Partition en chemins et théorème de Dilworth dans les graphes temporels

Dibyayan Chakraborty[1], Antoine Dailly[2],
Florent Foucaud[2], Ralf Klasing[3]

JGA 2023

[1] School of Computing, University of Leeds
[2] LIMOS, Clermont-Ferrand
[3] LaBRI, Bordeaux

ANR GRALMECO et TEMPOGRAL

Introduction : le théorème de Dilworth

Théorème [Dilworth, 1950]

La taille minimale d'une partition en chaines d'un poset fini est égal à la taille maximale d'une antichaine de ce poset.

Introduction : le théorème de Dilworth

Théorème [Dilworth, 1950]

La taille minimale d'une partition en chemins d'un DAG transitif est égal à la taille maximale d'une antichaine de ce DAG.

Reformulation en graphe...

Introduction : le théorème de Dilworth

Théorème [Dilworth, 1950]

La taille minimale d'une couverture par chemins d'un DAG est égal à la taille maximale d'une antichaine de ce DAG.

Reformulation en graphe...
... et en couverture.

Introduction : le théorème de Dilworth

Théorème [Dilworth, 1950]

La taille minimale d'une couverture par chemins d'un DAG est égal à la taille maximale d'une antichaine de ce DAG.

Reformulation en graphe...
... et en couverture.

Algorithmes :

- Preuve algorithmique (temps polynomial) [Fulkerson, 1956]

Introduction : le théorème de Dilworth

Théorème [Dilworth, 1950]

La taille minimale d'une couverture par chemins d'un DAG est égal à la taille maximale d'une antichaine de ce DAG.

Reformulation en graphe...
... et en couverture.

Algorithmes :

- Preuve algorithmique (temps polynomial) [Fulkerson, 1956]
- Constamment amélioré depuis, jusqu'à quasi-linéaire [Caceres, ICALP 2023]

Introduction : le théorème de Dilworth

Théorème [Dilworth, 1950]

La taille minimale d'une couverture par chemins d'un DAG est égal à la taille maximale d'une antichaine de ce DAG.

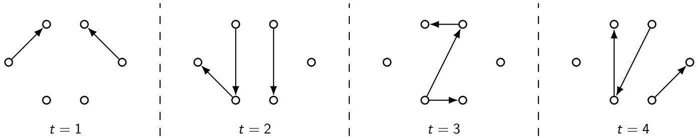
Reformulation en graphe...
... et en couverture.

Algorithmes :

- Preuve algorithmique (temps polynomial) [Fulkerson, 1956]
- Constamment amélioré depuis, jusqu'à quasi-linéaire [Caceres, ICALP 2023]
- NP-complet sur les graphes généraux

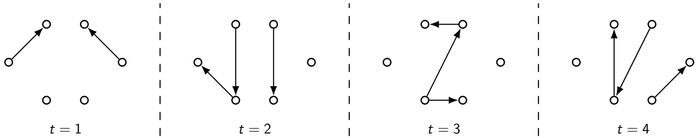
Introduction : (di)graphes temporels

$\mathcal{D} = (V, A_1, A_2, \ldots, A_k)$ [Ferreira & Viennot, 2002]

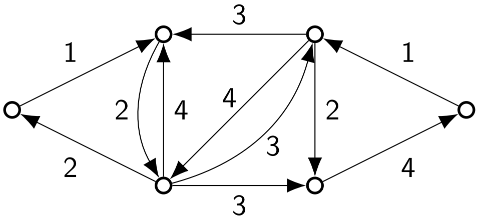


Introduction : (di)graphes temporels

$\mathcal{D} = (V, A_1, A_2, \ldots, A_k)$ [Ferreira & Viennot, 2002]

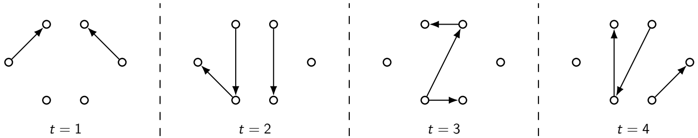


$\mathcal{D} = (V, A, \lambda)$ [Kempe, Kleinberg & Kumar, 2000]

