COURS 6 – OPTIMISATION ROBUSTE

IAD – RHAD P. PERNY ET O. SPANJAARD

LIP6 – Université Paris 6

Partie I : Robustesse d'une solution

L'IDÉE DE ROBUSTESSE EN OPTIMISATION

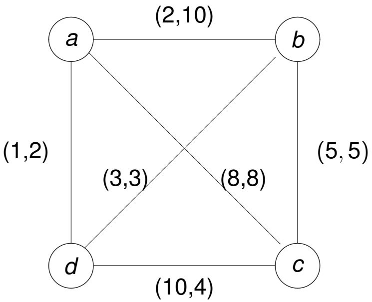
- "flexibilité" à des modifications éventuelles de l'instance ;
- "prudence" face à l'imprécision ou l'incertitude sur les coûts ;
- "aversion au risque ou à l'ambiguïté" en optimisation stochastique.

OPTIMISATION ROBUSTE

L'*optimisation robuste* concerne des situations où *différents scénarios* sur les données sont considérés et où l'objectif est de déterminer une solution qui reste "bonne" quel que soit le scénario considéré.

Liens avec la décision dans l'incertain et dans le risque.

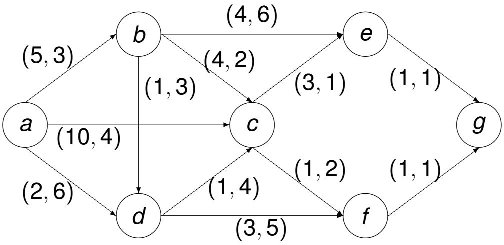
Arbre couvrant robuste (Vincke, 1999)



L'arbre $\{((a, b), (a, d), (b, c)\}$ est optimal pour le premier scénario mais mauvais pour le second.
L'arbre $\{(d, a), (d, b), (d, c)\}$ est optimal pour le second scénario mais mauvais pour le premier.

L'arbre $\{(a, d), (d, b), (b, c)\}$ peut être considéré comme un arbre robuste (même comme l'arbre le plus robuste).

PLUS COURT CHEMIN ROBUSTE



Chemins	*Sommets*	*Coûts*
1	(a, b, e, g)	(10,10)
2	(a, b, c, e, g)	(13,7)
3	(a, b, c, f, g)	(11,8)
4	(a, b, d, c, e, g)	(11,12)
5	(a, b, d, c, f, g)	(9,13)
6	(a, b, d, f, g)	(10,12)
7	(a, c, e, g)	(14,6)
8	(a, c, f, g)	(12,7)
9	(a, d, c, e, g)	(7,12)
10	(a, d, c, f, g)	(5,13)
11	(a, d, f, g)	(6,12)

Inadéquation des critères de décision habituels :

- Moyenne : chemin 10, le pire pour le scénario 2 ;
- Somme pondérée : chemins 1 et 3 inaccessibles ;
- Dominance : trop de solutions (10,11,1,3,8,7).

SOLUTIONS ROBUSTES

Soit S un ensemble de scénarios, Kouvelis et Yu (1997) mettent en avant deux concepts de solutions robustes :

- une solution est dite robuste au sens absolu si elle est de coût minimum en raisonnant au pire des cas, c'est-à-dire que la relation de préférence correspondante entre solutions est :

$$F_1 \succsim F_2 \iff \max_{s \in S} \varphi_s(F_1) \leq \max_{s \in S} \varphi_s(F_2)$$

en notant $\varphi_s(F)$ la valeur de la solution F dans le scénario s.

- une solution est dite robuste au sens relatif si le regret qu'elle est susceptible d'engendrer est minimum, c'est-à-dire que la relation de préférence correspondante entre solution est :

$$F_1 \succsim F_2 \iff \max_{s \in S}(\varphi_s(F_1) - \varphi_s(F_s^*)) \leq \max_{s \in S}(\varphi_s(F_2) - \varphi_s(F_s^*))$$

où F_s^* désigne la solution optimale si le scénario s se réalise.

DEUX APPROCHES

Deux approches selon la façon dont est défini l'ensemble des valeurs possibles des composantes élémentaires :

- par des fonctions $f : S \to \mathbb{N}$, où S peut être discret ou continu : variable exogène prenant un ensemble de valeurs connues représentées par S.
- par des ensembles de valeurs possibles : l'ensemble S des scénarios est alors l'ensemble des combinaisons possibles des coûts des différentes composantes élémentaires.

Première approche : ensemble fini $S = \{1, \ldots, q\}$ de scénarios. Valuations vectorielles sur les différentes composantes élémentaires.

Seconde approche : ensemble des scénarios défini comme un produit cartésien d'intervalles $S = \Pi_{e \in E}[l(e), u(e)]$, où l_e (resp. u_e) est la valeur minimum (resp. maximum) que peut prendre la composante élémentaire e.

Partie II : Valuations par intervalles

ARBRE COUVRANT ROBUSTE AU SENS ABSOLU

Soit $G = (V, E)$ un graphe. Un intervalle $[l(e), u(e)]$ est associé à chaque arête $e \in E$. Un scénario est une réalisation possible des coûts des arêtes, i.e. un coût $\varphi_s(e) \in [l(e), u(e)]$ est fixé pour chaque arête $e \in E$, avec $0 \leq l(e) < u(e)$.

DÉFINITION : PIRE SCÉNARIO ABSOLU

Étant donné un arbre couvrant T, un pire scénario absolu s_T^a est un scénario dans lequel le coût de T est maximum, i.e. $s_T^a \in \arg\max_{s\in S} \varphi_s(T)$.

