Master Informatique Théorique
Travaux d'Étude et de Recherche

Étude bibliographique

Algorithmique pour le consensus d'ordres

Auteurs
Himeur Areski
Picasarri-Arrieta Lucas

Encadrante
Bérard Sèverine
Institut des Sciences de l'Evolution-Montpellier

Sommaire

Sommaire

Carte génétique

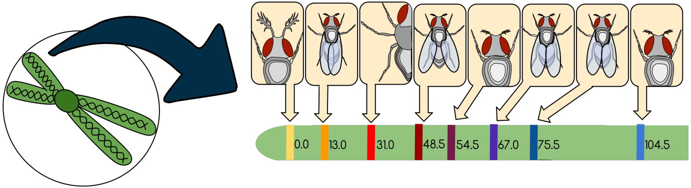

Figure – Illustration de la cartographie génétique de la drosophile [1]

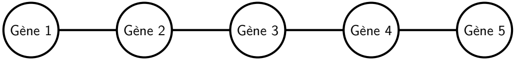

Figure – Représentation sous forme de graphe d'une carte génétique

1. Wikimedia : Drosophila Gene Linkage Map

La méthode *ARt-DeCo*

La méthode *ARt-DeCo* infère de nouvelles adjacences de gènes

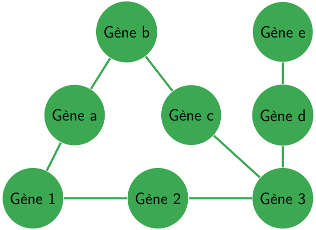

Figure – Représentation sous forme de graphe d'une *prédiction* de la méthode

Linéarisation et consensus d'ordre

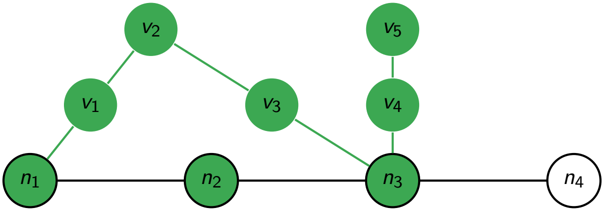

Figure – Multigraphe noir et vert

Représentation sous forme de multigraphes des deux sources :

- **sommets noirs** : locus connus d'après les cartes génétiques
- **sommets verts** : *prédictions* de la méthode *ARt-DeCo*

Linéarisation et consensus d'ordre

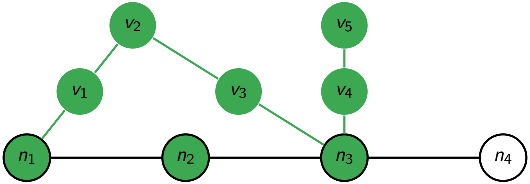

Figure – Multigraphe noir et vert

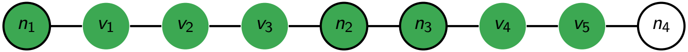

Figure – Multigraphe noir et vert linéaire [1]

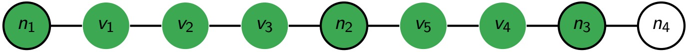

Figure – Multigraphe noir et vert linéaire [2]

Une **énumération linéaire** est une énumération des sommets du graphe vérifiant :

1. Pour chaque arête verte xy, x est à côté de y dans l'énumération.
2. Pour chaque arête noire xy, il n'y a aucun sommet noir entre x et y dans l'énumération.

Un graphe noir et vert admettant une énumération linéaire est dit **linéarisable**.

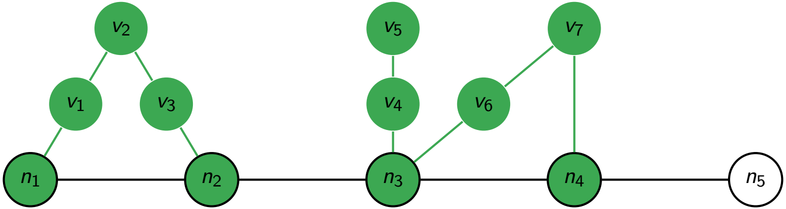

Figure – Exemple d'un graphe P-noir G_k-vert linéarisable

Énumération linéaire pour ce graphe : $n_1 v_1 v_2 v_3 n_2 v_5 v_4 n_3 v_6 v_7 n_4 n_5$

Sommaire

Problème à résoudre

- **Entrée :** un graphe noir et vert G
- **Sortie :** un ensemble *Cassure* d'arêtes vertes minimum tel que $G - \mathit{Cassure}$ soit linéarisable.

