
RAPPORT DE STAGE

Arc-partitions de graphes orientés

MASTER INFORMATIQUE THÉORIQUE

Auteur

PICASARRI-ARRIETA Lucas

Encadrants

BESSY Stéphane

GONÇALVES Daniel



UNIVERSITÉ
DE MONTPELLIER



LIRMM

Résumé

Ce rapport présente le stage que j'effectue au sein de l'équipe ALGCO au LIRMM, co-encadré par Stéphane BESSY et Daniel GONÇALVES. Ce stage s'inscrit dans le cadre du Master d'Informatique Théorique de l'Université de Montpellier.

L'objectif du stage est de faire un état des lieux sur les questions de partitionnement des arcs d'un graphe orienté en deux sous-ensembles devant induire chacun un graphe avec une propriété définie. Dans ce contexte, ce rapport vise à définir la notion d'arc-partitions d'un graphe orienté et à présenter les résultats obtenus.

REMERCIEMENTS

Je tiens à remercier sincèrement M. BESSY et M. GONÇALVES, avec qui j'ai eu l'occasion de travailler pendant 5 mois. Échanger avec vous fut l'occasion pour moi de bénéficier de conseils avisés, tant sur le bon déroulement du stage que sur la poursuite de ma carrière. Merci pour votre bienveillance et votre soutien constant à mon égard.

Table des matières

1	Introduction	3
1.1	Motivations	3
1.2	Formalisation et définitions	4
1.3	État de l'art	6
2	Quelques problèmes polynomiaux	7
2.1	Problèmes triviaux	7
2.2	Le problème $(acyclique, acyclique)$ -arc-partition	7
2.3	D'autres cas polynomiaux	11
3	Quelques problèmes NP-complets	14
3.1	Réductions depuis le problème du cycle hamiltonien sur les graphes orientés 2-réguliers	14
3.2	Les problèmes de $(acyclique, autre)$ -arc-partition	18
3.3	Les problèmes de $(biparti, autre)$ -arc-partition	19
4	Résultats de complexité sur des classes de graphes particulières	27
4.1	Graphes acycliques	27
4.2	Condition suffisante pour $(fort, connexe)$ -arc-partition	28
4.3	Condition suffisante pour $(fort, fort)$ -arc-partition	31
5	Conclusion et synthèse	36
	Bibliographie	38
A	Classification des problèmes de partition d'arcs	39

1 Introduction

1.1 Motivations

La théorie des graphes permet de modéliser de nombreux problèmes issus du monde réel, et plus particulièrement des problèmes liés à l'informatique. Plus spécifiquement, de nombreuses questions se modélisent efficacement par des graphes orientés. C'est le cas, par exemple, de l'architecture du web, des relations sociales sur certains réseaux sociaux (comme Twitter), des planifications de tâches (où les arcs représentent la précédence entre les tâches) ou d'architectures réseaux où les communications ne s'effectuent pas à double sens. Pourtant, tant structurellement qu'algorithmiquement, les graphes orientés restent moins étudiés que les graphes classiques, et de nombreuses questions traitées dans le cas non orienté restent ouvertes sur les graphes orientés.

Pour illustrer cette différence, intéressons-nous au partitionnement des arêtes d'un graphe en k sous-ensembles, de sorte que chacun de ces sous-ensembles induise une structure connexe du graphe. Si le graphe en question modélise une architecture réseau, ce problème est lié à sa tolérance aux pannes : s'il survient moins de k pannes sur le réseau, les machines restent toutes interconnectées sans avoir à recalculer un routage pour ces communications.

Si les communications s'effectuent à double sens, ce problème revient à calculer, dans un graphe non orienté, k arbres couvrants deux-à-deux arête-disjoints. On peut calculer cela en temps polynomial, et on sait que tout graphe $2k$ -connexe admet k arbres couvrants arête-disjoints (théorème de Tutte, voir [1]).

