Optimisation combinatoire multicritère et approximation avec garantie de performance

Laurent Gourvès

lgourves@lami.univ-evry.fr

LaMI, université d'Évry Val d'Essonne

en collaboration avec Eric Angel et Evripidis Bampis

Plan de l'exposé

1. Optimisation combinatoire multicritère
2. Approximation polynomiale et garantie de performance
3. Deux problèmes bicritères
   - 3.1 Coupe maximale pondérée
   - 3.2 Ordonnancement sur une machine
4. Conclusion

Plan de l'exposé

1. Optimisation combinatoire multicritère
2. Approximation polynomiale et garantie de performance
3. Deux problèmes bicritères
   - 3.1 Coupe maximale pondérée
   - 3.2 Ordonnancement sur une machine
4. Conclusion

Optimisation combinatoire

ensemble fini de solutions réalisables

$\Rightarrow$ trouver efficacement la meilleure

Optimisation combinatoire

ensemble fini de solutions réalisables

$\Rightarrow$ trouver efficacement la meilleure

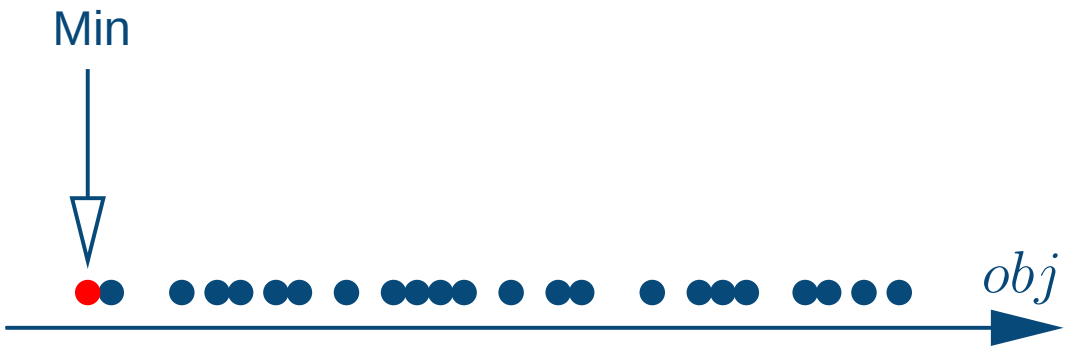
1 objectif obj

Optimisation combinatoire

ensemble fini de solutions réalisables

$\Rightarrow$ trouver efficacement la meilleure

1 objectif obj



Optimisation combinatoire multicritère

ensemble fini de solutions réalisables

$\Rightarrow$ trouver efficacement la meilleure

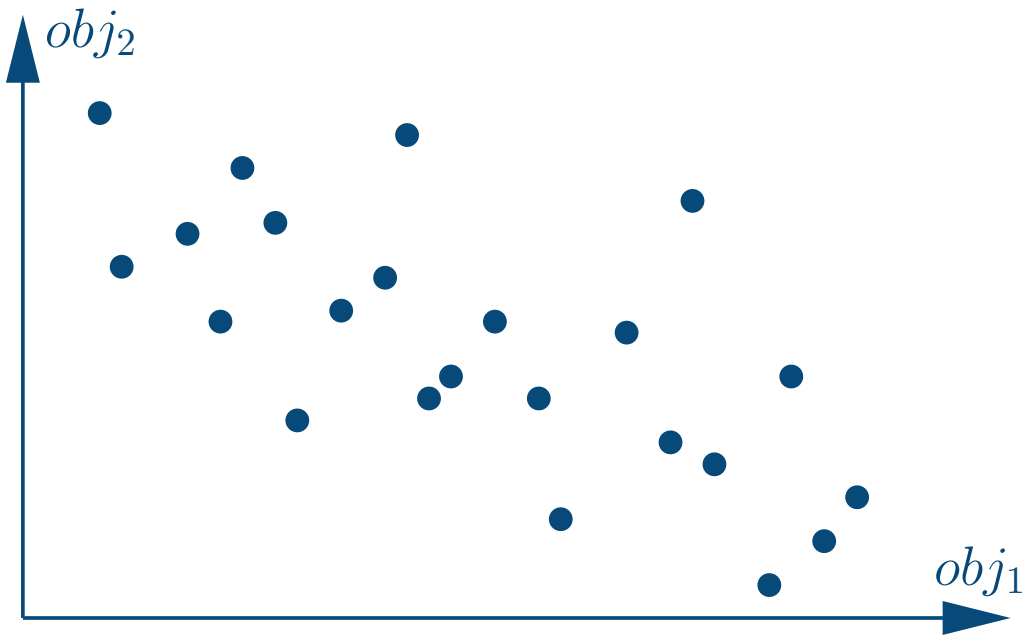
k objectifs $obj_1 \ldots obj_k$

Optimisation combinatoire multicritère

ensemble fini de solutions réalisables

$\Rightarrow$ trouver efficacement la meilleure

k objectifs $obj_1 \ldots obj_k$



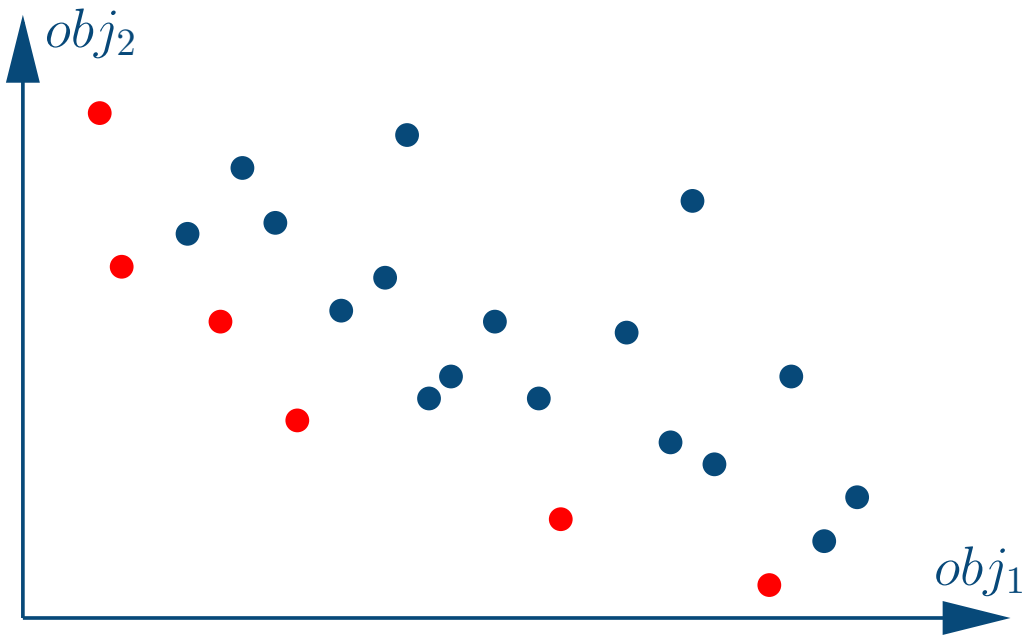
