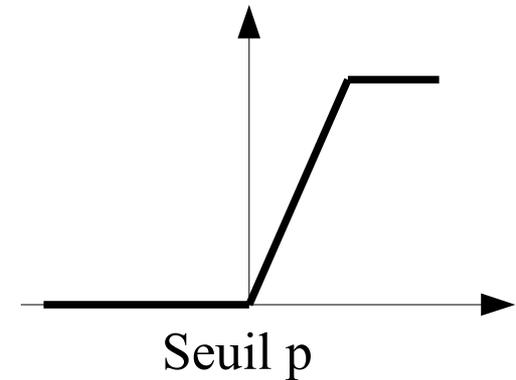
Type I: Critère usuel



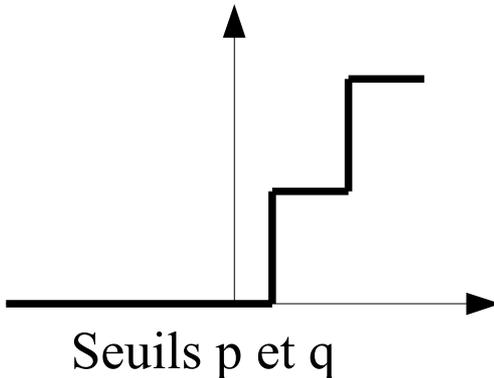
Type II: Critère en U



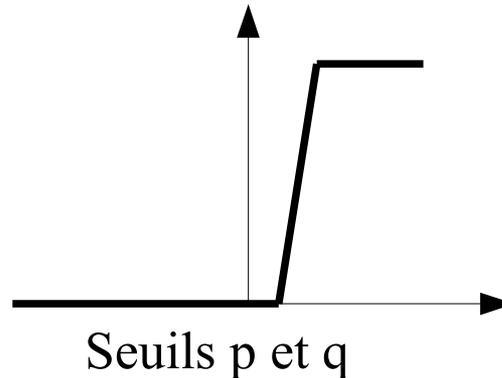
Type III: Critère en V



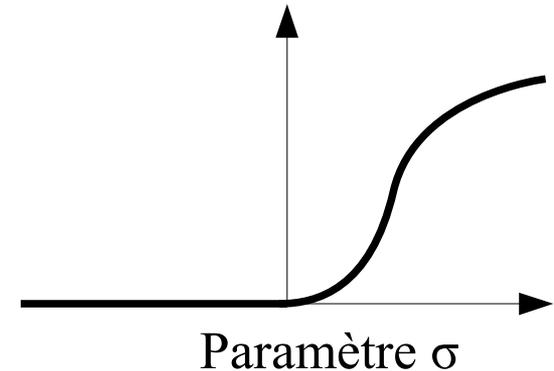
Type IV: Critère niveau



Type V: Critère Linéaire



Type VI: Critère Gaussien





# Choix des types de transformation - Exemple

- Critère opérationnel : **type III**
- Court-terme financier, coût d'acquisition, coût de construction : **type V**
- Long terme financier, coût de maintenance, coût d'un cycle de vie : **type IV**
- Ressources discrètes : **type II**
- Ecologie : **type I**
- Sécurité, Qualité : **type VI**



# Paramétrage

- Seuil d'indifférence :  $q$ 
  - Élevé si présence d'incertitude, faible précision des données
- Seuil de préférence :  $p$ 
  - Proche de la déviation max si les données sont précises



# La relation de préférence

Pour chaque critère  $C_i$  on associe une fonction de préférence  $P$ .

$$\Pi (a_1, a_2) = \sum w_i \cdot P_i (a_1, a_2)$$

(Différents poids)

$$\Pi (a_1, a_2) = (1/m) \cdot \sum P_i (a_1, a_2)$$

(Tous les poids sont égaux)



# La relation de préférence

On a :

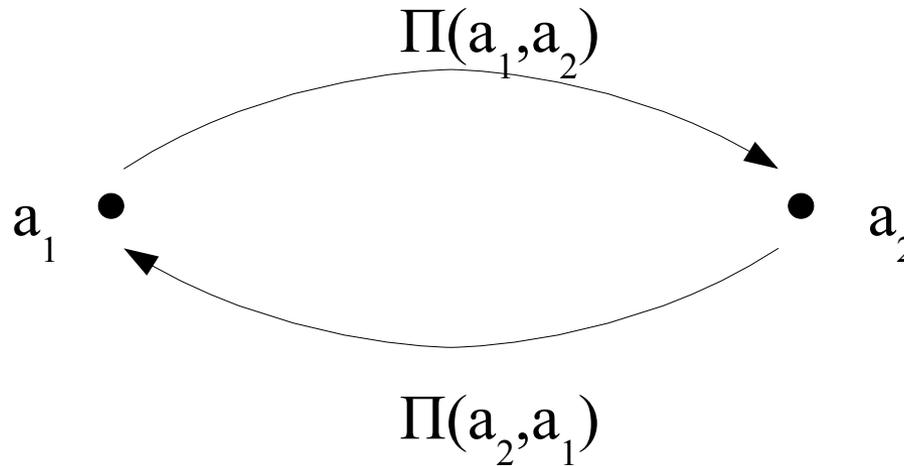
$$0 \leq \Pi(a_1, a_2) \leq 1$$

De plus,

- si  $\Pi(a_1, a_2) \approx 0$  légère préférence de  $a_1$  par rapport à  $a_2$
- si  $\Pi(a_1, a_2) \approx 1$  préférence forte de  $a_1$  par rapport à  $a_2$



# Evaluation des préférences



$$\text{Flux entrant : } \Phi^+(a_1) = \frac{1}{k-1} \sum_i \Pi(a_1, a_i)$$

$$\text{Flux sortant : } \Phi^-(a_1) = \frac{1}{k-1} \sum_i \Pi(a_i, a_1)$$

$$\text{Flux net : } \Phi(a_1) = \Phi^+(a_1) - \Phi^-(a_1)$$



# La relation de surclassement

$a_1 P a_2$  (l'action  $a_1$  surclasse l'action  $a_2$ ) si :

$a_1 P^+ a_2$  et  $a_1 P^- a_2$

$a_1 P^+ a_2$  et  $a_1 I^- a_2$

$a_1 I^+ a_2$  et  $a_1 P^- a_2$

$a_1 I a_2$  (l'action  $a_1$  et l'action  $a_2$  sont indifférentes) si :

$a_1 I^+ a_2$  et  $a_1 I^- a_2$

$a_1 R a_2$  (l'action  $a_1$  et l'action  $a_2$  sont incomparables) sinon

$a_1 P^+ a_2$  si  $\Phi^+(a_1) > \Phi^+(a_2)$

$a_1 I^+ a_2$  si  $\Phi^+(a_1) = \Phi^+(a_2)$

$a_1 P^- a_2$  si  $\Phi^-(a_2) > \Phi^-(a_1)$

$a_1 I^- a_2$  si  $\Phi^-(a_2) = \Phi^-(a_1)$



# La méthode Prométhée II

$a_1 P^{\text{II}} a_2$  (l'action  $a_1$  surclasse l'action  $a_2$ )

si  $\Phi(a_1) > \Phi(a_2)$

$a_1 I^{\text{III}} a_2$  (l'action  $a_1$  et l'action  $a_2$  sont indifférents)

si  $\Phi(a_1) = \Phi(a_2)$



# Sélection des critères généralisés

Critère	Type	Paramètres
$C_1$	<b>II</b>	$q=10$
$C_2$	<b>III</b>	$p=30$
$C_3$	<b>V</b>	$q=50; p=500$
$C_4$	<b>IV</b>	$q=10; p=60$
$C_5$	<b>I</b>	
$C_6$	<b>VI</b>	$\sigma=5$



# Les données

Critère	Min / Max	$a_1$	$a_2$	$a_3$	$a_4$	$a_5$	$a_6$
Critère 1	Min	80	65	83	40	52	94
Critère 2	Max	90	58	60	80	72	96
Critère 3	Min	600	200	400	1000	600	700
Critère 4	Min	54	97	72	75	20	36
Critère 5	Min	8	1	4	7	3	5
Critère 6	Max	5	1	7	10	8	6



Critère à minimiser  
ou maximiser



# Paramétrage des critères

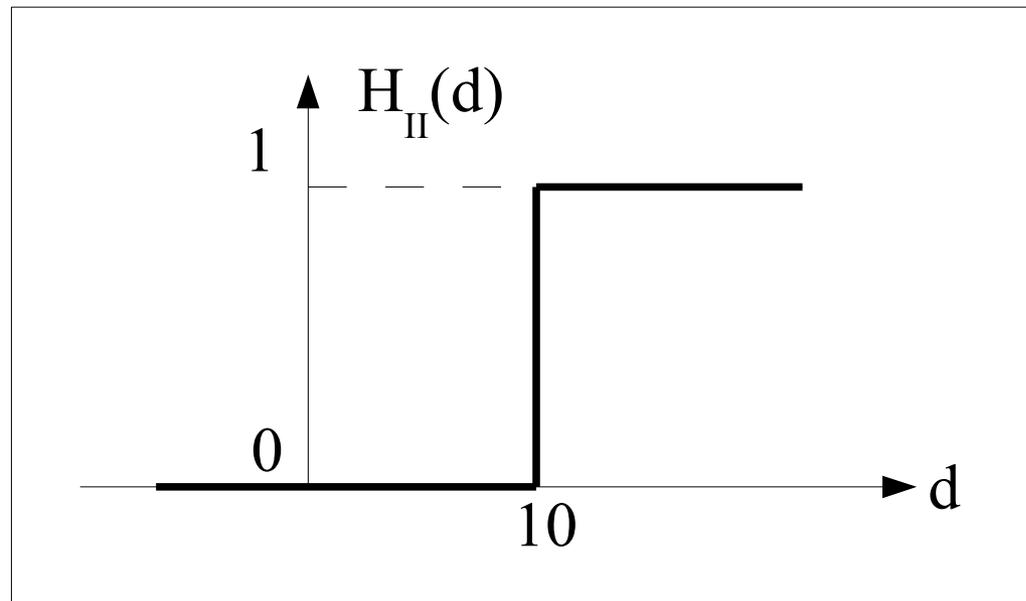
Critère  $C_1$  : Critère de type II – Critère en U

- C'est un critère à minimiser
- L'action  $a_1$  est moins bonne que l'action  $a_2$  :  $C_1(a_1) - C_1(a_2) = 15$

Par conséquent :

$$P_1(a_1, a_2) = 1$$

$$P_1(a_2, a_1) = 0$$





# Paramétrage des critères

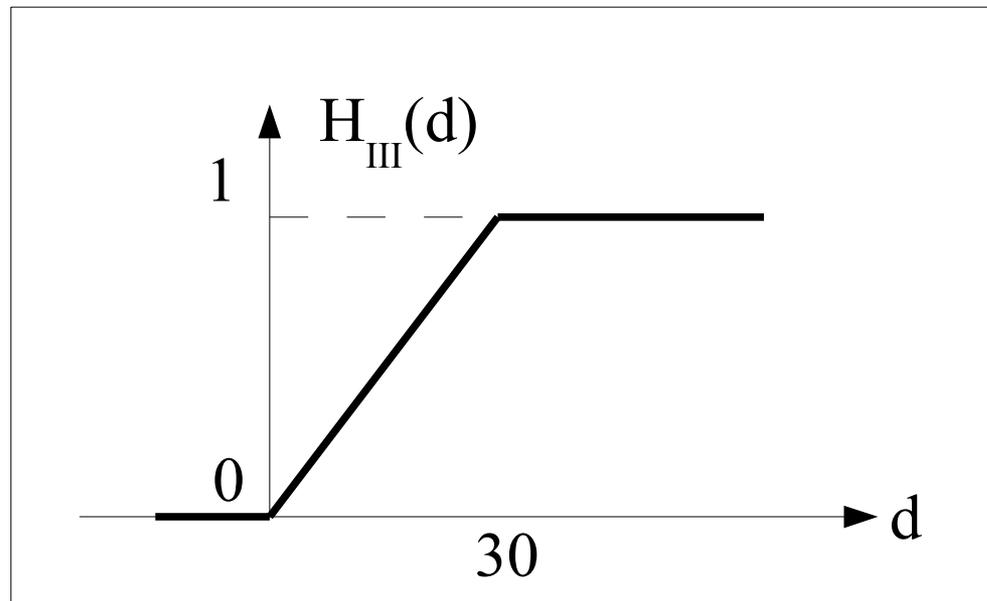
Critère  $C_2$  : Critère de type III – Critère en V

- C'est un critère à maximiser
- L'action  $a_1$  est meilleure que l'action  $a_2$  :  $C_2(a_1) - C_2(a_2) = 32$

Par conséquent :

$$P_2(a_1, a_2) = 1$$

$$P_2(a_2, a_1) = 0$$





# Paramétrage des critères

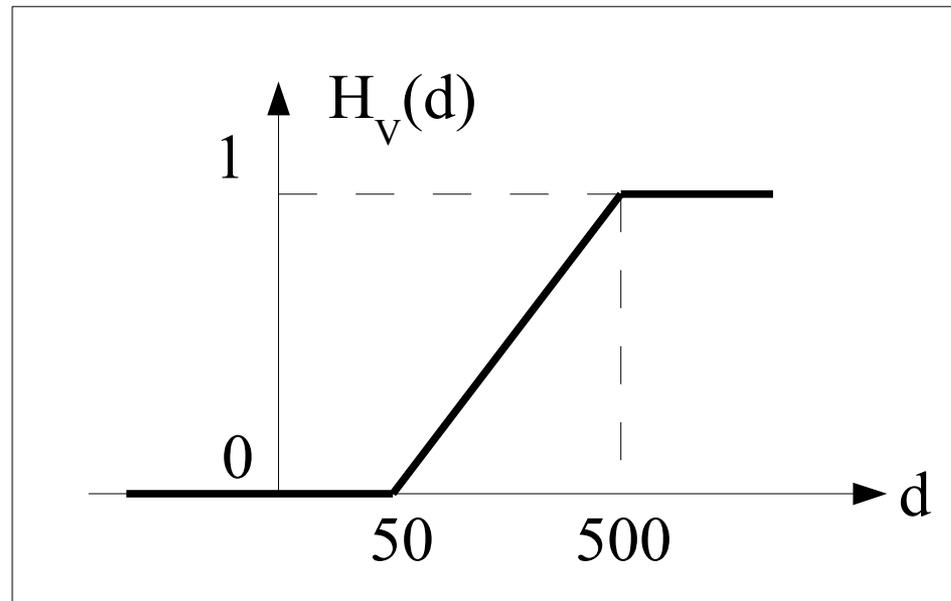
Critère  $C_3$  : Critère V – Critère Linéaire

- C'est un critère à minimiser
- L'action  $a_1$  est moins bonne que l'action  $a_2$  :  $C_3(a_1) - C_3(a_2) = 400$

Par conséquent :

$$P_3(a_1, a_2) = 0.778$$

$$P_3(a_2, a_1) = 0$$





# Paramétrage des critères

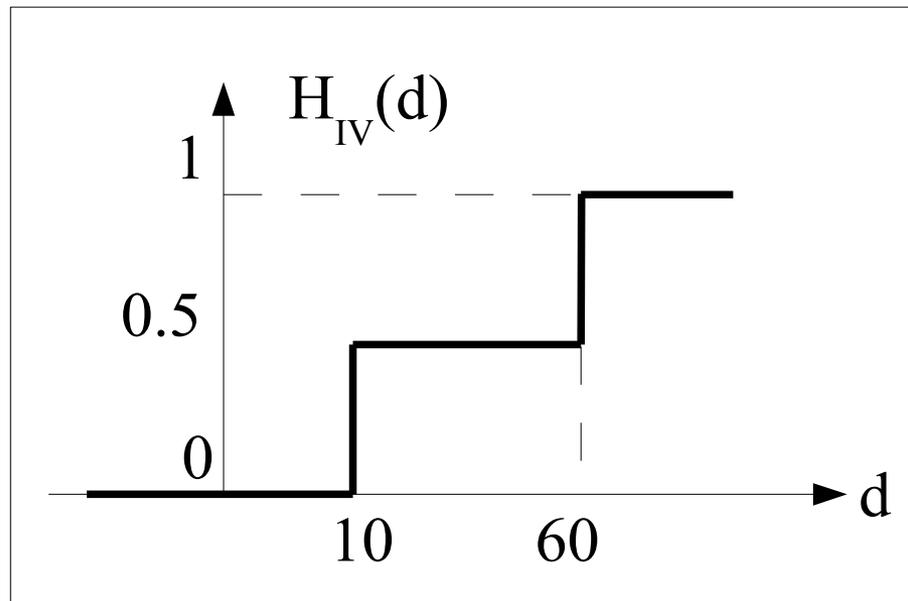
Critère  $C_4$  : Critère de type IV – Critère Niveau

- C'est un critère à minimiser
- L'action  $a_1$  est meilleure que l'action  $a_2$  :  $C_4(a_1) - C_4(a_2) = -43$

Par conséquent :

$$P_4(a_1, a_2) = 0$$

$$P_4(a_2, a_1) = 0.5$$





# Paramétrage des critères

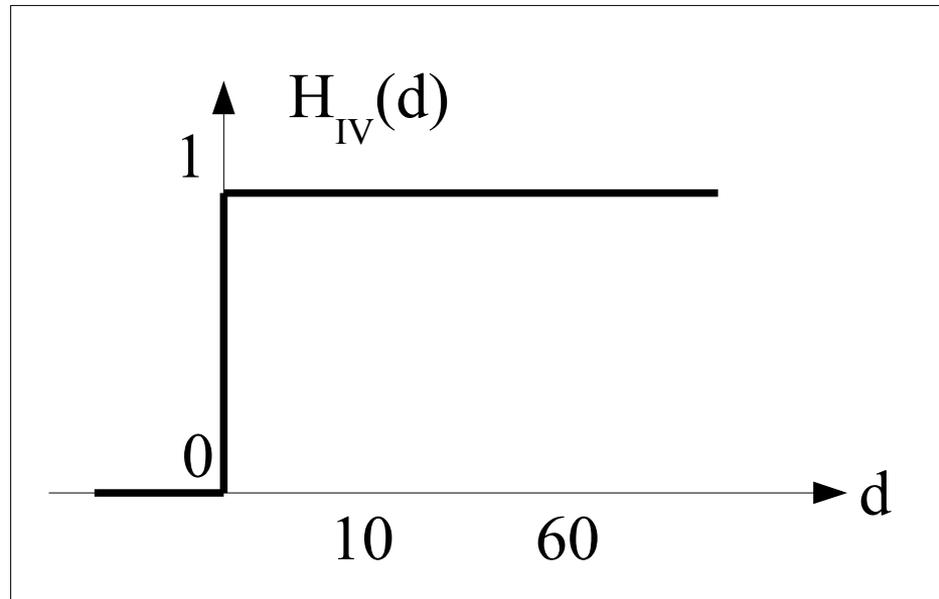
Critère  $C_5$  : Critère de type I – Critère Usuel

- C'est un critère à minimiser
- L'action  $a_1$  est moins bonne que l'action  $a_2$  :  $C_5(a_1) - C_5(a_2) = 7$

Par conséquent :

$$P_5(a_1, a_2) = 1$$

$$P_5(a_2, a_1) = 0$$





# Paramétrage des critères

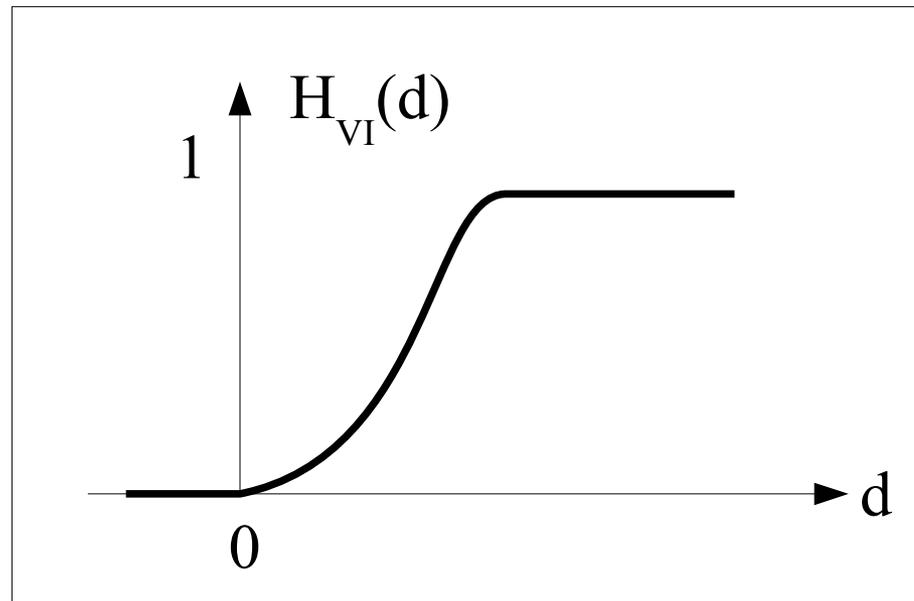
Critère  $C_6$  : Critère de type VI – Critère Gaussien

- C'est un critère à maximiser
- L'action  $a_1$  est meilleure que l'action  $a_2$

Par conséquent :

$$P_6(a_1, a_2) = 0.274$$

$$P_6(a_2, a_1) = 0$$





# Détermination de la table des flux

$$\text{Moyenne de } P_i(a_1, a_2) = 1/6 \cdot (1+1+0.778+0+1+0.274) = 0.675$$

$$\text{Moyenne de } P_i(a_2, a_1) = 1/6 \cdot (0+0+0+0.5+0+0) = 0.083$$

On applique ce raisonnement aux autres paires

	$a_1$	$a_2$	$a_3$	$a_4$	$a_5$	$a_6$
$a_1$		0.675	0.250	0.268	0.100	0.185
$a_2$	0.083		0.389	0.333	0.296	0.500
$a_3$	0.236	0.180		0.333	0.056	0.429
$a_4$	0.399	0.505	0.305		0.223	0.212
$a_5$	0.444	0.515	0.487	0.380		0.448
$a_6$	0.286	0.399	0.250	0.432	0.133	