En revanche, si les communications ne se font pas à double sens, l'architecture peut alors être modélisée par un graphe orienté, et on cherchera à calculer k ensembles d'arcs disjoints deux-à-deux tels que le graphe soit fortement connexe⁰ sur chacun de ces ensembles d'arcs. Ce problème est NP-complet [2], et on ne sait pas si des conditions de connexité assurent l'existence d'une telle partition.

Conjecture 1.1 (Bang-Jensen, Yeo [2]). Tout graphe orienté suffisamment arc-connexe¹ admet deux sous-graphes couvrants fortement connexes¹ arc-disjoints.

Remarquons que Thomassen a proposé une conjecture plus faible auparavant :

Conjecture 1.2 (Thomassen [3]). Tout graphe suffisamment arc-connexe admet une arborescence sortante¹ et une arborescence entrante¹ arc-disjointes.

Cette conjecture est plus faible que la première car tout graphe orienté fortement connexe contient une arborescence entrante et une arborescence sortante (à priori non disjointes).

La Conjecture 1.2 invite à s'intéresser au partitionnement des arcs d'un graphe orienté en k arborescences sortantes et k arborescences entrantes, toutes enracinées en un sommet r . C'est un problème NP-complet [4]. Si le graphe modélise les relations entre profils d'un réseau social, résoudre ce problème permet de mesurer l'influence du profil r sur un groupe de profils donné, et l'influence du groupe sur r . Ce problème est donc directement lié à un aspect majeur des

1. Notions définies plus loin

réseaux sociaux : les bulles de filtres et les chambres d'écho.

Comme dernier exemple, on peut vouloir partitionner les arcs d'un graphe orienté en deux sous-ensembles, de telle sorte que le graphe soit fortement connexe sur le premier ensemble, et que le deuxième ensemble induise une arborescence sortante, enracinée en un certain sommet r . Si le graphe modélise une architecture réseau, le premier ensemble est utilisé par les machines pour communiquer, et le second ensemble permet à la machine r de diffuser un message (détection d'une panne par exemple) aux autres. Ce dernier problème est également NP-complet.

Dans la suite, on s'intéressera donc au partitionnement d'arcs sur les graphes orientés en deux ensembles, de sorte que chacun de ces deux ensembles induise une propriété particulière. Ces propriétés seront entre autres liées aux exemples cités précédemment. On s'intéressera également aux Conjectures 1.1 et 1.2 sur des classes de graphes particulières.

1.2 Formalisation et définitions

Les définitions utilisées sont celles du livre [5]. Plus spécifiquement, on utilise certains résultats des chapitres 9 et 13 qui traitent d'arborescences et de partitionnement.

Définition 1.1 (fortement connexe). Un graphe D est **fortement connexe** si pour toute paire de sommets x et y , il existe un chemin orienté de x à y .

Définition 1.2 (degré). Soit $D = (V, A)$ et $v \in V$, le **degré** de v correspond au nombre d'arcs incidents (sortants ou entrants) à v , et se note $d(v)$. Similairement, $d^+(v)$ (respectivement $d^-(v)$) désigne le degré sortant de v (respectivement entrant). Si $A_1 \subseteq A$ est un sous-ensemble d'arcs de A , $d_{A_1}(v)$ désigne le nombre d'arcs incidents à v dans A_1 (de même pour $d_{A_1}^+(v)$ et $d_{A_1}^-(v)$).

Le degré minimum d'un graphe correspond au degré du sommet de plus petit degré dans D et se note $\delta(D)$ et similairement, le degré maximum d'un graphe se note $\Delta(D)$. On définit de façon analogue $\delta^+(D)$, $\delta^-(D)$, $\Delta^+(D)$ et $\Delta^-(D)$.

Définition 1.3 (arc-connexité). Un graphe orienté $D = (V, A)$ est **k -arc-connexe** si et seulement si quelque soit $B \subseteq A$, si $|B| \leq k - 1$, alors $D - B$ est fortement connexe.

Notons également que D est k -arc-connexe si et seulement si quelque soit $X \subseteq V$, si $X \neq \emptyset$ et $X \neq V$, alors il existe k arcs de X vers $V \setminus X$ (et k arcs de $V \setminus X$ vers X).