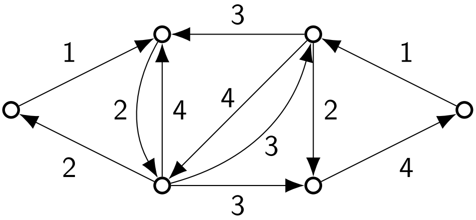


Introduction : (di)graphes temporels

$\mathcal{D} = (V, A_1, A_2, \ldots, A_k)$ [Ferreira & Viennot, 2002]



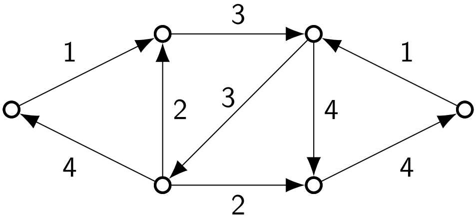
$\mathcal{D} = (V, A, \lambda)$ [Kempe, Kleinberg & Kumar, 2000]



Beaucoup de travaux en algorithmes distribués, réseaux dynamiques (de transport, sociaux, biologiques...), etc. Plus récemment, intérêt de la communauté d'algorithmes de graphes.

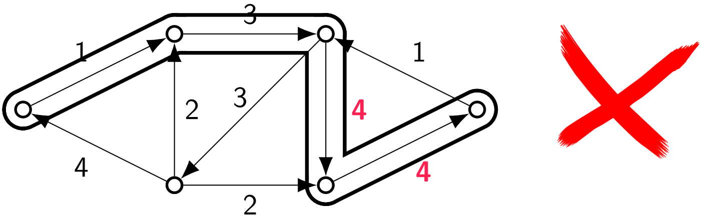
Quelques définitions

- Le (di)graphe sous-jacent est appelé l'empreinte. Un DAG (resp. arbre...) temporel est un (di)graphe temporel dont l'empreinte est un DAG (resp. arbre...).



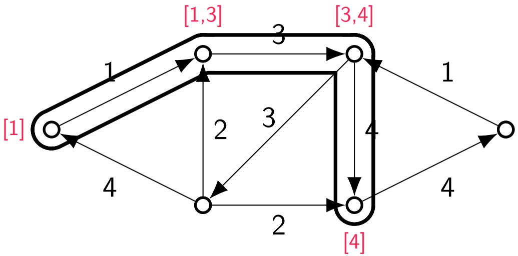
Quelques définitions

- Le (di)graphe sous-jacent est appelé l'empreinte. Un DAG (resp. arbre...) temporel est un (di)graphe temporel dont l'empreinte est un DAG (resp. arbre...).
- Chemin (dirigé) temporel : labels strictement croissants.



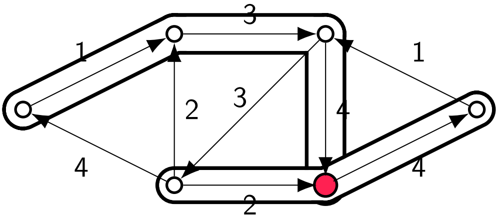
Quelques définitions

- Le (di)graphe sous-jacent est appelé l'empreinte. Un DAG (resp. arbre...) temporel est un (di)graphe temporel dont l'empreinte est un DAG (resp. arbre...).
- Chemin (dirigé) temporel : labels strictement croissants.
- Un chemin temporel occupe un sommet dans l'intervalle $[t_1, t_2]$ s'il y arrive en t_1 et en repart en t_2.



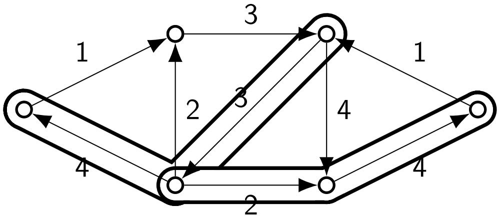
Quelques définitions

- Le (di)graphe sous-jacent est appelé l'empreinte. Un DAG (resp. arbre...) temporel est un (di)graphe temporel dont l'empreinte est un DAG (resp. arbre...).
- Chemin (dirigé) temporel : labels strictement croissants.
- Un chemin temporel occupe un sommet dans l'intervalle $[t_1, t_2]$ s'il y arrive en t_1 et en repart en t_2.
- Deux chemins temporels s'intersectent s'ils occupent un même sommet à des intervalles non-disjoints.