$k = 2$

Optimisation combinatoire multicritère

ensemble fini de solutions réalisables

$\Rightarrow$ trouver efficacement les solutions non dominées

k objectifs $obj_1 \ldots obj_k$



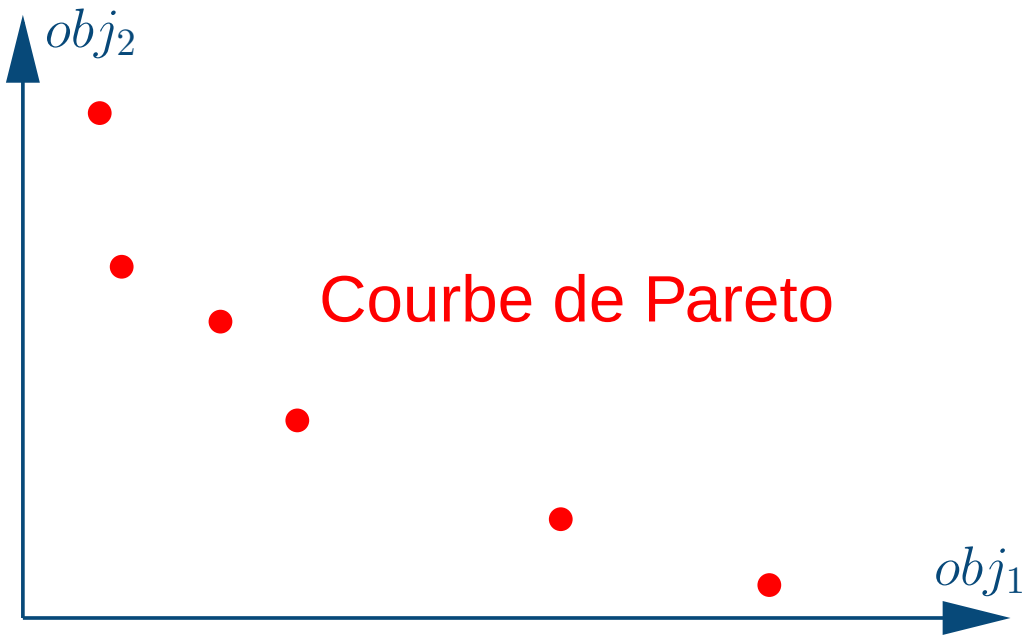
$k = 2$

Optimisation combinatoire multicritère

ensemble fini de solutions réalisables

$\Rightarrow$ trouver efficacement les solutions non dominées

k objectifs $obj_1 \ldots obj_k$



$k = 2$

Courbe de Pareto

Déterminer efficacement la courbe de Pareto est problématique

- taille potentiellement exponentielle
- décider si une solution appartient à la courbe de Pareto = problème généralement **NP**-complet

Courbe de Pareto

Déterminer efficacement la courbe de Pareto est problématique

- taille potentiellement exponentielle
- décider si une solution appartient à la courbe de Pareto = problème généralement **NP**-complet

⇓

APPROXIMATION POLYNOMIALE

Déterminer efficacement une courbe de Pareto approchée

Plan de l'exposé

1. Optimisation combinatoire multicritère
2. Approximation polynomiale et garantie de performance
3. Deux problèmes bicritères
   - 3.1 Coupe maximale pondérée
   - 3.2 Ordonnancement sur une machine
4. Conclusion

Approximation avec garantie de performance ε

minimiser 1 objectif obj

Une solution s est ε-approchée si pour toute solution réalisable s'

$$obj(s) \leq (1+\varepsilon)obj(s')$$

Approximation avec garantie de performance ε

minimiser 1 objectif obj

Une solution s est ε-approchée si pour toute solution réalisable s'