Remarque : le scénario dans lequel toutes les arêtes de G sont au maximum est un pire scénario absolu pour T.

DÉFINITION : ARBRE COUVRANT ROBUSTE AU SENS ABSOLU

Un arbre couvrant T^a minimisant le coût dans son pire scénario absolu est dit arbre couvrant robuste au sens absolu, i.e. $T^a \in \arg\min_{T\in G} \max_{s\in S} \varphi_s(T)$.

Cet arbre s'obtient en temps polynomial en appliquant un algorithme classique (Kruskal, Prim) sur le graphe où toute arête e est valuée par $u(e)$.

ARBRE COUVRANT ROBUSTE AU SENS RELATIF

DÉFINITION : PIRE SCÉNARIO RELATIF

Étant donné un arbre couvrant T, un pire scénario relatif s_T est un scénario dans lequel la différence entre le coût de T et le coût d'un arbre couvrant minimum est maximum, i.e. $s_T \in \arg\max_{s \in S}(\varphi_s(T) - \varphi_s(T^*_s))$, où T^*_s désigne l'arbre couvrant minimum pour le scénario s.

Remarque : le scénario dans lequel toutes les arêtes de T sont au maximum et celles de $E \setminus T$ sont au minimum est un pire scénario relatif pour T.

DÉFINITION : DÉVIATION ROBUSTE

La déviation robuste pour un arbre couvrant T est la différence $d(T) = \varphi_{s_T}(T) - \varphi_{s_T}(T^*_{s_T})$, où $T^*_{s_T}$ désigne l'arbre minimum pour s_T.

DÉFINITION : ARBRE COUVRANT ROBUSTE AU SENS RELATIF

Un arbre couvrant T^r dont la déviation robuste est minimum est dit arbre couvrant robuste au sens relatif, i.e. $T^r \in \arg\min_{T \in G} d(T)$.

La recherche de cet arbre ne peut se faire en temps polynomial !

COMPLEXITÉ DU PROBLÈME (1)

PROBLÈME : RSTID

Étant donné un graphe G à valuations intervalles et une valeur $D \in \mathbb{R}^+$, existe-t-il un arbre couvrant T pour lequel $d(T) \leq D$?

DÉFINITION : DISTANCE ENTRE DEUX ARBRES

La distance entre deux arbres couvrants correspond au nombre d'arêtes présentes dans l'un mais pas dans l'autre :

$$d(T_1, T_2) = |T_1 - T_2| = |T_2 - T_1|$$

PROBLÈME : ARBRE CENTRAL

Étant donné $k \leq n - 2$, existe-t-il T_0 tq $\max_{T \in G} d(T_0, T) \leq k$?

COMPLEXITÉ DU PROBLÈME (2)

THÉORÈME (BEZRUKOV ET AL., 1996)

Le problème ARBRE CENTRAL est NP-complet.

THÉORÈME (ARON ET VAN HENTENRYCK, 2004)

Le problème RSTID est NP-complet.

Idée de la preuve : Réduction depuis ARBRE CENTRAL dans un graphe valué par des intervalles $[0, 1]$.

Arêtes faibles et arêtes fortes

Définition : Arbre faible

Un arbre couvrant est un arbre faible ssi il existe un scénario pour lequel c'est un arbre couvrant minimum.

Définition : Arête faible

Une arête est une arête faible ssi il existe un arbre faible qui la contient.

Définition : Arête forte

Une arête est une arête forte ssi, dans tous les scénarios, il existe un arbre couvrant minimum qui la contient.

SCÉNARIOS INDUITS

DÉFINITION : SCÉNARIO FAVORABLE INDUIT

On appelle scénario favorable induit par un ensemble B d'arêtes est le scénario $s^*(B)$ suivant :

$$\begin{array}{rcll} \varphi_{s^*(B)}(e) & = & l(e) & \forall e \in B \\ \varphi_{s^*(B)}(e) & = & u(e) & \forall e \in E \setminus B \end{array}$$

DÉFINITION : SCÉNARIO DÉFAVORABLE INDUIT

On appelle scénario défavorable induit par un ensemble B d'arêtes est le scénario $s_*(B)$ suivant :

$$\begin{array}{rcll} \varphi_{s_*(B)}(e) & = & u(e) & \forall e \in B \\ \varphi_{s_*(B)}(e) & = & l(e) & \forall e \in E \setminus B \end{array}$$

IDENTIFICATION DES ARÊTES FAIBLES ET FORTES

THÉORÈME (YAMAN ET AL.,2001)

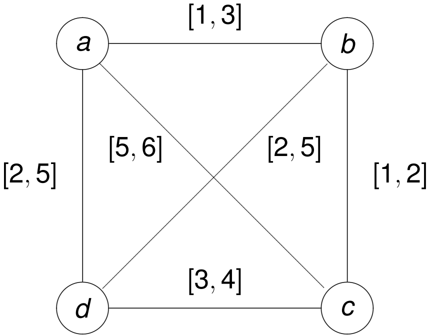
Une arête *e* est une arête faible ssi il existe un arbre minimum utilisant l'arête *e* dans le scénario favorable induit $s^*(\{e\})$.

THÉORÈME (YAMAN ET AL.,2001)

Une arête *e* est une arête forte ssi il existe un arbre minimum utilisant l'arête *e* dans le scénario défavorable induit $s_*(\{e\})$.

Conséquence : Les arêtes faibles et fortes peuvent être identifiées en temps polynomial en utilisant un algorithme de Kruskal favorisant une arête *e* donnée (i.e., en cas d'égalité on choisit *e* en priorité). On montre que si cet algorithme ne renvoie pas un arbre couvrant comportant *e*, alors il n'existe pas d'arbre couvrant minimum comportant *e*.