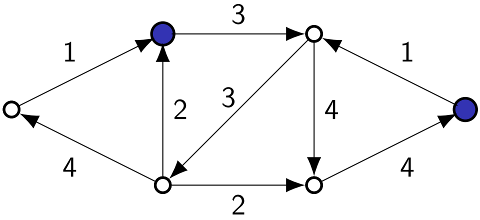
Quelques définitions

- Le (di)graphe sous-jacent est appelé l'empreinte. Un DAG (resp. arbre...) temporel est un (di)graphe temporel dont l'empreinte est un DAG (resp. arbre...).
- Chemin (dirigé) temporel : labels strictement croissants.
- Un chemin temporel occupe un sommet dans l'intervalle $[t_1, t_2]$ s'il y arrive en t_1 et en repart en t_2.
- Deux chemins temporels s'intersectent s'ils occupent un même sommet à des intervalles non-disjoints. Ils sont temporellement disjoints s'ils ne s'intersectent pas.



Quelques définitions

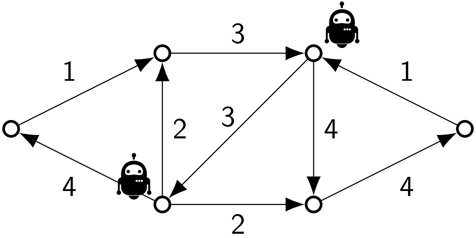
- Le (di)graphe sous-jacent est appelé l'empreinte. Un DAG (resp. arbre...) temporel est un (di)graphe temporel dont l'empreinte est un DAG (resp. arbre...).
- Chemin (dirigé) temporel : labels strictement croissants.
- Un chemin temporel occupe un sommet dans l'intervalle $[t_1, t_2]$ s'il y arrive en t_1 et en repart en t_2.
- Deux chemins temporels s'intersectent s'ils occupent un même sommet à des intervalles non-disjoints. Ils sont temporellement disjoints s'ils ne s'intersectent pas.
- Une antichaine temporelle est un ensemble de sommets qui n'ont pas de chemins temporels entre eux deux à deux.



Notre intérêt : les chemins temporellement disjoints

Historique

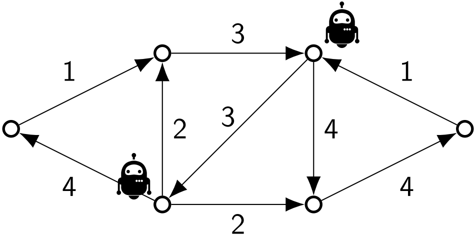
- Divers travaux sur les chemins dans les graphes temporels
- Les chemins temporellement disjoints modélisent le Multi Agent Path Finding [Stern *et al.*, 2019] dynamique



Notre intérêt : les chemins temporellement disjoints

Historique

- Divers travaux sur les chemins dans les graphes temporels
- Les chemins temporellement disjoints modélisent le Multi Agent Path Finding [Stern *et al.*, 2019] dynamique



- Marches Temporellement Disjointes : W[1]-dur et XP (nombre de marches) [Klobas *et al.*, IJCAI 2021]
- Chemins Temporellement Disjoints : NP-complet et W[1]-dur (nombre de sommets) sur les étoiles temporelles [Kunz, Molter & Zehavi, IJCAI 2023]

Un théorème de Dilworth... temporel ?

Propriété de Dilworth

La taille minimale d'une partition en chemins est égale à la taille maximale d'une antichaine.

Un théorème de Dilworth... temporel ?

Propriété de Dilworth temporelle

La taille minimale d'une partition en chemins temporellement disjoints est égale à la taille maximale d'une antichaine temporelle.

Un théorème de Dilworth... temporel ?

Propriété de Dilworth temporelle

La taille minimale d'une partition en chemins temporellement disjoints est égale à la taille maximale d'une antichaine temporelle.

Deux questions :

Quelles familles de digraphes temporels ont la propriété de Dilworth ?

$\Rightarrow$ Aspect **combinatoire**

Quelle est la complexité de calculer une partition en chemins temporellement disjoints optimale ?