Définition 1.4 (arborescence sortante). Une **arborescence sortante** enracinée en r d'un graphe orienté $D = (V, A)$ est un sous-graphe connexe et couvrant B_r^+ de D dans lequel chaque sommet a exactement un arc entrant, à l'exception de r qui n'en a aucun.

On définit de façon similaire une **arborescence entrante**.

Définition 1.5 (acyclique). Un graphe D est **acyclique** s'il ne contient pas de cycle orienté.

Définition 1.6 (couvrant). Un graphe $D = (V, A)$ est **couvrant** si tout sommet de V a au moins un voisin (sortant ou entrant).

Définition 1.7 (hamiltonien). Un graphe $D = (V, A)$ est hamiltonien s'il admet un cycle orienté

de taille $n = |V|$. Dans la suite, en l'absence de précision, n correspond au nombre de sommets du graphe.

Définition 1.8 (graphe sous-jacent). Le graphe **sous-jacent** d'un graphe orienté D est le graphe obtenu à partir de D en supprimant l'orientation des arcs.

Définition 1.9 (multigraphe). Un **multigraphe** orienté est un graphe orienté $D = (V, A)$ muni d'un ou plusieurs arcs multiples. Dans un multigraphe, la multiplicité d'un arc désigne le nombre d'occurrences de cet arc.

Définition 1.10 (arc-partition). Une **arc-partition** d'un graphe orienté $D = (V, A)$ est une partition de ses arcs A en deux sous-ensembles A_1 et A_2 .

À partir de deux propriétés $\mathcal{P}_1, \mathcal{P}_2$ et d'un graphe orienté $D = (V, A)$, une **$(\mathcal{P}_1, \mathcal{P}_2)$ -arc-partition** de D est une arc-partition (A_1, A_2) de D telle que (V, A_i) vérifie la propriété \mathcal{P}_i pour $i \in \{1, 2\}$.

Le stage porte alors sur l'étude des problèmes de décision suivants :

Entrée : un graphe orienté $D = (V, A)$

Question : D admet-il une $(\mathcal{P}_1, \mathcal{P}_2)$ -arc-partition ?

où \mathcal{P}_1 et \mathcal{P}_2 peuvent appartenir à l'ensemble des propriétés suivantes :

- *biparti* : Être biparti,
- *connexe* : Être connexe (au sens du graphe sous-jacent),
- *fort* : Être fortement connexe,
- *acyclique* : Être acyclique,
- *acyclique couvrant* : Être acyclique et couvrant,
- *out-branching* : Contenir une arborescence sortante,
- *in-branching* : Contenir une arborescence entrante,
- *cycle factor* : Tous les sommets ont exactement un arc entrant et un arc sortant,
- $\delta^- \geq k$: Tous les sommets ont au moins k arcs entrants,
- $\delta^+ \geq k$: Tous les sommets ont au moins k arcs sortants,
- $\leq k$ arcs : Le graphe contient au plus k arcs,
- $\geq k$ arcs : Le graphe contient au moins k arcs,
- *équilibré* : Tous les sommets ont le même nombre d'arcs sortants et d'arcs entrants. Un graphe orienté équilibré peut être vu comme l'union arc-disjointe d'un ensemble de cycles orientés,
- *eulérien* : Être eulérien, c'est-à-dire être équilibré et connexe,
- *cycle* : Être un cycle.

La Figure 1 présente un exemple de graphe orienté admettant une $(\textit{fort}, \textit{in-branching})$ -arc-partition (la partie verte est fortement connexe, la partie orange contient une arborescence entrante enracinée en 1).

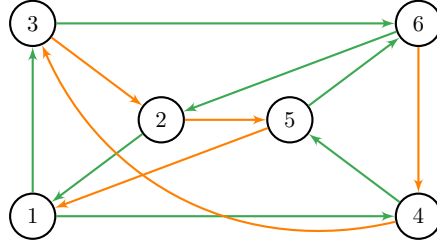


FIGURE 1 – Un graphe orienté admettant une $(\text{fort}, \text{in-branching})$ -arc-partition

Ainsi, pour chaque choix de $(\mathcal{P}_1, \mathcal{P}_2)$, on s'intéresse à la complexité du problème de décision associé. On peut également s'intéresser à la complexité du problème sur des classes de graphes particulières, et chercher des conditions qui assurent l'existence d'une telle partition.