Ce problème est $\mathcal{NP}$-difficile [2]

2. L. De Mattéo, *Étude d'un problème de graphe concernant la compatibilité de données génomiques*, Université de Montpellier, 2016.

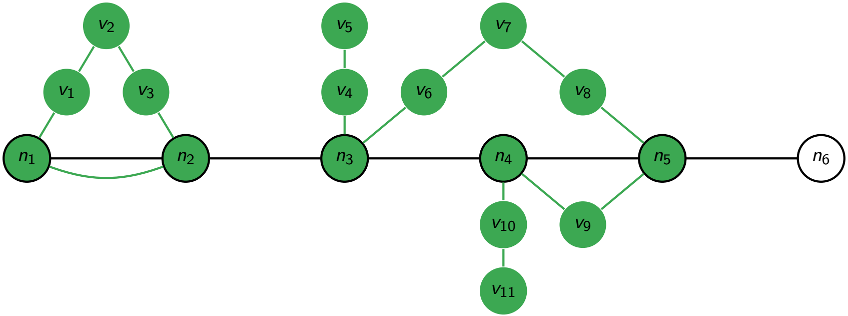

Figure – Exemple d'un graphe P-noir G_k-vert

Un cycle est un mauvais cycle s'il est la concaténation de 2 chaines P_1 et P_2 où les arêtes de P_1 sont toutes noires, et où les sommets de P_2 sont tous verts, avec $|P_1| \geq 2$.

Caractérisation

Théorème

Un graphe noir et vert est linéarisable si et seulement si :

1. *Chaque sommet admet au plus 2 voisins verts*
2. *Il n'admet aucun cycle vert (i.e. composé uniquement d'arêtes vertes)*
3. *Il n'admet aucun mauvais cycle*

L. De Mattéo, *Étude d'un problème de graphe concernant la compatibilité de données génomiques*, Université de Montpellier, 2016.

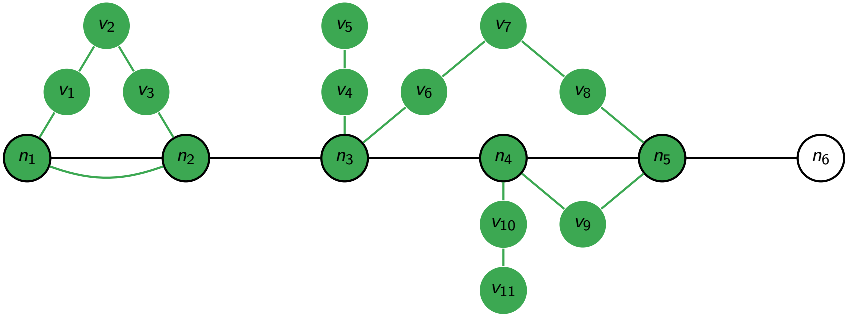

Figure – Exemple d'un graphe P-noir G_k-vert

Exemple de solution pour le problème : $Cassure = \{v_3 n_2, v_6 v_7\}$

Sommaire

Heuristique de linéarisation d'un multigraphe P-noir G_k-vert

Entrée: G un multigraphe P-noir G_k-vert

Sortie: Un ensemble *Cassure* d'arêtes vertes de G tel que $G - Cassures$ soit linéarisable.

Bijection $w : E(G_i) \rightarrow \mathbb{N}, \forall i$

pour $uv \in E(G_i), \forall i$ **faire**

    $w(uv) \leftarrow 0$

    si $|N^v(u)| > 2$ **alors**

        $w(uv) \leftarrow w(uv) + |N^v(u)| - 2$

    si $|N^v(v)| > 2$ **alors**

        $w(uv) \leftarrow w(uv) + |N^v(v)| - 2$

    $c_{vert} \leftarrow \#\{$ Cycle vert C, C contenant uv $\}$

    $w(uv) \leftarrow w(uv) + c_{vert}$

    $c_{mauvais} \leftarrow \#\{$ Mauvais cycle C, C contenant uv $\}$

    $w(uv) \leftarrow w(uv) + c_{mauvais}$

...