EXEMPLE



arêtes faibles : $E \setminus \{(a, c)\}$

arêtes fortes : $\{(b, c)\}$

PRÉTRAITEMENT

PROPOSITION (YAMAN ET AL., 2001)

Un arbre couvrant robuste au sens relatif est un arbre faible.

Conséquence : On peut exclure les arêtes non-faibles de l'arbre recherché.

PROPOSITION (YAMAN ET AL., 2001)

Il existe un arbre couvrant robuste au sens relatif comportant toutes les arêtes fortes.

Conséquence : On peut inclure les arêtes fortes dans l'arbre recherché.

Après ces décisions préliminaires on peut développer une recherche arborescente ou par programmation linéaire.

SÉPARATION ET ÉVALUATION (MONTEMANNI ET GAMBARDELLA, 2005)

Chaque nœud n est caractérisé par les éléments suivants :

- $in(n)$ l'ensemble des arêtes que l'on a décidé d'inclure dans l'arbre ;
- $out(n)$ l'ensemble des arêtes que l'on a décidé d'exclure de l'arbre ;

  REM : $in(n)$ et $out(n)$ définissent implicitement un sous-ensemble $S(n)$ d'arbres couvrants
- $T(n)$ l'arbre couvrant minimum dans $S(n)$ pour le scénario $s_*(E)$ (son coût sera noté (de coût $CostST(s_*, in, out)$)
- $lb(n)$ une borne inférieure de la déviation robuste minimum parmi les arbres de $S(n)$.

Borne inférieure (1/2)

Borne inférieure en n

La borne inférieure $lb(n)$ utilisée en n est :

$$CostST(s_*(E), in(n), out(n)) - CostST(s_*(E \setminus out(n)), \emptyset, \emptyset)$$

En pratique, $lb(n)$ correspond donc à la valeur de $T(n)$ moins la valeur de l'arbre couvrant minimum pour le scénario $s_*(E \setminus out(n))$.

BORNE INFÉRIEURE (2/2)

Soit $SubT(n) = \{p : in(p) \supseteq in(n), out(p) \supseteq out(n)\}$

THÉORÈME (MONTEMANNI ET GAMBARDELLA, 2005)

$CostST(s_*(E), in(n), out(n)) - CostST(s_*(E \setminus out(n)), \emptyset, \emptyset) \leq d(T(p))$ pour tout $p \in SubT(n)$.

Idée de la preuve : Par définition de d on a :

$$d(T(p)) = CostST(s_*(E), in(p), out(p)) - CostST(s_*(T(p)), \emptyset, \emptyset)$$

Comme $in(n) \subseteq in(p)$ et $out(n) \subseteq out(p)$:

$$CostST(s_*(E), in(n), out(n)) \leq CostST(s_*(E), in(p), out(p))$$

Comme $T(p) \subseteq E \setminus out(p)$:

$$CostST(s_*(E \setminus out(n)), \emptyset, \emptyset) \geq CostST(s_*(T(p)), \emptyset, \emptyset)$$

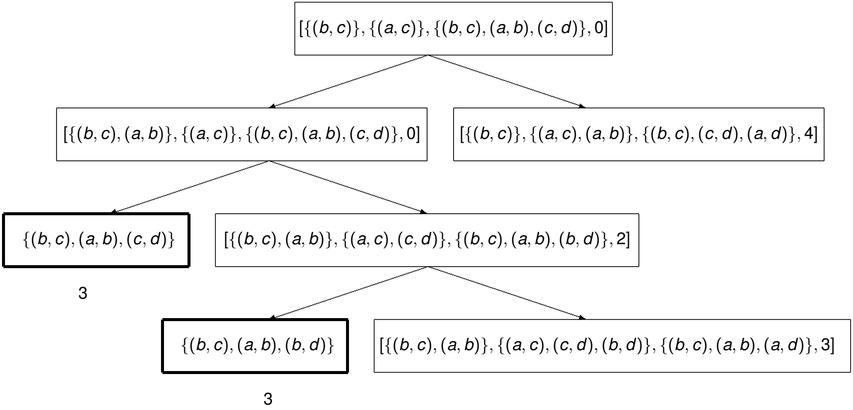
STRATÉGIE DE BRANCHEMENT

Développement de l'arborescence de recherche :

- Racine de l'arborescence :
  $out(r) = \{e : e \text{ est une arête faible}\}$,
  $in(r) = \{e : e \text{ est une arête forte}\}$,
  $lb(r) = 0$
- On développe le nœud n avec la borne la plus basse
- En un nœud n, on choisit de mettre dans *in* ou *out* l'arête $e \in T(n) \setminus in(n)$ dont l'interdiction pénalise le plus la valeur de $T(n)$
- On obtient un noeud n' et un noeud n'' tels que :

$$\begin{array}{ll} in(n') = in(n) & out(n') = out(n) \cup \{e\} \\ in(n'') = in(n) \cup \{e\} & out(n'') = out(n) \end{array}$$

EXEMPLE


$[\{(b,c)\},\{(a,c)\},\{(b,c),(a,b),(c,d)\},0]$
$[\{(b,c),(a,b)\},\{(a,c)\},\{(b,c),(a,b),(c,d)\},0]$
$[\{(b,c)\},\{(a,c),(a,b)\},\{(b,c),(c,d),(a,d)\},4]$
$\{(b,c),(a,b),(c,d)\}$
$[\{(b,c),(a,b)\},\{(a,c),(c,d)\},\{(b,c),(a,b),(b,d)\},2]$
3
$\{(b,c),(a,b),(b,d)\}$
$[\{(b,c),(a,b)\},\{(a,c),(c,d),(b,d)\},\{(b,c),(a,b),(a,d)\},3]$
3

CHEMIN ROBUSTE AU SENS RELATIF

Soit $G = (V, A)$ un graphe orienté. Un intervalle $[l(a), u(a)]$ est associé à chaque arc $a \in A$. Un scénario est une réalisation possible des coûts des arêtes, i.e. un coût $\varphi_s(a) \in [l(a), u(a)]$ est fixé pour chaque arc $a \in A$, avec $0 \leq l(a) < u(a)$.

