

L'art de reconfigurer un noeud de routage

Francois CLAD, Pascal MERINDOL et Jean-Jacques PANSIOT

ICube - Université de Strasbourg

En collaboration avec Pierre Francois (IMDEA, Madrid) et Stefano Vissicchio
(UCL, Louvain-la-Neuve)

*16èmes Rencontres Francophones pour les
Aspects Algorithmiques des Télécommunications*

Le Bois-Plage-en-Ré, 3 – 5 juin 2014

- 1 Introduction
- 2 Boucles transitoires
- 3 Reconfiguration d'un routeur
- 4 Conclusion

Contexte

- Routage intra-domaine dans les réseaux d'opérateurs;
 - Protocoles à état des liens (OSPF, IS-IS)
- Changements topologiques fréquents;
 - Opérations de maintenance sur les liens ou noeuds;
 - Ingénierie de trafic (modification de poids);
 - Pannes
- Autant de périodes de convergence;
 - Routage transitoirement inconsistant;
 - Perturbations du trafic.

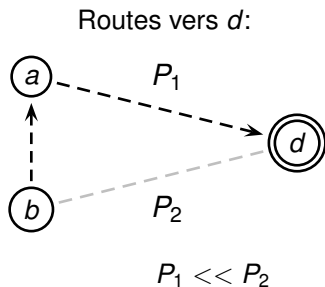
- 1 Introduction
- 2 Boucles transitoires**
- 3 Reconfiguration d'un routeur
- 4 Conclusion

Comment peuvent-elles apparaître ?

Ordre de mise à jour des routeurs *non contrôlé*
(dépend de la propagation des messages de signalisation et du temps de mise à jour des tables de routage)

Exemple:

- Initialement, a et b atteignent d par a ;

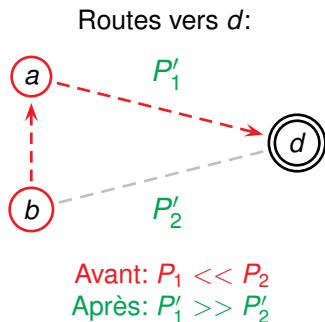


Comment peuvent-elles apparaître ?

Ordre de mise à jour des routeurs *non contrôlé*
(dépend de la propagation des messages de signalisation et du temps de mise à jour des tables de routage)

Exemple:

- Initialement, a et b atteignent d par a ;
- Un changement survient sur le réseau;
Le chemin par b est plus intéressant, même pour a ;

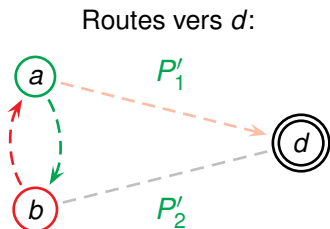


Comment peuvent-elles apparaître ?

Ordre de mise à jour des routeurs *non contrôlé*
(dépend de la propagation des messages de signalisation et du temps de mise à jour des tables de routage)

Exemple:

- Initialement, a et b atteignent d par a ;
- Un changement survient sur le réseau;
Le chemin par b est plus intéressant, même pour a ;
- Si a se met à jour en premier et **envoie son trafic pour d via b** , **alors que b passe toujours par a** ;



Avant: $P_1 \ll P_2$

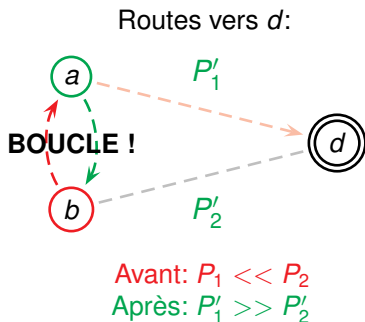
Après: $P'_1 \gg P'_2$

Comment peuvent-elles apparaître ?

Ordre de mise à jour des routeurs *non contrôlé*
(dépend de la propagation des messages de signalisation et du temps de mise à jour des tables de routage)

Exemple:

- Initialement, a et b atteignent d par a ;
- Un changement survient sur le réseau;
Le chemin par b est plus intéressant, même pour a ;
- Si a se met à jour en premier et **envoie son trafic pour d via b** , **alors que b passe toujours par a** ;
- Une **boucle transitoire** apparaît sur (a, b) ;

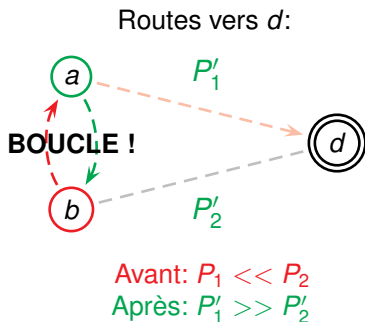


Comment peuvent-elles apparaître ?

Ordre de mise à jour des routeurs *non contrôlé*
(dépend de la propagation des messages de signalisation et du temps de mise à jour des tables de routage)

Exemple:

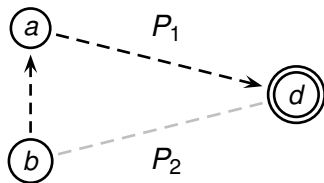
- Initialement, a et b atteignent d par a ;
- Un changement survient sur le réseau;
Le chemin par b est plus intéressant, même pour a ;
- Si a se met à jour en premier et **envoie son trafic pour d via b** , **alors que b passe toujours par a** ;
- Une **boucle transitoire** apparaît sur (a, b) ;
 - ▷ Augmentation des délais;
 - ▷ Perte de paquets.



Comment les prévenir ?

Forcer les routeurs à se mettre à jour dans le *bon* ordre.

- Initialement, a et b atteignent d par a ;

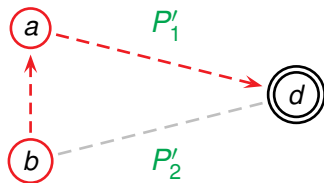


$$P_1 + w(b, a) < P_2$$

Comment les prévenir ?

Forcer les routeurs à se mettre à jour dans le *bon* ordre.

- Initialement, a et b atteignent d par a ;
- Le même changement survient;



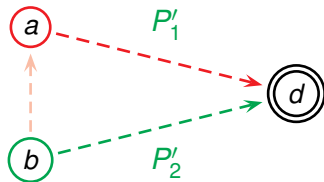
Avant: $P_1 + w(b, a) < P_2$

Après: $P'_1 > w(a, b) + P'_2$

Comment les prévenir ?