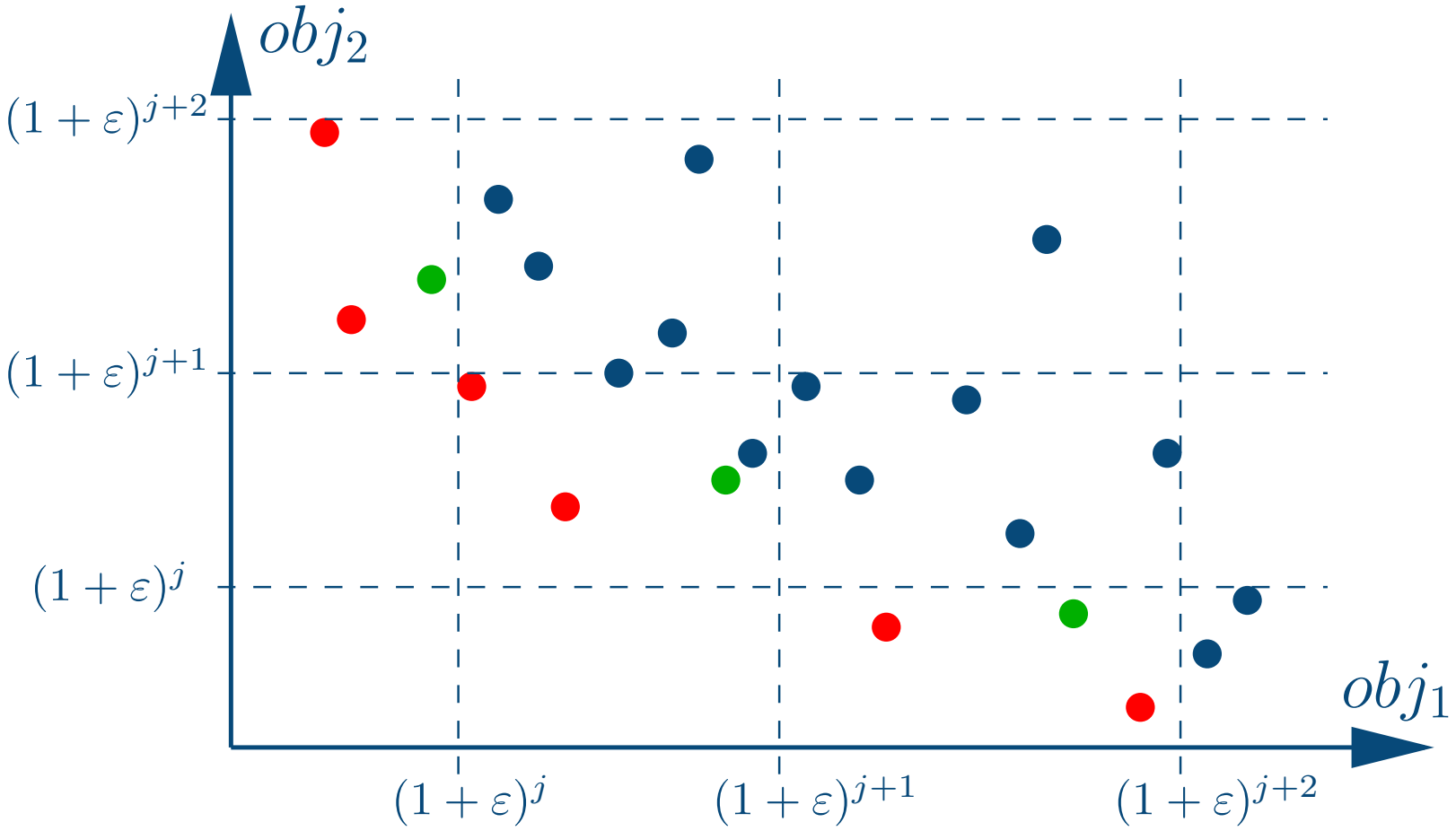
$$obj(s) \leq (1+\varepsilon) obj(s')$$

minimiser k objectifs $obj_1 \ldots obj_k$

Un ensemble de solutions P est une courbe de Pareto ε-approchée si pour toute solution réalisable s', il existe une solution $s \in P$ telle que

$$obj_i(s) \leq (1+\varepsilon) obj_i(s'),\ i = 1, \ldots, k$$

Courbe de Pareto ε-approchée



Courbe de Pareto ε-approchée

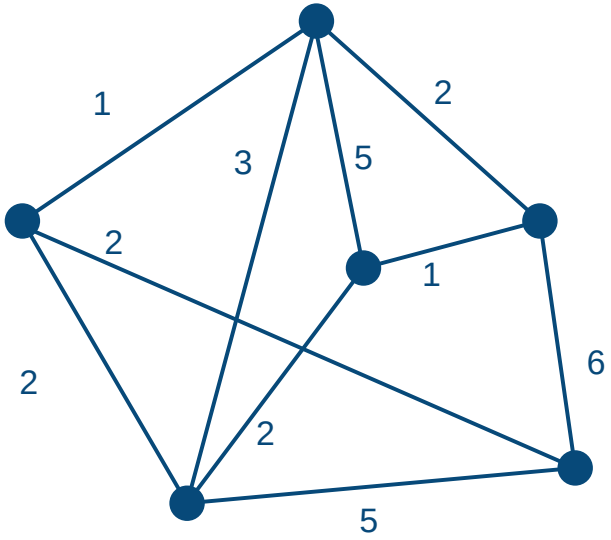
Théorème : Un nombre polynomial de solutions suffit pour approcher la courbe de Pareto avec une précision aussi petite que l'on veut. [Papadimitriou & Yannakakis, 2000]

Recherche d'algorithmes polynomiaux qui génèrent une courbe de Pareto ε-approchée

Plan de l'exposé

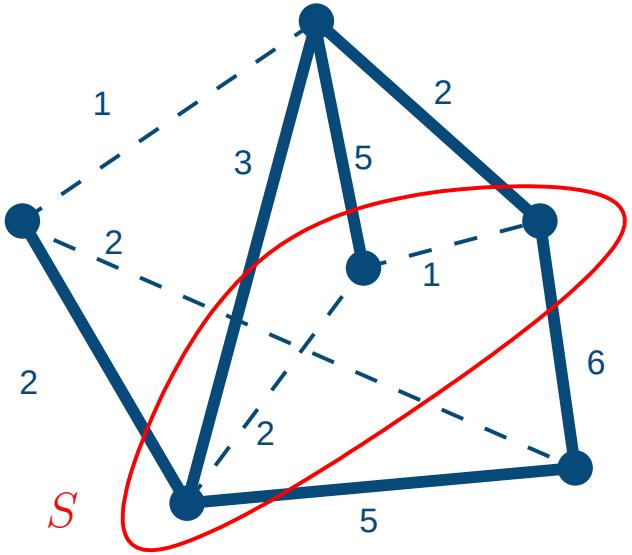
1. Optimisation combinatoire multicritère
2. Approximation polynomiale et garantie de performance
3. Deux problèmes bicritères
   - 3.1 Coupe maximale pondérée
   - 3.2 Ordonnancement sur une machine
4. Conclusion

Coupe maximale pondérée (MAX-CUT)



un poids w_{ij} pour toute arête $[ij]$

Coupe maximale pondérée (MAX-CUT)

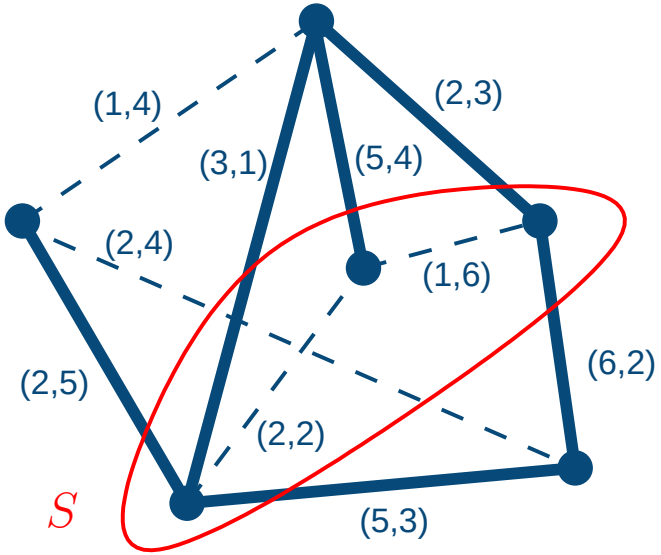


Couper les sommets en deux ensembles $(S, \overline{S})$ afin de maximiser le poids total $W(S, \overline{S}) = \sum_{i \in S, j \in \overline{S}} w_{ij}$

NP-difficile [Karp, 1972]

algorithme 0.878-approché [Goemans & Williamson, 1995]