1.3 État de l'art

L'article [6] propose une caractérisation des graphes orientés admettant k arborescences sortantes arc-disjointes, et un algorithme permettant de décider en temps polynomial si un graphe admet ou non une telle structure.

L'article [2] résout la Conjecture 1.1 sur les tournois : tout tournoi k -arc-connexe avec un degré sortant et un degré entrant supérieur à $37k$ admet k sous-graphes couvrants arc-disjointes fortement connexes. Il traite également de la Conjecture 1.2, impliquée par la Conjecture 1.1.

L'article [7] montre que les problèmes $(\text{out-branching}, \text{connexe})$ -arc-partition et $(\text{fort}, \text{connexe})$ -arc-partition sont NP-complets. Il montre plus spécifiquement que le second problème reste NP-complet sur les graphes 2-réguliers. Suite à cet article, on trouve dans l'article [8] une étude des problèmes $(\text{out-branching}, \text{in-branching})$ -arc-partition et $(\text{out-branching}, \text{connexe})$ -arc-partition exclusivement sur les graphes orientés 2-réguliers : le premier reste NP-complet, alors que le second devient polynomial.

L'article [9] étudie les questions de partitionnement spécifiquement sur les graphes n'admettant aucun indépendant de taille 3 : si un tel graphe est 2-arc-connexe, et que son degré entrant minimal est supérieur à 5, il admet une $(\text{out-branching}, \text{fort})$ -arc-partition. De plus, tout graphe orienté 2-arc-connexe sans indépendant de taille 3 sur au moins 14 sommets admet une $(\text{connexe}, \text{fort})$ -arc-partition.

Enfin, les articles [10] et [11] traitent du partitionnement de sommets. Si les résultats ne sont pas directement liés, on s'inspire néanmoins de la méthodologie utilisée.

Dans la suite de ce rapport, on présente dans un premier temps quelques problèmes de partition d'arcs polynomiaux. Ensuite, on présente quelques réductions de problèmes connus NP-complets vers des problèmes de partition d'arcs. Ensuite, on présente l'évolution de la complexité de certains de ces problèmes de partition sur des classes de graphes particulières. Enfin, on effectue dans une dernière partie une synthèse de la complexité de l'ensemble des problèmes de partition d'arcs, donc chaque résultat est justifié brièvement en Annexe A.

2 Quelques problèmes polynomiaux

2.1 Problèmes triviaux

Certains des problèmes de décision présentés dans la partie précédente peuvent être résolus de manière triviale. Fixons \mathcal{P}_1 parmi les propriétés suivantes, closes par sous-ensembles d'arcs :

- *biparti*,
- *acyclique*,
- $\leq k$ arcs,
- *équilibré*

et \mathcal{P}_2 parmi les propriétés suivantes, closes par sur-ensembles d'arcs :

- *connexe*,
- *fort*,
- *out-branching*,
- *in-branching*,
- $\delta^+ \geq k$,
- $\delta^- \geq k$,
- $\geq k$ arcs

alors $D = (V, A)$ est une instance positive au problème de décision de $(\mathcal{P}_1, \mathcal{P}_2)$ -arc-partition si et seulement si D vérifie la propriété \mathcal{P}_2 :

- * \mathcal{P}_1 est vérifiée sur les indépendants (graphes sans arc). Donc, si D vérifie la propriété \mathcal{P}_2 , (\emptyset, A) est une $(\mathcal{P}_1, \mathcal{P}_2)$ -arc-partition de D , et D est une instance positive.
- * À l'inverse, \mathcal{P}_2 est close pour l'ajout d'arc. Donc, si $D = (V, A)$ est une instance positive, et que (A_1, A_2) est une $(\mathcal{P}_1, \mathcal{P}_2)$ -arc-partition de D , alors (\emptyset, A) est également une $(\mathcal{P}_1, \mathcal{P}_2)$ -arc-partition de D (car $A_2 \subseteq A$). Ainsi, D vérifie la propriété \mathcal{P}_2 .