Forcer les routeurs à se mettre à jour dans le *bon* ordre.

- Initialement, a et b atteignent d par a ;
- Le même changement survient;
- Mais b se met à jour en premier;



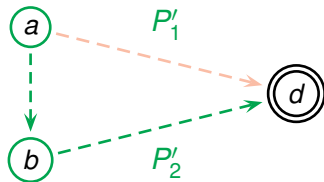
Avant: $P_1 + w(b, a) < P_2$

Après: $P'_1 > w(a, b) + P'_2$

Comment les prévenir ?

Forcer les routeurs à se mettre à jour dans le *bon* ordre.

- Initialement, a et b atteignent d par a ;
- Le même changement survient;
- Mais b se met à jour en premier;
- Puis a , et aucune boucle n'apparaît.



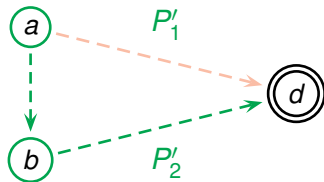
Avant: $P_1 + w(b, a) < P_2$

Après: $P'_1 > w(a, b) + P'_2$

Comment les prévenir ?

Forcer les routeurs à se mettre à jour dans le *bon* ordre.

- Initialement, a et b atteignent d par a ;
- Le même changement survient;
- Mais b se met à jour en premier;
- Puis a , et aucune boucle n'apparaît.



Avant: $P_1 + w(b, a) < P_2$

Après: $P'_1 > w(a, b) + P'_2$

Plusieurs approches possibles.

État de l'art

- Ordered FIB [*INFOCOM'05, TON'07*]
 - Ordonnancement explicite des mises à jour;
 - Nécessite des extensions de protocoles;
 - Pas de déploiement incrémental possible;
- IGP migration [*SIGCOMM'12*]
 - Prévu pour la migration de tout le réseau;
 - Nécessite de maintenir deux plans de contrôle;
 - Surcout important pour des modifications mineures;
- Metric increment - Link shut [*INFOCOM'07, TON'13*]
 - Modification progressive du poids des liens;
 - Adapté pour des changements sur un unique lien;

État de l'art

- Ordered FIB [*INFOCOM'05, TON'07*]
 - Ordonnancement explicite des mises à jour;
 - Nécessite des extensions de protocoles;
 - Pas de déploiement incrémental possible;
- IGP migration [*SIGCOMM'12*]
 - Prévu pour la migration de tout le réseau;
 - Nécessite de maintenir deux plans de contrôle;
 - Surcout important pour des modifications mineures;
- Metric increment - Link shut [*INFOCOM'07, TON'13*]
 - Modification progressive du poids des liens;
 - Adapté pour des changements sur un unique lien;
 - **Extension pour la reconfiguration de routeurs.**

Mises à jour progressives

Idée

Diviser la modification en une séquence de mises à jour *sans boucle*.

Objectif

Calculer une séquence de mises à jour intermédiaires de sorte qu'aucune boucle n'apparaisse entre deux mises à jour consécutives.

Défis

- Séquence de taille minimale (impact opérationnel);
- Algorithme efficace (calcul par les routeurs).

Illustration: incréments de distance

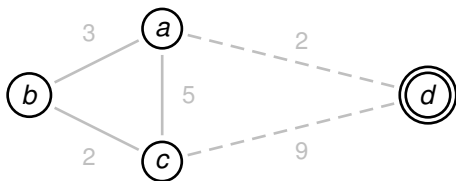


Illustration: incréments de distance

- Initialement, a , b et c atteignent d via a .

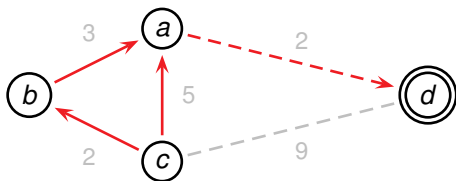


Illustration: incréments de distance

- Initialement, a , b et c atteignent d via a .
- Si un changement sur le chemin $P(a, d)$ augmente son coût à 50...

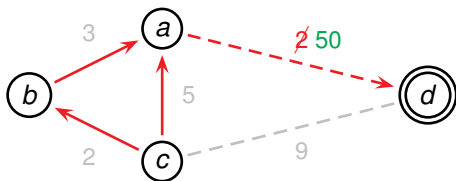


Illustration: incréments de distance

- Initialement, a , b et c atteignent d via a .
- Si un changement sur le chemin $P(a, d)$ augmente son coût à 50, les trois routeurs passeront par c ...

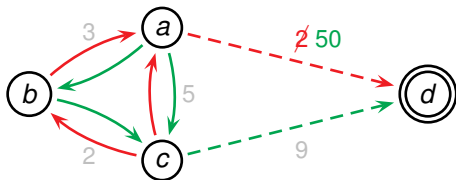


Illustration: incréments de distance

- Initialement, a , b et c atteignent d via a .
- Si un changement sur le chemin $P(a, d)$ augmente son coût à 50, les trois routeurs passeront par c et des boucles pourraient apparaître.

Avec des incréments
intermédiaires:

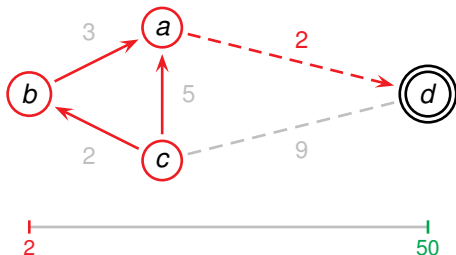


Illustration: incréments de distance

- Initialement, a , b et c atteignent d via a .
- Si un changement sur le chemin $P(a, d)$ augmente son coût à 50, les trois routeurs passeront par c et des boucles pourraient apparaître.