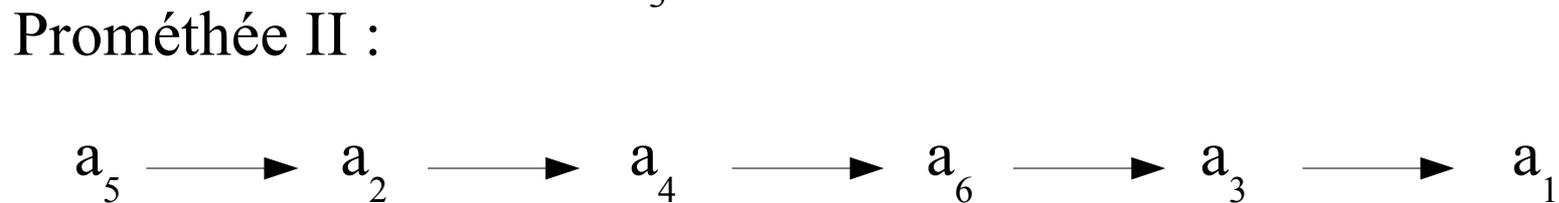
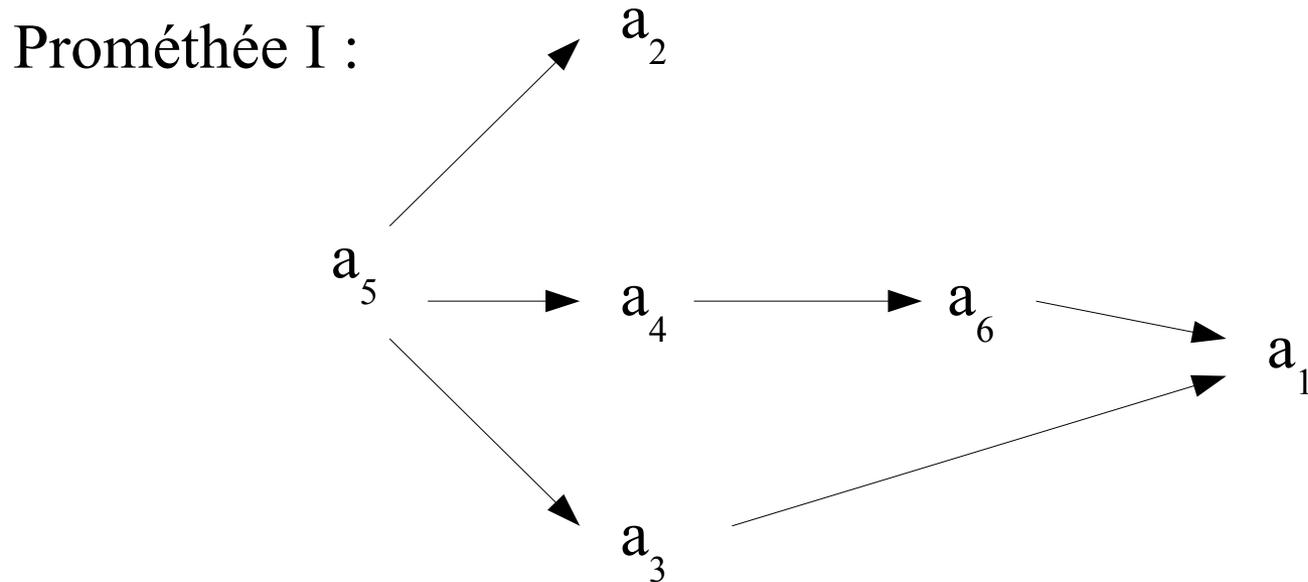
# Détermination de la table des flux

Flux net = flux entrant – flux sortant /  $\Phi = \Phi^+ - \Phi^-$

	$a_1$	$a_2$	$a_3$	$a_4$	$a_5$	$a_6$	$\Phi^+$	$\Phi$
$a_1$		0.296	0.250	0.268	0.100	0.185	0.220	-0.146
$a_2$	0.462		0.389	0.333	0.296	0.500	0.396	0.017
$a_3$	0.236	0.180		0.333	0.056	0.429	0.247	-0.089
$a_4$	0.399	0.505	0.305		0.223	0.212	0.329	-0.020
$a_5$	0.444	0.515	0.487	0.380		0.448	0.455	0.293
$a_6$	0.286	0.399	0.250	0.432	0.133		0.300	-0.055
$\Phi^-$	0.366	0.379	0.336	0.349	0.162	0.355		



# Classement par les méthodes Prométhée I et II





# Aide à la décision

## Les méthodes ELECTRE



# Electre I

## Exemple

Les critères:

$C_1$  la moyenne des notes sur l'ensemble des matières

$C_2$  la note minimal doit être supérieure à 8

$C_3$  la variation des notes autour de la moyenne doit être la plus petite possible

	$M_1$	$M_2$	$M_3$	$M_4$	$M_5$
$E_1$	7	13	8	12	11
$E_2$	8	11	11	12	11
$E_3$	20	2	10	3	15
$E_4$	16	14	16	14	13
$E_5$	12	12	8	8	10

$$C_1(E) = \frac{\sum_{\text{matière}} \text{Note}(E, \text{matière})}{5}$$

$$C_2(E) = \begin{cases} 10 & \text{si } \min_{\text{matière}} (\text{Note}(E, \text{matière})) > 8 \\ 0 & \text{sinon} \end{cases}$$

$$C_3(E) = \frac{\sqrt{\sum_{\text{matière}} (\text{Note}(E, \text{matière}) - C_1(E))^2}}{5}$$



# Electre I Exemple

	$C_1$	$C_2$	$C_3$
$E_1$	10.2	0	1.035
$E_2$	10.6	10	0.606
$E_3$	10	0	3.08
$E_4$	14.6	10	0.5366
$E_5$	10	10	0.8

Les critères non  
normalisés

	$C'_1$	$C'_2$	$C'_3$
$E_1$	0.87	0	16.08
$E_2$	2.61	20	19.45
$E_3$	0	0	0
$E_4$	20	20	20
$E_5$	0	20	17.9

Les critères  
normalisés

Cette étape n'est pas obligatoire



# Electre I Principe

La méthode Electre I permet d'effectuer une sélection

1. Affectation des poids aux critères
2. Comparaisons entre actions
3. Conversion en valeurs numériques
4. Calcul des indices de concordance / discordance
5. Filtrage des actions
6. Représentation sous la forme de graphe



# Electre I Principe

On affecte un poids  $P_i$  à chaque critères  $g_i$

On effectue des comparaisons entre actions. On calcule les ensembles de comparaison suivant:

$$J^+(a_i, a_k) = \{j \text{ tq } g_j(a_i) > g_j(a_k)\} \leftarrow$$

$$J^-(a_i, a_k) = \{j \text{ tq } g_j(a_i) = g_j(a_k)\}$$

$$J^-(a_i, a_k) = \{j \text{ tq } g_j(a_i) < g_j(a_k)\}$$

Ensemble des indices des critères pour lesquels il y a préférence

Conversion des relations en valeur numérique:

$$P^+(a_i, a_k) = \sum_j p_j \text{ avec } j \in J^+(a_i, a_k) \leftarrow$$

$$P^-(a_i, a_k) = \sum_j p_j \text{ avec } j \in J^-(a_i, a_k)$$

Somme des poids des critères pour lesquels il y a préférence

$$P^-(a_i, a_k) = \sum_j p_j \text{ avec } j \in J^-(a_i, a_k)$$



# Electre I

## Principe

Calcul de indices de concordance et de discordance :

$$C_{ik} = \frac{P^+(a_i, a_k) + P^-(a_i, a_k)}{P^+(a_i, a_k) + P^-(a_i, a_k) + P^{\bar{}}(a_i, a_k)}$$

$$D_{ik} = \begin{cases} 0 & \text{si } J^-(a_i, a_k) = \emptyset \\ \frac{1}{\delta_j} \max(g_j(a_k) - g_j(a_i)) & \text{avec } j \in J^-(a_i, a_k) \text{ sinon} \end{cases}$$

Filtrage des actions :

$$\left. \begin{array}{l} C_{ik} \geq c \\ D_{ik} \leq d \end{array} \right\} \rightarrow a_i S a_k$$

c : seuil de concordance (~0.7)  
d : seuil de discordance (~0.3)



# Electre I Résultats

	$a_1$	$a_2$	$a_3$	$a_4$	$a_5$
$a_1$		$J^-$	$J^+$	$J^-$	$J^+$
$a_2$	$J^+$		$J^+$	$J^-$	$J^+$
$a_3$	$J^-$	$J^-$		$J^-$	$J^=$
$a_4$	$J^+$	$J^+$	$J^+$		$J^+$
$a_5$	$J^-$	$J^-$	$J^=$	$J^-$	

Critère  $C_1$  ↑

	$a_1$	$a_2$	$a_3$	$a_4$	$a_5$
$a_1$		$J^-$	$J^=$	$J^-$	$J^-$
$a_2$	$J^+$		$J^+$	$J^=$	$J^=$
$a_3$	$J^=$	$J^-$		$J^-$	$J^-$
$a_4$	$J^+$	$J^=$	$J^+$		$J^=$
$a_5$	$J^+$	$J^=$	$J^+$	$J^=$	

Critère  $C_2$  ↑

	$a_1$	$a_2$	$a_3$	$a_4$	$a_5$
$a_1$		$J^-$	$J^+$	$J^-$	$J^-$
$a_2$	$J^+$		$J^+$	$J^-$	$J^+$
$a_3$	$J^-$	$J^-$		$J^-$	$J^-$
$a_4$	$J^+$	$J^+$	$J^+$		$J^+$
$a_5$	$J^+$	$J^-$	$J^+$	$J^-$	

Critère  $C_3$  ↑

	$a_1$	$a_2$	$a_3$	$a_4$	$a_5$
$a_1$		$\emptyset$	$\{1, 3\}$	$\emptyset$	$\{1\}$
$a_2$	$\{1, 2, 3\}$		$\{1, 2, 3\}$	$\emptyset$	$\{1, 3\}$
$a_3$	$\emptyset$	$\emptyset$		$\emptyset$	$\emptyset$
$a_4$	$\{1, 2, 3\}$	$\{1, 3\}$	$\{1, 2, 3\}$		$\{1, 3\}$
$a_5$	$\{2, 3\}$	$\emptyset$	$\{2, 3\}$	$\emptyset$	

Résumé pour  $J^+$  ↑

	$a_1$	$a_2$	$a_3$	$a_4$	$a_5$
$a_1$		$\{1, 2, 3\}$	$\emptyset$	$\{1, 2, 3\}$	$\{2, 3\}$
$a_2$	$\emptyset$		$\emptyset$	$\{1, 3\}$	$\emptyset$
$a_3$	$\{1, 3\}$	$\{1, 2, 3\}$		$\{1, 2, 3\}$	$\{2, 3\}$
$a_4$	$\emptyset$	$\emptyset$	$\emptyset$		$\emptyset$
$a_5$	$\{1\}$	$\{1, 3\}$	$\emptyset$	$\{1, 3\}$	

Résumé pour  $J^-$  ↑

	$a_1$	$a_2$	$a_3$	$a_4$	$a_5$
$a_1$		$\emptyset$	$\{2\}$	$\emptyset$	$\emptyset$
$a_2$	$\emptyset$		$\emptyset$	$\{2\}$	$\{2\}$
$a_3$	$\{2\}$	$\emptyset$		$\emptyset$	$\{1\}$
$a_4$	$\emptyset$	$\{2\}$	$\emptyset$		$\{2\}$
$a_5$	$\emptyset$	$\{2\}$	$\{1\}$	$\{2\}$	

Résumé pour  $J^=$  ←



# Electre I Résultats

	$a_1$	$a_2$	$a_3$	$a_4$	$a_5$
$a_1$		$\emptyset$	{1, 2, 3}	$\emptyset$	{1}
$a_2$	{1, 2, 3}		{1, 2, 3}	{2}	{1, 2, 3}
$a_3$	{2}	$\emptyset$		$\emptyset$	{1}
$a_4$	{1, 2, 3}	{1, 2, 3}	{1, 2, 3}		{1, 2, 3}
$a_5$	{1, 2, 3}	{2}	{1, 2, 3}	{2}	

## Résumé sur $J^+$ union $J^-$

	$a_1$	$a_2$	$a_3$	$a_4$	$a_5$
$a_1$		0	0.75	0	0.5
$a_2$	1		1	0	0.75
$a_3$	0	0		0	0
$a_4$	1	0.75	1		0.75
$a_5$	0.5	0	0.5	0	

## Résumé pour $P_{ik}^+$

	$a_1$	$a_2$	$a_3$	$a_4$	$a_5$
$a_1$		0	0.25	0	0
$a_2$	0		0	0.25	0.25
$a_3$	0.25	0		0	0.5
$a_4$	0	0.25	0		0.25
$a_5$	0	0.25	0.5	0.25	

## Résumé pour $P_{ik}^-$



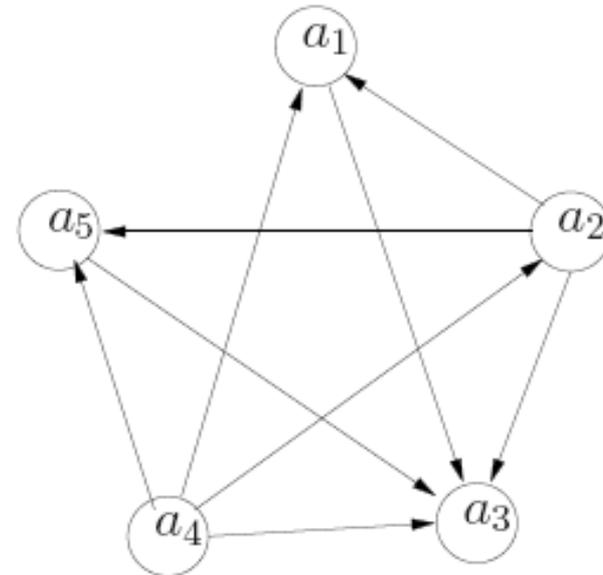
# Electre I Résultats

	$a_1$	$a_2$	$a_3$	$a_4$	$a_5$
$a_1$		0	1	0	0.5
$a_2$	1		1	0.25	1
$a_3$	0.25	0		0	0.5
$a_4$	1	0.75	1		1
$a_5$	0.5	0.25	1	0.25	

Résumé concernant  
l'indice de concordance  $C_{ik}$

	$a_1$	$a_2$	$a_3$	$a_4$	$a_5$
$a_1$		0.087	0	0.196	0.196
$a_2$	0		0	0.13	0
$a_3$	0.0435	0.1305		1	0.895
$a_4$	0	0	0		0
$a_5$	0.0435	0.0775	0	0.105	

Résumé concernant  
l'indice de discordance  $D_{ik}$



Graphe des relations entre  
actions



# Les méthodes Electre

Elles permettent de réaliser un certains nombre d'actions:

**Sélection** : ELECTRE I et IS

**Affectation** : ELECTRE II, III et IV

**Classement** : ELECTRE TRI



**RENAULT**

**Les Applications**

Y. Collette - 64240



**Le problème de séquençement de véhicules RENAULT :  
un problème benchmark revisité d'optimisation sous  
contraintes et une nouvelle aire de jeux pour les  
métaheuristiques ainsi que les méthodes de recherche locales**

**Présentation extraite d'un article de O. Briant, D. Naddef et G. Mounié**

**Y. Collette**



# Plan

- Présentation du problème
- Algorithmes implémentés par les candidats du challenge ROADEF 2005
- Présentation de l'approche recuit simulé



# Le processus de planification de la journée de production dans les usines

Echelle de temps : jour J+5 à mois M+4

Où : usines d'assemblage

Quand : tous les jours

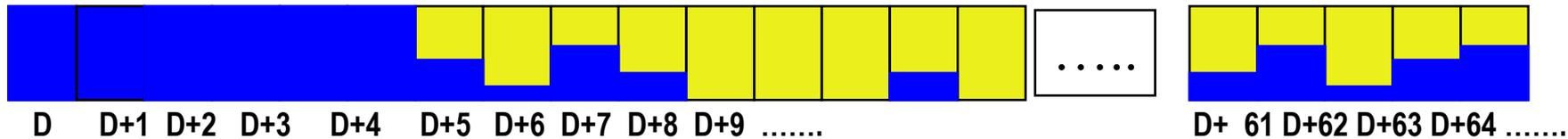


Commandes clients



Commandes pour le stock

## Plan de Production



### Contraintes : Tenir compte des capacités

- des lignes de montage
- des fournisseurs
- des unités de travail

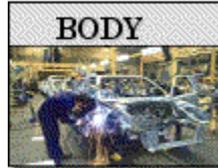
### Objectifs :

- Maximiser le nombre de véh. Accepter dans le plan de production
- Satisfaire les délais de livraison clients
- Assurer la stabilité du plan de prod.



# Plannification d'un jour de production

Séquence de véh. →



↙  
**Lots de caisses**

↕  
**Lots de caisses peintes**

↘  
**Contraintes de Ratio**

▶ **Lots de caisses**

- ✓ Lots de véhicules ayant les mêmes caractéristiques (3-portes, toit ouvrant ...)
- ✓ Minimiser les changements d'outils qui requière des opérations manuelles

▶ **Lors de caisses peintes**

- ✓ Lots de véhicules ayant la même couleur
- ✓ Minimiser la consommation de solvants dus aux changements de couleurs

▶ **Contraintes de Ratio (espacement)**

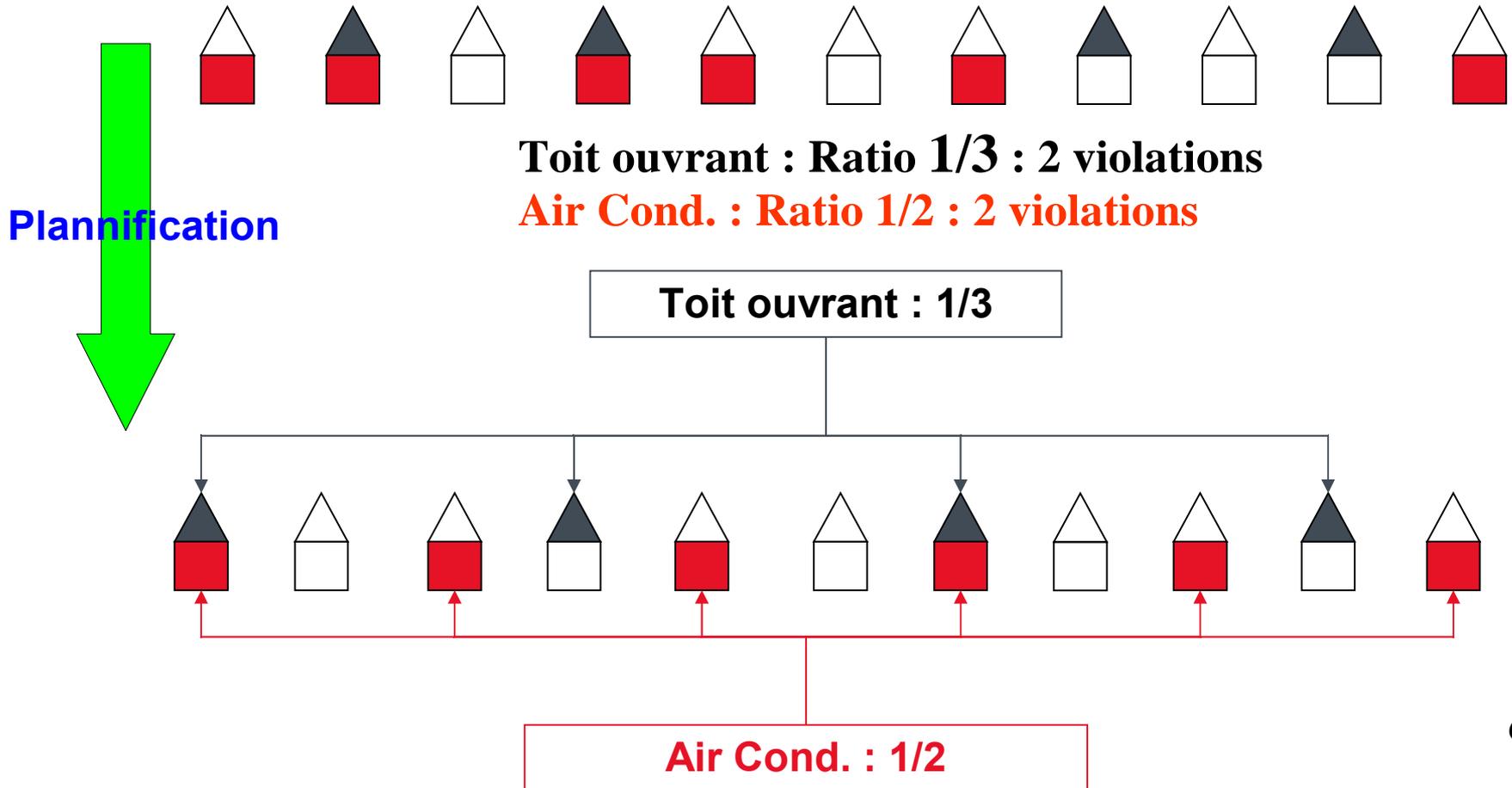
- ✓ Ratio  $N/P$  pour l'option  $i$  : pour chaque séquence consécutives de  $P$  véhicules, au plus  $N$  véhicules requière l'option  $i$
- ✓ Lisser la charge de travail sur la ligne d'assemblage
- ✓ Ne pas surcharger les unités de travail critiques



# Contraintes de Ratio

## Les Écarts de critère

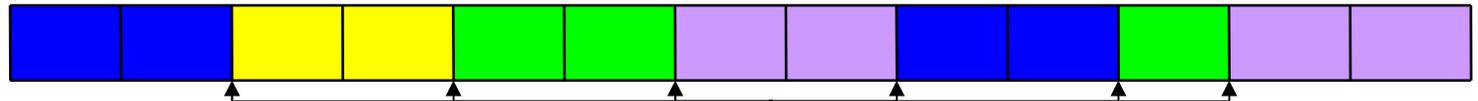
Comment réorganiser une séquence de montage





# Lots de caisses peintes

Comment réorganiser une séquence de peinture



Changement de couleurs : 6

Plannification



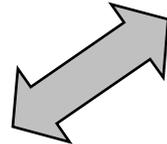
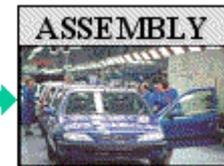
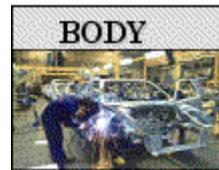
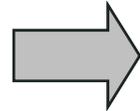
Changement de couleurs : 3



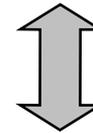


# Le problème soumis au challenge ROADEF

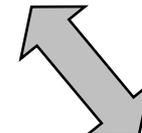
Séquence de véh.