Puisque \mathcal{P}_2 peut être vérifiée temps polynomial, chacun de ces 28 problèmes de $(\mathcal{P}_1, \mathcal{P}_2)$ -arc-partition peut être résolu en temps polynomial.

2.2 Le problème $(acyclique, acyclique)$ -arc-partition

Tout graphe orienté $D = (V, A)$ admet une telle arc-partition. Il suffit de prendre un ordre $x_1 x_2 \dots x_n$ quelconque sur les sommets, et de fixer :

$$A_1 = \{x_i x_j \in A \mid i < j\} \text{ et } A_2 = \{x_i x_j \in A \mid i > j\}$$

(A_1, A_2) forme alors une $(acyclique, acyclique)$ -arc-partition de D (voir Figure 2).

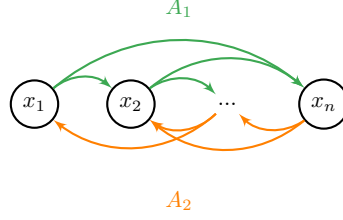


FIGURE 2 – Construction d’une $(acyclic, acyclic)$ -arc-partition

On peut montrer que le problème reste polynomial si on impose à une ou aux deux parties d’être couvrantes.

Théorème 2.1. Soit $D = (V, A)$ un graphe orienté, alors D admet une $(acyclic, acyclic \text{ couvrant})$ -arc-partition si et seulement si D n’admet pas de sommet isolé (c’est-à-dire de sommet sans arc incident).

Démonstration. Soit (A_1, A_2) la partition construite précédemment. Tant que V admet un sommet v non couvert par A_2 , on passe un arc incident à v de A_1 à A_2 . Cette opération ne crée pas de cycle dans A_1 , et elle ne crée pas non plus de cycle dans A_2 car l’arc ajouté est le seul arc incident à v . Après au plus $|V|$ étapes, (A_1, A_2) forme la partition voulue. \square

Théorème 2.2. Soit $D = (V, A)$ un graphe orienté connexe, alors D admet une $(acyclic \text{ couvrant}, acyclic \text{ couvrant})$ -arc-partition si et seulement si $\delta(D) \geq 2$ et si D ne correspond pas à l’orientation d’un cycle impair.

Démonstration. Si $D = (V, A)$ admet une telle partition, alors $\delta(D) \geq 2$ car pour tout $v \in V$, $d_{A_1}(v) \geq 1$ et $d_{A_2}(v) \geq 1$. On peut remarquer facilement que si D correspond à l’orientation d’un cycle impair, D n’admet pas de telle partition.

Supposons donc que $\delta(D) \geq 2$ et que D ne soit pas l’orientation d’un cycle impair. Si D correspond à l’orientation d’un cycle pair, on obtient la partition voulue en alternant A_1 et A_2 le long du cycle.

On peut donc supposer que D n’est pas l’orientation d’un cycle. Soit alors (A_1, A_2) une $(acyclic \text{ couvrant}, acyclic)$ -arc-partition qui minimise le nombre de sommets non couverts par A_2 (cette partition existe par le Théorème 2.2). Supposons qu’il existe $v \in V$ un sommet non couvert par A_2 , alors :

- * D ne contient aucun chemin non-orienté $P = u_0 u_1 \dots u_k$ partant de $v = u_0$, tel que $u_{k-1} u_k \in A_1$ et $d_{A_1}(u_k) > 1$, ou tel que $u_{k-1} u_k \in A_2$ et $d_{A_2}(u_k) > 1$. En effet, si D admet un tel chemin, soit P un de ces chemins de taille minimum. On remarque que $u_0 u_1 \in A_1$ et que par minimalité de P , les arcs de P alternent entre A_1 et A_2 . On peut

alors poser :

$$\begin{aligned} A'_1 &= (A_1 \cup A(P)) \setminus (A_1 \cap A(P)) \\ A'_2 &= (A_2 \cup A(P)) \setminus (A_1 \cap A(P)) \end{aligned}$$

Moins formellement, on «inverse» les arcs de A_1 et A_2 le long de P (voir Figure 3).