Coupe maximale pondérée bicritère



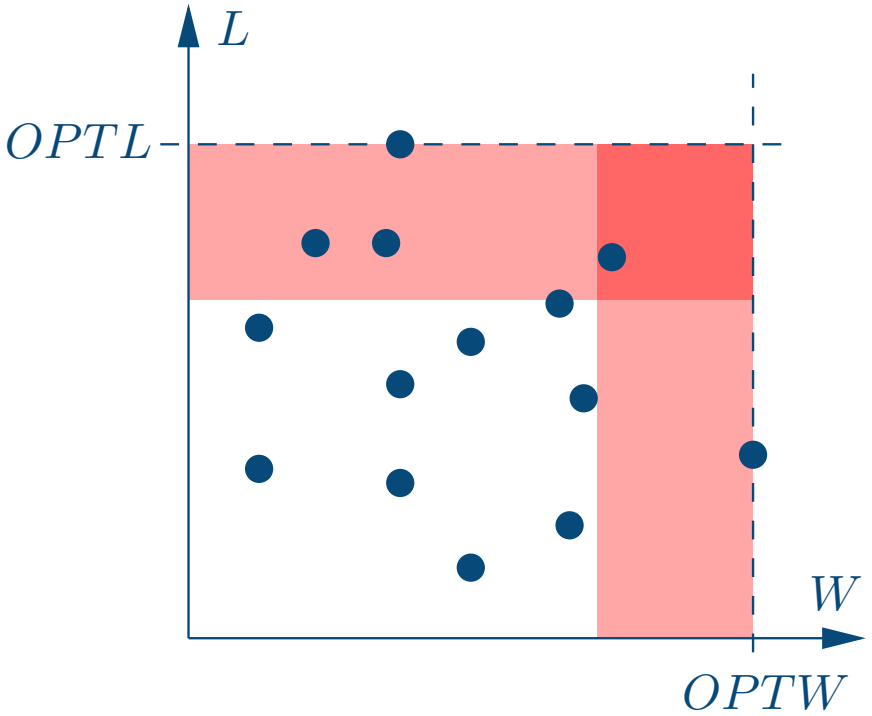
un poids w_{ij} et une longueur l_{ij} pour toute arête $[ij]$

Maximiser le poids total $W(S, \overline{S}) = \sum_{i \in S, j \in \overline{S}} w_{ij}$

ET la longueur totale $L(S, \overline{S}) = \sum_{i \in S, j \in \overline{S}} l_{ij}$

Approche

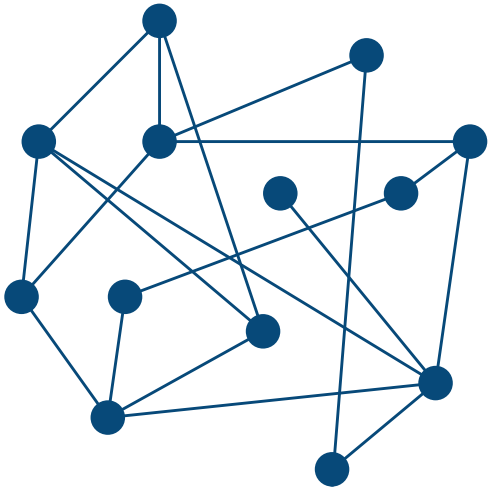
Approximation de la courbe de Pareto avec une seule coupe $(S, \overline{S})$



Algorithme bicritère α-approché si $W(S, \overline{S}) \geq \alpha\, OPTW$

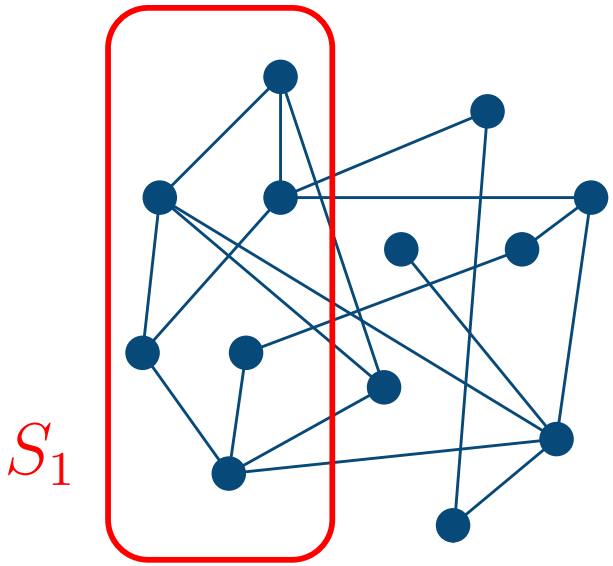
et $L(S, \overline{S}) \geq \alpha\, OPTL$

Algorithme **Bi-Approx**



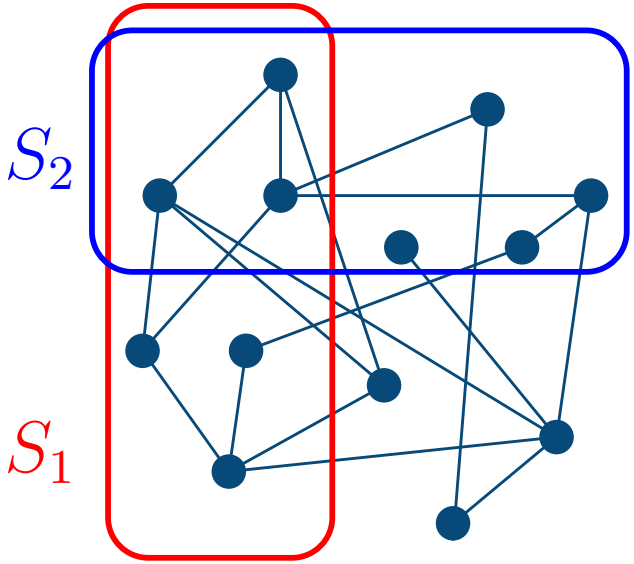
Hypothèse : on dispose d'un algorithme *boite noire* γ-approché pour le problème MAX-CUT classique

Algorithme **Bi-Approx**



Boite noire : une coupe $(S_1, \overline{S_1})$ t.q. $W(S_1, \overline{S_1}) \geq \gamma\, OPTW$