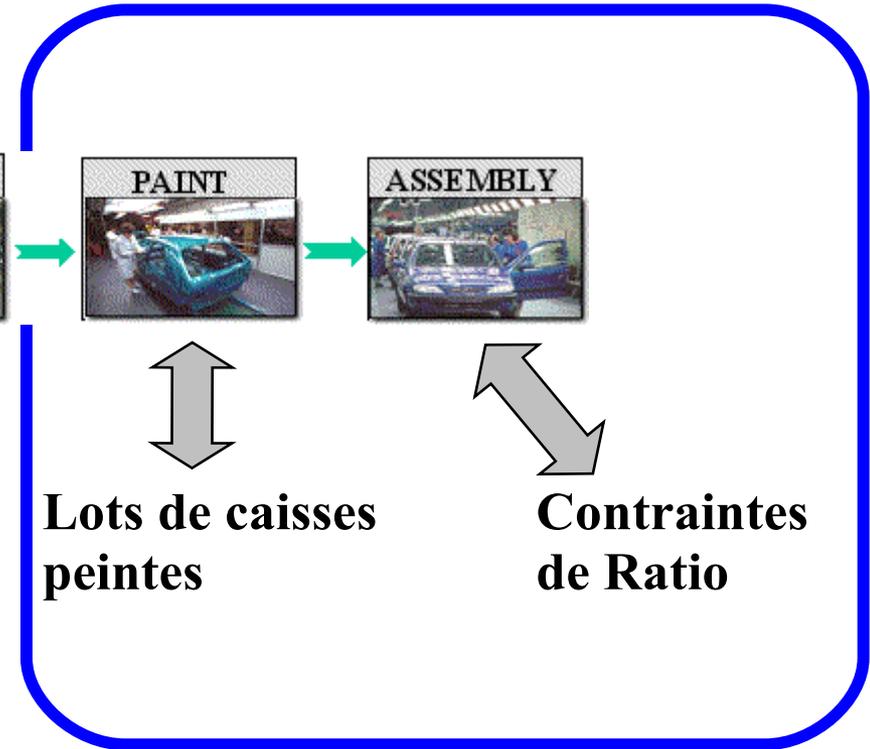
Lots de caisses



Lots de caisses  
peintes



Contraintes  
de Ratio



Challenge **ROADEF 2005**



# Plannification d'un jour de production

## **Objectif**

Construire une séquence de véh. :

Qui maximise des contraintes « douces » de l'atelier d'assemblage

Qui minimise les coûts de production de l'atelier de peinture

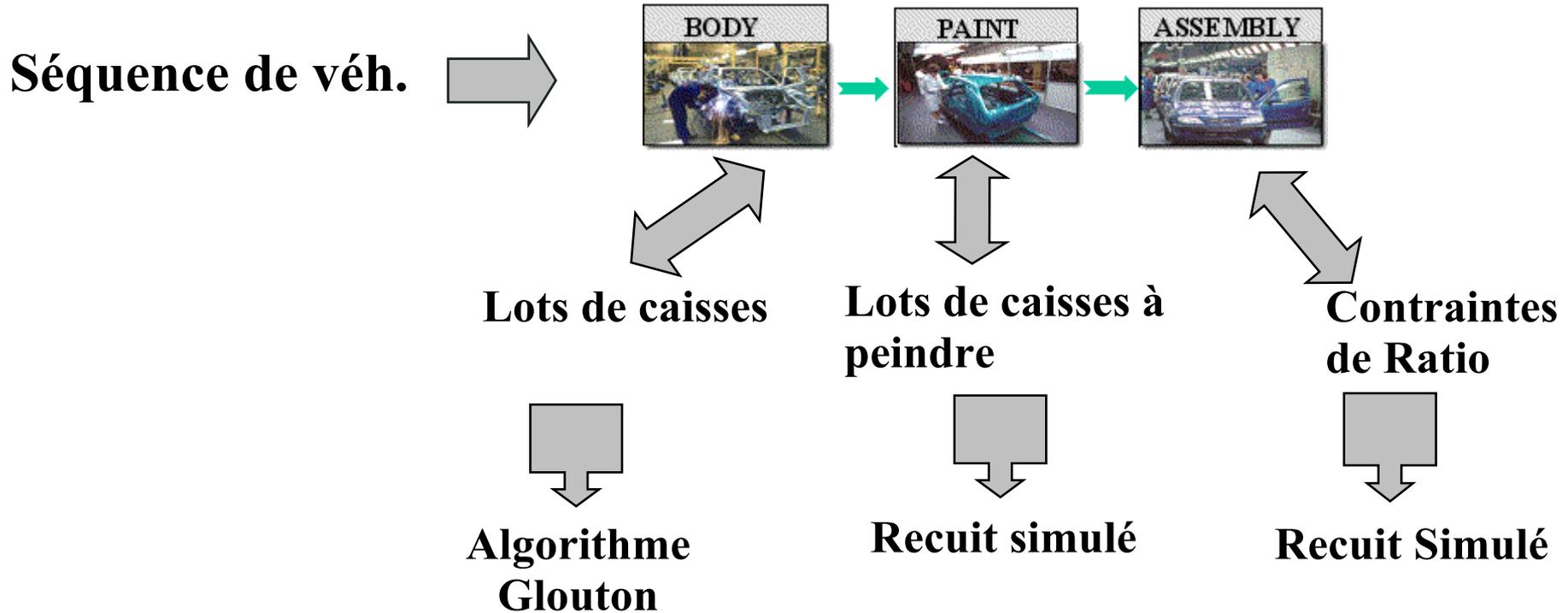
## **Contraintes fortes**

Longueur maximale de la séquence de caisses à peindre

(due aux exigences de nettoyage des installations de peinture)



# Approche RENAULT



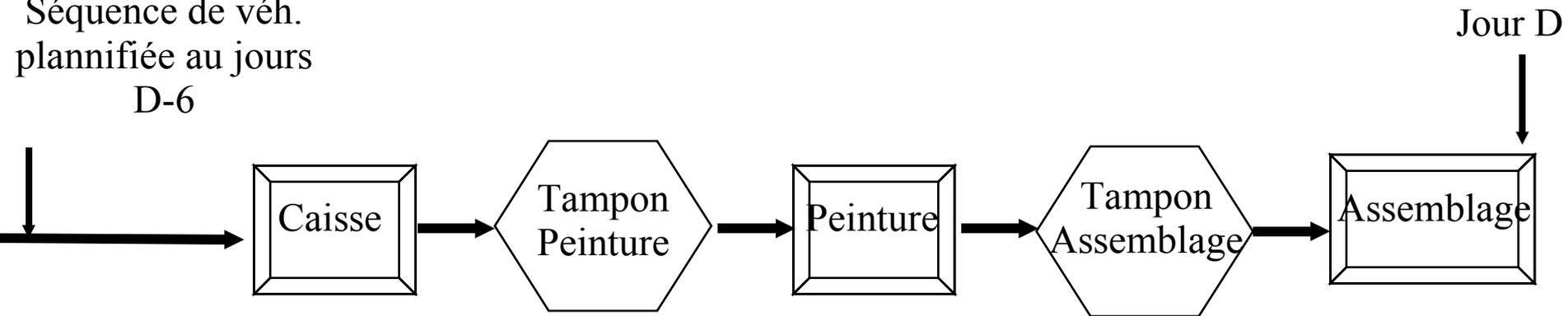
Ce système de séquençement de véhicule est en fonction depuis 1993 dans l'ensemble des usines RENAULT

⇒ Pourquoi un nouvel algorithme de séquençement de véhicules ?



# Le séquençement de véhicules: un nouveau processus critique pour RENAULT

Séquence de véh.  
plannée au jours  
D-6



## ► Stratégie RENAULT : Une compliance stricte vis-à-vis de la séquence de véhicules initiale

- ✓ De façon à assurer une synchronisation avec la livraison des pièces fournisseur (fenêtre de 2 heures émise au jour D-6)
- ✓ Le tampon peinture est limité en taille et le tampon assemblage réordonne la séquence de façon à reproduire la planification initiale
- ✓ La performance de l'algorithme de séquençage est maintenant critique !



# Optimisation Multiobjectif

- Optimisation multicritère lexicographique
  - Pas de compensation entre les objectifs (Contraintes de Ratio et ensuite de caisses à peindre)
  - La priorité entre les fonctions objectifs est donnée par chacune des usines
- Configuration des usines

## Usines à main-d'oeuvre coûteuse (e.g. Europe de l'Ouest)

- Contraintes de ratio à forte priorité (demandes fortes de la part des ateliers de montage)
- Lots de caisses à peindre
- Contraintes de ratio à faible priorité

## Usines à main-d'oeuvre peu coûteuse (e.g. Roumanie/Turquie)

- Lots de caisses à peindre
- Contraintes de ratio à forte priorité
- Contraintes de ratio à faible priorité



# Contraintes de Ratio

- Sens industriel d'une contrainte de ratio  $1/N$  sur l'option  $o_i$ 
  - Pour espacer les véh. ayant l'option  $o_i$  autant que possible à  $N$  positions
  - Quand ce n'est pas possible pour certains véh., les espacer à  $N-1$  positions, puis à  $N-2$  positions etc
  
- Maximiser l'espace entre des véh. qui violent les contraintes de ratio
  - ⇒ Compter les violations en utilisant une fenêtre glissante
  
- Affecter des pénalités lourdes aux véhicules qui sont trop proches

Exemple Contrainte de Ratio =  $1/5$

Séquences (a)  $\_ X \_ \_ \_ X \_$  (b)  $\_ \_ X \_ \_ X \_$

⇒ La séquence (a) est meilleure que la séquence (b)

(f)  $\_ X \_ \_ \_ X \_ : 1$  violation dans la fenêtre  $\_ [X \_ \_ \_ X] \_$

(g)  $\_ \_ X \_ \_ X \_ : 1$  violation dans la fenêtre  $\_ [ \_ X \_ \_ X ] \_$   
 + 1 violation dans la fenêtre  $\_ \_ [X \_ \_ X \_ ]$

⇒ nb. de séquence violant la contrainte pour (a) < nb. de séquences violant la contrainte pour (b)



# Fonction objectif pour les contraintes de Ratio

## Notations

$n_{car}$  : le nombre total de véhicules

$n_{opt}$  : le nombre total d'options

Contraintes de Ratio  $i$  :  $p_i / q_i$

$c_{ki}$  : Paramètre booléen indiquant si le véh.  $k$  requière l'option  $i$

$V_{ij}$  : Nombre de violations de contraintes de ratio  $i$  dans la séquence commençant à la position  $j$  et de longueur  $q_i$

$Y_{kj}$  : Paramètre booléen indiquant si le véh. est positionné en  $j$

$$\text{Min } F = \sum_{i=1}^{n_{opt}} \sum_{j=1}^{n_{car}-q_i+1} V_{ij}$$

Minimiser les violations de contraintes sur l'ensemble des véh. et l'ensemble des options

S'il n'y a pas de violation de contraintes, on met 0

où  $V_{ij} = 0$  si  $\sum_{s=j}^{j+q_i-1} \sum_{k=1}^{n_{car}} c_{ki} * Y_{ks} \leq p_i$

S'il y a violation de contraintes, on met le dépassement de la contrainte

$= \sum_{s=j}^{j+q_i-1} \sum_{k=1}^{n_{car}} c_{ki} * Y_{ks} - p_i$  autrement (1 dans le modèle classique)

Ancienne méthode RENAULT de décompte

⇒ La fonction objectif est le nombre de véhicules qui violent la contrainte de ratio (et non pas le nombre de contraintes de ratio violées comme dans le modèle classique !)



# Contraintes de Ratio

## Exemple de calcul

Exemple Contrainte de Ratio = 1/5

Séquences (a) XX X\_ \_ et (b) \_ X \_ X \_ X \_

⇒ séquence (b) est meilleure que la séquence (a)

- (a)
- \_ \_ X X \_ X \_ \_
  - [\_ \_ XX] \_ X \_ \_ : 1 violation RENAULT (1 violation dans le cas classique)
  - [\_ \_ XX \_] X \_ \_ : 1 violation RENAULT (1 violation dans le cas classique)
  - \_ [XX \_ X] \_ \_ : 2 violations RENAULT (1 violation dans le cas classique)
  - \_ \_ [XX \_ X \_] \_ : 2 violation sRENAULT (1 violation dans le cas classique)
  - \_ \_ X [X \_ X \_ \_] : 1 violation RENAULT (1 violation dans le cas classique)

total : **7 violations RENAULT** (5 violations dans le cas classique)

- (b)
- \_ X \_ X \_ X \_
  - [\_ X \_ X] \_ X \_ : 1 violation RENAULT (1 violation dans le cas classique)
  - [X \_ X \_] X \_ \_ : 1 violation RENAULT (1 violation dans le cas classique)
  - \_ [X \_ X \_ X] \_ \_ : 2 violations RENAULT (1 violation dans le cas classique)
  - \_ X [X \_ X \_] \_ : 1 violation RENAULT (1 violation dans le cas classique)
  - \_ X \_ [X \_ X \_ \_] : 1 violation RENAULT (1 violation dans le cas classique)

total : **6 violations RENAULT** (5 violations dans le cas classique)

⇒ nb. de séquences de violations (b) < nb. de séquences de violations (a)



# Environnement industriel

- Limite temporelle : **10 minutes** de temps CPU sur un Pentium-IV
- Problème de grande tailles

Usines RENAULT	Véh.	options & couleurs	ratio avec taille max block	option/configurations couleurs
022	485	11 options / 15 couleurs	10/15	58
064-2	335	6 options / 12 couleurs	1/100	37
064-1	875	9 options / 14 couleurs	1/100	133
039	954	5 options / 14 couleurs	1/297	113
048	600	17 options / 12 couleurs	1/274	229
024	1260	13 options / 13 couleurs	1/20	237
025	485	22 options / 24 couleurs	1/76	287

CSP lib	Véh.	options & couleurs	ratio avec taille max block	option/configurations couleurs
Instancesclassiques	100-200	5 options	2/5	17-30
Nouvelles instances	200-400	5 options	2/5	19-26



# Differences avec le problème classique de séquençement de véhicules

- ✓ Optimisation multiobjectif
- ✓ Prise en compte des objectifs de l'atelier de peinture
- ✓ Fonction objectif modifiée pour les contraintes de Ratio
- ✓ Prise en compte des séquences à jour D-1
- ✓ Problèmes de grandes tailles
- ✓ Un problème d'optimisation et non pas un problème de satisfaction de contraintes
  - ⇒ *Même s'il peut subsister des violations de contraintes de ratio, la séquence de véhicules doit être produite !*



- Présentation du problème
- Algorithmes implémentés par les candidats du challenge ROADEF 2005
- Présentation de l'approche recuit simulé



# Méthodes implémentées par les 25 équipes candidates

Méthodes	Nombre d'équipes
Méthode Glouton pour construire la solution initiale	Toutes sauf 1
Recuit simulé	5
Recherche Tabou	7
Voisinage variable / Recherche à large voisinage / Recherche locale itérée	13
Algorithme génétique	1
Colonies de fourmis	1
Programmation linéaire en nombres entiers	1



# Méthodes des 12 meilleures équipes

Place	Méthodes	Equipe
1	Algorithme Glouton + Recherche locale	Estellon, Gardi et Nouioua
2	Algorithme Glouton + Recherche à voisinage variable + Recherche locale itérée	Aloise, Noronha, Rocha, Ribeiro et Urrutia
3	Algorithme Glouton + Recuit simulé	Briant, Naddef et Mounié
4	Algorithme Glouton + Recuit simulé	Bloemen
5	Algorithme Glouton + Recuit simulé	Kuipers
6	Algorithme glouton + Recherche à voisinage variable + Recherche Tabou	Gavranovic
7	Recherche Tabou	Cordeau, Laporte et Pasin
8	Algorithme Glouton + Recherche à voisinage variable + Recuit simulé	Riesler, Chiarandini, Paquete, Schiavinotto et Stütze
9	Algorithme Glouton + Recherche Tabou	Craciunas, Gendreau et Potvin
10	Algorithme Glouton + Recherche locale	Pawlak, Rucinski, Piechowiak, Plaza
11	Algorithme Glouton + Colonie de fourmis + Rech. Tabou + Recherche à voisinage variable	Gravel, Gagné, Price, Krajecki et Jaillet
12	Algorithme Glouton + Recherche locale	Benoist



- Présentation du problème
- Algorithmes implémentés par les candidats du challenge ROADEF 2005
- **Présentation de l'approche recuit simulé**



# Méthode d'optimisation pour le séquençement des véhicules

Algorithme en 2 phases:

Phase 1: Algorithme glouton (pour les couleurs et les ratios prioritaires)

Phase 2: Recuit simulé multiobjectif

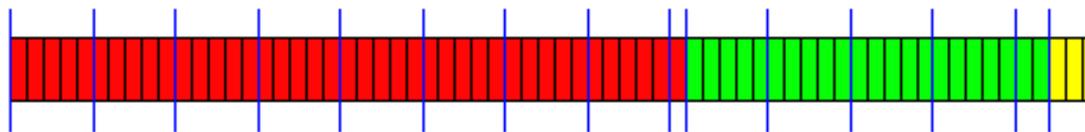


# Algorithme Glouton pour les couleurs

Trie des voitures par couleurs:



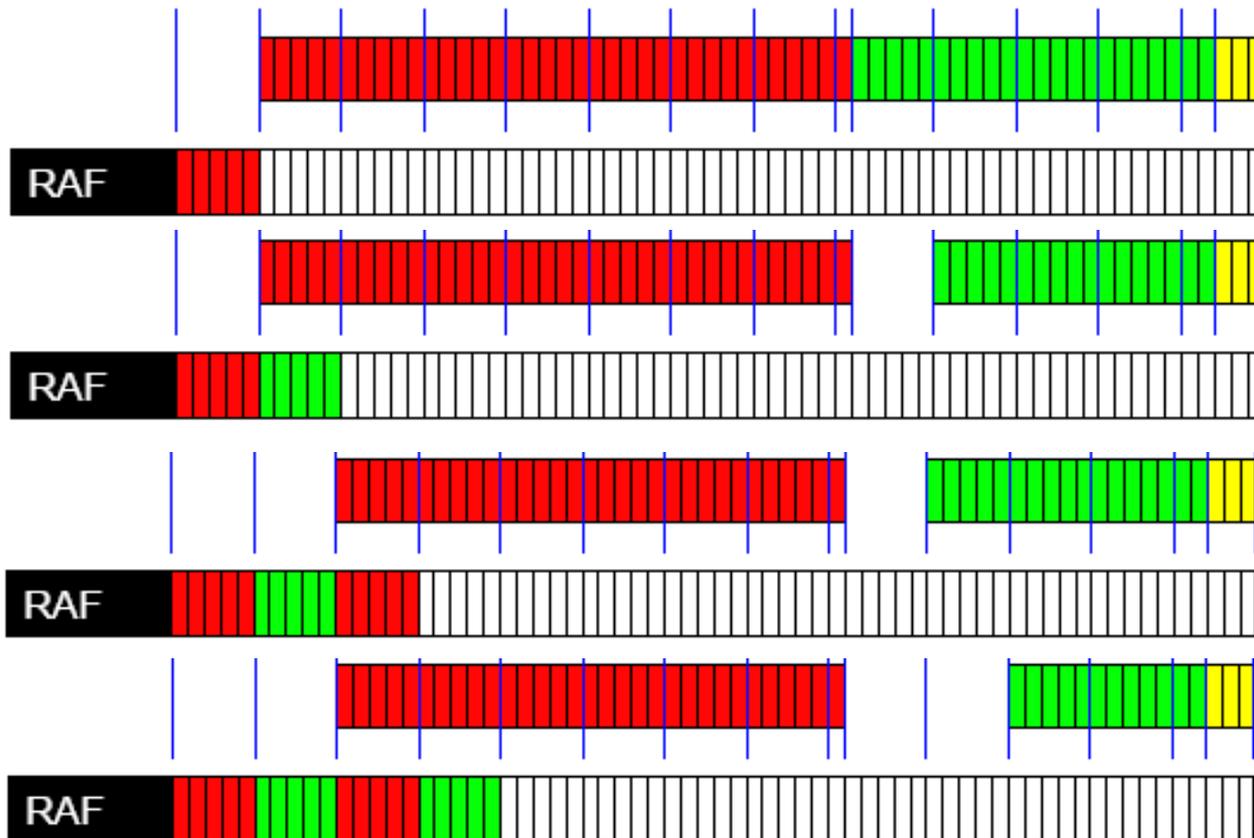
Sélection de blocs d'une même couleur de la plus grande taille possible:





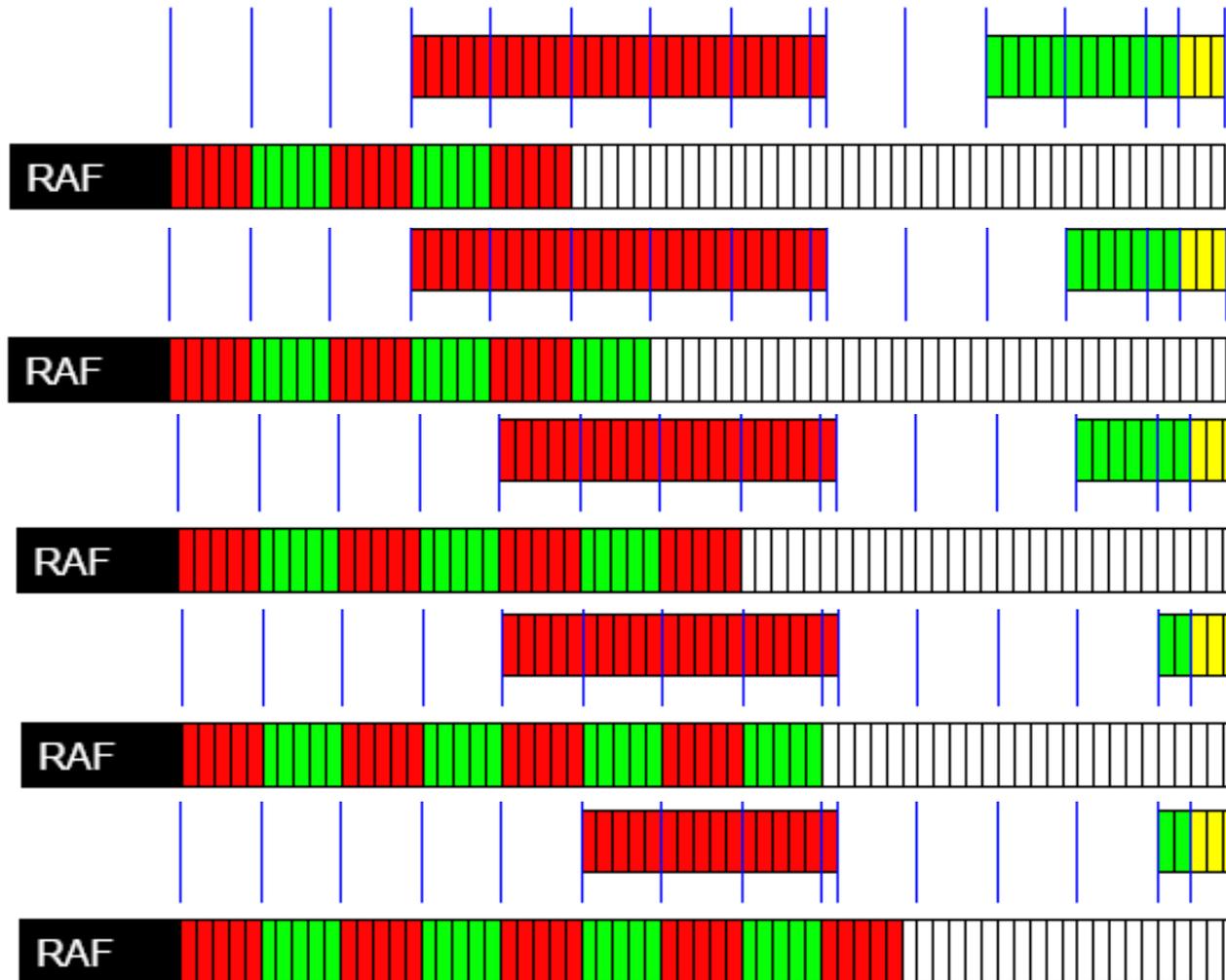
# Algorithme Glouton pour les couleurs

Allocation du bloc de couleur qui reste le plus et qui est différent du dernier bloc



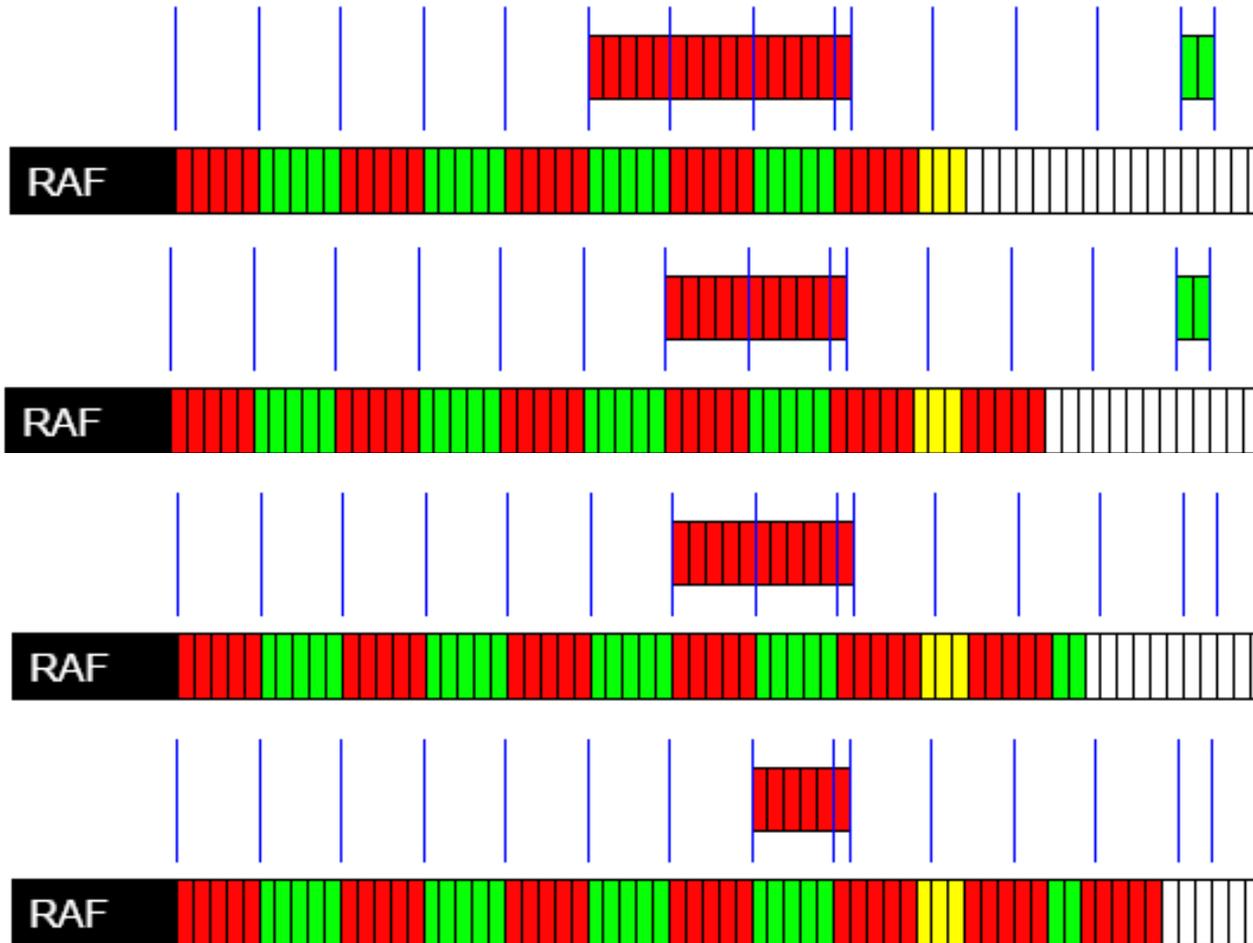


# Algorithme Glouton pour les couleurs





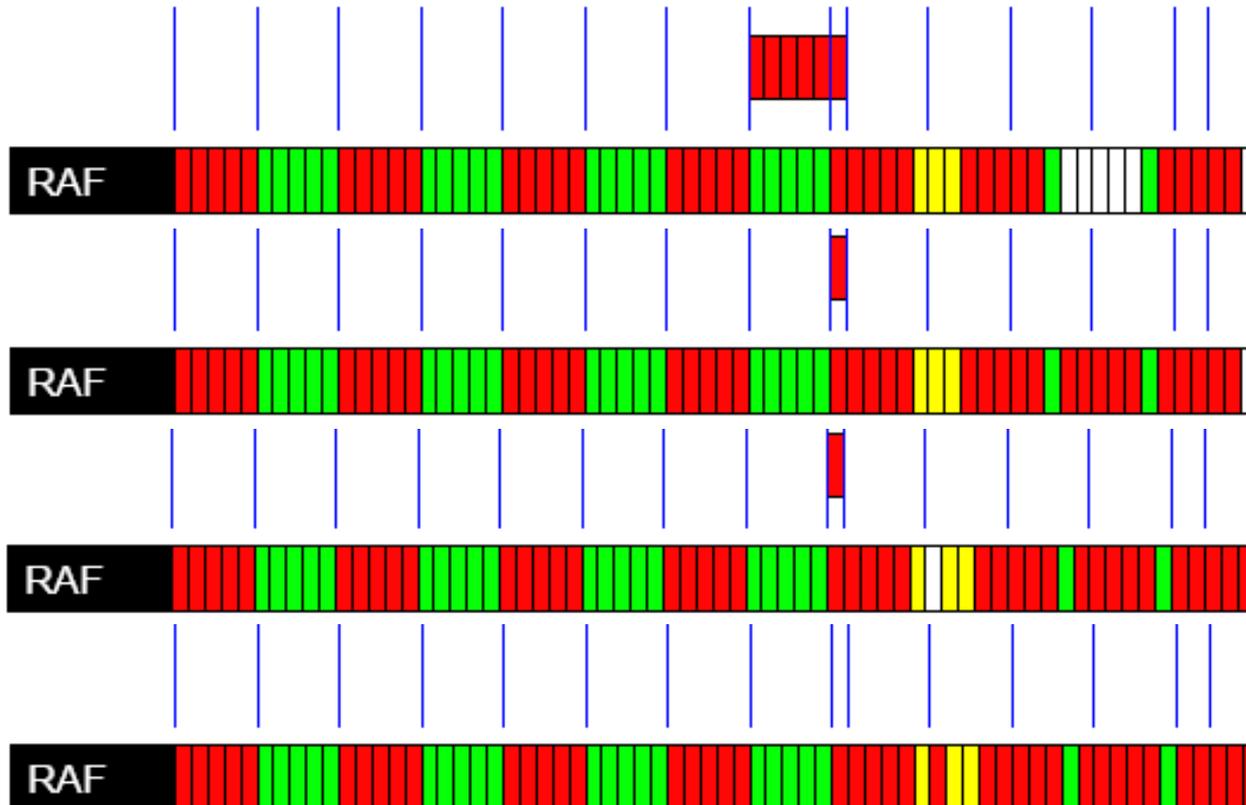
# Algorithme Glouton pour les couleurs





# Algorithme Glouton pour les couleurs

S'il reste un bloc d'une même couleur, découper un bloc d'une autre couleur



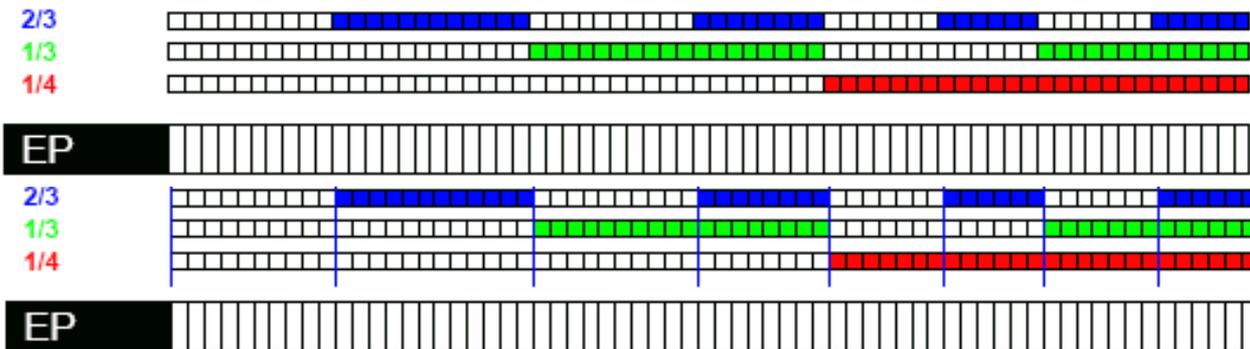


# Algorithme Glouton pour les ratios

Algorithme glouton avec priorisation dynamique



Les véhicules sont classés par famille. Les véhicules concernés par le même ratio font parti de la même famille



EP: ratio prioritaire

ENP: ratio moins prioritaire

RAF: nombre de purges



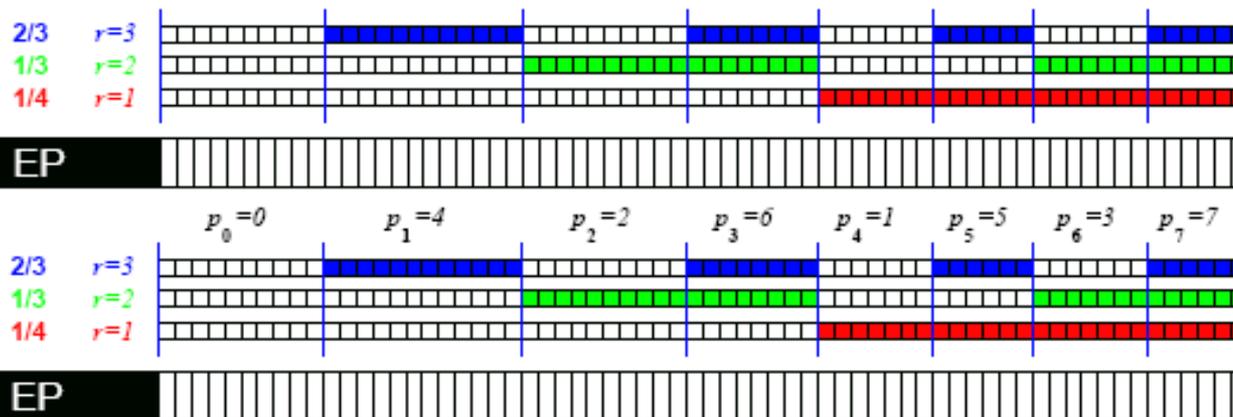
# Algorithme Glouton pour les ratios

A intervalles réguliers, les ratios sont classés en ordre croissants de distance par rapport à la borne de la contrainte.

Soit  $r_j$  le rang du ratio  $j$ . Chaque famille  $k$  de véhicules se voit affecté une priorité

$$p_k = \sum \left\{ 2^{r_j - 1} : \text{la famille } k \text{ contient le ratio } j \right\}$$

On planifie ensuite les véhicules les uns après les autres.



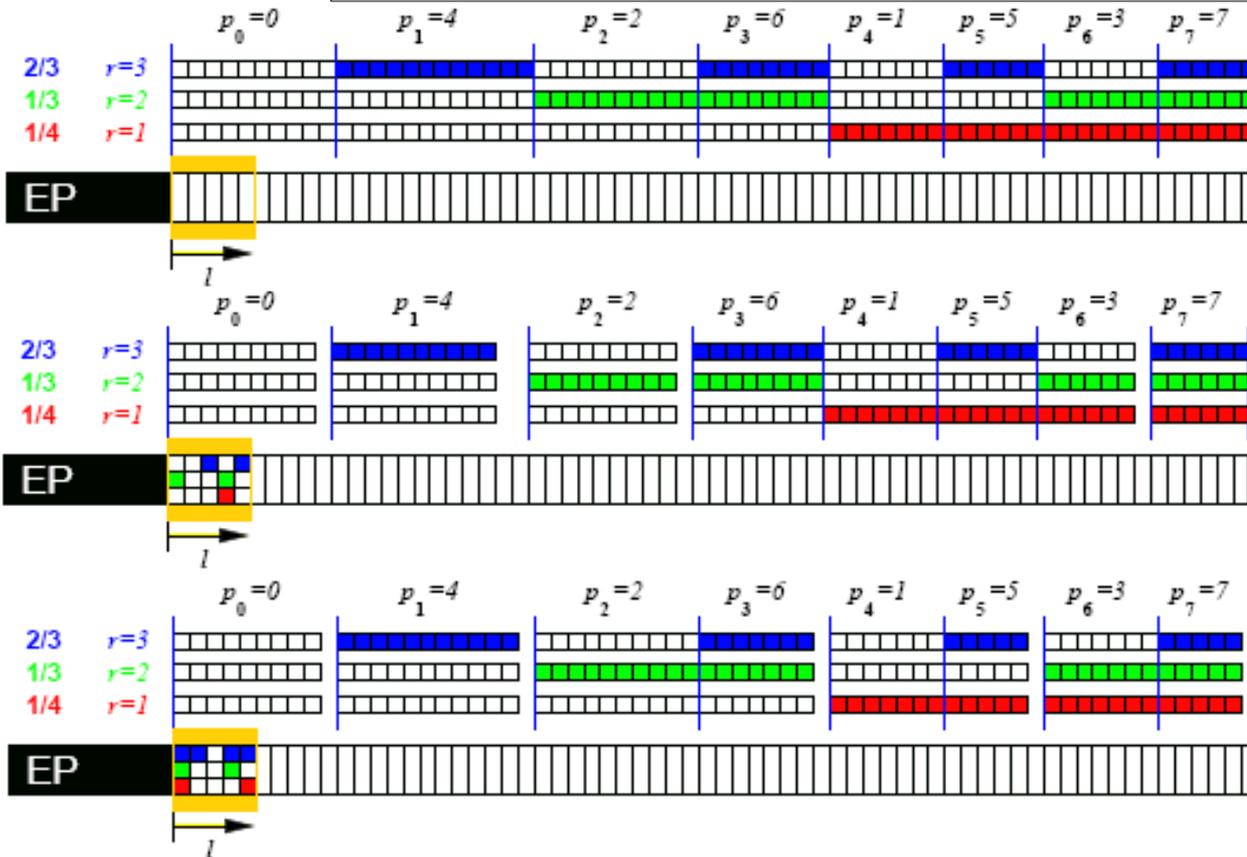
Une famille est d'autant plus prioritaire qu'elle est composée de contraintes de ratio de rang élevé



# Algorithme Glouton pour les ratios

Exploration de toutes les combinaisons d'affectation  $\langle k_1, k_2, \dots, k_l \rangle$  sans violation dans les prochaines  $l$  positions, et stocker les affectations qui maximisent:

$$f(\langle k_1, k_2, \dots, k_l \rangle) = \sum_{i=1}^l \alpha^i p_{k_i} \text{ avec } \alpha < 1$$



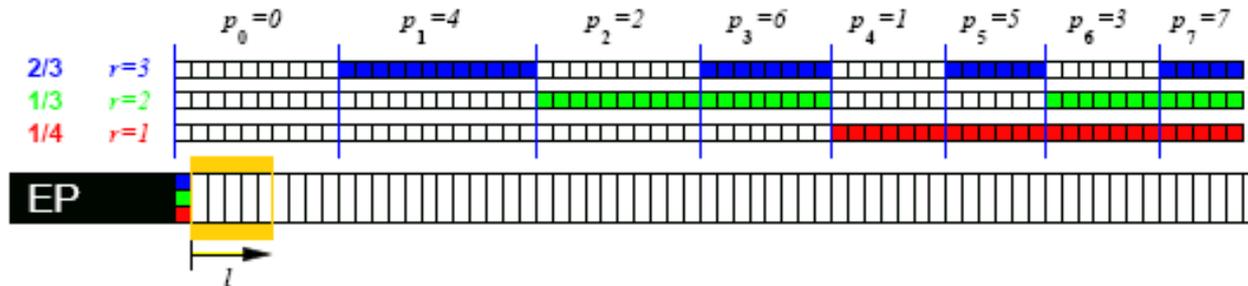
Cette fonction objectif permet de caser les familles non-prioritaires au début de la séquence

De cette façon, on construit « facilement » des séquences de longueur  $l$  qui ne violent pas les contraintes de ratio



# Algorithme Glouton pour les ratios

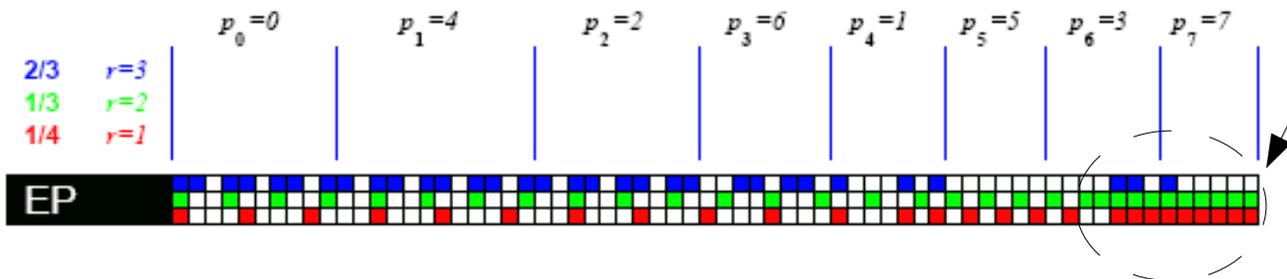
On affecte les premiers  $k_1$  véhicules de cette séquence



La valeur de  $l$  dépend du nombre de ratios. Une valeur entre 4 et 7 semble bien marcher. Si on ne peut pas éviter les violations, on sélectionne l'affectation qui minimise cette violation.

Cette procédure prédictive tend à repousser un maximum la violation des contraintes. On se retrouve alors avec les violations à la fin.

Il semble que cette structure de solution soit efficacement traitée par les méthodes d'amélioration comme le recuit simulé.



On se retrouve avec une séquence bonne au début mais dont la structure ne vérifie pas les contraintes de ratio à la fin !