Avec des incréments intermédiaires:

- c sera mis à jour d'abord;

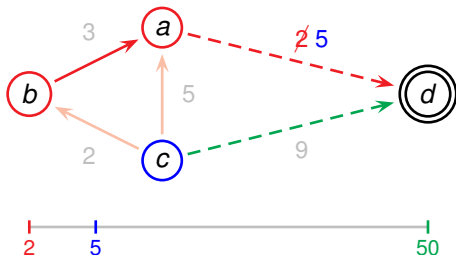


Illustration: incréments de distance

- Initialement, a , b et c atteignent d via a .
- Si un changement sur le chemin $P(a, d)$ augmente son coût à 50, les trois routeurs passeront par c et des boucles pourraient apparaître.

Avec des incréments intermédiaires:

- c sera mis à jour d'abord;
- Puis b ,

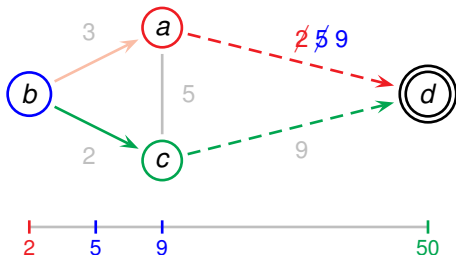


Illustration: incréments de distance

- Initialement, a , b et c atteignent d via a .
- Si un changement sur le chemin $P(a, d)$ augmente son coût à 50, les trois routeurs passeront par c et des boucles pourraient apparaître.

Avec des incréments intermédiaires:

- c sera mis à jour d'abord;
- Puis b , et enfin a ;

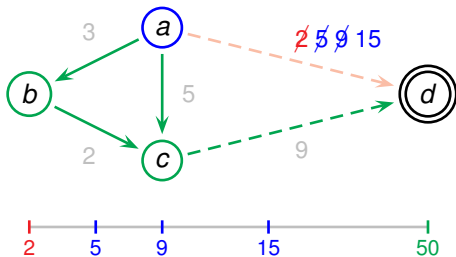


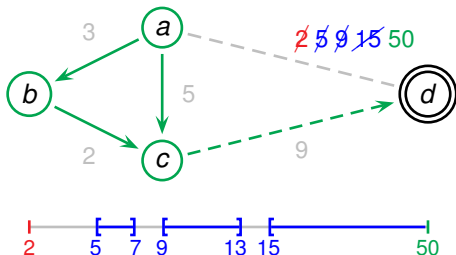
Illustration: incréments de distance

- Initialement, a , b et c atteignent d via a .
- Si un changement sur le chemin $P(a, d)$ augmente son coût à 50, les trois routeurs passeront par c et des boucles pourraient apparaître.

Avec des incréments
intermédiaires:

- c sera mis à jour d'abord;
- Puis b , et enfin a ;

Transition sans boucle vers 50
pour la destination d .



- 1 Introduction
- 2 Boucles transitoires
- 3 Reconfiguration d'un routeur**
- 4 Conclusion

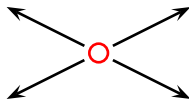
Le problème

Objectif : permettre l'ajout ou la suppression d'un routeur entier sans créer de perturbations.

Le problème

Objectif : permettre l'ajout ou la suppression d'un routeur entier sans créer de perturbations.

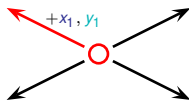
- Solution simple: couper les liens un par un
 - ▷ Nombre d'étapes intermédiaires proportionnel au degré du noeud



Le problème

Objectif : permettre l'ajout ou la suppression d'un routeur entier sans créer de perturbations.

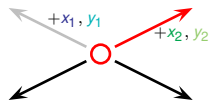
- Solution simple: couper les liens un par un
 - ▷ Nombre d'étapes intermédiaires proportionnel au degré du noeud



Le problème

Objectif : permettre l'ajout ou la suppression d'un routeur entier sans créer de perturbations.

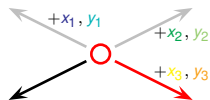
- Solution simple: couper les liens un par un
 - ▷ Nombre d'étapes intermédiaires proportionnel au degré du noeud



Le problème

Objectif : permettre l'ajout ou la suppression d'un routeur entier sans créer de perturbations.

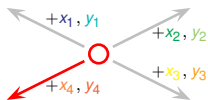
- Solution simple: couper les liens un par un
 - ▷ Nombre d'étapes intermédiaires proportionnel au degré du noeud



Le problème

Objectif : permettre l'ajout ou la suppression d'un routeur entier sans créer de perturbations.

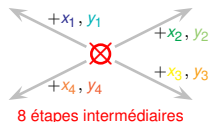
- Solution simple: couper les liens un par un
 - ▷ Nombre d'étapes intermédiaires proportionnel au degré du noeud



Le problème

Objectif : permettre l'ajout ou la suppression d'un routeur entier sans créer de perturbations.

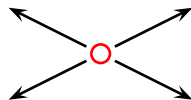
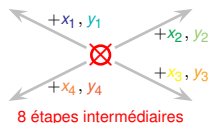
- Solution simple: couper les liens un par un
 - ▷ Nombre d'étapes intermédiaires proportionnel au degré du noeud



Le problème

Objectif : permettre l'ajout ou la suppression d'un routeur entier sans créer de perturbations.

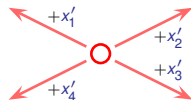
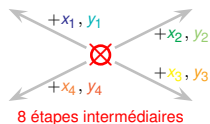
- Solution simple: couper les liens un par un
 - ▷ Nombre d'étapes intermédiaires proportionnel au degré du noeud
- Meilleure solution: modifier les poids simultanément
 - ▷ Problème multi-dimensionnel



Le problème

Objectif : permettre l'ajout ou la suppression d'un routeur entier sans créer de perturbations.

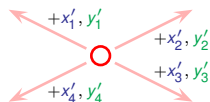
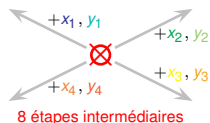
- Solution simple: couper les liens un par un
 - ▷ Nombre d'étapes intermédiaires proportionnel au degré du noeud
- Meilleure solution: modifier les poids simultanément
 - ▷ Problème multi-dimensionnel



Le problème

Objectif : permettre l'ajout ou la suppression d'un routeur entier sans créer de perturbations.