PROBLÈME : CHEMIN ROBUSTE AU SENS RELATIF

Données : G à valuations intervalles, une source s et un puit t.
Solutions réalisables : ensemble $\mathcal{P}$ des chemins de s à t.
Objectif : trouver un chemin P^r robuste au sens relatif, i.e. $P^r \in \arg\min_{P \in \mathcal{P}} d(P)$.

Remarque :
Le scénario $s_*(P)$ est un pire scénario relatif pour P.

RÉSOLUTION (1)

Notations :

- $C^P(Q)$: coût d'un chemin Q dans le scénario $s_*(P)$;
- $SP(P)$: plus court chemin de s à t dans le scénario $s_*(P)$;
- $LC(P)$: coût d'un chemin P dans le scénario $s^*(E)$;
- $UC(P)$: coût d'un chemin P dans le scénario $s_*(E)$;
- P^i : i^e plus court chemin dans le scénario $s_*(E)$.

Idée de l'algorithme : on énumère les chemins dans l'ordre croissant selon le scénario $s_*(E)$.

LEMME (MONTEMANNI ET GAMBARDELLA, 2004)

$UC(P^1) \geq C^P(SP(P)) \ \ \forall P \in \mathcal{P}$

Preuve : $\sum_{a \in P^1} u(a) \geq \sum_{a \in P^1} \varphi_{s_*(P)}(a) \geq \sum_{a \in SP(P)} \varphi_{s_*(P)}(a) = C^P(SP(P))$

RÉSOLUTION (2)

THÉORÈME (MONTEMANNI ET GAMBARDELLA, 2004)

$d(P^k) \geq UC(P^i) - UC(P^1) \;\; \forall k \geq i$

Preuve : $d(P^k) = UC(P^k) - C^{P^k}(SP(P^k)) \geq UC(P^k) - UC(P^1) \; \forall P^k \in \mathcal{P}$
Par définition on a : $UC(P^k) \geq UC(P^i) \;\; \forall k \geq i$
Finalement : $d(P^k) \geq UC(P^k) - UC(P^1) \geq UC(P^i) - UC(P^1) \;\; \forall k \geq i$

PROPOSITION (MONTEMANNI ET GAMBARDELLA, 2004)

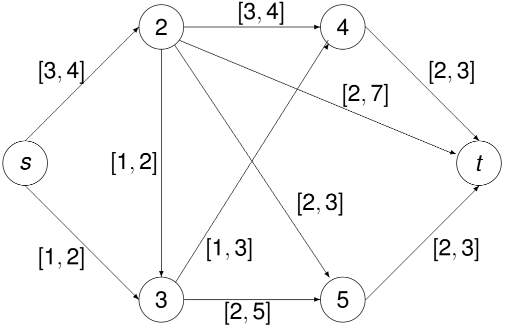
Si $UC(P^i) - UC(P^1) \geq \min_{j \leq i}\{d(P^j)\}$ alors $\arg\min_{j \leq i}\{d(P^j)\}$ robuste relatif.

Preuve : D'après le théorème : $d(P^k) \geq UC(P^i) - UC(P^1) \;\; \forall k \geq i$
Par hypothèse : $UC(P^i) - UC(P^1) \geq \min_{j \leq i}\{d(P^j)\} \; \forall k \geq i$
Finalement : $d(P^k) \geq \min_{j \leq i}\{d(P^j)\} \;\; \forall k \geq i$

ALGORITHME

1. $P^* \leftarrow P^1$, $d^* \leftarrow d^1$
2. Enumerer les chemins P^i par ordre croissant de $UC(P^i)$
3. à l'étape i courante, calculer $d(P^i)$ et mettre à jour : si $d(P^i) < d^*$ alors $d^* \leftarrow d(P^i)$ et $P^* \leftarrow P^i$
4. on stoppe l'énumération si $UC(P^i) - UC(P^1) \geq d^*$

EXEMPLE



P	$UC(P)$	$d(P)$
$(s, 2, 4, t)$	11	6
$(s, 2, t)$	11	
$(s, 2, 5, t)$	10	6
$(s, 2, 3, 4, t)$	12	
$(s, 2, 3, 5, t)$	14	
$(s, 3, 4, t)$	8	3
$(s, 3, 5, t)$	10	5

Le chemin $(s, 3, 4, t)$ est robuste au sens relatif.