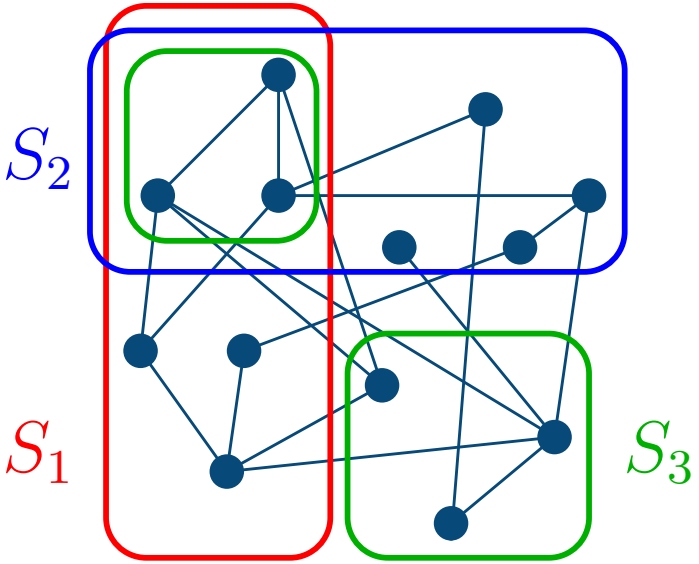
Algorithme **Bi-Approx**



Boite noire : une coupe $(S_1, \overline{S_1})$ t.q. $W(S_1, \overline{S_1}) \geq \gamma OPTW$

Boite noire : une coupe $(S_2, \overline{S_2})$ t.q. $L(S_2, \overline{S_2}) \geq \gamma OPTL$

Algorithme **Bi-Approx**



Boite noire : une coupe $(S_1, \overline{S_1})$ t.q. $W(S_1, \overline{S_1}) \geq \gamma OPTW$

Boite noire : une coupe $(S_2, \overline{S_2})$ t.q. $L(S_2, \overline{S_2}) \geq \gamma OPTL$

une coupe $(S_3, \overline{S_3})$ telle que $S_3 = (S_1 \cap S_2) \cup (\overline{S_1} \cap \overline{S_2})$

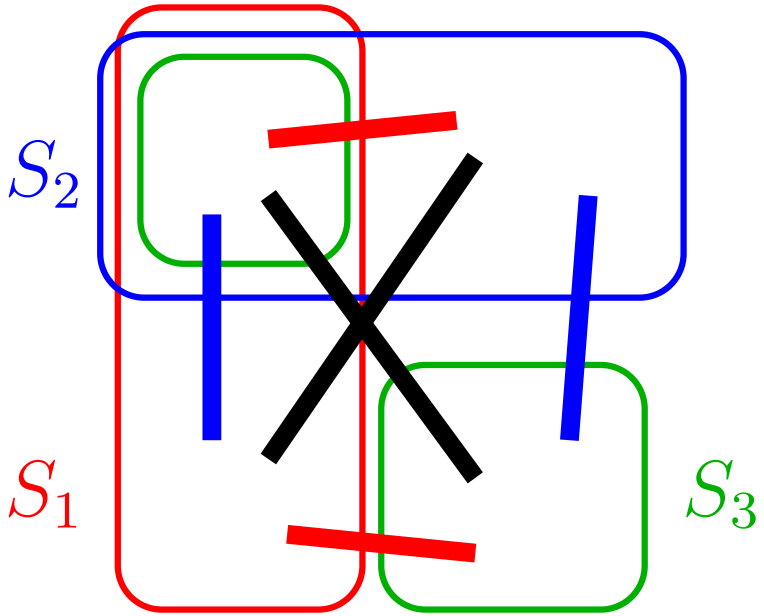
Algorithme **Bi-Approx**

Si	$L(S_1, \overline{S_1}) \geq 0.5\, L(S_2, \overline{S_2})$
Alors	Retourner $(S_1, \overline{S_1})$
Sinon	*Si* $W(S_2, \overline{S_2}) \geq 0.5\, W(S_1, \overline{S_1})$
	Alors Retourner $(S_2, \overline{S_2})$
	Sinon Retourner $(S_3, \overline{S_3})$

Boite noire : algorithme monocritère γ-approché

Théorème : **Bi-Approx** est un algorithme bicritère $\frac{\gamma}{2}$-approché pour le problème de la coupe maximale bicritère [WG 2005]

Élément de preuve



Si	$L(S_1, \overline{S_1}) \geq 0.5\, L(S_2, \overline{S_2})$
Alors	Retourner $(S_1, \overline{S_1})$
Sinon	*Si* $W(S_2, \overline{S_2}) \geq 0.5\, W(S_1, \overline{S_1})$
	Alors Retourner $(S_2, \overline{S_2})$
	Sinon Retourner $(S_3, \overline{S_3})$

$(S_1, \overline{S_1})$: arêtes rouges + arêtes noires

$(S_2, \overline{S_2})$: arêtes bleues + arêtes noires

$(S_3, \overline{S_3})$: arêtes bleues + arêtes rouges

Conséquences positives

Approximabilité

Bi-Approx 0.439-approché

- boite noire = algorithme 0.878-approché [Goemans & Williamson,1995]

Existence

Toute instance du problème bicritère admet une solution réalisable 0.5-approchée

- boite noire = algorithme exact

Lien avec l'ordonnancement

2 critères : somme pondérée des dates de terminaison
makespan

[Stein & Wein, 1997]

$\forall \delta > 0$, construction d'un ordonnancement N à la fois $\alpha(1+\delta)$-approché pour le makespan et $\beta(\frac{1+\delta}{\delta})$-approché pour la somme pondérée des dates de terminaison à partir de

- M, un ordonnancement α-approché pour le makespan
- T, un ordonnancement β-approché pour la somme pondérée des dates de terminaison

Lien avec l'ordonnancement

2 critères : somme pondérée des dates de terminaison
makespan

[Stein & Wein, 1997]

$\forall \delta > 0$, construction d'un ordonnancement N à la fois $\alpha(1+\delta)$-approché pour le makespan et $\beta(\frac{1+\delta}{\delta})$-approché pour la somme pondérée des dates de terminaison à partir de

- M, un ordonnancement α-approché pour le makespan
- T, un ordonnancement β-approché pour la somme pondérée des dates de terminaison

Approche simultanée : construction d'un compromis à partir de solutions "bonnes" pour chacun des critères pris séparément

Plan de l'exposé

1. Optimisation combinatoire multicritère
2. Approximation polynomiale et garantie de performance
3. Deux problèmes bicritères
   - 3.1 Coupe maximale pondérée
   - 3.2 Ordonnancement sur une machine
4. Conclusion

Problème

Exécuter n tâches sur 1 machine

Une solution est une permutation π des tâches
$\rightarrow$ pas de temps d'attente