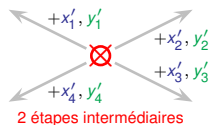
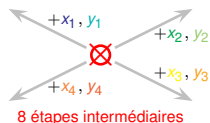
- Solution simple: couper les liens un par un
 - ▷ Nombre d'étapes intermédiaires proportionnel au degré du noeud
- Meilleure solution: modifier les poids simultanément
 - ▷ Problème multi-dimensionnel



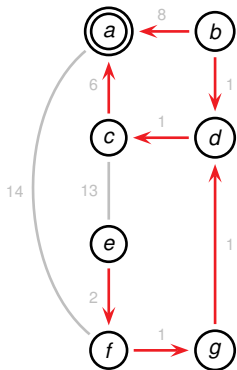
Le problème

Objectif : permettre l'ajout ou la suppression d'un routeur entier sans créer de perturbations.

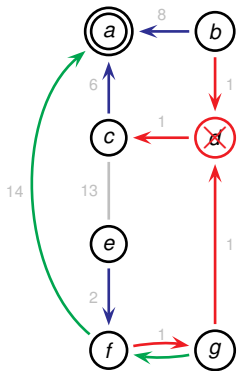
- Solution simple: couper les liens un par un
 - ▷ Nombre d'étapes intermédiaires proportionnel au degré du noeud
- Meilleure solution: modifier les poids simultanément
 - ▷ Problème multi-dimensionnel



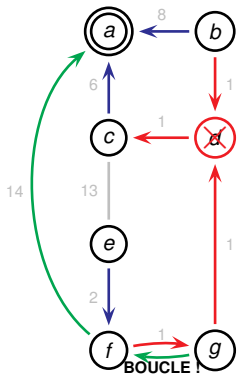
Incréments multi-dimensionnels



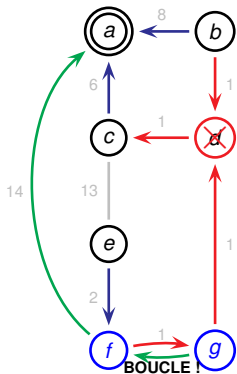
Incréments multi-dimensionnels



Incréments multi-dimensionnels



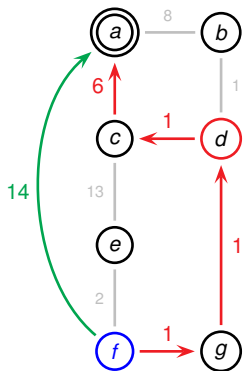
Incréments multi-dimensionnels



Vecteur d'incrément minimum tel que le noeud x utilise un **nouveau chemin**, pas via n , pour atteindre d .

$$\Delta_d^n(x)[i] = C'(x, d) - C(x, i, d)$$

Incréments multi-dimensionnels

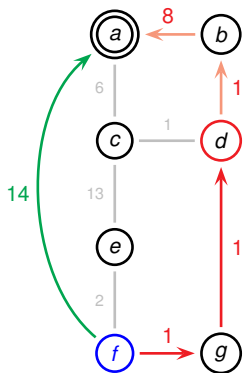


Vecteur d'incrément minimum tel que le noeud x utilise un **nouveau chemin**, pas via n , pour atteindre d .

$$\Delta_d^n(x)[i] = C'(x, d) - C(x, i, d)$$

- $\Delta_a^d(f) = (14 - (1 + 1 + 1 + 6))$

Incréments multi-dimensionnels

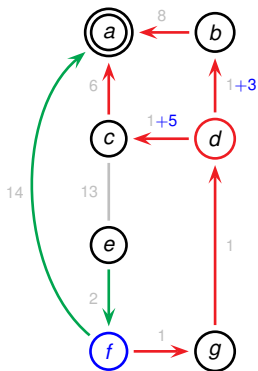


Vecteur d'incrément minimum tel que le noeud x utilise un **nouveau chemin**, pas via n , pour atteindre d .

$$\Delta_d^n(x)[i] = C'(x, d) - C(x, i, d)$$

- $$\Delta_a^d(f) = \begin{pmatrix} 14 - (1 + 1 + 1 + 6) \\ 14 - (1 + 1 + 1 + 8) \end{pmatrix}$$

Incréments multi-dimensionnels

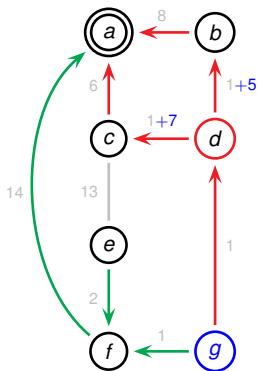


Vecteur d'incrément minimum tel que le noeud x utilise un **nouveau chemin**, pas via n , pour atteindre d .

$$\Delta_d^n(x)[i] = C'(x, d) - C(x, i, d)$$

- $$\Delta_a^d(f) = \begin{pmatrix} 14 - (1 + 1 + 1 + 6) \\ 14 - (1 + 1 + 1 + 8) \end{pmatrix} = \begin{pmatrix} 5 \\ 3 \end{pmatrix}$$

Incréments multi-dimensionnels

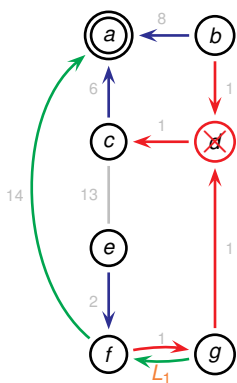


Vecteur d'incrément minimum tel que le noeud x utilise un **nouveau chemin**, pas via n , pour atteindre d .

$$\Delta_d^n(x)[i] = C'(x, d) - C(x, i, d)$$

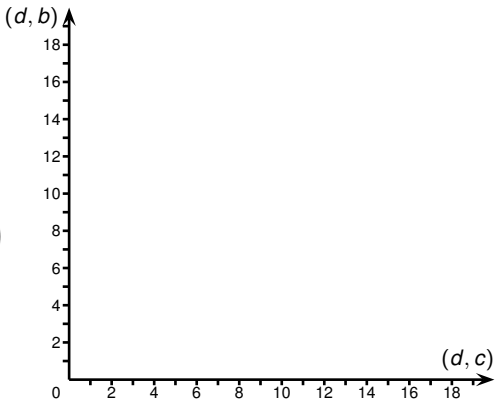
- $\Delta_a^d(f) = \begin{pmatrix} 14 - (1 + 1 + 1 + 6) \\ 14 - (1 + 1 + 1 + 8) \end{pmatrix} = \begin{pmatrix} 5 \\ 3 \end{pmatrix}$
- $\Delta_a^d(g) = \begin{pmatrix} 15 - 8 \\ 15 - 10 \end{pmatrix} = \begin{pmatrix} 7 \\ 5 \end{pmatrix}$