Partie III : Valuation par des vecteurs

PLUS COURT CHEMIN ROBUSTE

PROBLÈME : CHEMIN ROBUSTE AU SENS ABSOLU

Données : un graphe orienté fini $G = (V, A)$ sans circuit, deux sommets s et t inclus dans V, un ensemble $S = \{1, \ldots, m\}$ de scénarios ;
Solutions réalisables : l'ensemble $\mathcal{P}$ des chemins de s à t ;
Objectif : on cherche à déterminer le chemin P qui minimise :

$$\max_{s \in S} \sum_{a \in P} \varphi_s(a)$$

PROBLÈME : CHEMIN ROBUSTE AU SENS RELATIF

Données : un graphe orienté fini $G = (V, A)$ sans circuit, deux sommets s et t inclus dans V, un ensemble $S = \{1, \ldots, m\}$ de scénarios ;
Solutions réalisables : l'ensemble $\mathcal{P}$ des chemins de s à t ;
Objectif : on cherche à déterminer le chemin P qui minimise :

$$\max_{s \in S} \sum_{a \in P} (\varphi_s(a) - \varphi_s^*)$$

où φ_s^* désigne la valeur du plus court chemin pour le scénario s.

Difficulté : Violation du principe d'optimalité !

Complexité du problème

On appelle ARSP (Absolute Robust Shortest Path) et RDSP (Robust Deviation Shortest Path) les versions décisions des deux problèmes précédents.

Problème : Partition

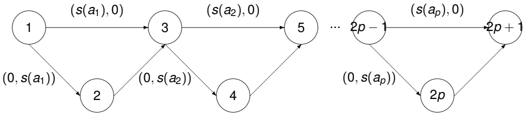
Instance : Un ensemble fini $A = \{a_1, \ldots, a_p\}$ et une taille $s(a) \in \mathbb{Z}^+$ pour tout $a \in A$.

Question : Existe-t-il un sous-ensemble $A' \subseteq A$ tel que $\sum_{a \in A'} s(a) = \sum_{a \in A - A'} s(a)$?

Théorème (Yu et Yang, 1998)

Le problème ARSP est NP-complet.

Idée de la preuve : Réduction depuis PARTITION.



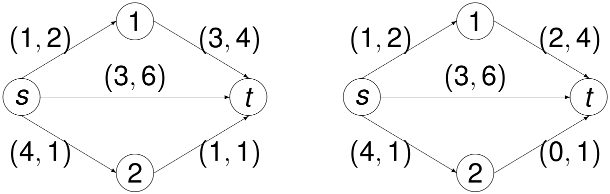
LIEN ROBUSTE ABSOLU - ROBUSTE RELATIF

On peut transformer un problème de plus court chemin robuste au sens relatif en un problème de plus court chemin robuste au sens absolu via la transformation suivante de l'instance :

$$\forall a \in A',\ \varphi'_s(a) = M + \varphi_s(a) - \varphi^*_s$$

où $A' \subseteq A$ l'ensemble des arcs menant à t et $M = \max\{\max_{a \in A', s \in S}(\varphi^*_s - \varphi_s(a)); 0\}$.

Exemple : $\varphi^*_1 = 3$, $\varphi^*_2 = 2$, $M = 2$



Résolution de ARSP

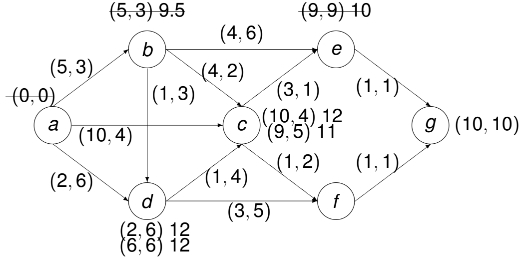
Murthy et Her (1992) ont incorporé à un algorithme classique par étiquetages des techniques de coupe pour un chemin efficace *P* de *s* à un sommet quelconque :

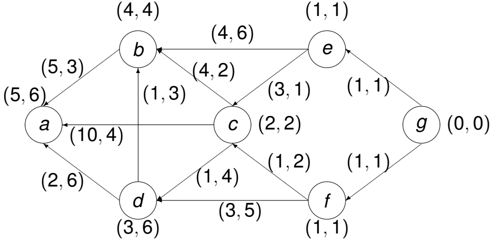
- première relaxation possible (= supprimer des contraintes dans le PL) : déterminer les plus courts chemins pour les différents scénarios du puits aux autres sommets dans le graphe inversé

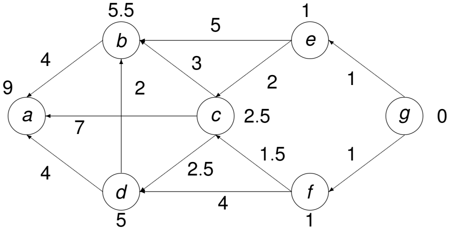
  $\hookrightarrow$ minorant de la valeur pour le scénario
- autre relaxation possible (= relaxation lagrangienne dans le PL) : déterminer un plus court chemin selon une somme pondérée des scénarios, ce qui est réalisable en temps polynomial par un algorithme classique

  $\hookrightarrow$ minorant de la valeur minmax

EXEMPLE


~~(5, 3)~~ 9.5
~~(9, 9)~~ 10
(4, 6)
(5, 3)
(4, 2)
(1, 3)
(3, 1)
(1, 1)
~~(0, 0)~~
(10, 4) 12
(9, 5) 11
(10, 10)
(10, 4)
(1, 2)
(1, 1)
(2, 6)
(1, 4)
(3, 5)
(2, 6) 12
(6, 6) 12
a
b
c
d
e
f
g


(4, 4)
(1, 1)
(4, 6)
(5, 3)
(4, 2)
(5, 6)
(1, 3)
(3, 1)
(1, 1)
(10, 4)
(2, 2)
(0, 0)
(1, 2)
(1, 1)
(2, 6)
(1, 4)
(3, 5)
(3, 6)
(1, 1)
a
b
c
d
e
f
g


