

# OPTIMISATION DISCRÈTE

## Compléments



Yannick Kergosien  
Polytech Tours  
LIFAT EA 6300– Equipe ROOT



# Plan

- ❑ Les données
- ❑ Tester et analyser
- ❑ Intégration de l'incertitude et au côté dynamique

# Les données

## □ Sources des données

### ■ Benchmark disponibles

- Avantage : comparaison des résultats avec la littérature
- Inconvénient : rare ou ne « colle » pas parfaitement au problème

### ■ Instances « réelles »

- Avantage : comparaison avec la méthode actuelle
- Inconvénient : difficile à obtenir

### ■ Génération d'instances aléatoires ou « semi-aléatoire »

- Avantage : avoir ses propres jeux de données
- Inconvénient : difficulté à générer des instances pertinentes

## □ Automatisation

## □ Eviter la génération à la volée

# Les données

## ❑ Notions importantes

- Quantité suffisante
  - ❑ Pour le paramétrage
  - ❑ Pour les tests, les comparaisons et la validation
- Taille des instances, généralement 3 catégories :
  - ❑ Petites : pour comparaison à l'exacte
  - ❑ Moyennes : pour valider/comparer la ou les méthodes
  - ❑ Grandes : montrer les limites de la ou les méthodes
- Diversification et représentativité
- Des instances réalisables
- Difficulté des instances, quelques techniques :
  - ❑ Liens/ratios entre certains paramètres d'instances (ex : charge de travail / ressource disponible)
  - ❑ Construire une solution puis en déduire l'instance
  - ❑ Vérification après résolution des solutions trouvées (ex : pourcentage d'inactifé des ressources)

# Tester et analyser

## □ Tests unitaires et fonctionnels

- Analyse à la main des solutions trouvées
- Développer une fonction indépendante de vérification de solutions trouvées
- Comparaison des résultats aux bornes inférieures, aux méthodes exactes, à la littérature, etc.

## □ Paramétrage d'une méthode

- Nécessite du temps => automatisation des tests
- Complexe à cause des liens des paramètres (commencer les paramètres les plus « violent »)
- Lier les paramètres à la taille et nature des instances
- Attention au sur-paramétrage

# Tester et analyser

## □ Principaux critères d'évaluations :

- Résultats\* (qualité des solutions trouvées, temps de calculs, etc.)
- Convergence\*
- Stabilité\*
- Paramétrage
- Complexité à implémenter
- Flexibilité/générique
- Parallélisable ou non

\* : nécessite une campagne de tests sur de nombreuses données

# Tester et analyser

## □ Outils de présentation des résultats :

### ■ Le traditionnel tableau

- Par paquet d'instances de même nature/taille
- Des moyennes et des écarts types (avec intervalle de confiance)

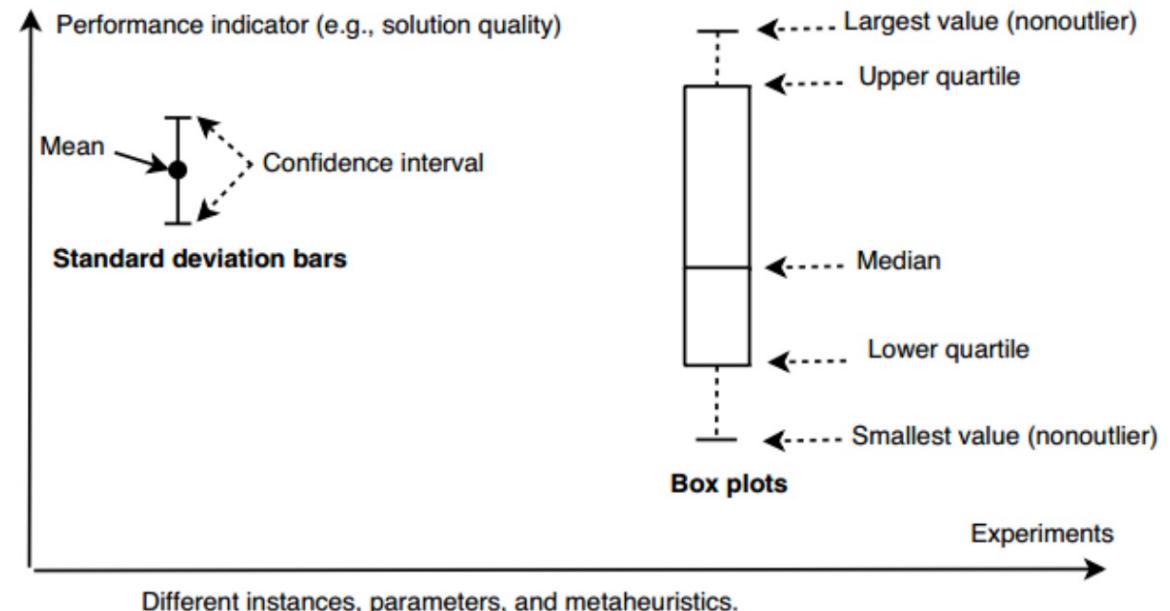
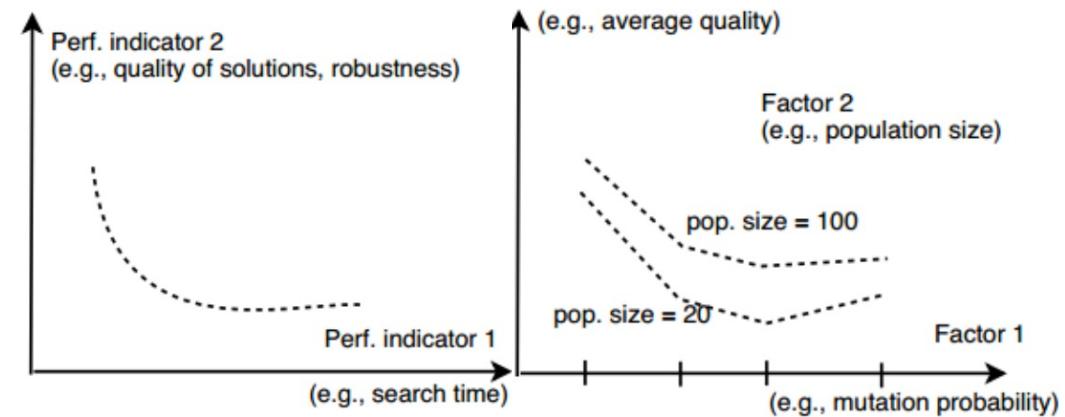
### ■ Les courbes classiques

### ■ Les box plots

## □ Sans oublier le « fameux » GAP :

$$\frac{\|f(s) - f(s^*)\|}{f(s^*)}$$

$$\frac{\|f(s) - f(s^*)\|}{\|f_{worst} - f(s^*)\|}$$



# Intégration de l'incertitude et au côté dynamique

## □ Constats :

- Certains problèmes ont des données incertaines
- Et la nature des problèmes est dynamiques

## □ Types d'approches robustes:

- les approches proactives visent à créer des solutions robustes qui anticipent les perturbations
- les approches réactives permettent de réviser intelligemment les solutions en prenant en compte les perturbations qui arrivent

## □ Quelques solutions d'adaptation des méthodes de résolution

- Planification à horizon flottant
- Approche par scénario
- « Chance constraint method »