L. De Mattéo, *Étude d'un problème de graphe concernant la compatibilité de données génomiques*, Université de Montpellier, 2016.

Heuristique de linéarisation d'un multigraphe P-noir G_k-vert

Entrée: G un multigraphe P-noir G_k-vert
Sortie: Un ensemble *Cassure* d'arêtes vertes de G tel que $G - Cassures$ soit linéarisable.

...
Trier dans le tableau T les arêtes de w par ordre croissant.
Ensemble $Cassures \leftarrow \emptyset$
Multigraphe $G' \leftarrow P$
pour tout $uv \in T$ **faire**
  si $G' \cup \{uv\}$ est linéarisable **alors**
    $G' \leftarrow G' \cup \{uv\}$
  sinon
    $Cassures \leftarrow Cassures \cup \{uv\}$
renvoyer $Cassures$;

L. De Mattéo, *Étude d'un problème de graphe concernant la compatibilité de données génomiques*, Université de Montpellier, 2016.

Algorithme glouton

Algorithme glouton de linéarisation d'un multigraphe P-noir G_k-vert

Entrée: G un multigraphe P-noir G_k-vert
Sortie: Un ensemble $Cassure$ d'arêtes vertes de G tel que $G - Cassure$ soit linéarisable.

Ensemble $Cassure \leftarrow \emptyset$
Multigraphe $G' \leftarrow P$
pour toute arête verte uv de G **faire**
    Ajouter dans G' l'arête uv
    si G' n'est plus linéarisable **alors**
        $Cassure \leftarrow Cassure \cup \{uv\}$
        Retirer de G' l'arête uv
renvoyer $Cassure$;

Algorithme exact

Linéarisation d'un multigraphe P-noir G_k-vert

Entrée: G un multigraphe P-noir G_k-vert
Sortie: Un ensemble *Cassure* minimum d'arêtes vertes de G tel que $G - Cassure$ soit linéarisable.

```
l ← 0
Trouve ← Faux
tant que (non Trouve) faire
    pour tout F sous-ensemble des arêtes vertes de taille l faire
        si G − F est linéarisable alors
            Cassure ← F
            Trouve ← Vrai
    l ← l + 1
renvoyer Cassure ;
```

Optimisation de l'algorithme exact

Théorème

Soit G un multigraphe P-noir G_k-vert connexe, et H l'ensemble des arêtes vertes uv de G telles que :

- $deg(u) \leq 2$
- $deg(v) \leq 2$
- *u et v sont des sommets verts*

Alors il existe une solution F au problème de la linéarisation tel que $F \cap H = \emptyset$

Optimisation de l'algorithme exact

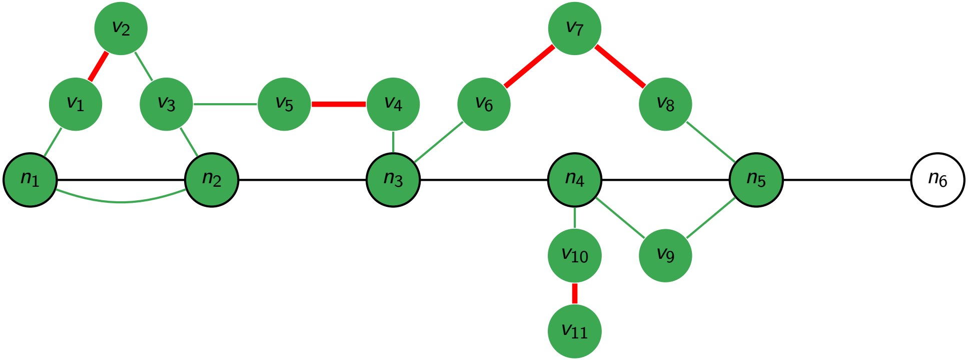

Figure – Exemple d'optimisation sur un graphe.
Les arêtes en rouges sont retirées de l'ensemble des arêtes vertes à explorer

Complexités

On peut déterminer si un graphe est linéarisable en $\mathcal{O}(n+m)$

- *Compléxité de l'heuristique :* dépend du dénombrement des cycles
- *Algorithme glouton :* polynomial, $\mathcal{O}(m \cdot (n+m))$
- *Algorithmes exacts :* exponentielle
  - *Sans optimisation :* $\mathcal{O}(2^n \cdot (n+m))$
  - *Avec optimisation :* $\mathcal{O}(2^{n-|H|} \cdot (n+m))$

Sommaire

Dénombrement des cycles pour l'heuristique

L'implémentation proposé par L. De Mattéo dénombre tous les cycles, elle est donc polynomiale.