5.5
1
5
4
3
2
1
9
2
7
2.5
0
1.5
1
4
2.5
4
5
1
a
b
c
d
e
f
g

Arbre couvrant robuste

Problème : Arbre couvrant robuste au sens absolu

Données : un graphe non-orienté fini $G = (V, E)$, un ensemble $S = \{1, \ldots, m\}$ de scénarios ;
Solutions réalisables : l'ensemble $\mathcal{T}$ des arbres couvrants de G ;
Objectif : on cherche à déterminer l'arbre couvrant T qui minimise :

$$\max_{s \in S} \sum_{e \in T} \varphi_s(e)$$

Problème : Arbre couvrant robuste au sens relatif

Données : un graphe non-orienté fini $G = (V, E)$, un ensemble $S = \{1, \ldots, m\}$ de scénarios ;
Solutions réalisables : l'ensemble $\mathcal{T}$ des arbres couvrants de G ;
Objectif : on cherche à déterminer l'arbre couvrant T qui minimise :

$$\max_{s \in S} \sum_{e \in T} (\varphi_s(e) - \varphi_s^*)$$

où φ_s^* désigne la valeur de l'arbre couvrant min pour le scénario s.

Difficulté : Violation du principe glouton !

COMPLEXITÉ DU PROBLÈME

On appelle ARST (Absolute Robust Spanning Tree) et RDST (Robust Deviation Spanning Tree) les versions décisions des deux problèmes précédents.

PROBLÈME : PARTITION

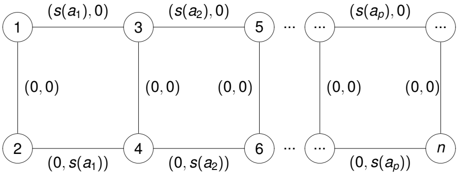
Instance : Un ensemble fini $A = \{a_1, \ldots, a_p\}$ et $s(a) \in \mathbb{Z}^+$ pour tout $a \in A$.

Question : Existe-t-il $A' \subseteq A$ tel que $\sum_{a \in A'} s(a) = \sum_{a \in A - A'} s(a)$?

THÉORÈME (HAMACHER ET RUHE, 1994 ; YU, 1998)

Le problème ARST est NP-complet.

Idée de la preuve : Réduction depuis PARTITION.



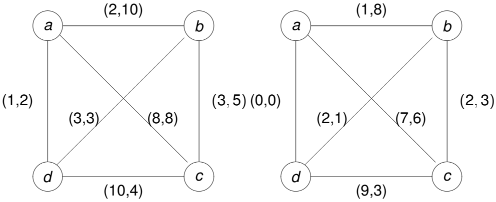
LIEN ROBUSTE ABSOLU - ROBUSTE RELATIF

On peut transformer un problème d'arbre couvrant robuste au sens relatif en un problème d'arbre couvrant robuste au sens absolu via la transformation suivante de l'instance :

$$\forall a \in A,\ \varphi_s(a) = M + \varphi_s(a) - \frac{\varphi_s^*}{n-1}$$

où $M = \max\{\max_{a \in A, s \in S}(\frac{\varphi_s^*}{n-1} - \varphi_s(a)); 0\}$.

Exemple : $\varphi_1^* = 6$, $\varphi_2^* = 9$, $M = 1$



Résultats préliminaires pour résoudre Arst

Hamacher et Ruhe (1994) ont proposé un algorithme consistant à énumérer les arbres couvrants dans l'ordre croissant selon une somme pondérée des critères.

Lemme (Hamacher et Ruhe, 1994)

Pour tout $T \in \mathcal{T}$ et pour tout vecteur $\lambda = (\lambda_1, \ldots, \lambda_m)$ de réels positifs satisfaisant $\sum_{s=1}^{m} \lambda_s = 1$ on a :

$$\sum_{s=1}^{m} \lambda_s \varphi_s(T) \leq \max\{\varphi_1(T), \ldots, \varphi_m(T)\}$$

Par conséquent :

Lemme (Hamacher et Ruhe, 1994)

Soit T^1 un arbre minimal selon le vecteur de poids λ et T^* un arbre robuste au sens absolu. Alors :

$$\varphi(T^1) = \sum_{s=1}^{m} \lambda_s \varphi_s(T^1) \leq \sum_{s=1}^{m} \lambda_s \varphi_s(T^*) \leq \max\{\varphi_1(T^*), \ldots, \varphi_m(T^*)\}$$

RÉSOLUTION DE ARST

En notant T^k le k^e arbre selon λ et $\max(T) = \max\{\varphi_1(T), \ldots, \varphi_m(T)\}$) :

THÉORÈME : HAMACHER ET RUHE (1994)

Soit k un entier positif tel que :

$$\varphi(T) = \sum_{s=1}^m \lambda_s \varphi_s(T^{k-1}) < \min\{\max(T^i) : i = 1, \ldots, k-1\}$$

et $$\sum_{s=1}^m \lambda_s \varphi_s(T^k) \geq \min\{\max(T^i) : i = 1, \ldots, k\}$$

alors $T^* \in \arg\min\{\varphi(T^i) : i = 1, \ldots, k\}$ est robuste au sens absolu.

Idée de la preuve : On établit :

$$\begin{array}{rcll} \max(T^*) & = & \min\{\max(T^i) : i = 1, \ldots, k\} & \\ & \leq & \sum_{s=1}^m \lambda_s \varphi_s(T^k) & \text{(par hypothèse)} \\ & \leq & \sum_{s=1}^m \lambda_s \varphi_s(T) & \text{(par définition des k meilleurs)} \\ & \leq & \max(T) & \end{array}$$

pour tout arbre couvrant $T \neq T^1, \ldots, T^k$.