Chaque tâche t_j a
- une durée d'exécution p_j
- un poids w_j

2 critères :
- Poids total : $w(\pi) = \sum_{j=1}^{n} w_j C_j^{\pi}$
- Coût total : $c(\pi) = \sum_{j=1}^{n} C_j^{\pi}$

$$1||(\sum_j C_j, \sum_j w_j C_j)$$

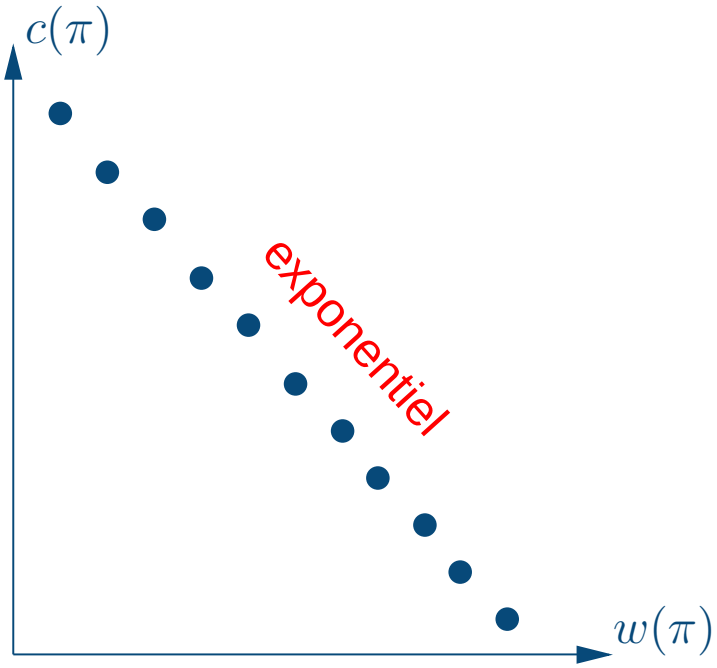
Complexité

1 critère : résolution en ordonnant les tâches par w_j/p_j décroissants [Smith, 1956]

2 critères : **NP**-difficile [Hoogeveen, 1992]

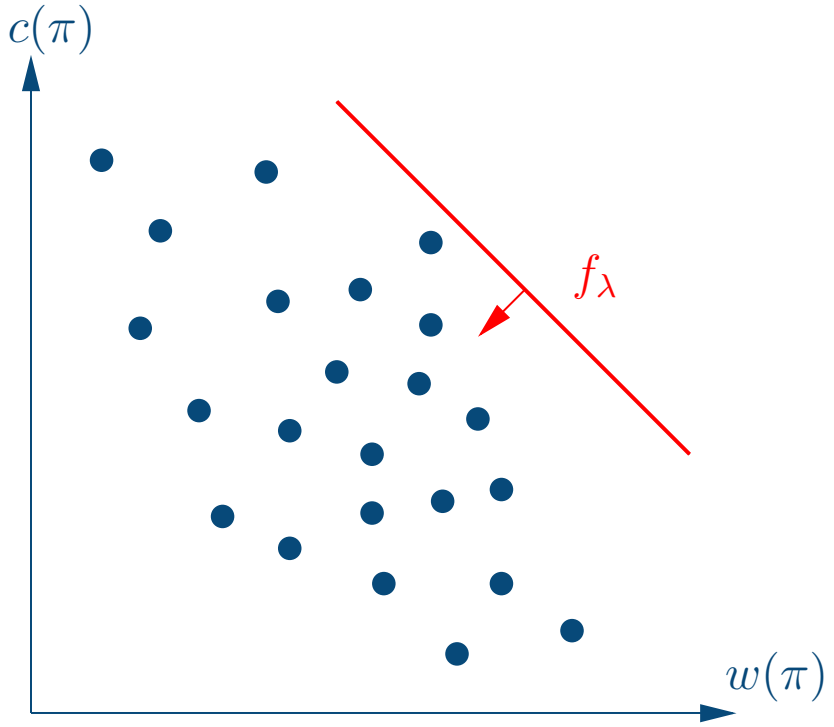
nombre de solutions parfois exponentiel [IPL, 2005]

	t_j
p_j	n^{j-1}
w_j	$n^j - 1$



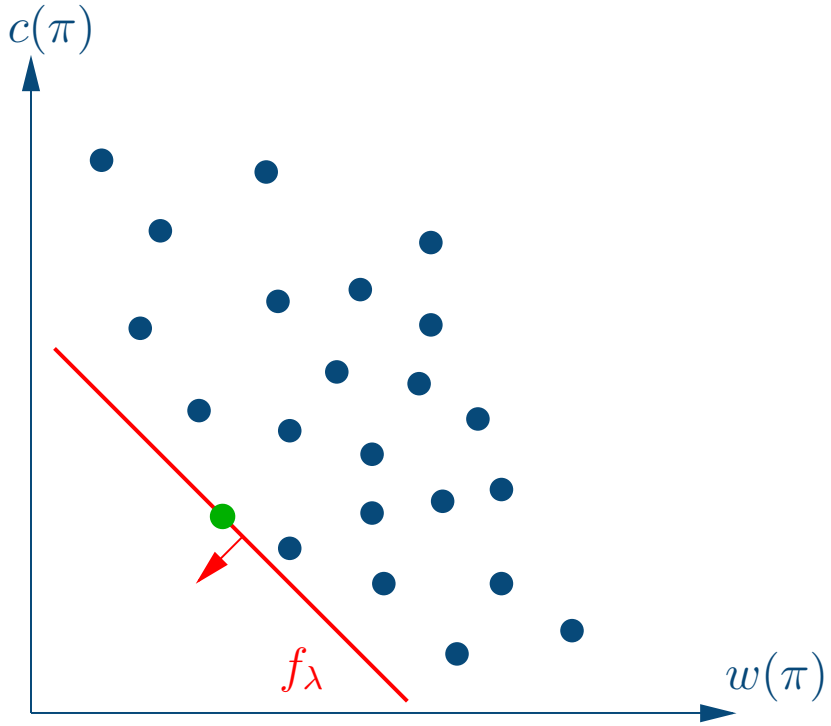
Approche par pondération des critères

Minimiser $f_\lambda(\pi) = \lambda w(\pi) + (1-\lambda)c(\pi)$ t.q. $0 \leq \lambda \leq 1$



Approche par pondération des critères

Minimiser $f_\lambda(\pi) = \lambda w(\pi) + (1-\lambda)c(\pi)$ t.q. $0 \leq \lambda \leq 1$