Nous proposons une nouvelle version qui ne dénombre que les cycles simples.

Le dénombrement des cycles de taille ℓ paramétré par Δ le degré maximal du graphe en entrée est dans la classe FPT [3] :

$$\mathcal{O}(f(\Delta) \cdot |V(G)|)$$

avec $f(\Delta) = \Delta + (\ell^{-1}\Delta + \ell^{\omega-2})\Delta^{\ell-1}$

3. P.-L. Giscard, N. Kriege et R. C. Wilson, *A general purpose algorithm for counting simple cycles and simple paths of any length* Algorithmica, 2019.

Les algorithmes testés

- *Heuristique basée sur le dénombrement de cycles*
  - **[HeuTous]** Avec le dénombrement de tous les cycles verts et mauvais cycles *(Implémentation de L. De Mattéo)*
  - **[HeuSimpl]** Avec le dénombrement de tous les cycles simples
  - **[HeuSimpl10]** Avec le dénombrement des cycles simples de taille inférieure à 10
- *Algorithme exact*
  - **[Exact]** Sur toutes les arêtes vertes
  - **[ExactOpt]** Avec l'optimisation présentée
- **[Glouton]** *Algorithme glouton*

Jeux de données

Nom du graphe	Nombre de sommets	Nombre d'arêtes vertes
G1	17	15
G2	22	26
G3	34	28
G4	28	29
G5	34	32
G6	34	33
Graphe extrait du moustique	462	347
Graphe extrait du canard	1873	988

Tableau – Caractéristiques des graphes qui vont être testés

Résultats

Instance :	G1		G2		G3		G4		G5		G6	
	S.	Durée	S.	Durée	S.	Durée	S.	Durée	S.	Durée	S.	Durée
HeuTous	**2**	30ms	**9**	1s	**3**	60ms	**10**	394ms	**8**	185ms	**9**	246ms
HeuSimpl10	**2**	200ms	**9**	30s	**3**	1s	**10**	287s	**8**	73s	**8**	103s
HeuSimpl	**2**	350ms	**9**	267s	**3**	4s	**10**	8h	**8**	5h	-	-
Exact	**2**	1ms	**9**	30s	**3**	92ms	**10**	152s	**7**	41s	**8**	104s
ExactOpt	**2**	1ms	**9**	14s	**3**	30ms	**10**	4s	**7**	17s	**8**	35s
Glouton	**2**	1ms	**10**	2 ms	**3**	4ms	**10**	2ms	**8**	2ms	**8**	2ms

Tableau – Résultats des tests sur G1, G2, G3, G4, G5 et G6
S. est la taille de l'ensemble cassure de la solution.

Résultats

	Extrait du moustique		Extrait du canard	
	Taille de la solution	Durée	Taille de la solution	Durée
HeuTous	**6**	44s	**87**	57s
HeuSimpl10	**6**	1h30mn	-	-
HeuSimpl	-	-	-	-
Exact	-	-	-	-
ExactOpt	**6**	38mn	-	-
Glouton	**6**	180ms	**90**	5s

Tableau – Résultats des tests sur l'instance extrait des données génétiques du moustique et du canard

Sommaire

Conclusion

Bilan de nos travaux

- *Partie théorique :* théorèmes, algorithmes et démonstrations
- *Développement :* notre projet en C++ disponible sur Github
- *Expérimentation :* questionnement de l'heuristique et mesure de l'optimisation

Perspectives

Nos travaux motivent à poursuivre l'étude et la recherche sur ce sujet avec des retours de chercheurs en biologie et de nouvelles optimisations pour l'algorithme exact.