# Recuit simulé multiobjectif

Procédure d'optimisation:

- Optimisation de la fonction objectif  $k=2$ , puis 3
- Aucune détérioration de la fonction objectif  $k_0 < k$  n'est autorisée
- On passe le même temps pour l'optimisation de chacun des objectifs

Réglage du recuit simulé:

- La température initiale est choisie de façon à ce que la probabilité d'accepter un accroissement d'objectif de 1% soit moins que 0.5%
- Le coefficient de réduction de la température est choisi de façon à ce qu'après  $2/3$  du temps d'affectation pour l'objectif  $k$ , le recuit est gelé

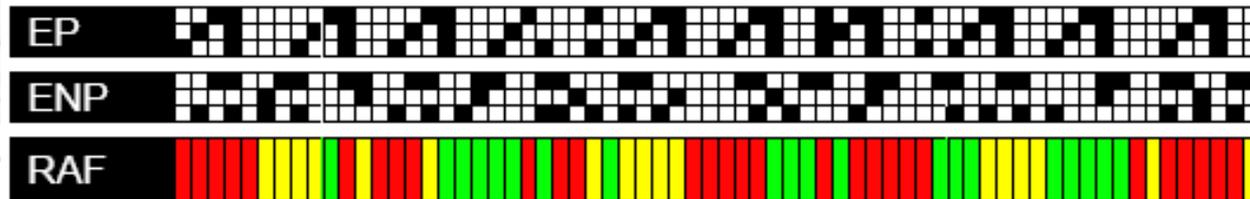


# Recuit simulé multiobjectif

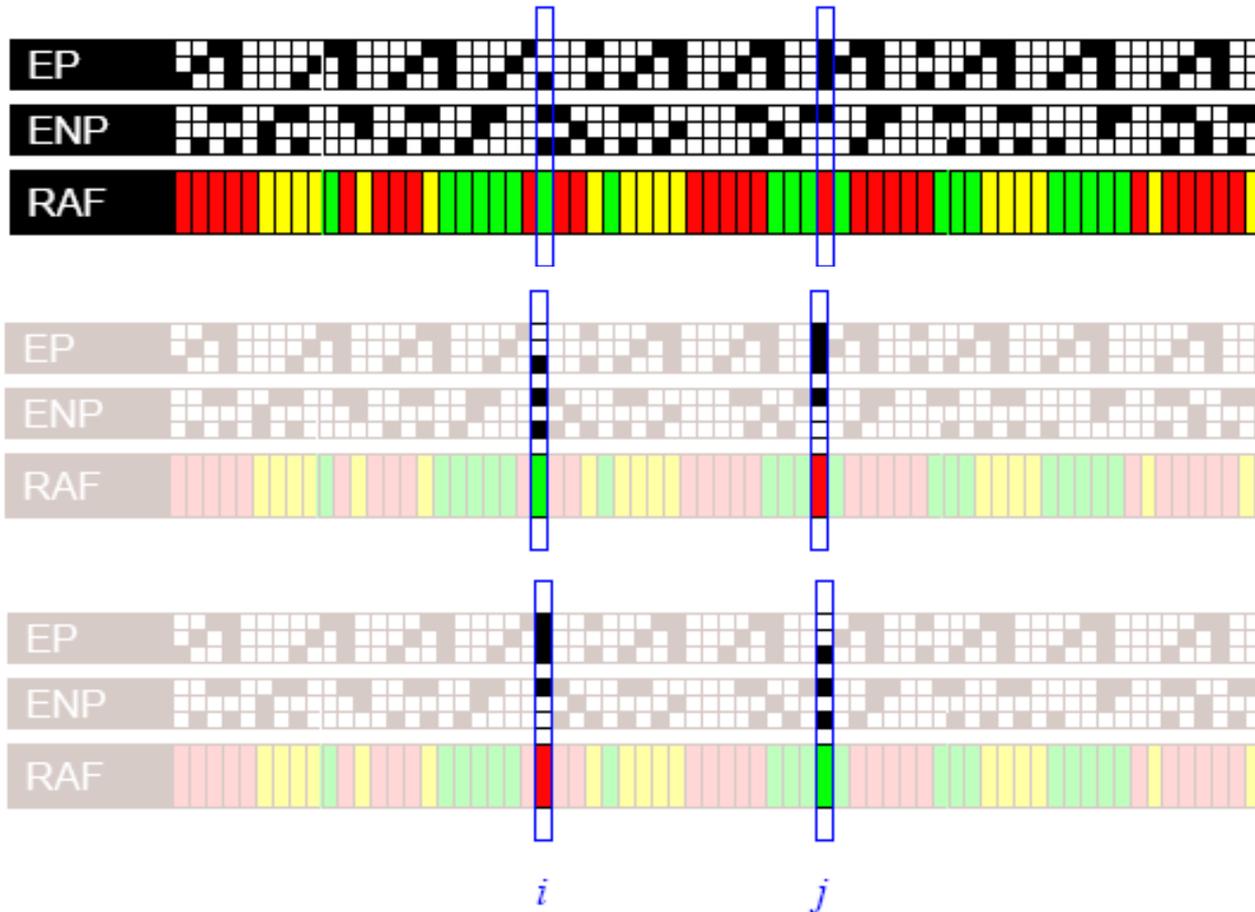
Comment construire un opérateur de voisinage pour le recuit simulé ?



# Recuit simulé multiobjectif



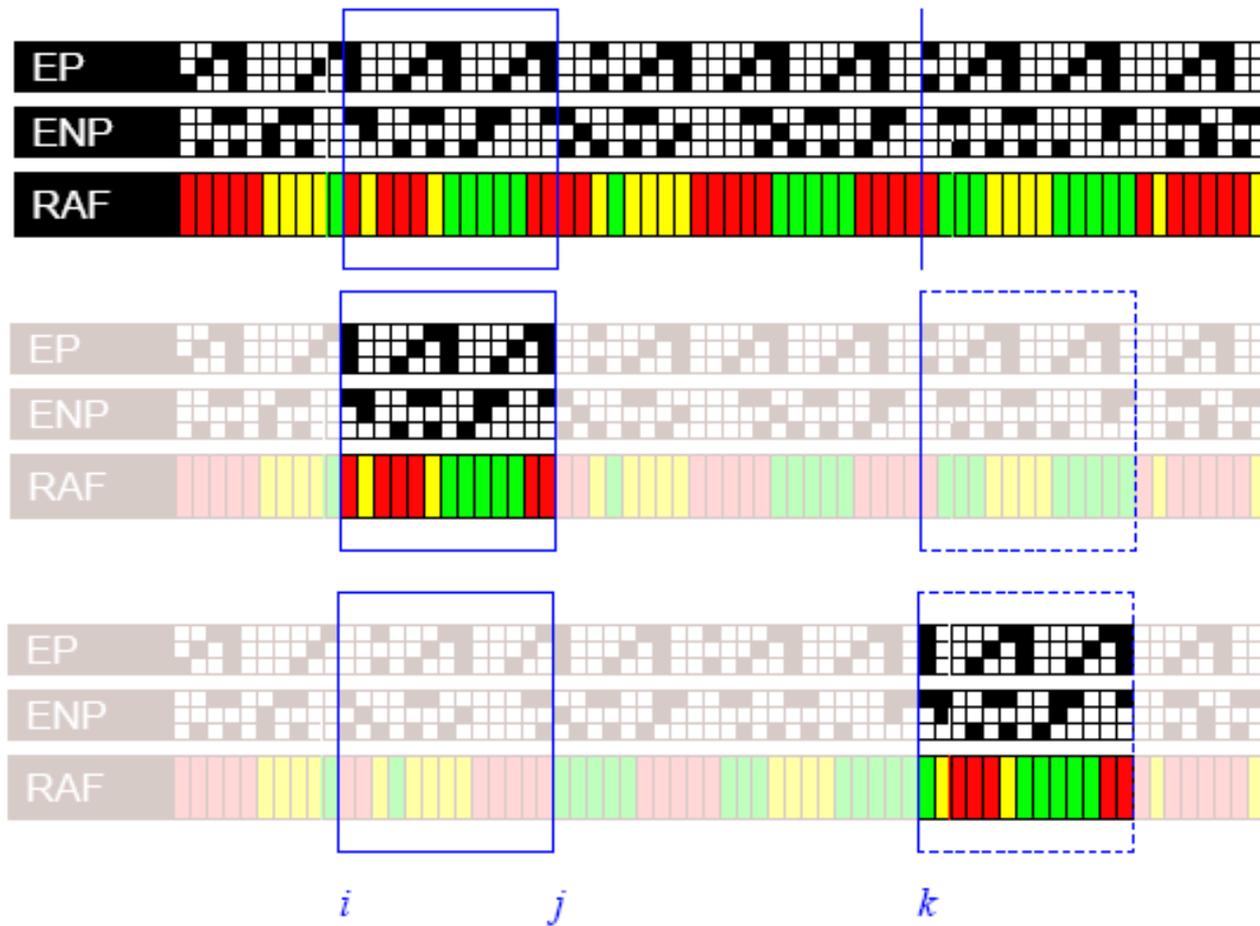
ECHBIN: échange de 2 véhicules  $i$  et  $j$





# Recuit simulé multiobjectif

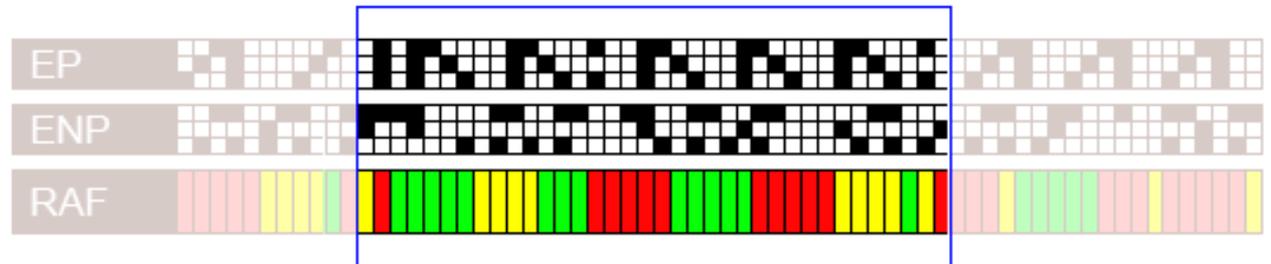
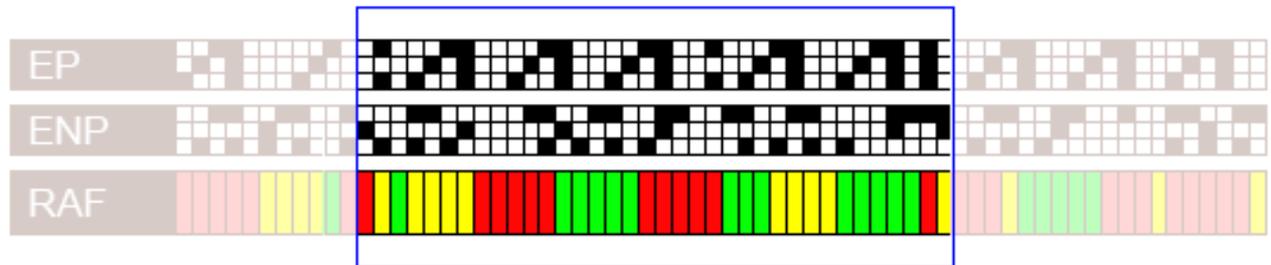
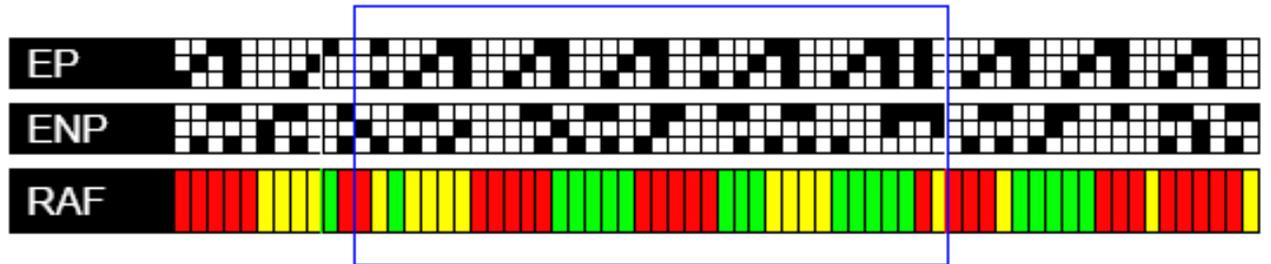
DECBLOC: décalage d'un bloc  $[i,j]$  en position  $k$





# Recuit simulé multiobjectif

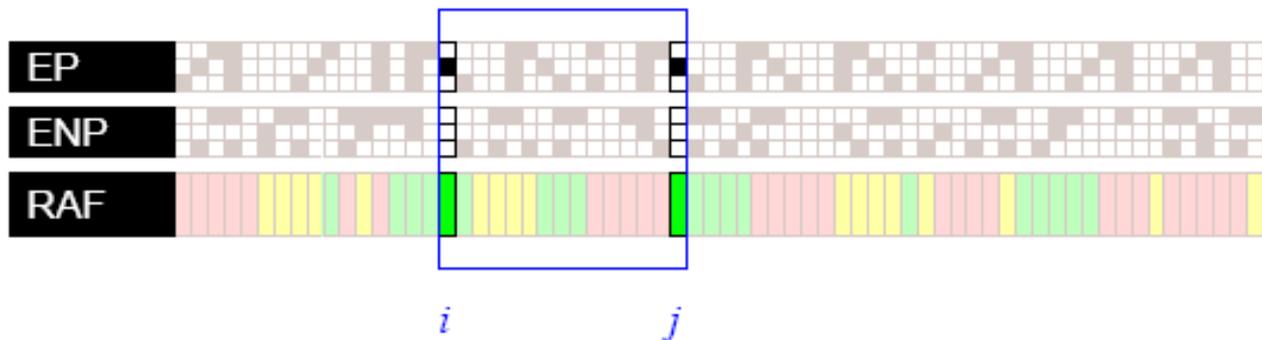
INVBLOC: inversion d'un bloc [i,j]



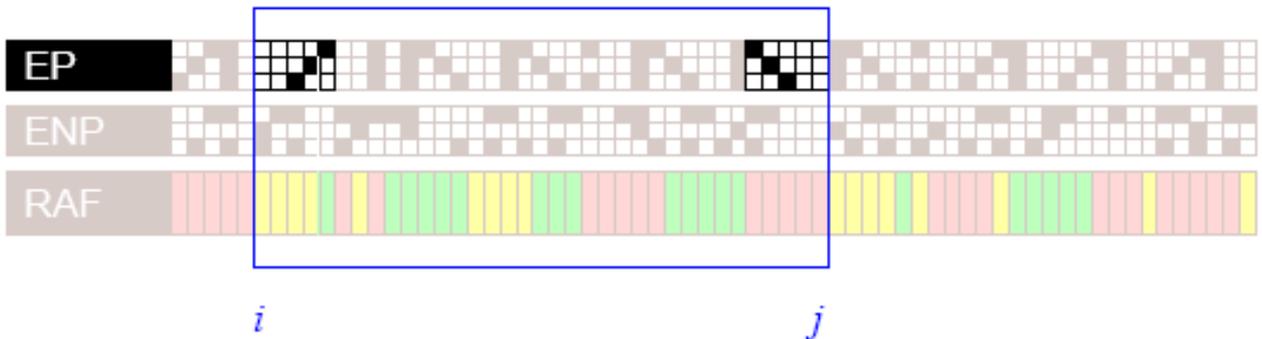


# Recuit simulé multiobjectif

INV\_TYPE:  $i$  et  $j$  du même type (EP, ENP, RAF)



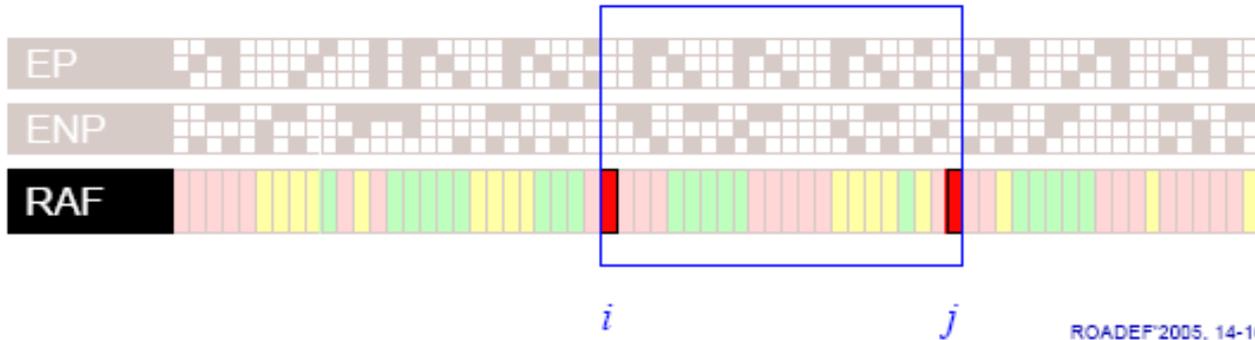
INV\_EP:  $i$  et  $j$  du même type EP



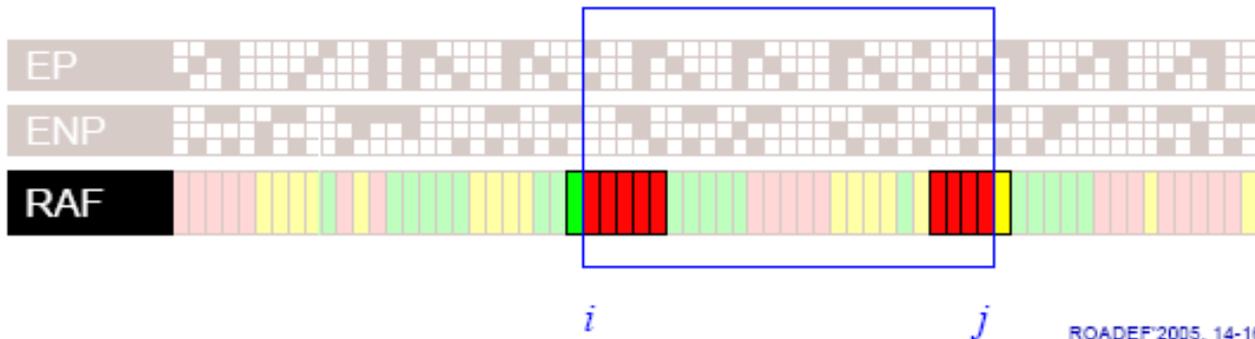


# Recuit simulé multiobjectif

INV\_RAF:  $i$  et  $j$  de la même couleur



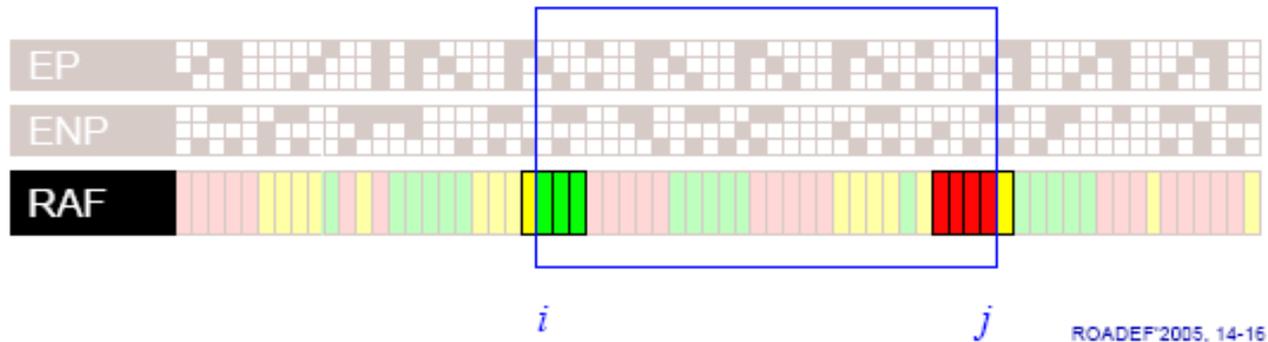
INV\_RAF\_BL1:  $i$  et  $j$  sont les extrémités ayant la même couleur, de blocs de la même couleur





# Recuit simulé multiobjectif

INV\_RAF\_BL2:  $i$  et  $j$  sont les extrémités ayant des couleurs différentes de blocs de la même couleur





# Recuit simulé multiobjectif

Comment choisir le meilleur opérateur de voisinage pour le recuit simulé ?



# Recuit simulé multiobjectif

Test de 200 ECHBIN après chaque mouvement

Au début de la phase 2, tous les mouvements (à part ECHBIN) ont la même priorité ( $\text{Pr}[i] = 1/7$ )

Chaque seconde, après 20 mouvements efficaces, mise à jour des probabilités:

$$P[i] = \beta P[i] + (1 - \beta) P'[i] \text{ avec } P'[i] = \frac{\text{nbSuccess}[i]/P[i]}{\sum_j \text{nbSuccess}[j]/P[j]}$$

et  $\beta$  est mis à jour de la façon suivante:  $\beta = \beta \cdot 0.992$

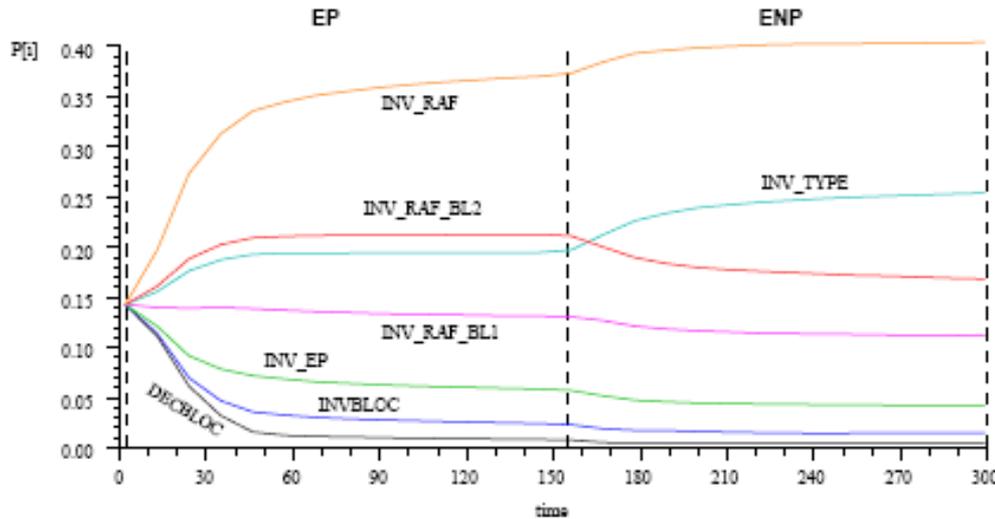
Au début de l'optimisation de l'objectif  $k$ ,  $\beta = 0.95$  et  $\text{nbSuccess}[i]=0$  pour chaque mouvement.