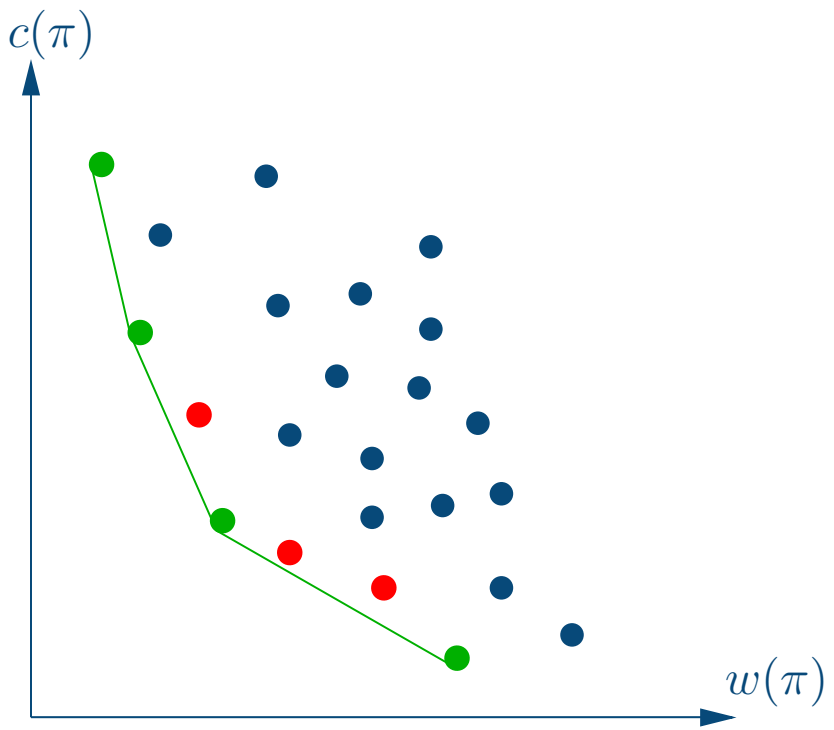


Règle de Smith : ordonner selon les $\frac{\lambda w_j + (1-\lambda)}{p_j}$ décroissants

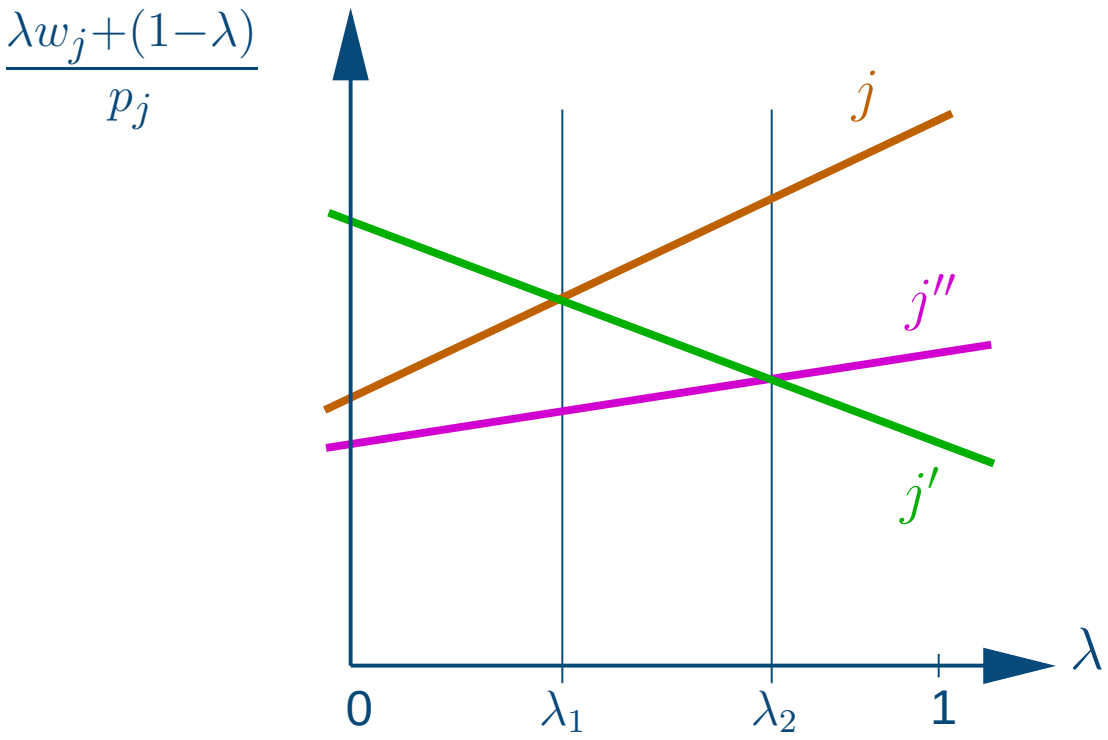
Approche par pondération des critères

Courbe de Pareto : solutions supportées ●

solutions non supportées ●



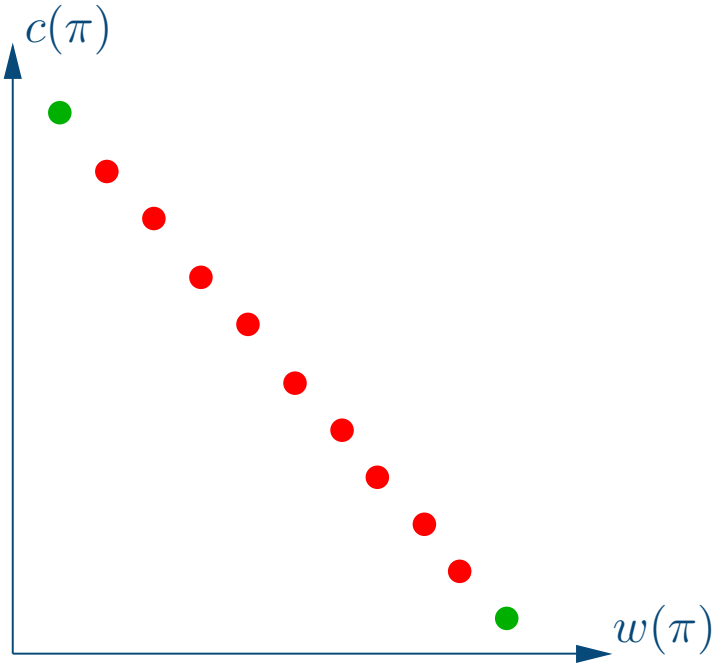
Générer les solutions supportées



Algorithme :