Modélisation des boucles sous forme de contraintes



$$\Delta_a^d(f) = \begin{pmatrix} 5 \\ 3 \end{pmatrix}$$

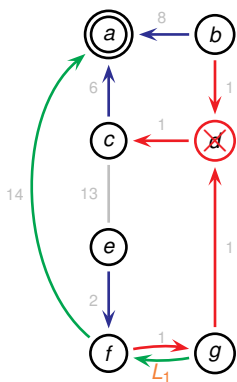
$$\Delta_a^d(g) = \begin{pmatrix} 7 \\ 5 \end{pmatrix}$$



Contrainte c associée à une boucle L .

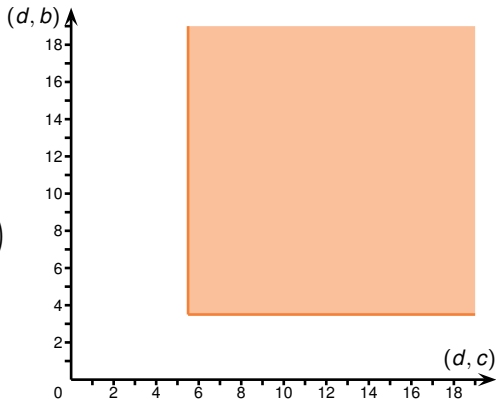
$$c := (\min_{\forall x \in L} (\Delta(x)), \max_{\forall x \in L} (\Delta(x)))$$

Modélisation des boucles sous forme de contraintes



$$\Delta_a^d(f) = \begin{pmatrix} 5 \\ 3 \end{pmatrix}$$

$$\Delta_a^d(g) = \begin{pmatrix} 7 \\ 5 \end{pmatrix}$$

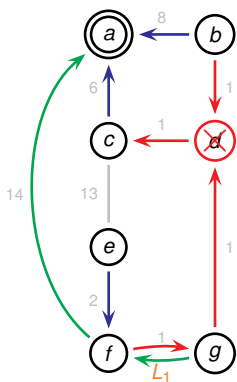


Contrainte c associée à une boucle L .

$$c := (\min_{\forall x \in L} (\Delta(x)), \max_{\forall x \in L} (\Delta(x)))$$

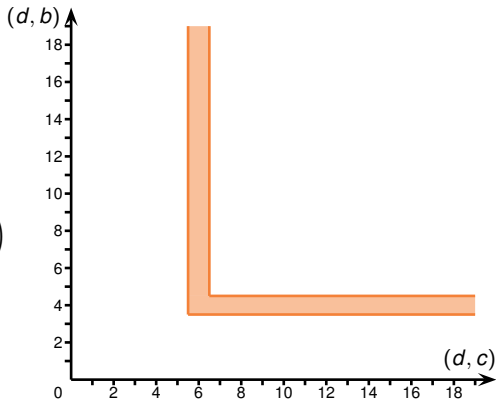
$$c_1 = \left(\begin{pmatrix} 5 \\ 3 \end{pmatrix} \right)$$

Modélisation des boucles sous forme de contraintes



$$\Delta_a^d(f) = \begin{pmatrix} 5 \\ 3 \end{pmatrix}$$

$$\Delta_a^d(g) = \begin{pmatrix} 7 \\ 5 \end{pmatrix}$$

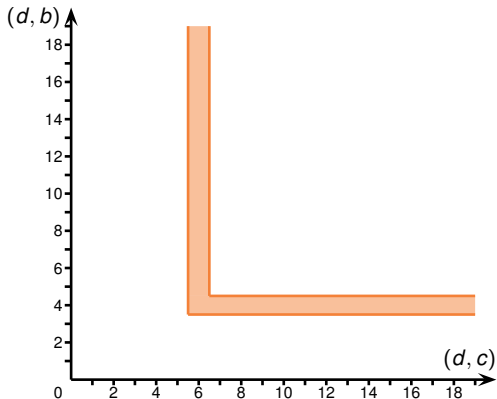
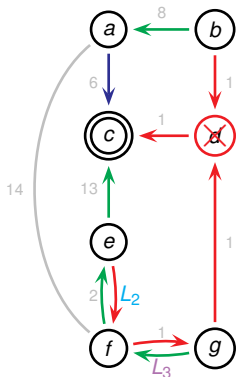


Contrainte c associée à une boucle L .

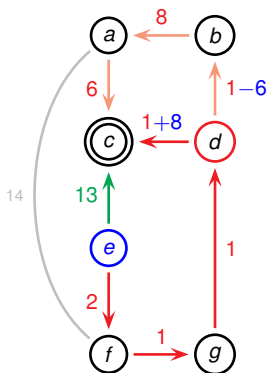
$$c := (\min_{\forall x \in L} (\Delta(x)), \max_{\forall x \in L} (\Delta(x)))$$

$$c_1 = \left(\begin{pmatrix} 5 \\ 3 \end{pmatrix}, \begin{pmatrix} 7 \\ 5 \end{pmatrix} \right)$$

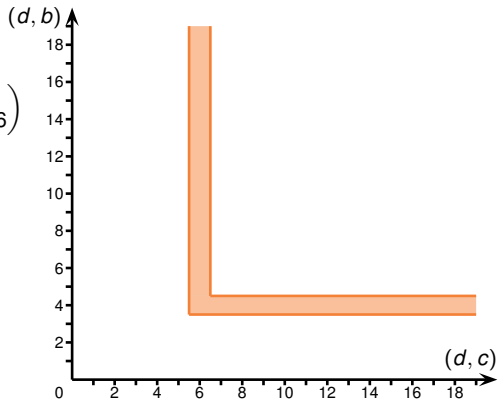
Modélisation des boucles sous forme de contraintes