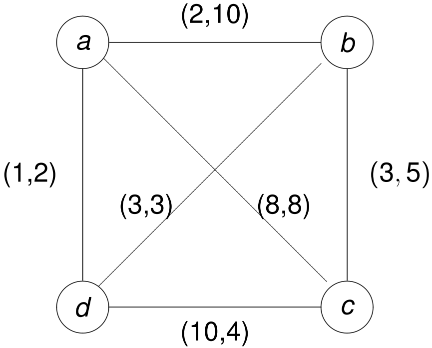
ALGORITHME DE HAMACHER ET RUHE

On note $\max(T)$ la quantité $\max_{s \in S} \varphi_s(T)$

1. $T^* \leftarrow P^1$, $\max^* \leftarrow \max(T^*)$
2. Enumerer les arbres T^i par ordre croissant de $\varphi(T^i)$
3. à l'étape i courante, calculer $\varphi(T^i)$ et mettre à jour : si $\max(T^i) < \max^*$ alors $\max^* \leftarrow \max(T^i)$ et $T^* \leftarrow T^i$
4. on stoppe l'énumération si $\varphi(T^i) \geq \max^*$

En effet, si on sort à l'itération i, on a pour $k > i$: $\max(T^k) \geq \phi(T^k) \geq \phi(T^i) \geq max^*$ donc la solution T^k n'est pas intéressante.

EXEMPLE



L'arbre couvrant $\{(a,d),(b,d),(b,c)\}$ est l'arbre couvrant minimum pour la moyenne, avec un coût de 8.5. Le vecteur associé est $(7,10)$. C'est l'arbre couvrant robuste au sens absolu.

Approche axiomatique

On cherche une relation de préférence $\succsim$ satisfaisant les axiomes suivants :

Axiome : $\succsim_P$-monotonie

Pour tout $x, y \in \mathbb{R}_+^q$, $(x \succsim_P y \Rightarrow x \succsim y)$ et $(x \succ_P y \Rightarrow x \succ y)$.

Axiome : Principe de transfert

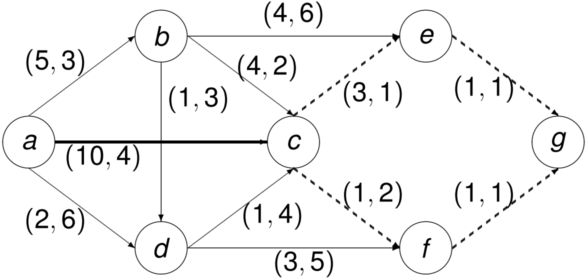
Soit $x \in \mathbb{R}_+^q$ tel que $x_i > x_j$ pour un couple (i, j). Alors pour tout ε tel que $0 \leq \varepsilon \leq x_i - x_j$, $\;x - \varepsilon e_i + \varepsilon e_j \succsim x$ où e_i (resp. e_j) est le vecteur dont la $i^{\text{ème}}$ (resp. $j^{\text{ème}}$) composante est égale à 1, toutes les autres étant nulles.

Axiome : Symétrie

Pour tout $x \in \mathbb{R}_+^q$, pour toute permutation π de $\{1, \ldots, q\}$, $(x_{\pi(1)}, \ldots, x_{\pi(q)}) \sim (x_1, \ldots, x_q)$

Remarque : Ce dernier axiome est induit par le principe de transfert.

$\succsim_P$-MONOTONIE ET PRINCIPE DE TRANSFERT



Les vecteur-coûts sont respectivement $(13, 5)$ et $(11, 6)$. D'une part $(11, 6) \succ_P (12, 6)$ et donc $(11, 6) \succ (12, 6)$ par $\succsim_P$-monotonie ; d'autre part, $(12, 6) \succsim (13, 5)$ par le principe de transfert appliqué pour le transfert $(13 - 1, 5 + 1) = (12, 6)$. Ainsi, on obtient : $(11, 6) \succ (13, 5)$ par transitivité.

UN RÉSULTAT DE CARACTÉRISATION

DÉFINITION : VECTEUR DE LORENZ

Pour tout $x \in \mathbb{R}_+^q$, le *vecteur de Lorenz généralisé* associé à x est le vecteur :

$$L(x) = (x_{[1]}, x_{[1]} + x_{[2]}, \ldots, x_{[1]} + x_{[2]} + \ldots + x_{[q]})$$

où $x_{[1]} \geq x_{[2]} \geq \ldots \geq x_{[q]}$ représentent les composantes de x rangées par ordre décroissant. La $k^{\text{ième}}$ composante de $L(x)$ est $L_k(x) = \sum_{i=1}^{k} x_{[i]}$.

DÉFINITION : DOMINANCE DE LORENZ

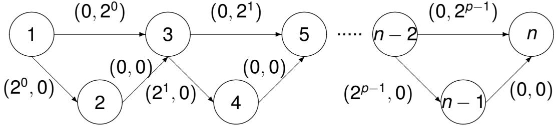
La dominance de Lorenz généralisée (que nous appelons $\succsim_L$-dominance en abrégé) sur $\mathbb{R}_+^q$ est définie par : $\forall x, y \in \mathbb{R}_+^q,\ x \succsim_L y \iff L(x) \succsim_P L(y)$

THÉORÈME (CHONG, 1976)

Pour toute paire de vecteurs distincts $x, y \in \mathbb{R}_+^q$, si $x \succsim_P y$, ou si x est obtenu à partir de y par un transfert admissible, alors $x \succsim_L y$. Inversement, si $x \succsim_L y$, alors il existe une séquence de transferts admissibles et/ou d'améliorations de Pareto qui transforme y en x.