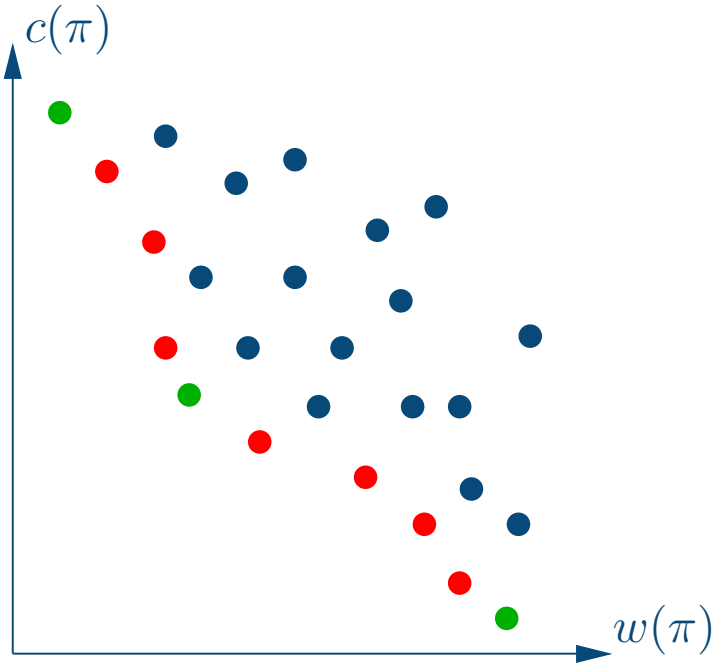
- appliquer la règle de Smith pour une valeur de λ dans chaque intervalle
- au plus $n(n-1)/2+1$ permutations

Garantie de performance

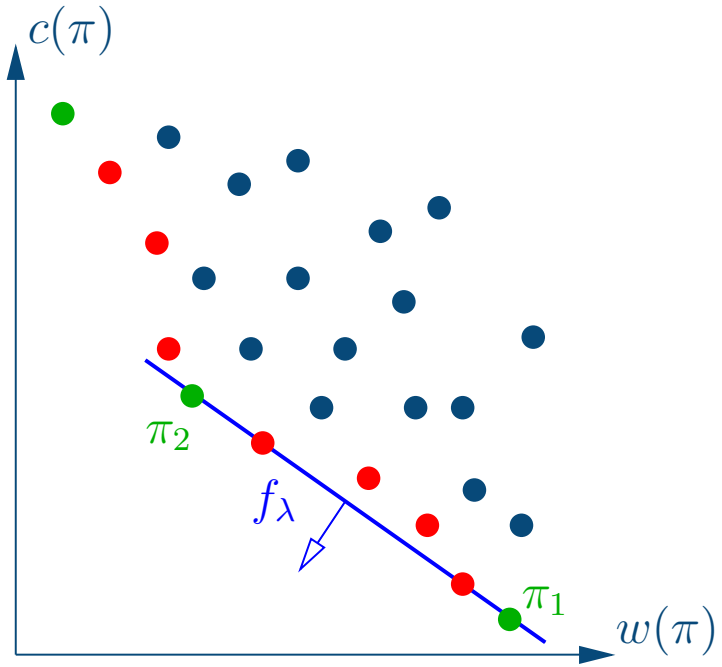


$p_j = n^{j-1}$ et $w_j = n^j - 1$

Garantie de performance

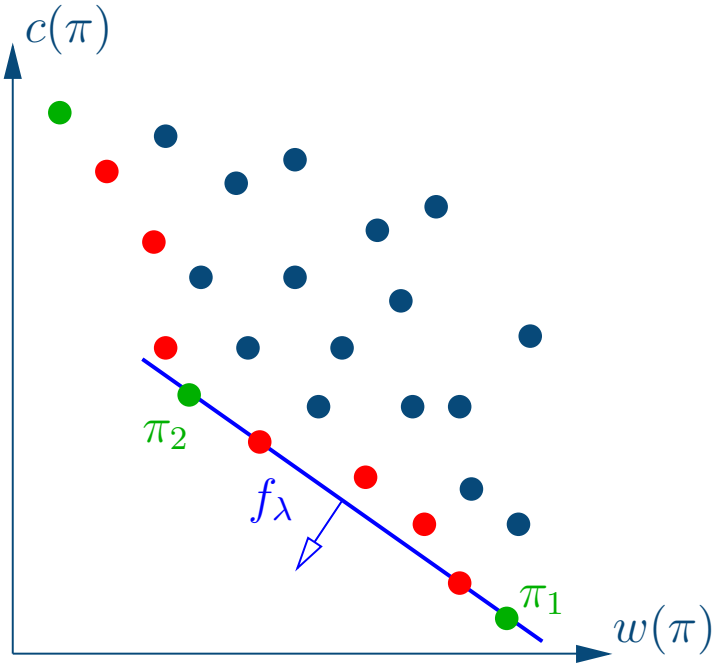


Garantie de performance



π_1 et π_2 minimisent la fonction f_λ

Garantie de performance



π_1 et π_2 minimisent la fonction f_λ

Plusieurs tâches adjacentes ont le même rapport

$$\frac{\lambda w_j + (1-\lambda)}{p_j}$$

$\Downarrow$

swap de ces tâches adjacentes pour obtenir des ordonnancements intermédiaires

Algorithme

1. Générer les permutations supportés $\{\pi_1, \pi_2, \ldots, \pi_l\}$
   (permutations classées par coûts croissants)
2. Générer des permutations intermédiaires entre π_i et π_{i+1} par swaps successifs de tâches adjacentes

Théorème : l'algorithme s'exécute en temps polynomial et l'ensemble des permutations générées constitue une courbe de Pareto $(1, 0)$-approchée pour le problème $1||(\sum_j C_j, \sum_j w_j C_j)$

[IPL, 2005]

Plan de l'exposé

1. Optimisation combinatoire multicritère
2. Approximation polynomiale et garantie de performance
3. Deux problèmes bicritères
   - 3.1 Coupe maximale pondérée
   - 3.2 Ordonnancement sur une machine
4. Conclusion

Approximation de la courbe de Pareto

Approximation avec 1 solution

Approche simultanée : construction d'un compromis à partir de solutions "bonnes" pour chacun des critères pris séparément

limite : nombre de critères

Approximation avec plusieurs solutions

Approche par pondération des critères : possible lorsque les critères sont homogènes et existence d'un algorithme monocritère exact efficace

Autres approches : adaptation d'algorithmes monocritères pour le cas multicritère (ex : recherche locale)

- $\text{TSP}(1,2)$ multicritère [TCS, 2004]

Perspectives

Étude de problèmes combinatoires multicritères

- approximation avec garantie de performance
- plus de critères
- critères hétérogènes
- inapproximabilité
  - argument de complexité **P** $\neq$ **NP**
  - non existence

Questions