$\Rightarrow$ Aspect **algorithmique**

Nos résultats

Classe temporelle	Propriété de Dilworth	Algorithme
Chemins orientés	OUI	$\mathcal{O}(\ell n)$
Arbres enracinés	OUI	$\mathcal{O}(\ell n^2)$
Arbres orientés	NON	NP-complet
DAGs*	NON	NP-complet
Digraphes	NON	FPT (tw et $t_{\max}$) $2^{\mathcal{O}(\mathrm{tw}^2\, t_{\max} \log(\mathrm{tw} + t_{\max}))} n$

* planaires, subcubiques, bipartis, maille 10, $\ell = 1$, $t_{\max} = 2$

$n =$ nombre de sommets

$\ell =$ nombre de labels temporels par arc (non-triés)

$t_{\max} =$ nombre total de temps

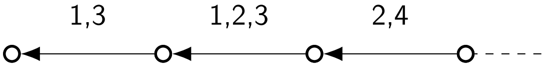
Chemins orientés temporels

Théorème [CDFK, 2023+]

Les chemins orientés temporels vérifient la propriété de Dilworth, et on peut calculer une partition en chemins temporellement disjoints de taille minimale en temps $\mathcal{O}(\ell n)$.

Algorithme

Prendre un chemin temporel maximal qui contient une feuille.



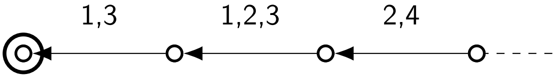
Chemins orientés temporels

Théorème [CDFK, 2023+]

Les chemins orientés temporels vérifient la propriété de Dilworth, et on peut calculer une partition en chemins temporellement disjoints de taille minimale en temps $\mathcal{O}(\ell n)$.

Algorithme

Prendre un chemin temporel maximal qui contient une feuille.



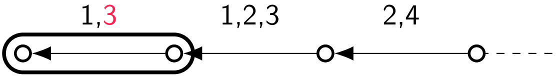
Chemins orientés temporels

Théorème [CDFK, 2023+]

Les chemins orientés temporels vérifient la propriété de Dilworth, et on peut calculer une partition en chemins temporellement disjoints de taille minimale en temps $\mathcal{O}(\ell n)$.

Algorithme

Prendre un chemin temporel maximal qui contient une feuille.



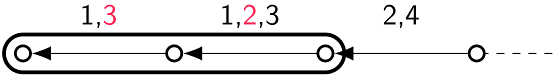
Chemins orientés temporels

Théorème [CDFK, 2023+]

Les chemins orientés temporels vérifient la propriété de Dilworth, et on peut calculer une partition en chemins temporellement disjoints de taille minimale en temps $\mathcal{O}(\ell n)$.

Algorithme

Prendre un chemin temporel maximal qui contient une feuille.



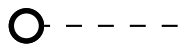
Chemins orientés temporels

Théorème [CDFK, 2023+]

Les chemins orientés temporels vérifient la propriété de Dilworth, et on peut calculer une partition en chemins temporellement disjoints de taille minimale en temps $\mathcal{O}(\ell n)$.

Algorithme

Prendre un chemin temporel maximal qui contient une feuille.

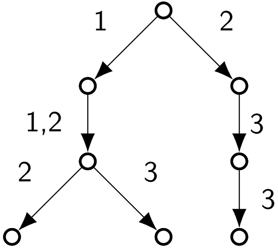


Puis itérer. Les feuilles successives forment une antichaine !

Arbres enracinés temporels

Théorème [CDFK, 2023+]

Les arbres enracinés temporels vérifient la propriété de Dilworth, et on peut calculer une partition en chemins temporellement disjoints de taille minimale en temps $\mathcal{O}(\ell n^2)$.



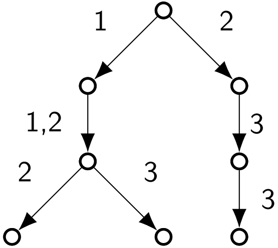
Arbres enracinés temporels

Théorème [CDFK, 2023+]

Les arbres enracinés temporels vérifient la propriété de Dilworth, et on peut calculer une partition en chemins temporellement disjoints de taille minimale en temps $\mathcal{O}(\ell n^2)$.