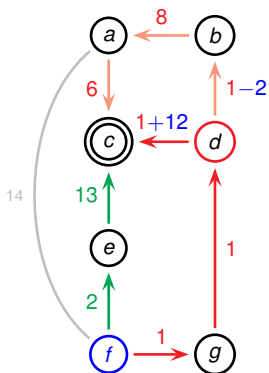
Modélisation des boucles sous forme de contraintes



$$\Delta_2^3(e) = \begin{pmatrix} 8 \\ -6 \end{pmatrix}$$

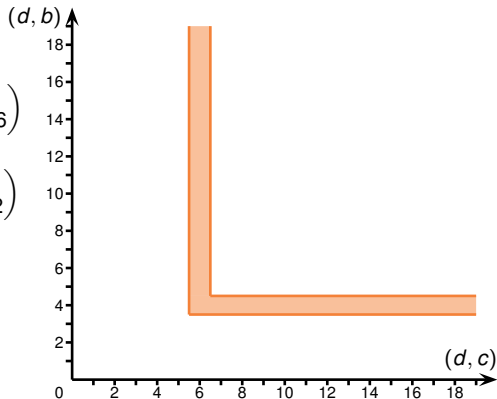


Modélisation des boucles sous forme de contraintes

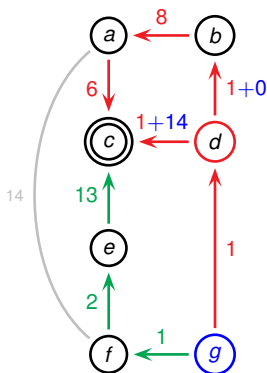


$$\Delta_2^3(e) = \begin{pmatrix} 8 \\ -6 \end{pmatrix}$$

$$\Delta_2^3(f) = \begin{pmatrix} 12 \\ -2 \end{pmatrix}$$



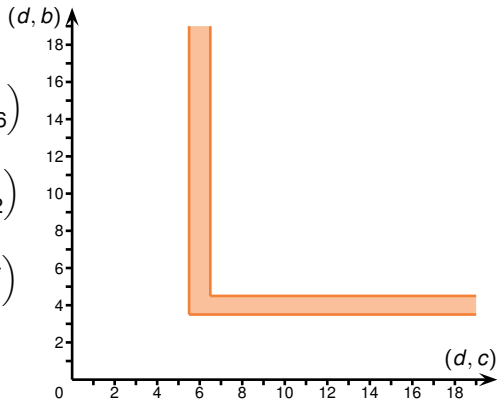
Modélisation des boucles sous forme de contraintes



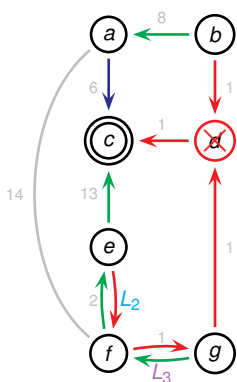
$$\Delta_2^3(e) = \begin{pmatrix} 8 \\ -6 \end{pmatrix}$$

$$\Delta_2^3(f) = \begin{pmatrix} 12 \\ -2 \end{pmatrix}$$

$$\Delta_2^3(g) = \begin{pmatrix} 14 \\ 0 \end{pmatrix}$$



Modélisation des boucles sous forme de contraintes

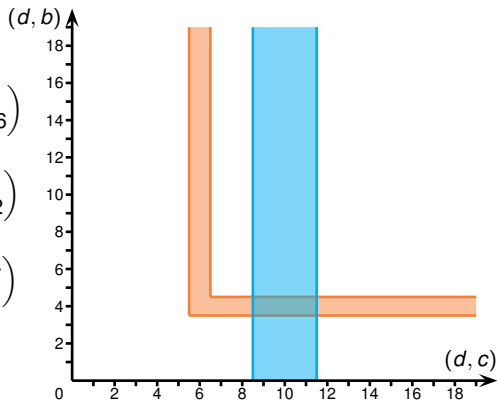


$$\Delta_2^3(e) = \begin{pmatrix} 8 \\ -6 \end{pmatrix}$$

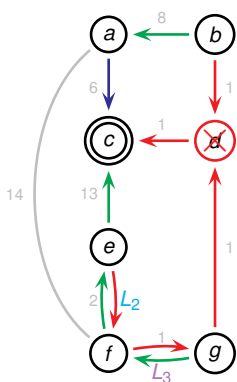
$$\Delta_2^3(f) = \begin{pmatrix} 12 \\ -2 \end{pmatrix}$$

$$\Delta_2^3(g) = \begin{pmatrix} 14 \\ 0 \end{pmatrix}$$

$$c_2 = \left(\begin{pmatrix} 8 \\ -6 \end{pmatrix}, \begin{pmatrix} 12 \\ -2 \end{pmatrix} \right)$$



Modélisation des boucles sous forme de contraintes



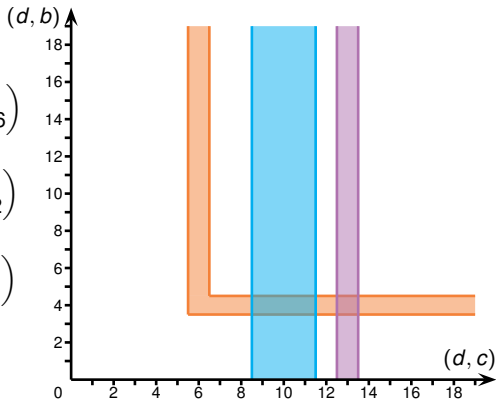
$$\Delta_2^3(e) = \begin{pmatrix} 8 \\ -6 \end{pmatrix}$$

$$\Delta_2^3(f) = \begin{pmatrix} 12 \\ -2 \end{pmatrix}$$

$$\Delta_2^3(g) = \begin{pmatrix} 14 \\ 0 \end{pmatrix}$$

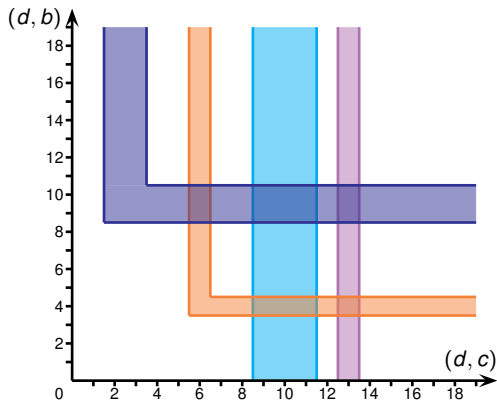
$$c_2 = \left(\begin{pmatrix} 8 \\ -6 \end{pmatrix}, \begin{pmatrix} 12 \\ -2 \end{pmatrix} \right)$$

$$c_3 = \left(\begin{pmatrix} 12 \\ -2 \end{pmatrix}, \begin{pmatrix} 14 \\ 0 \end{pmatrix} \right)$$