La probabilité dépend de:

- L'objectif déjà optimisé  $k' < k$ , et de l'objectif  $k$
- La complexité de l'instance elle-même

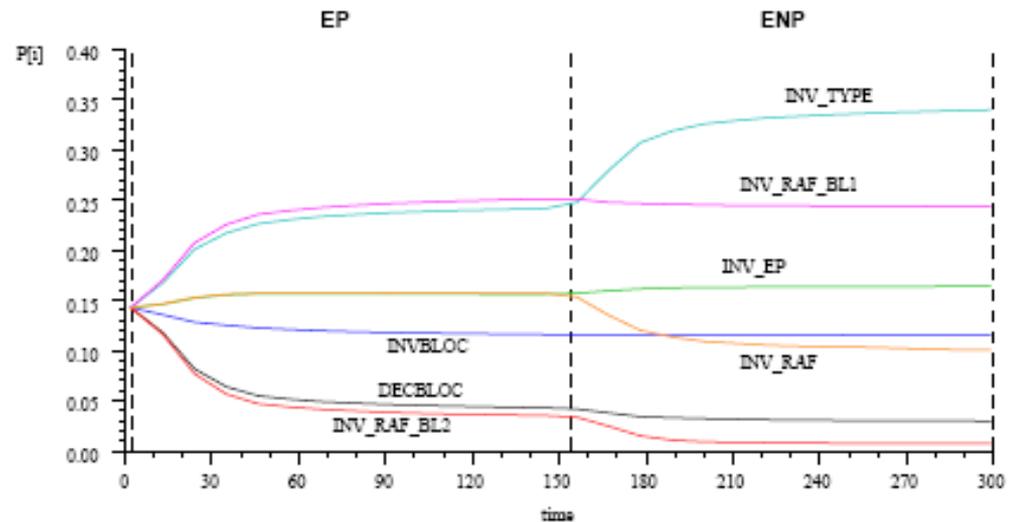


# Evolution du type de voisinage utilisé



Sur l'instance n°1

Sur l'instance n°2



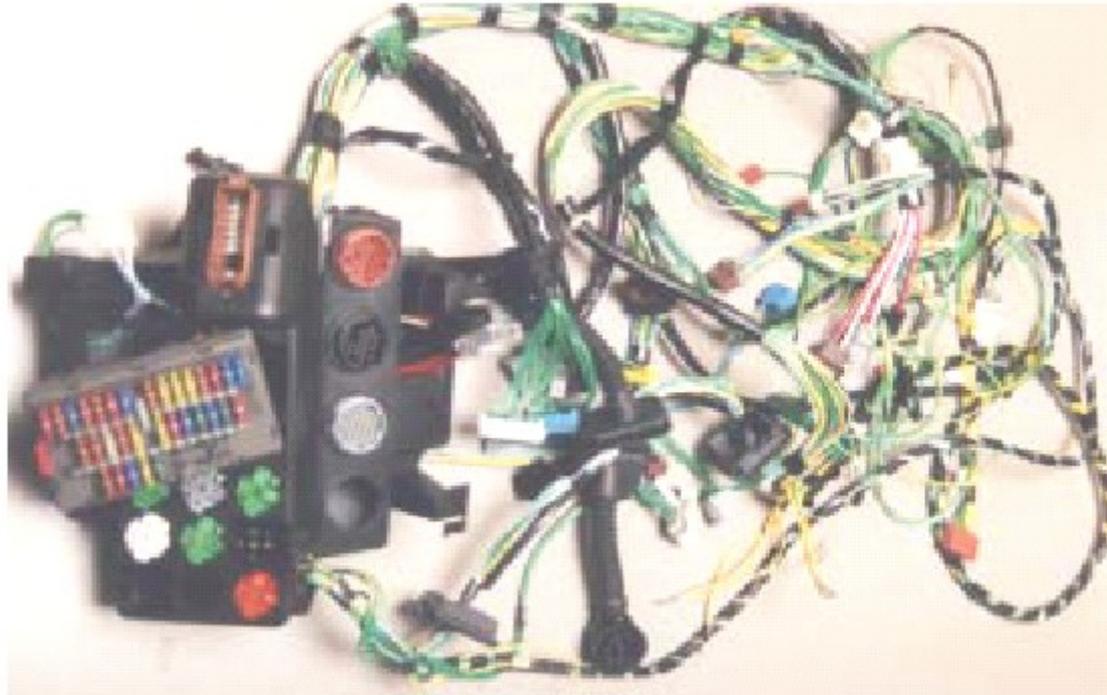


# Conclusions

- Déploiement de la méthode de recherche locale issue du challenge ROADEF dans les usines RENAULT le 15 Nov 2005
- Nouvelles contraintes à explorer :
  - Contrainte sur la position d'un véhicule (*p.ex. Les véhicules avec la nouvelle option i doivent être produits le matin*)
  - Succession de couleurs interdites (*p.ex. Un véhicule noir ne doit pas suivre ou précéder un véhicule blanc*)
  - Plusieurs lignes de peinture pour la même ligne de montage (*nouvelle méthode de calcul pour le changement de peinture*)

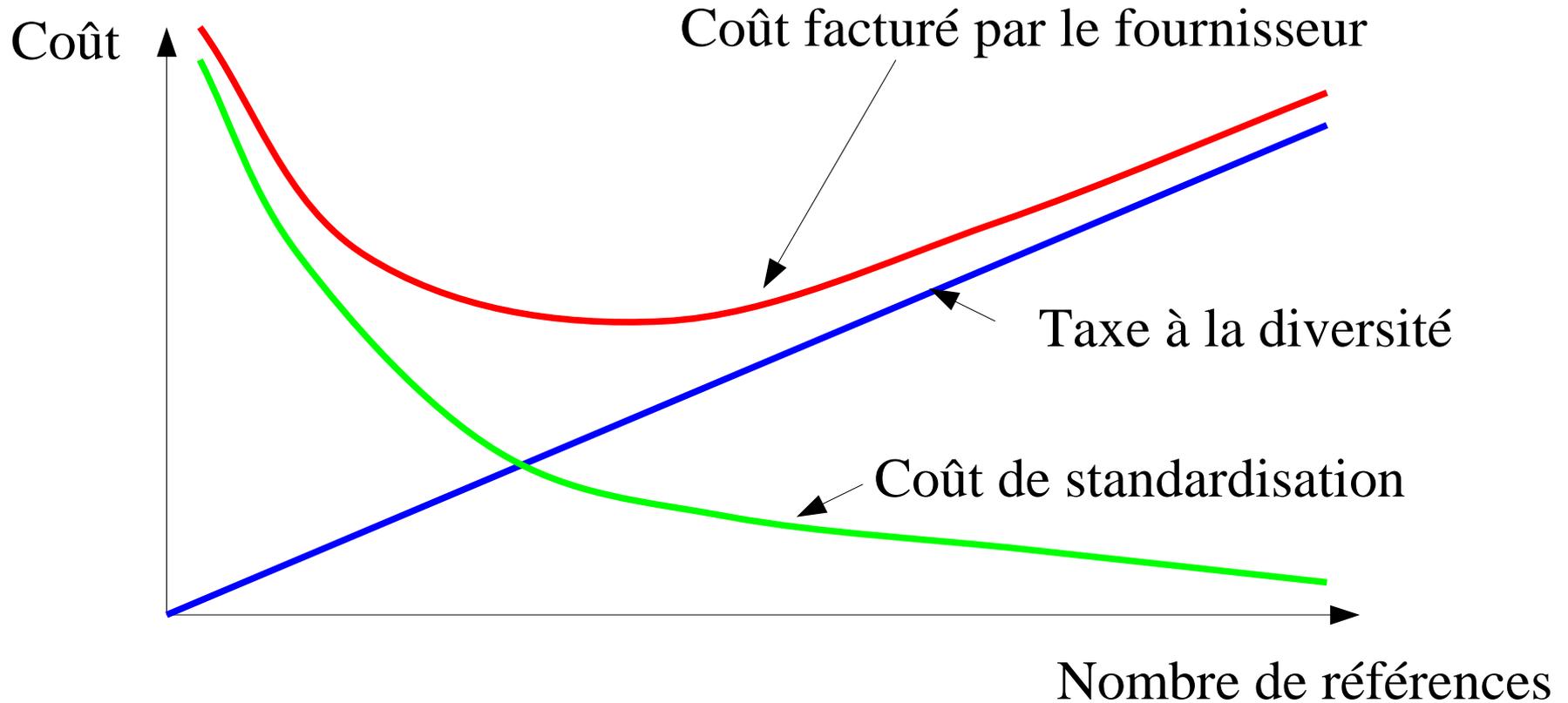


# TOSCA



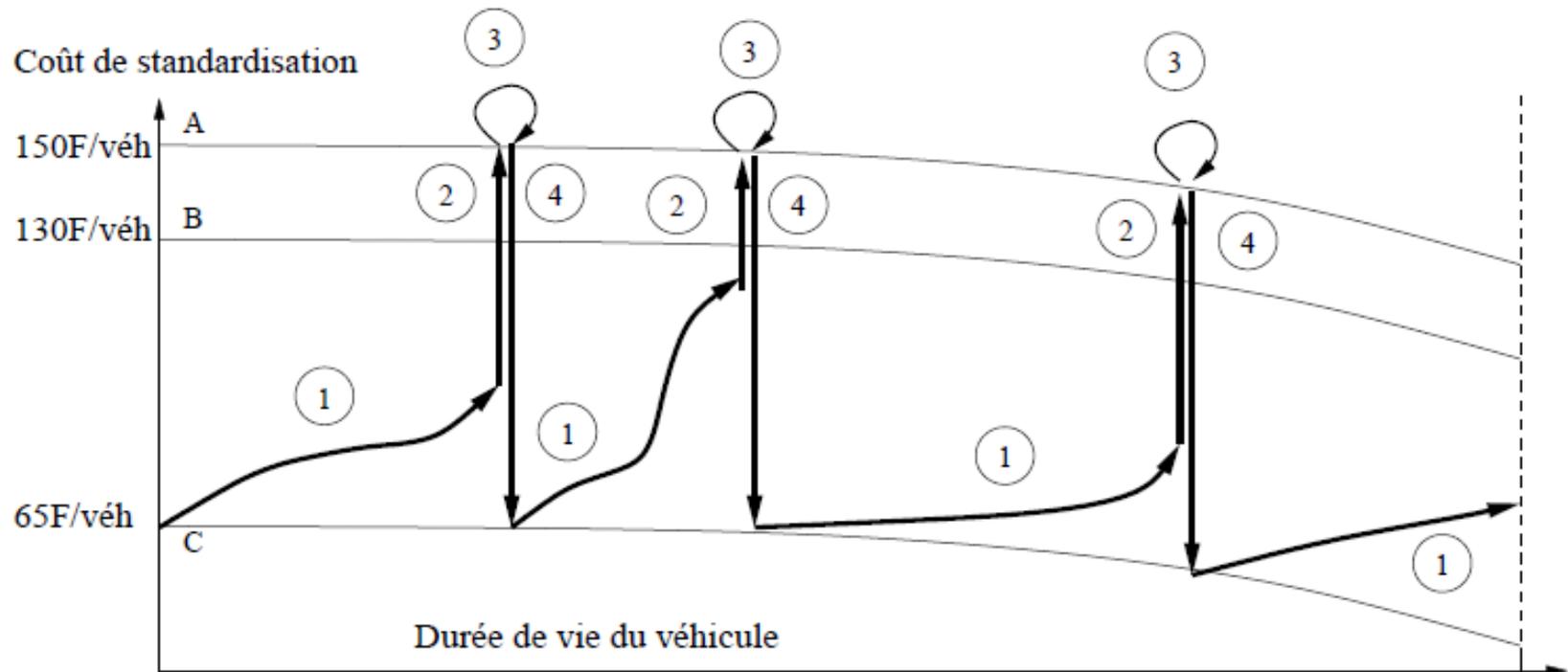


# TOSCA: Le coût de production des câblages





# TOSCA: Le problème





# Plan de la présentation

- 1 Problématique TOSCA
- 2 Méthode de résolution actuelle
- 3 Le voisinage pour le recuit simulé
- 4 Perspectives



# 1 Problématique TOSCA

- Nombre de modules :  $N_{\text{Mod}} = 4$
- Nombre de demandes :  $N_{\text{Dem}} = 5$
- Coûts d'immobilisation :  $I^m = (0.7 \quad 0.4 \quad 0.4 \quad 0.1)$
- Coûts de production :  $P^m = (1.4 \quad 1.8 \quad 1.6 \quad 1.4)$
- Quantités de demandes + Demandes :

$$Q_d = \begin{pmatrix} 7 \\ 6 \\ 3 \\ 1 \\ 24 \end{pmatrix} \quad D_d^m = \begin{pmatrix} 0 & 0 & 1 & 1 \\ 1 & 0 & 0 & 1 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \end{pmatrix}$$

- A nombre fixé de références, quelles sont les **références optimales** minimisant le coût de standardisation ?



# Plan de la présentation

- 1 Problématique TOSCA
- 2 Méthode de résolution actuelle
- 3 Le voisinage pour le recuit simulé
- 4 Perspectives

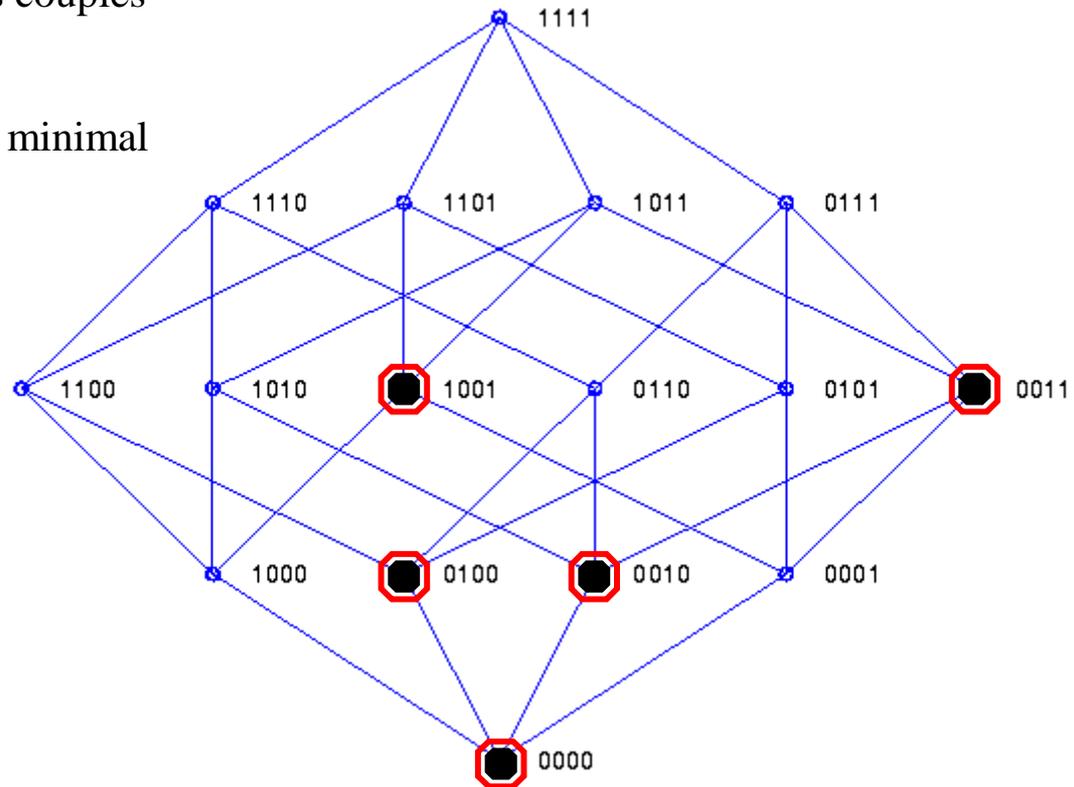


## 2 Méthode de résolution actuelle algorithme glouton

- Recherche incrémentale de la solution :
  - listing des fusions de tous les couples de références
  - choix de la fusion de surcoût minimal

$$R = D = \begin{pmatrix} 0 & 0 & 1 & 1 \\ 1 & 0 & 0 & 1 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \end{pmatrix}$$

Algorithme glouton: algo qui construit séquentiellement une solution sans revenir sur sa décision

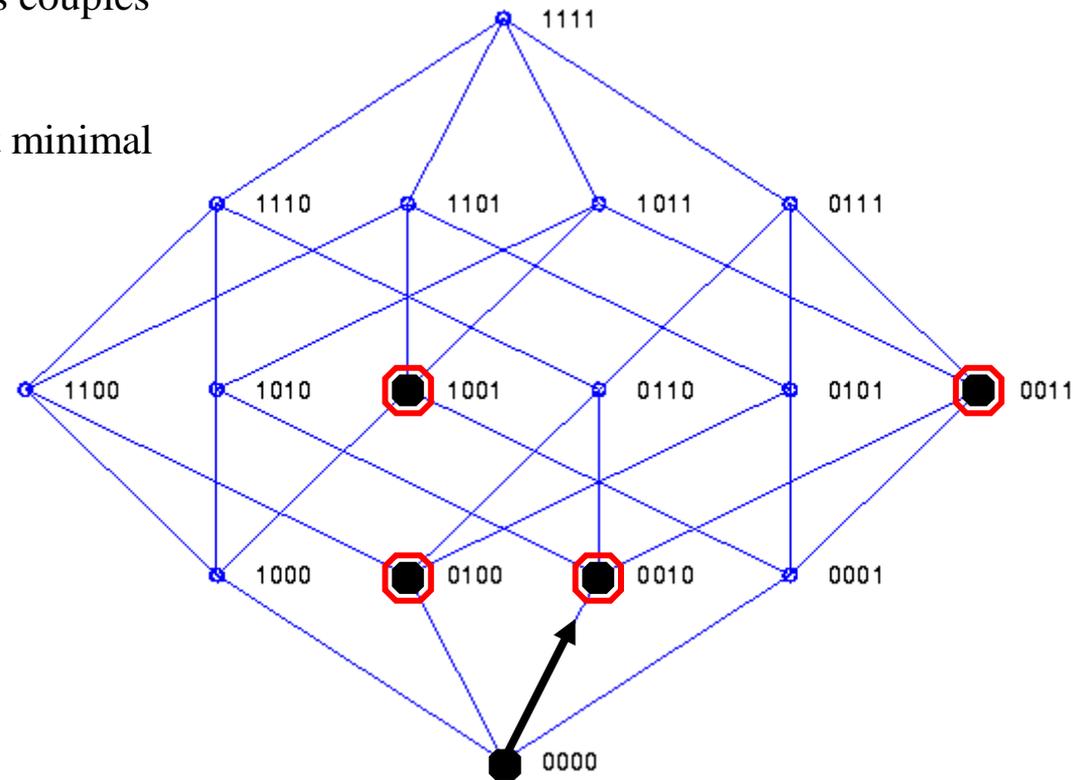




## 2 Méthode de résolution actuelle algorithme glouton

- Recherche incrémentale de la solution :
  - listing des fusions de tous les couples de références
  - choix de la fusion de surcoût minimal

$$R = \begin{pmatrix} 0 & 0 & 1 & 1 \\ 1 & 0 & 0 & 1 \\ 0 & 0 & 1 & 0 \\ * & * & * & * \\ 0 & 1 & 0 & 0 \end{pmatrix}$$

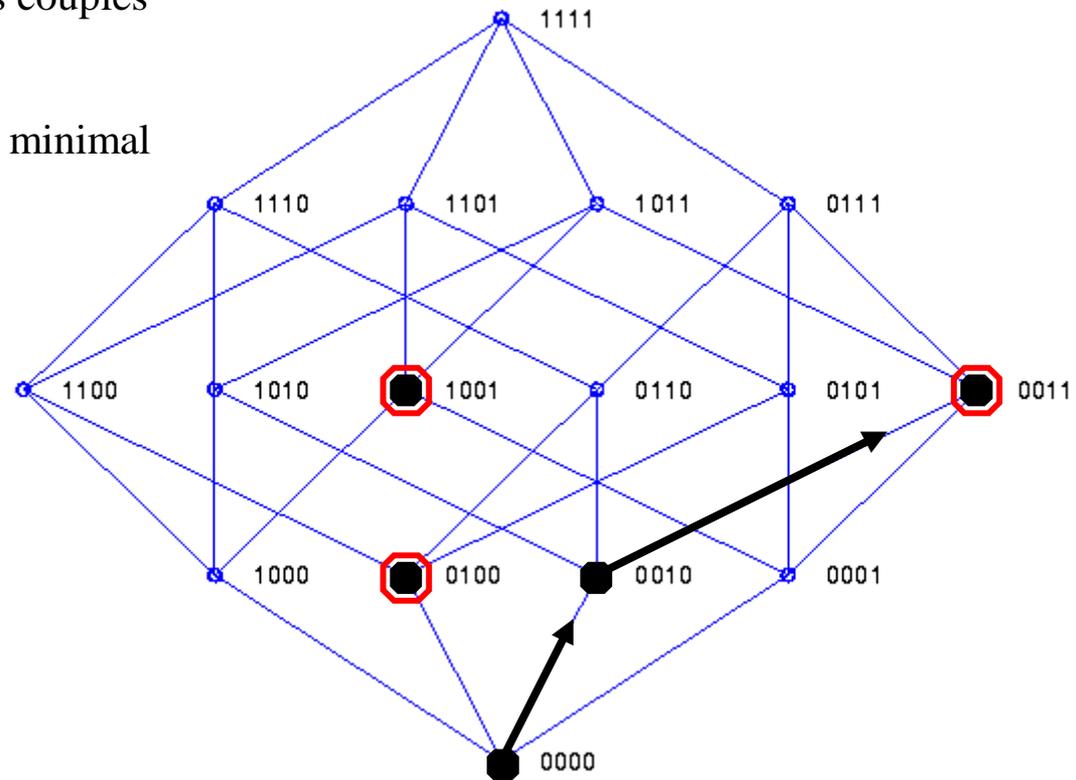




## 2 Méthode de résolution actuelle algorithme glouton

- Recherche incrémentale de la solution :
  - listing des fusions de tous les couples de références
  - choix de la fusion de surcoût minimal

$$R = \begin{pmatrix} 0 & 0 & 1 & 1 \\ 1 & 0 & 0 & 1 \\ * & * & * & * \\ * & * & * & * \\ 0 & 1 & 0 & 0 \end{pmatrix}$$