Algorithme

- Même principe que pour les chemins



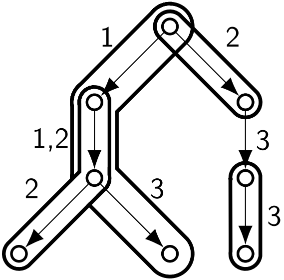
Arbres enracinés temporels

Théorème [CDFK, 2023+]

Les arbres enracinés temporels vérifient la propriété de Dilworth, et on peut calculer une partition en chemins temporellement disjoints de taille minimale en temps $\mathcal{O}(\ell n^2)$.

Algorithme

- Même principe que pour les chemins



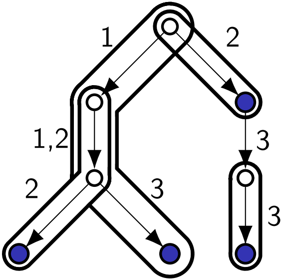
Arbres enracinés temporels

Théorème [CDFK, 2023+]

Les arbres enracinés temporels vérifient la propriété de Dilworth, et on peut calculer une partition en chemins temporellement disjoints de taille minimale en temps $\mathcal{O}(\ell n^2)$.

Algorithme

- Même principe que pour les chemins : les feuilles forment une antichaine



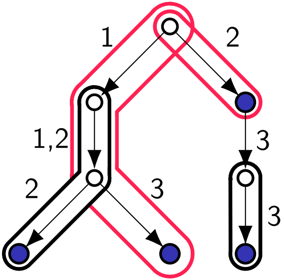
Arbres enracinés temporels

Théorème [CDFK, 2023+]

Les arbres enracinés temporels vérifient la propriété de Dilworth, et on peut calculer une partition en chemins temporellement disjoints de taille minimale en temps $\mathcal{O}(\ell n^2)$.

Algorithme

- Même principe que pour les chemins : les feuilles forment une antichaine
- En partant de la racine, résoudre les conflits : si deux chemins s'intersectent, soit ils commencent en même temps



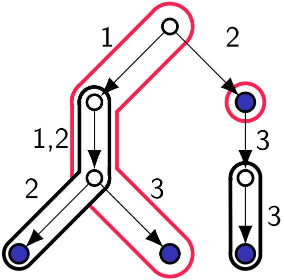
Arbres enracinés temporels

Théorème [CDFK, 2023+]

Les arbres enracinés temporels vérifient la propriété de Dilworth, et on peut calculer une partition en chemins temporellement disjoints de taille minimale en temps $\mathcal{O}(\ell n^2)$.

Algorithme

- Même principe que pour les chemins : les feuilles forment une antichaine
- En partant de la racine, résoudre les conflits : si deux chemins s'intersectent, soit ils commencent en même temps



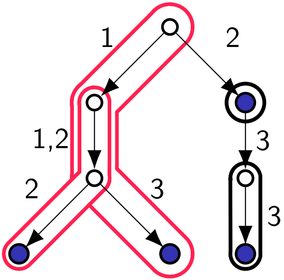
Arbres enracinés temporels

Théorème [CDFK, 2023+]

Les arbres enracinés temporels vérifient la propriété de Dilworth, et on peut calculer une partition en chemins temporellement disjoints de taille minimale en temps $\mathcal{O}(\ell n^2)$.

Algorithme

- Même principe que pour les chemins : les feuilles forment une antichaine
- En partant de la racine, résoudre les conflits : si deux chemins s'intersectent, soit ils commencent en même temps, soit un commence avant l'autre



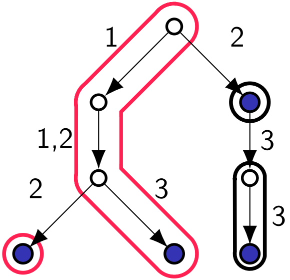
Arbres enracinés temporels

Théorème [CDFK, 2023+]

Les arbres enracinés temporels vérifient la propriété de Dilworth, et on peut calculer une partition en chemins temporellement disjoints de taille minimale en temps $\mathcal{O}(\ell n^2)$.