Modélisation des boucles sous forme de contraintes

$$C_4 = \left(\begin{pmatrix} 1 \\ 8 \end{pmatrix}, \begin{pmatrix} 4 \\ 11 \end{pmatrix} \right)$$

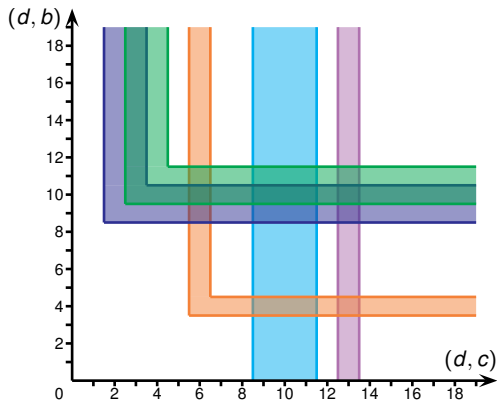


Une séquence s prévient une boucle L si et seulement si s contient au moins un vecteur satisfaisant la contrainte correspondante.

Modélisation des boucles sous forme de contraintes

$$C_4 = \left(\begin{pmatrix} 1 \\ 8 \end{pmatrix}, \begin{pmatrix} 4 \\ 11 \end{pmatrix} \right)$$

$$C_5 = \left(\begin{pmatrix} 2 \\ 9 \end{pmatrix}, \begin{pmatrix} 5 \\ 12 \end{pmatrix} \right)$$

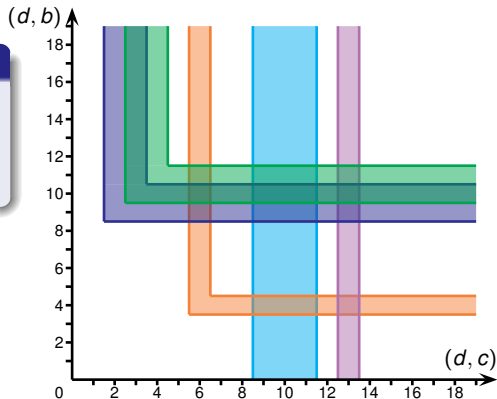


Une séquence s prévient une boucle L si et seulement si s contient au moins un vecteur satisfaisant la contrainte correspondante.

Calcul des séquences d'incrément

Greedy Backward Algorithm (GBA)

A chaque itération, extraire sur chaque index la valeur maximale parmi les bornes inférieures des contraintes restantes.

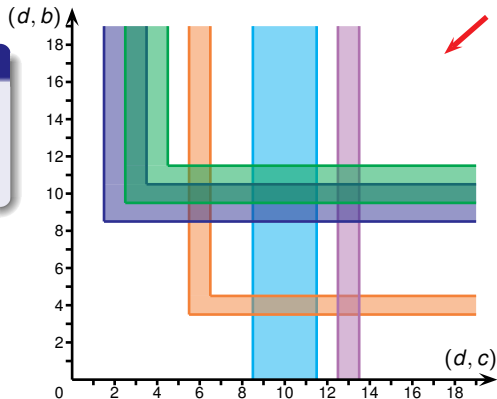


Calcul des séquences d'incrément

Greedy Backward Algorithm (GBA)

A chaque itération, extraire sur chaque index la valeur maximale parmi les bornes inférieures des contraintes restantes.

Sequence:

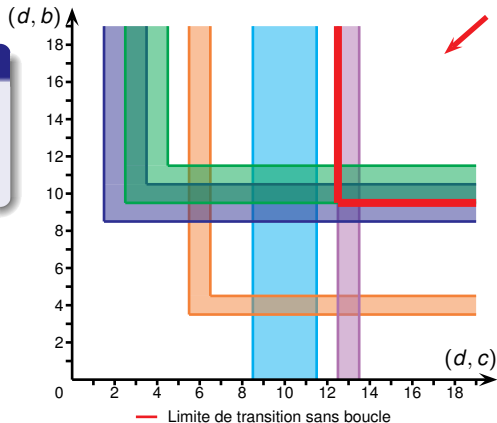


Calcul des séquences d'incrément

Greedy Backward Algorithm (GBA)

A chaque itération, extraire sur chaque index la valeur maximale parmi les bornes inférieures des contraintes restantes.

Sequence:



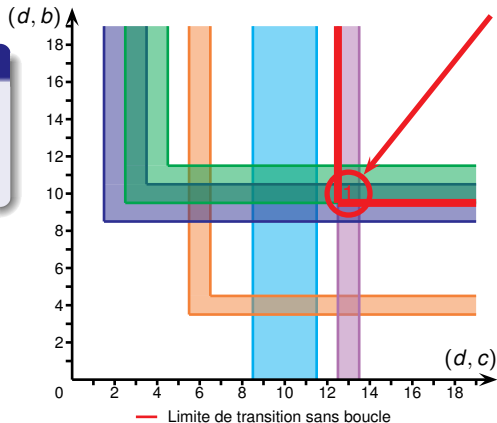
Calcul des séquences d'incrément

Greedy Backward Algorithm (GBA)

A chaque itération, extraire sur chaque index la valeur maximale parmi les bornes inférieures des contraintes restantes.