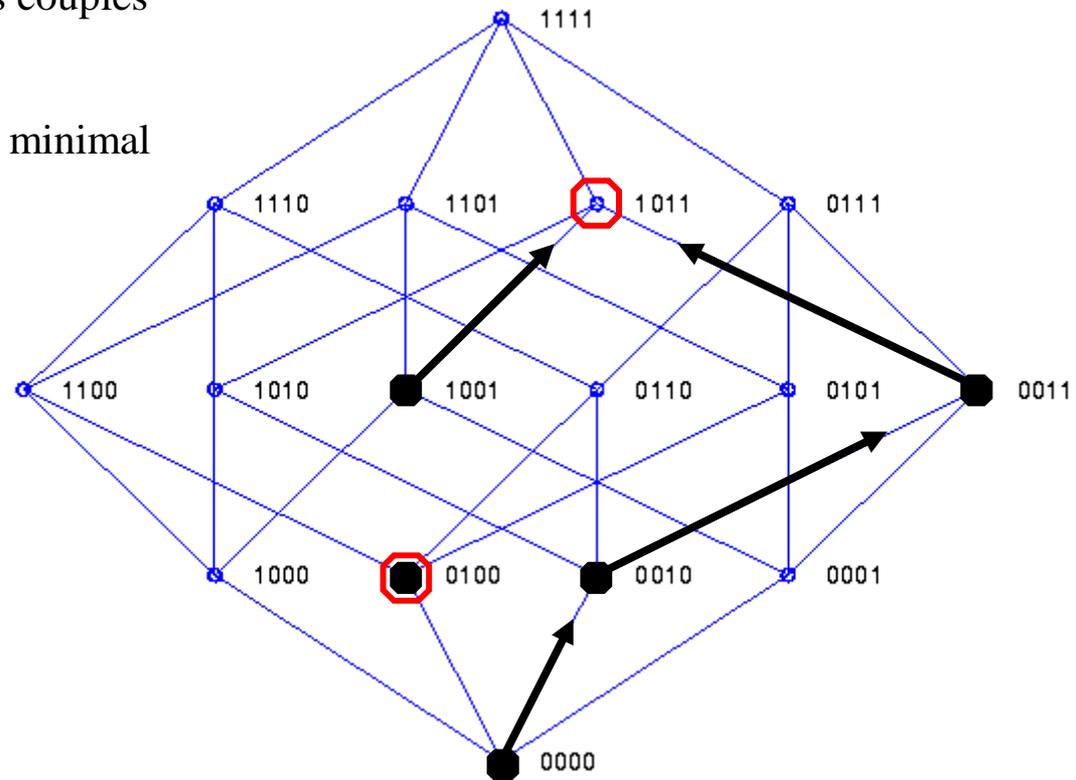




## 2 Méthode de résolution actuelle algorithme glouton

- Recherche incrémentale de la solution :
  - listing des fusions de tous les couples de références
  - choix de la fusion de surcoût minimal

$$R = \begin{pmatrix} 1 & 0 & 1 & 1 \\ * & * & * & * \\ * & * & * & * \\ * & * & * & * \\ 0 & 1 & 0 & 0 \end{pmatrix}$$

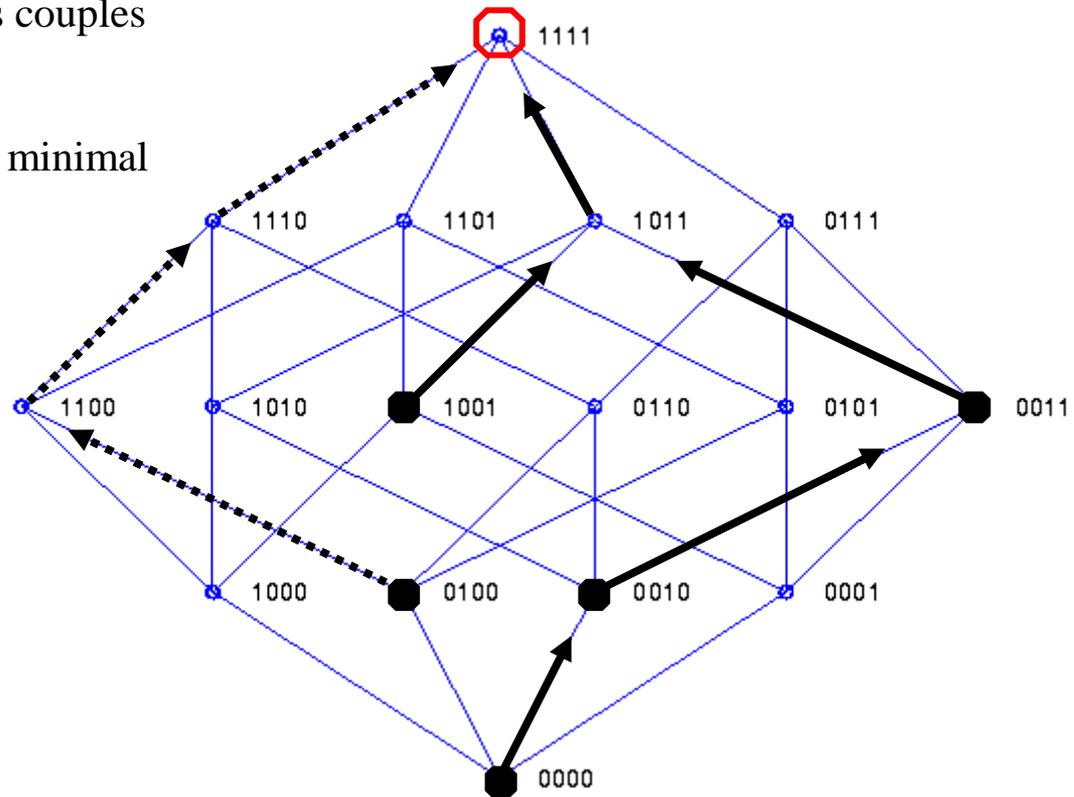




## 2 Méthode de résolution actuelle algorithme glouton

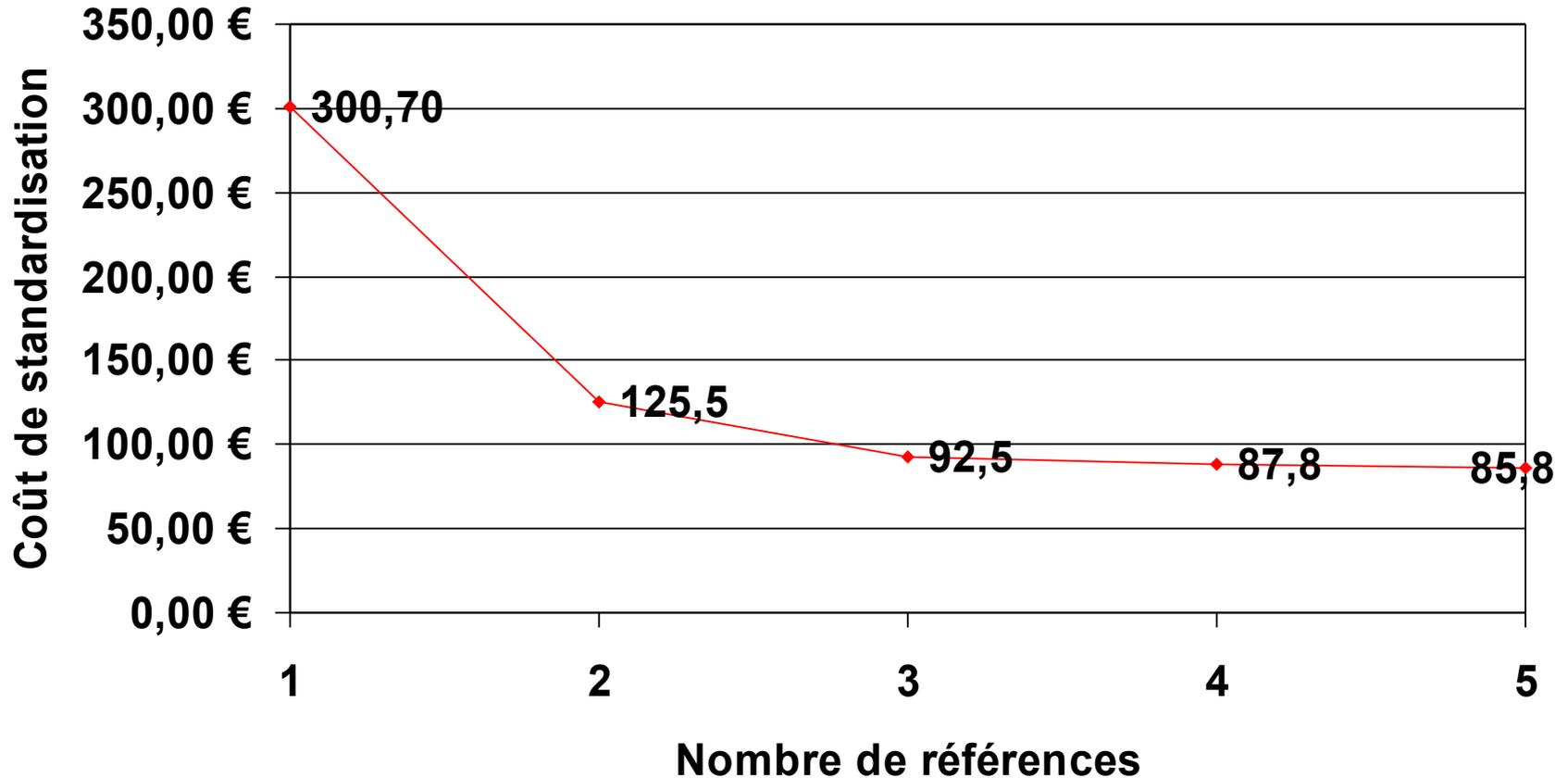
- Recherche incrémentale de la solution :
  - listing des fusions de tous les couples de références
  - choix de la fusion de surcoût minimal

$$R = \begin{pmatrix} 1 & 1 & 1 & 1 \\ * & * & * & * \\ * & * & * & * \\ * & * & * & * \\ * & * & * & * \end{pmatrix}$$





## 2 Méthode de résolution actuelle algorithme glouton





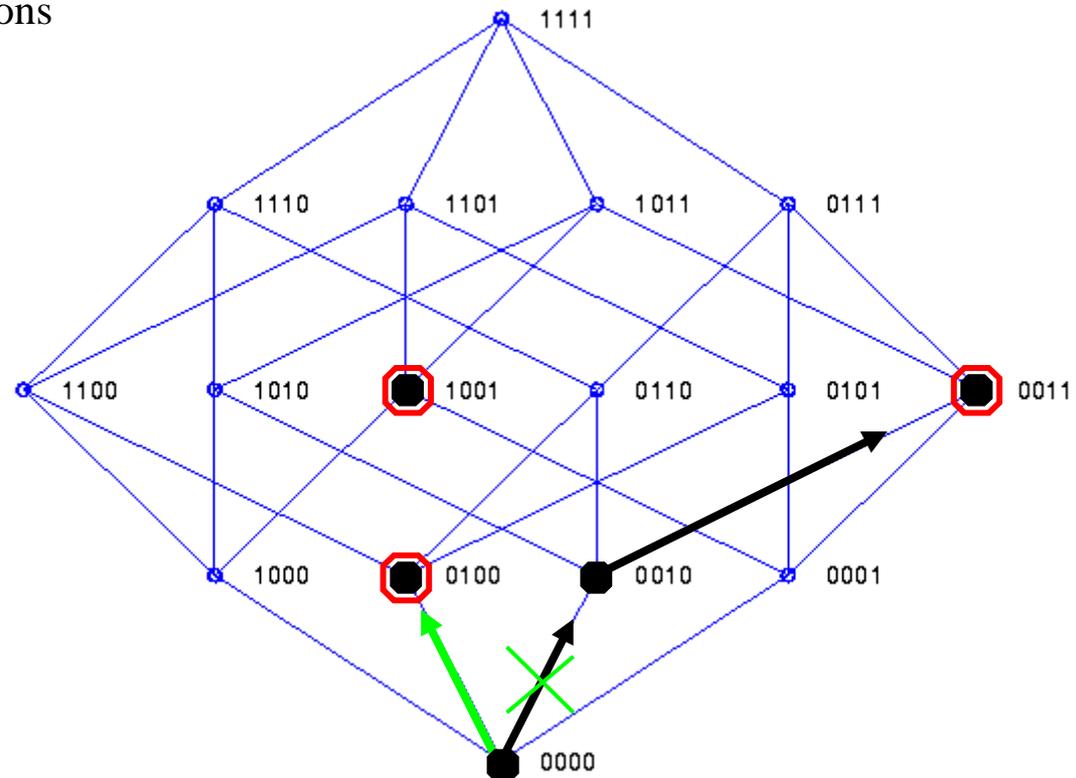
## 2 Méthode de résolution actuelle les limites

- Pas de réaffectation régulière des demandes aux références

=> approximative du surcoût à chaque itération

=> choix non optimal des fusions

=> dégradation de la solution

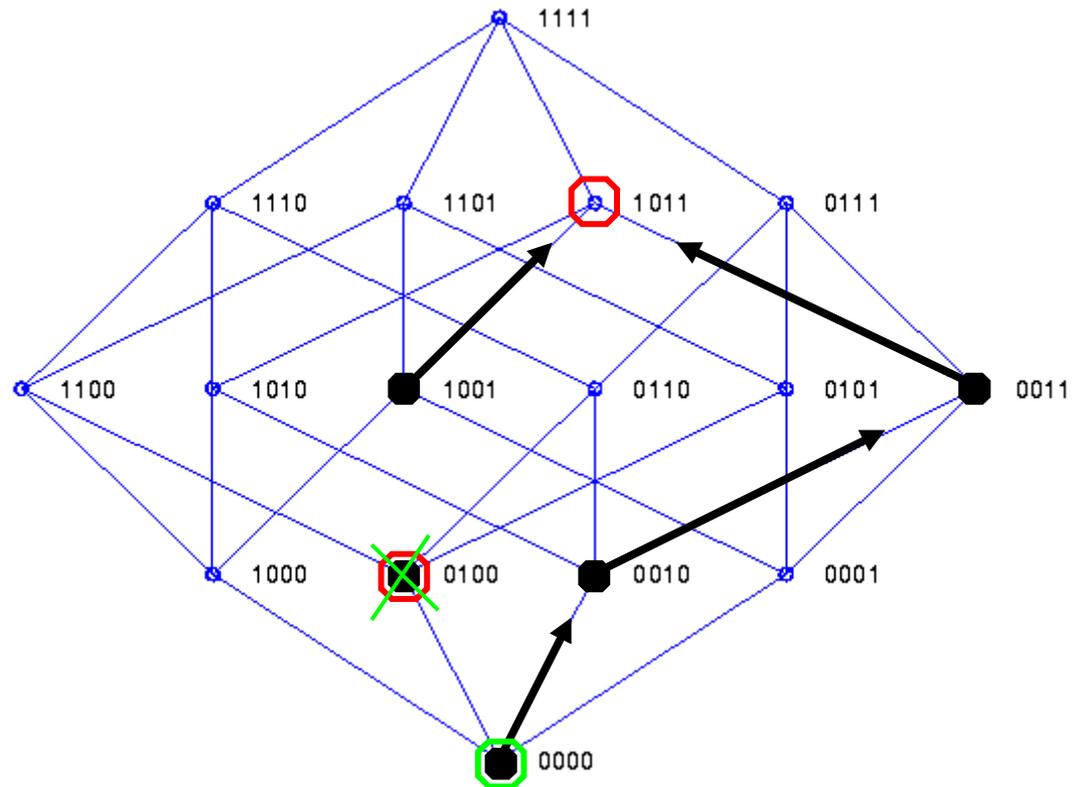




## 2 Méthode de résolution actuelle les limites

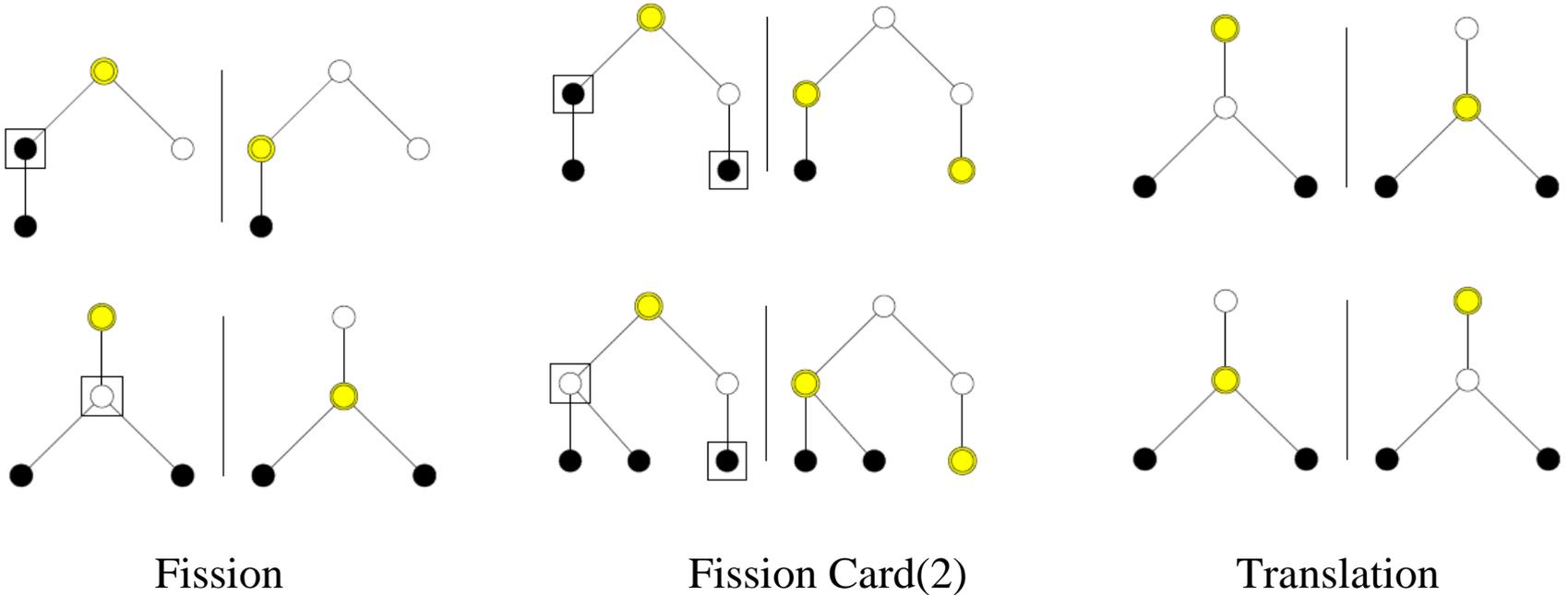
- **Construction itérative** de la solution

=> irréversibilité d'une fusion (une référence supprimée à une itération ne pourra plus jamais être sélectionnée ensuite).





# TOSCA : Le voisinage



Fusion



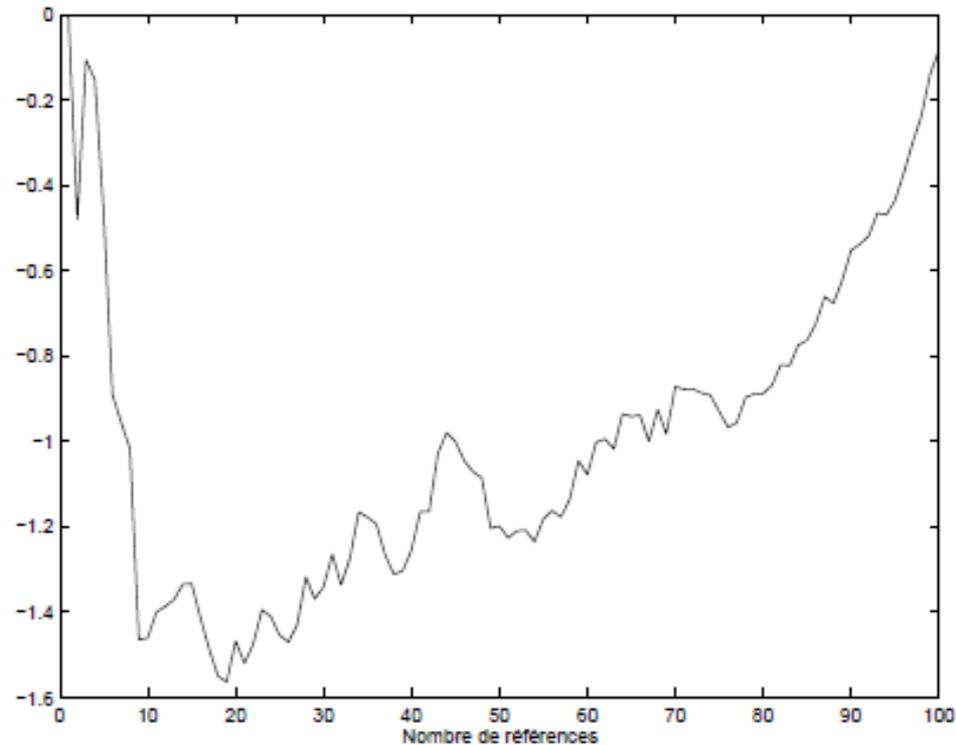
# TOSCA: Le recuit simulé

- Recuit simulé basé sur les précédents mouvements
- La température initiale vaut 0 !
- Le recuit est appliqué à la fin de la passe de l'algorithme glouton