Algorithme

- Même principe que pour les chemins : les feuilles forment une antichaine
- En partant de la racine, résoudre les conflits : si deux chemins s'intersectent, soit ils commencent en même temps, soit un commence avant l'autre



Arbres orientés temporels : NP-complétude

Théorème [CDFK, 2023+]

La partition en chemins temporellement disjoints est NP-difficile dans les arbres orientés temporels.

Arbres orientés temporels : NP-complétude

Théorème [CDFK, 2023+]

La partition en chemins temporellement disjoints est NP-difficile dans les arbres orientés temporels.

Réduction depuis Bin Packing Unaire

Objets de taille $x_1, \ldots, x_n$; b boites de taille B

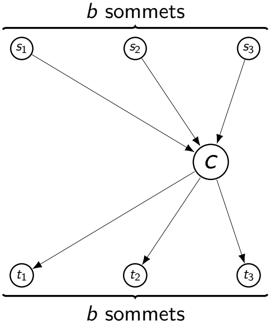
Arbres orientés temporels : NP-complétude

Théorème [CDFK, 2023+]

La partition en chemins temporellement disjoints est NP-difficile dans les arbres orientés temporels.

Réduction depuis Bin Packing Unaire

Objets de taille $x_1, \ldots, x_n$; b boites de taille B



Chaque $(s_i, t_i) \equiv$ une boite

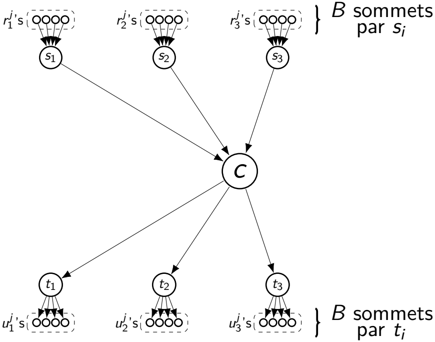
Arbres orientés temporels : NP-complétude

Théorème [CDFK, 2023+]

La partition en chemins temporellement disjoints est NP-difficile dans les arbres orientés temporels.

Réduction depuis Bin Packing Unaire

Objets de taille $x_1, \ldots, x_n$; b boites de taille B



Chaque $(s_i, t_i) \equiv$ une boite

Chaque boite doit être remplie

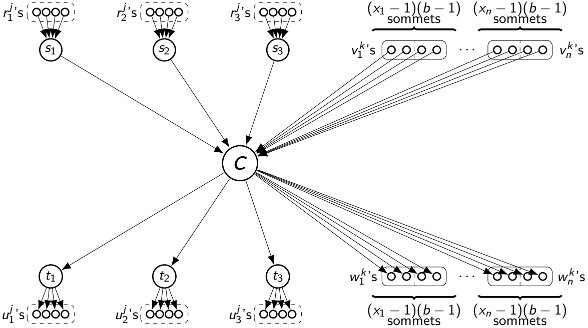
Arbres orientés temporels : NP-complétude

Théorème [CDFK, 2023+]

La partition en chemins temporellement disjoints est NP-difficile dans les arbres orientés temporels.

Réduction depuis BIN PACKING UNAIRE

Objets de taille $x_1, \ldots, x_n$; b boites de taille B



Chaque $(s_i, t_i) \equiv$ une boite

Chaque boite doit être remplie

$(v_i, w_i) \equiv$ boites non-utilisées par l'objet i

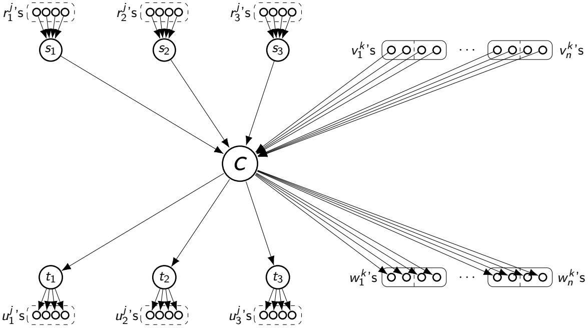
Arbres orientés temporels : NP-complétude

Théorème [CDFK, 2023+]

La partition en chemins temporellement disjoints est NP-difficile dans les arbres orientés temporels.