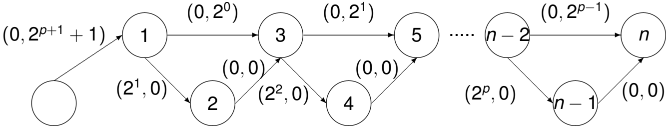
NOMBRE DE SOLUTIONS LORENZ-OPTIMALES

Une instance pathologique pour Pareto :



$$\{(x, 2^p - 1 - x),\ x \in \{0, \ldots, 2^p - 1\}\} \text{ pour } n = 2p + 1$$

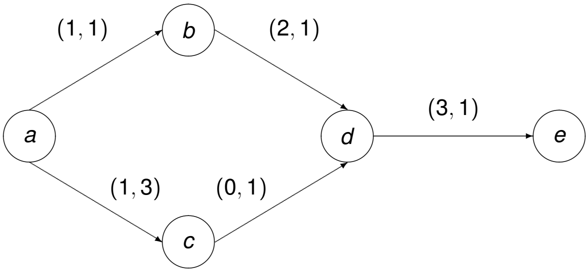
Une instance pathologique pour Lorenz :



ens des vecteurs-coûts : $\{(2x, 3 \times 2^p - x),\ x \in \{0, \ldots, 2^p - 1\}\}$
ens des L-vecteurs : $\{(3 \times 2^p - x, 3 \times 2^p + x),\ x \in \{0, \ldots, 2^p - 1\}\}$

VIOLATION DU PRINCIPE D'OPTIMALITÉ

Le chemin (a, c, d, e) est Lorenz-optimal et pourtant le sous-chemin (a, c, d) est dominé au sens de Lorenz.



On ne peut appliquer la programmation dynamique !

ALGORITHME DE RÉSOLUTION

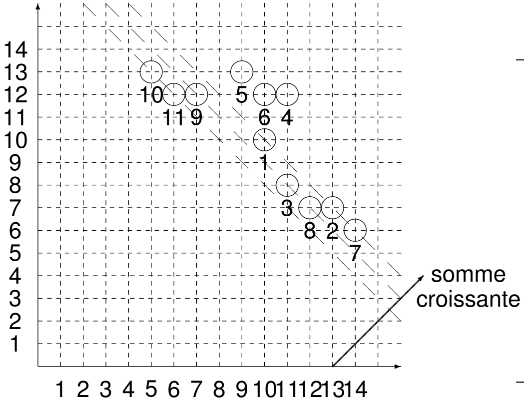
PROPOSITION

Un vecteur-coût $(x_1, \ldots, x_q)$ $\succsim_L$-domine tout vecteur-coût $(y_1, \ldots, y_q)$ tel que $\sum_{i=1}^{q} y_i > q.x_{[1]}$.

ALGORITHME : LORENZ OPTIMISATION

$X^1 \leftarrow \arg\min_{X \in \mathcal{F}} \sum_{i=1}^{p} x_i$;
Poser $b \leftarrow \max_{i=1,\ldots,p} x_i^1$; $LL \leftarrow \emptyset$; $YY \leftarrow \{X^1\}$; $k \leftarrow 2$;
Tant que $\sum_{i=1}^{p} x_i^{k-1} \leq p \times b$
    Calculer la k^e meilleure solution X^k ;
    Si $\max_{i=1,\ldots,p} x_i^k < b$ alors $b \leftarrow \max_{i=1,\ldots,p} x_i^k$;
    Si $\sum_{i=1}^{p} x_i^k > \sum_{i=1}^{p} x_i^{k-1}$ alors
        Compléter LL avec les sol. dans YY qui sont L-eff. dans $LL \cup YY$;
        $YY \leftarrow \emptyset$;
    $YY \leftarrow YY \cup \{X^k\}$;
    $k \leftarrow k + 1$;
Renvoyer l'ens LL des solutions L-efficaces ;

EXEMPLE



Path	*Vertices*	*Costs*
1	(a, b, e, g)	(10,10)
2	(a, b, c, e, g)	(13,7)
3	(a, b, c, f, g)	(11,8)
4	(a, b, d, c, e, g)	(11,12)
5	(a, b, d, c, f, g)	(9,13)
6	(a, b, d, f, g)	(10,12)
7	(a, c, e, g)	(14,6)
8	(a, c, f, g)	(12,7)
9	(a, d, c, e, g)	(7,12)
10	(a, d, c, f, g)	(5,13)
11	(a, d, f, g)	(6,12)

Rafinement de la dominance de Lorenz

Neutrality. For all x, y in X, $L(x) = L(y) \Rightarrow x \sim y$.
Complete weak-order. $\succsim$ is reflexive, transitive and complete.

Continuity. Let $L, M, N \in L(X)$ such that $L \succ M \succ N$. There exists $\alpha, \beta \in]0,1[$ such that :

$$\alpha L + (1-\alpha)N \succ M \succ \beta L + (1-\beta)N$$

Independence. Let L, M, N belong to $L(X)$. Then, for all $\alpha \in]0,1[$:

$$L \succ M \Longrightarrow \alpha L + (1-\alpha)N \succ \alpha M + (1-\alpha)N$$

$\Rightarrow$ Utilisation d'un OWA avec : $w_1 > w_2 > \ldots > w_m$

POUR ALLER PLUS LOIN...

Optimisation robuste dans des contextes informationnels plus riches :

- Optimisation pour un agent averse au risque
  - SSD généralise la dominance de Lorenz
  - RDU généralise OWA
- Optimisation pour un agent averse à l'ambiguité
- Optimisation dans l'incertain non-probabiliste...