# TOSCA : Résultat

Gain relatif par rapport à la méthode actuelle





# Génération de Cartographies pour calculateur

La génération de cartographies perturbées



# Le système de contrôle : exemple diesel

Calculateur ABS / ESP

Boutonerie régulateur vitesse

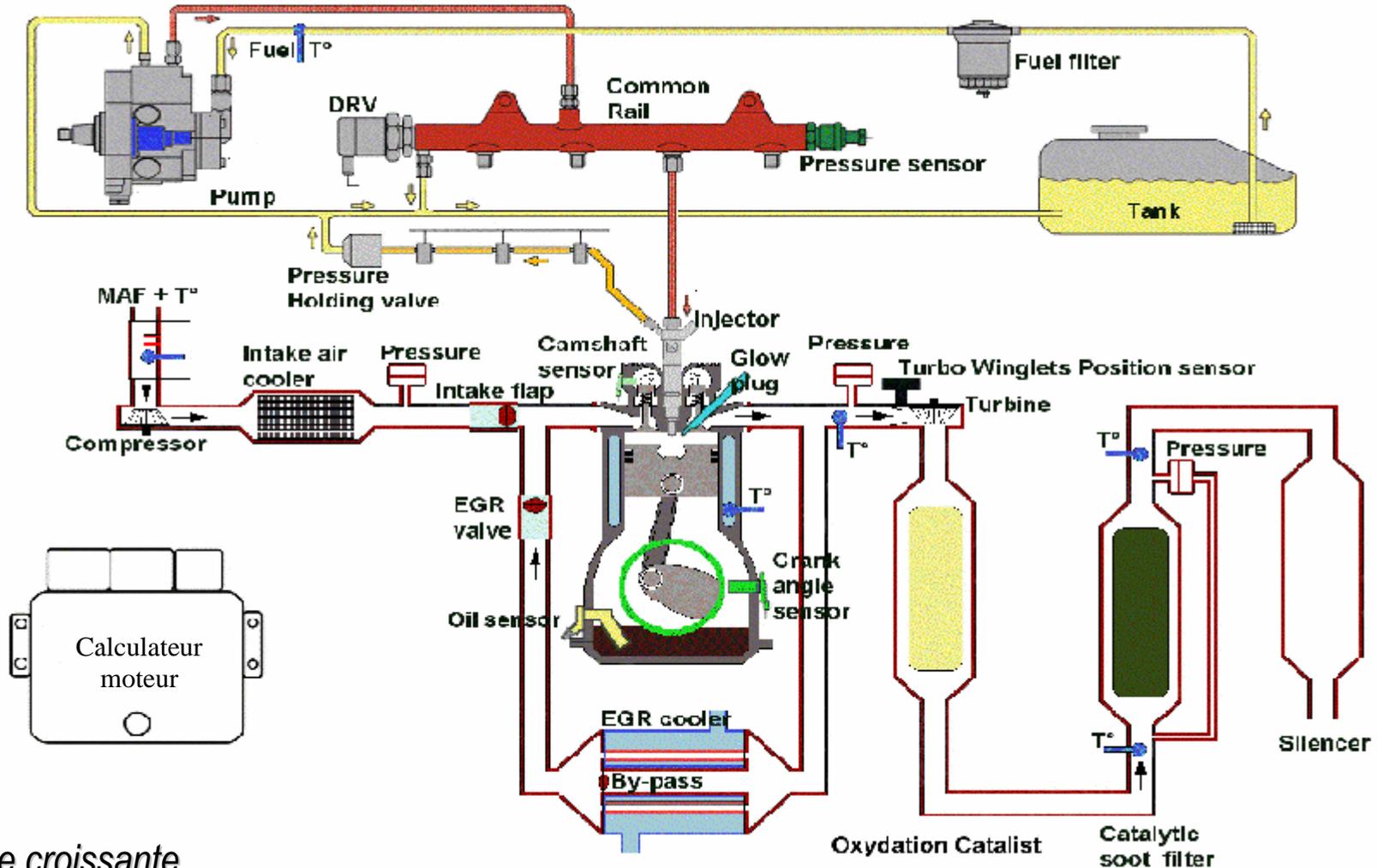
Calculateur CLIM

Calculateur BVA

VERLOG

ACC

Tableau de bord ADAC



*Une importance croissante de l'intersystèmes*



# Complexité croissante et impact calibrations

Nb of calibration parameters

+

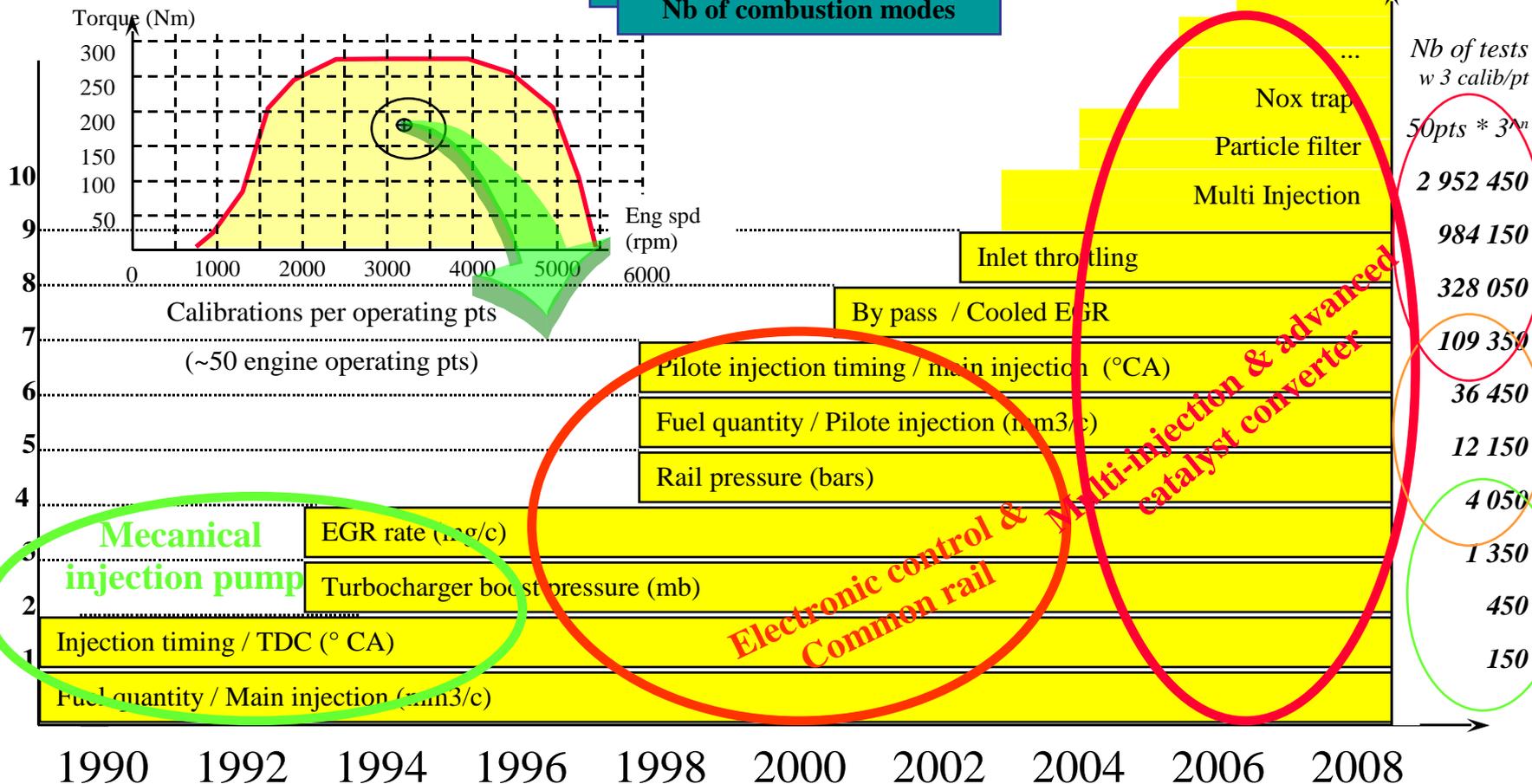
Nb of combustion modes

Nb of combustion modes

Nb of combustion modes

=>

Nb of engine test points

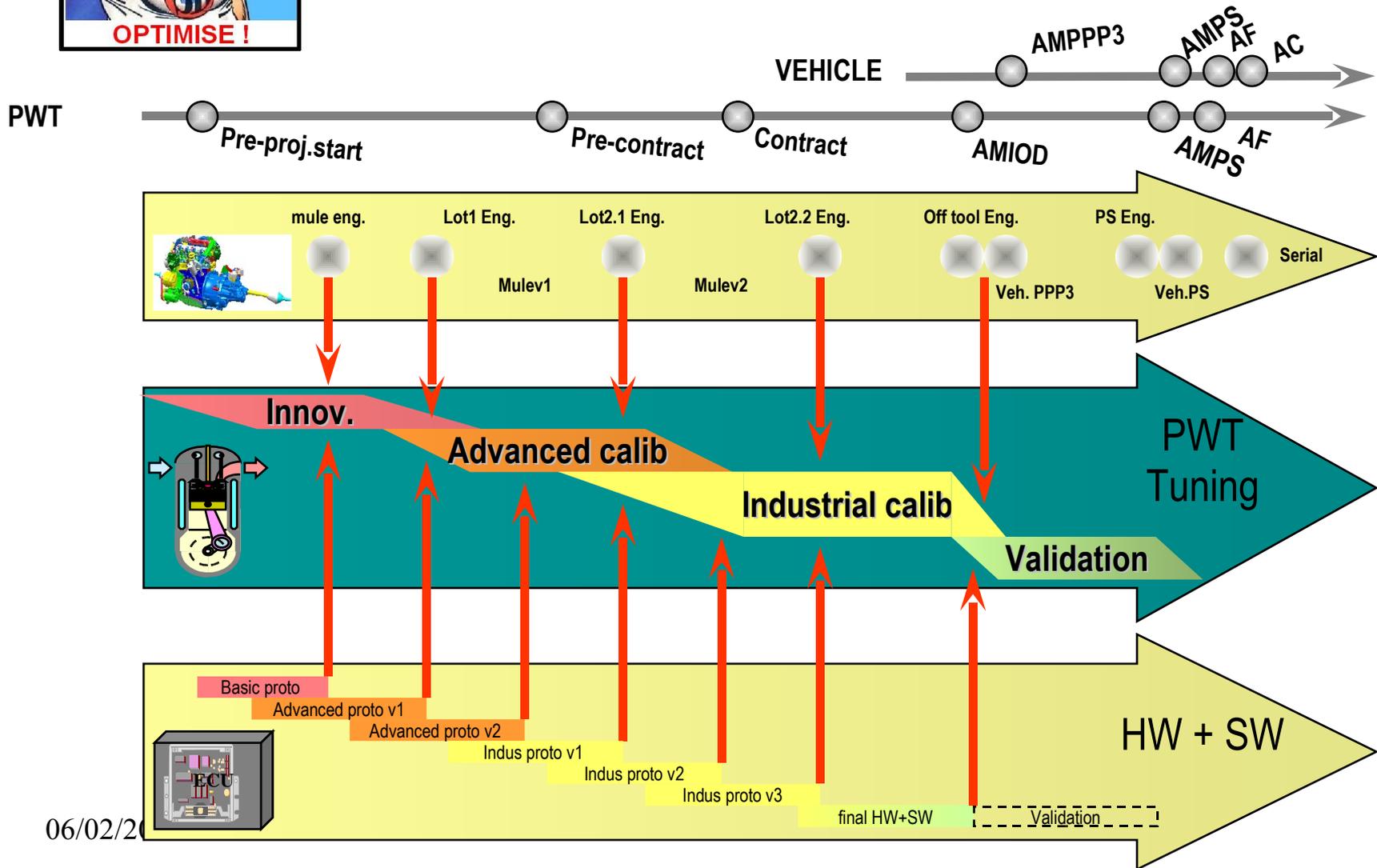


Classical method

Computing (DoE, models)



# Tuning process





# General Process Diesel emission optimisation

Steady state calibrations  
DoE

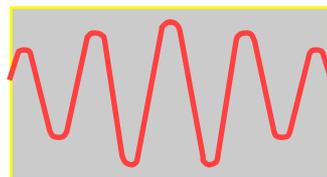
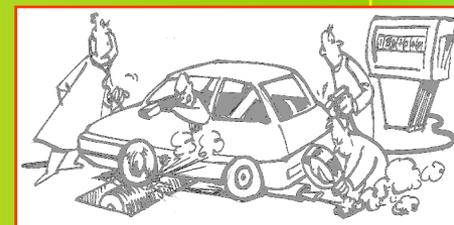
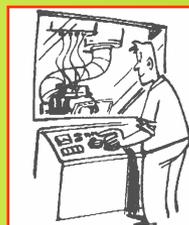
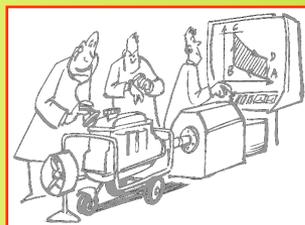
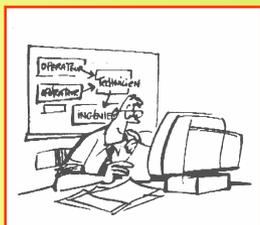
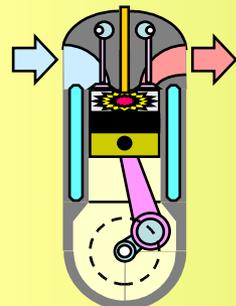
Validation & certification

Dynamic calibrations

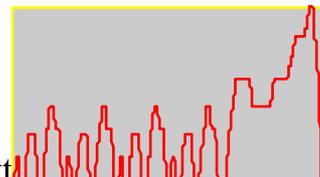
Cold transient (20°C => 90°C)

Warm engine (90°C)

Customer conditions



Colletti



time  
65



# Le Banc Moteur



06/02/2006

Y. Collette



# La génération de cartographies perturbées

**But :** Pour un réglage donné, générer un ensemble de  $N$  cartographies couvrant un domaine compris entre une cartographie de borne min et une cartographie de borne max

**Solution :** Utilisation d'un algorithme génétique multiobjectif

**Fonction objectif 1 :** maximiser le minimum d'entropie sur l'ensemble des PF

**Fonction objectif 2 :** maximiser le minimum de variance sur l'ensemble des cartographies

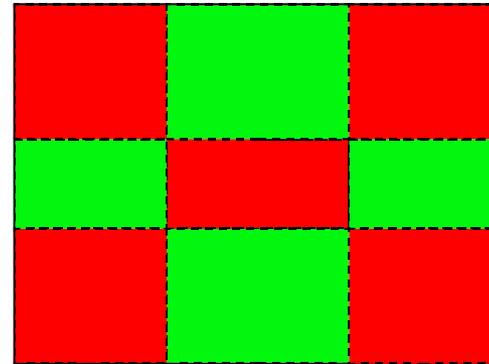
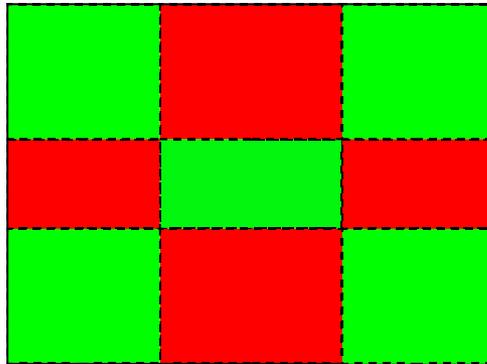
**Contrainte 1 :** sur l'ensemble des cartographies et pour chaque PF, on doit retrouver la borne min et la borne max

**Contrainte 2 :** contrainte de lissage respectée



# Les algorithmes génétiques

## Le croisement de deux cartographies



$$\begin{aligned}\text{Green} &= (1-A).\text{Green} + A.\text{Red} \\ \text{Red} &= A.\text{Green} + (1-A).\text{Red}\end{aligned}$$

$$-1 < A < 2$$

Corrections:

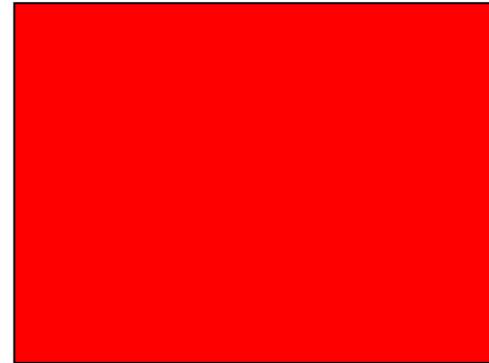
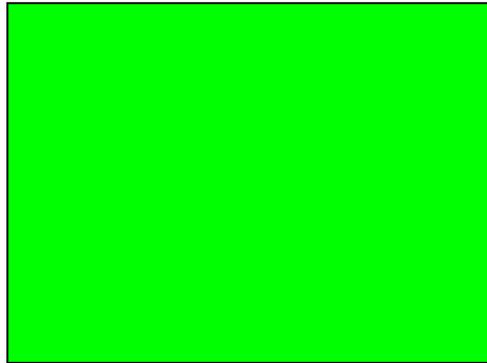
Vérification contrainte de lissage

Vérification contrainte de bornes



# Les algorithmes génétiques

## Le croisement de deux cartographies



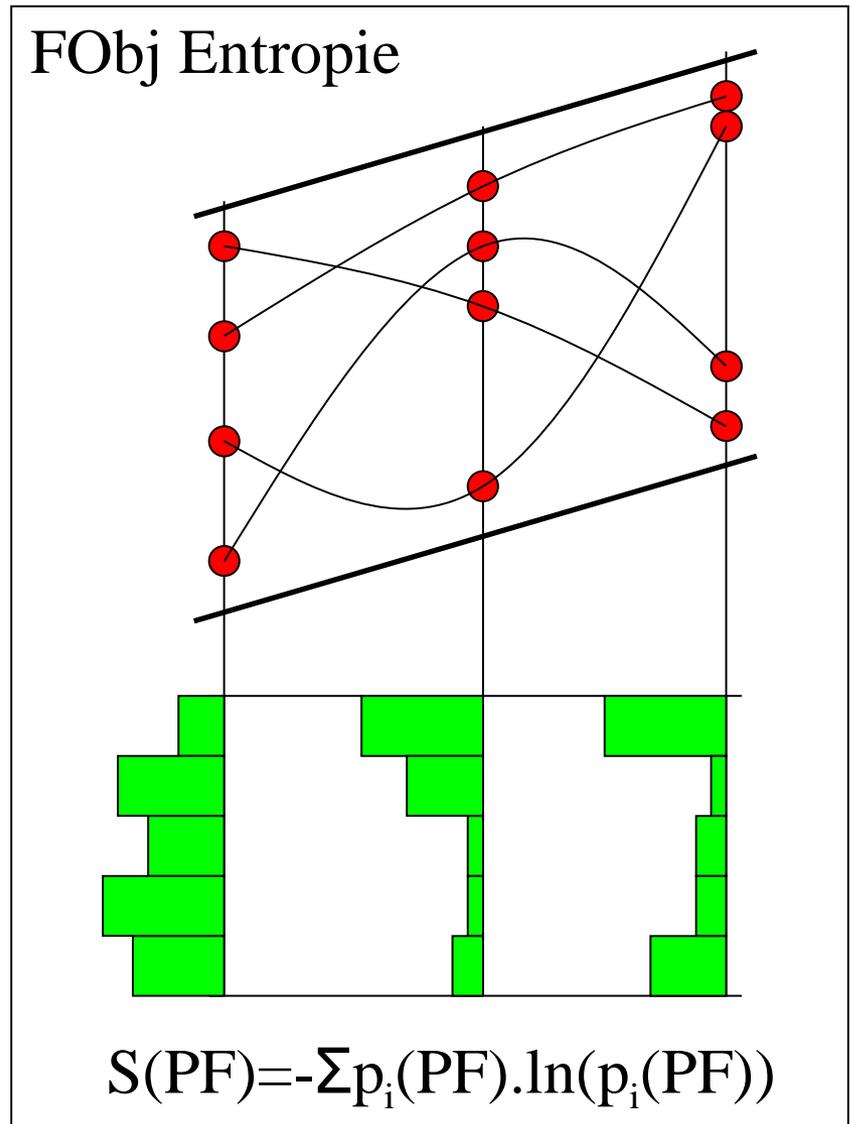
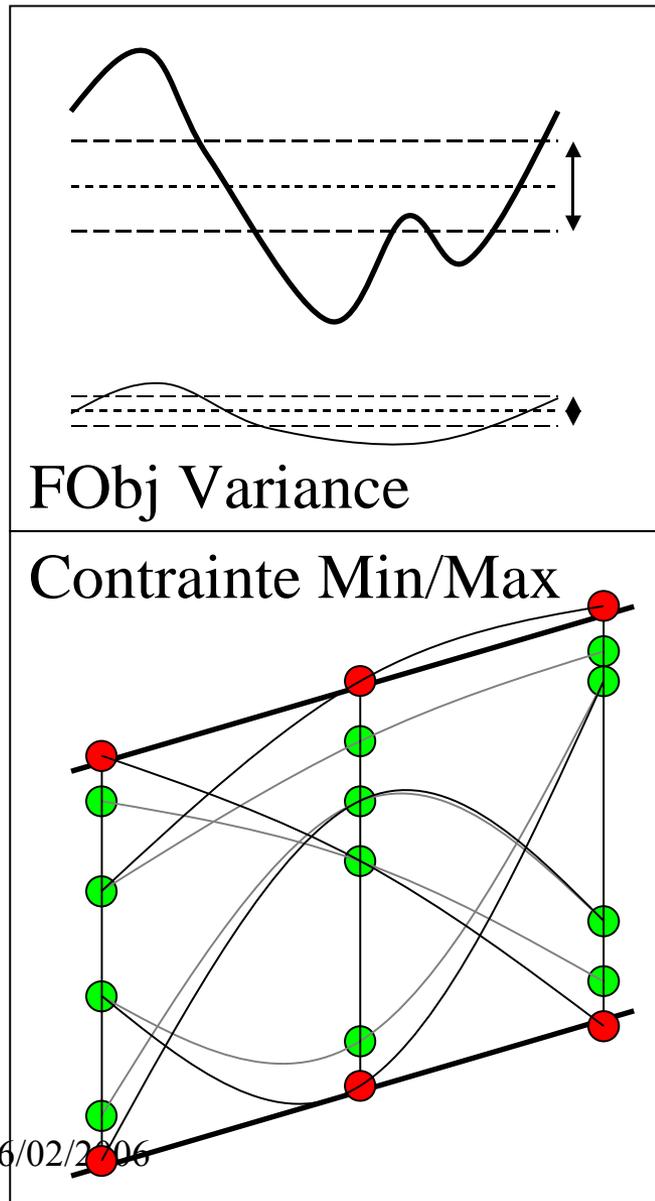
$$\begin{aligned}\text{Green} &= (1-A).\text{Green} + A.\text{Red} \\ \text{Red} &= A.\text{Green} + (1-A).\text{Red}\end{aligned}$$

$$0 < A < 1$$

Correction:  
Vérification contrainte de bornes

Avantage:  
Si la population initiale est lisse  
Les enfants seront lisses

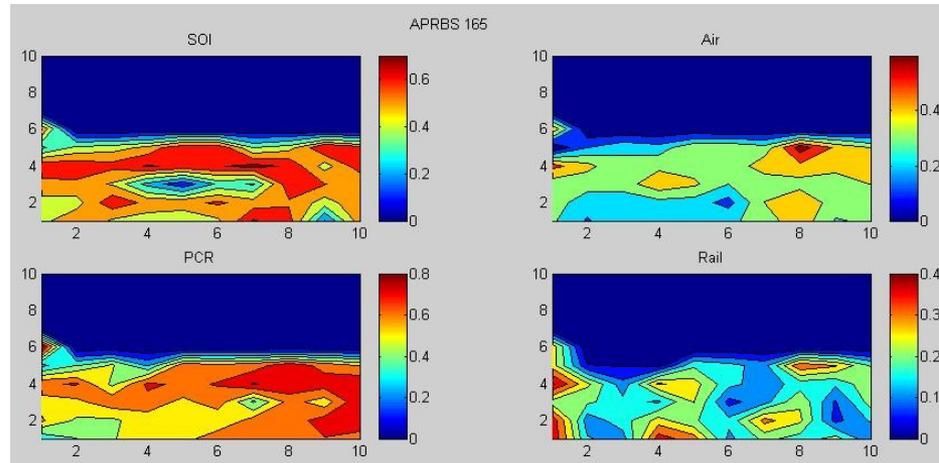
# La génération de cartographies perturbées





# La génération de cartographies perturbées

APRBS



MAPDYN