Réduction depuis Bin Packing Unaire

Objets de taille $x_1, \ldots, x_n$; b boites de taille B



Chaque $(s_i, t_i) \equiv$ une boite

Chaque boite doit être remplie

$(v_i, w_i) \equiv$ boites non-utilisées par l'objet i

On crée des **couches temporelles** pour chaque objet i

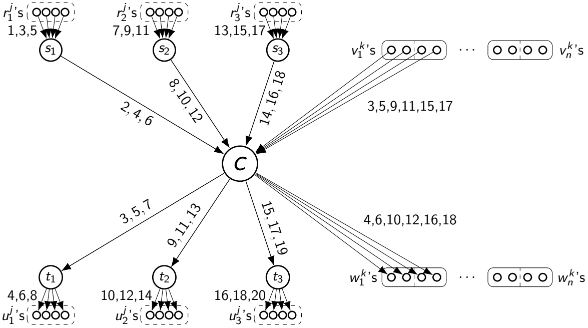
Arbres orientés temporels : NP-complétude

Théorème [CDFK, 2023+]

La partition en chemins temporellement disjoints est NP-difficile dans les arbres orientés temporels.

Réduction depuis Bin Packing Unaire

Objets de taille $x_1, \ldots, x_n$; b boites de taille B



Chaque $(s_i, t_i) \equiv$ une boite

Chaque boite doit être remplie

$(v_i, w_i) \equiv$ boites non-utilisées par l'objet i

On crée des **couches temporelles** pour chaque objet i

Ici, $x_1 = 3$

Arbres orientés et DAGs temporels : propriété de Dilworth

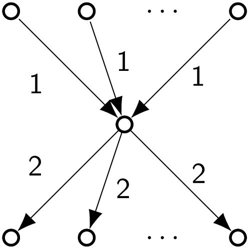
Théorème [CDFK, 2023+]

Les arbres orientés temporels ne vérifient pas la propriété de Dilworth.

Arbres orientés et DAGs temporels : propriété de Dilworth

Théorème [CDFK, 2023+]

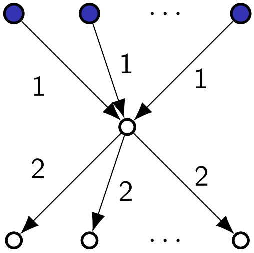
Les arbres orientés temporels ne vérifient pas la propriété de Dilworth.



Arbres orientés et DAGs temporels : propriété de Dilworth

Théorème [CDFK, 2023+]

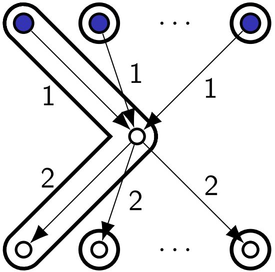
Les arbres orientés temporels ne vérifient pas la propriété de Dilworth.



Arbres orientés et DAGs temporels : propriété de Dilworth

Théorème [CDFK, 2023+]

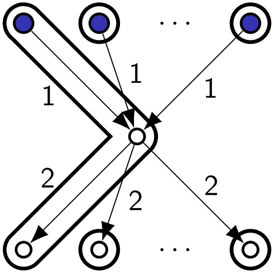
Les arbres orientés temporels ne vérifient pas la propriété de Dilworth.



Arbres orientés et DAGs temporels : propriété de Dilworth

Théorème [CDFK, 2023+]

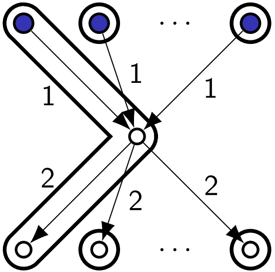
Les arbres orientés temporels ne vérifient pas la propriété de Dilworth. De plus, l'écart entre antichaine et partition est arbitrairement grand dans les DAGs temporels.

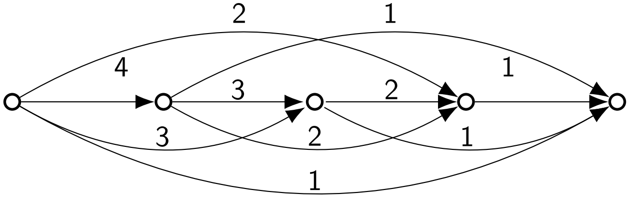


Arbres orientés et DAGs temporels : propriété de Dilworth

Théorème [CDFK, 2023+]

Les arbres orientés temporels ne vérifient pas la propriété de Dilworth. De plus, l'écart entre antichaine et partition est arbitrairement grand dans les DAGs temporels.