Sequence:

1 c_3 , c_4 , c_5



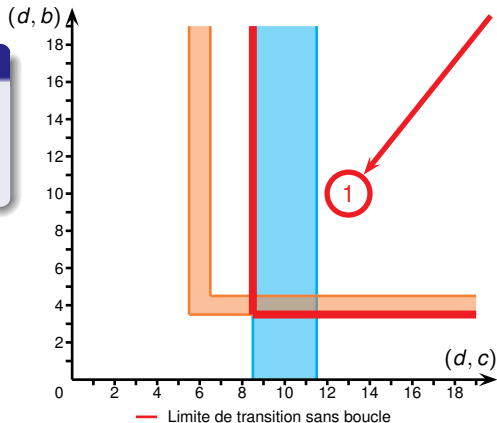
Calcul des séquences d'incrément

Greedy Backward Algorithm (GBA)

A chaque itération, extraire sur chaque index la valeur maximale parmi les bornes inférieures des contraintes restantes.

Sequence:

1 C_3 , C_4 , C_5



Calcul des séquences d'incrément

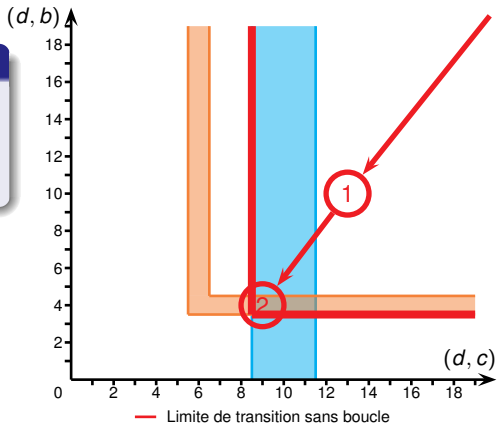
Greedy Backward Algorithm (GBA)

A chaque itération, extraire sur chaque index la valeur maximale parmi les bornes inférieures des contraintes restantes.

Sequence:

1 c_3, c_4, c_5

2 c_1, c_2



Calcul des séquences d'incrément

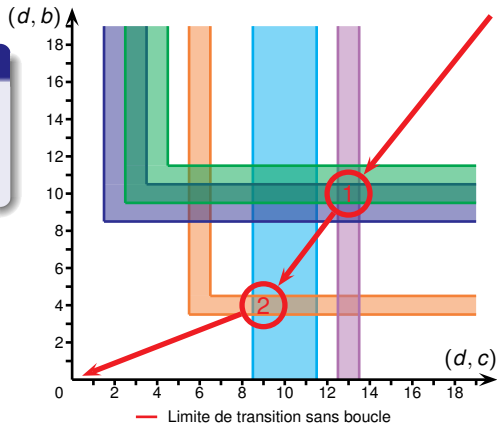
Greedy Backward Algorithm (GBA)

A chaque itération, extraire sur chaque index la valeur maximale parmi les bornes inférieures des contraintes restantes.

Sequence:

1 c_3, c_4, c_5

2 c_1, c_2



Calcul des séquences d'incrémentations

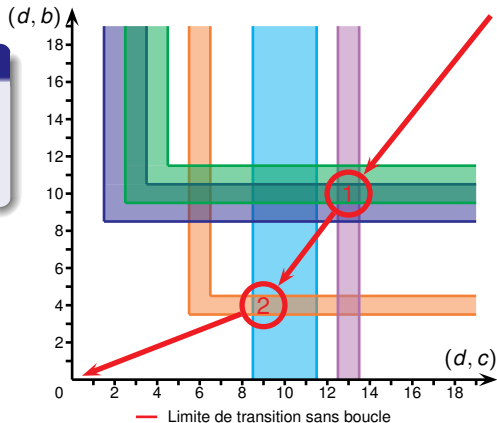
Greedy Backward Algorithm (GBA)

A chaque itération, extraire sur chaque index la valeur maximale parmi les bornes inférieures des contraintes restantes.

Sequence:

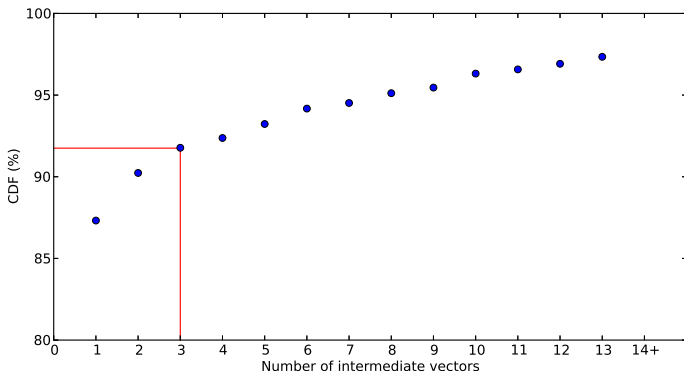
1 c_3, c_4, c_5

2 c_1, c_2



Soit un ensemble de contraintes, GBA calcule une séquence d'incrémentations intermédiaires sans boucle de longueur minimale.

Longueur des séquences sur un réseau d'opérateur



- ▶ Plus de 1000 routeurs et 4000 liens;
- ▶ 90% des noeuds requièrent moins de 3 étapes intermédiaires avant de pouvoir être retirés du réseau.

- 1 Introduction
- 2 Boucles transitoires
- 3 Reconfiguration d'un routeur
- 4 Conclusion**

Conclusion

- Solution minimale;
- Algorithme efficace;
- Approche générique;

Conclusion

- ✓ Solution minimale;
 - ▷ Plus courtes séquences possibles;
- Algorithme efficace;
- Approche générique;

Conclusion

- ✓ Solution minimale;
 - ▷ Plus courtes séquences possibles;
- ✓ Algorithme efficace;
 - ▷ Complexité polynomiale;
- Approche générique;

Conclusion

- ✓ Solution minimale;
 - ▷ Plus courtes séquences possibles;

- ✓ Algorithme efficace;
 - ▷ Complexité polynomiale;

- ✓ Approche générique;
 - ▷ Couvre le cas de la modification d'un unique lien.

Conclusion

- ✓ Solution minimale;
 - ▷ Plus courtes séquences possibles;
- ✓ Algorithme efficace;
 - ▷ Complexité polynomiale;
- ✓ Approche générique;
 - ▷ Couvre le cas de la modification d'un unique lien.

Perspectives

- ▷ Implémentation dans Quagga;
- ▷ Expérimentations sur un vrai réseau.

Merci pour votre attention.



5 Appendix