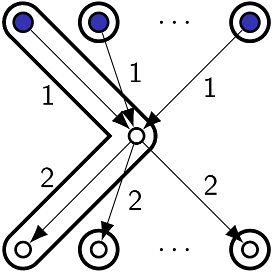


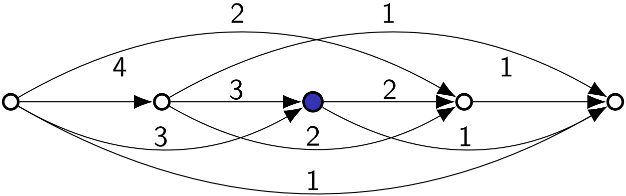


Arbres orientés et DAGs temporels : propriété de Dilworth

Théorème [CDFK, 2023+]

Les arbres orientés temporels ne vérifient pas la propriété de Dilworth. De plus, l'écart entre antichaine et partition est arbitrairement grand dans les DAGs temporels.

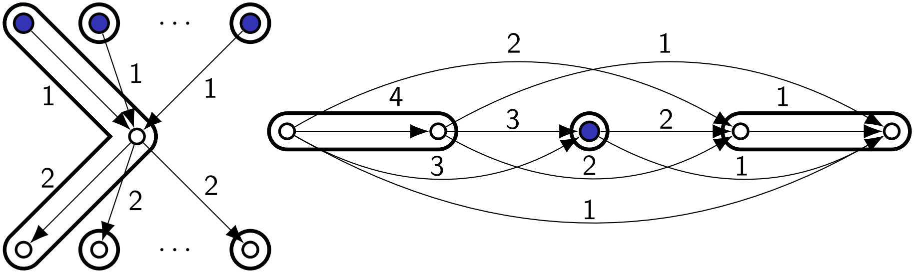




Arbres orientés et DAGs temporels : propriété de Dilworth

Théorème [CDFK, 2023+]

Les arbres orientés temporels ne vérifient pas la propriété de Dilworth. De plus, l'écart entre antichaine et partition est arbitrairement grand dans les DAGs temporels.



Conclusion, perspectives

Classe temporelle	**Propriété de Dilworth**	**Algorithme**
Chemins orientés	OUI	$\mathcal{O}(\ell n)$
Arbres enracinés	OUI	$\mathcal{O}(\ell n^2)$
Arbres orientés & DAGs	NON	NP-complet
Digraphes	NON	FPT (tw et $t_{\max}$) $2^{\mathcal{O}(\mathrm{tw}^2\, t_{\max} \log(\mathrm{tw} + t_{\max}))}\, n$

Conclusion, perspectives

Classe temporelle	**Propriété de Dilworth**	**Algorithme**
Chemins orientés	OUI	$\mathcal{O}(\ell n)$
Arbres enracinés	OUI	$\mathcal{O}(\ell n^2)$
Arbres orientés & DAGs	NON	NP-complet
Digraphes	NON	FPT (tw et $t_{\max}$) $2^{\mathcal{O}(\mathrm{tw}^2\, t_{\max} \log(\mathrm{tw} + t_{\max}))}\, n$

Perspectives

- Classes d'arbres orientés ou DAGs polynomiales ou vérifiant la propriété de Dilworth ?
- Meilleur algorithme FPT ?
- Approximation ?
- Cas non-orienté ?

Conclusion, perspectives

Classe temporelle	**Propriété de Dilworth**	**Algorithme**
Chemins orientés	OUI	$\mathcal{O}(\ell n)$
Arbres enracinés	OUI	$\mathcal{O}(\ell n^2)$
Arbres orientés & DAGs	NON	NP-complet
Digraphes	NON	FPT (tw et $t_{\max}$) $2^{\mathcal{O}(\mathrm{tw}^2\, t_{\max} \log(\mathrm{tw} + t_{\max}))}\, n$

Perspectives

- Classes d'arbres orientés ou DAGs polynomiales ou vérifiant la propriété de Dilworth ?
- Meilleur algorithme FPT ?
- Approximation ?
- Cas non-orienté ?