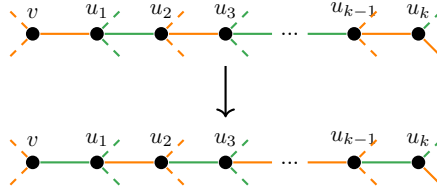


FIGURE 3 – Opération d'inversion le long de P (A_1 est en orange, A_2 en vert)

Cette opération ne crée pas de cycle dans A_1 ni dans A_2 . En effet, P étant minimum, on sait que pour tout sommet interne u_i de P , si $u_{i-1}u_i \in A_1$ alors $d_{A_1}(u_i) = 1$. De même, si $u_{i-1}u_i \in A_2$ alors $d_{A_2}(u_i) = 1$. Après application de l'opération, si $u_i u_{i+1} \in A_1$ alors $d_{A_1}(u_i) = 1$ (de même pour A_2). De plus, $u_0 u_1$ est le seul arc de A'_2 incident à u_0 (car u_0 n'était pas couvert par A_2 par hypothèse).

Chaque arc le long de P n'appartient donc à aucun cycle de A'_1 ni A'_2 , les graphes (V, A'_1) et (V, A'_2) sont donc bien acycliques. De plus, A'_1 couvre tous les sommets, et A'_2 couvre strictement plus de sommets que A_2 , car A'_2 couvre v .

Le choix de (A_1, A_2) interdit donc bien l'existence d'un tel chemin P .

- * v appartient à tous les cycles non-orientés de D . En effet, supposons que D admette un cycle C ne contenant pas v , et prenons P un chemin non-orienté de v à u un sommet quelconque de C (P existe car D est connexe par hypothèse).

Par le point précédent, on peut remarquer que C est un cycle impair et qu'à l'exception de u , tous les sommets de C ont exactement deux voisins, avec des arcs incidents dans des parties A_i différentes (sinon, en prolongeant P , on obtient un chemin avec les caractéristiques voulues, voir Figure 4).

On peut alors effectuer une opération semblable à celle du point précédent, le long de P et de C :

$$\begin{aligned} A'_1 &= (A_1 \cup A(P) \cup A(C)) \setminus (A_1 \cap (A(P) \cup A(C))) \\ A'_2 &= (A_2 \cup A(P) \cup A(C)) \setminus (A_2 \cap (A(P) \cup A(C))) \end{aligned}$$

De la même façon, (A'_1, A'_2) contredit le choix de (A_1, A_2) (voir Figure 4).

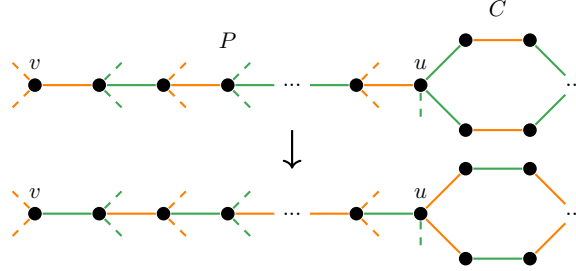


FIGURE 4 – Opération d’inversion le long de P et C (A_1 est en orange, A_2 en vert)

- * Il n’existe pas dans D deux cycles non-orientés arête-disjoints.

En effet, puisque tous les cycles passent par v , alors on peut inverser les arcs le long de l’un des deux cycles pour obtenir (A'_1, A'_2) une partition qui contredit le choix de (A_1, A_2) (voir Figure 5).

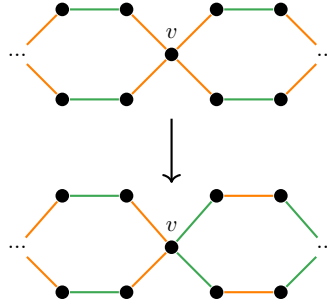


FIGURE 5 – Opération d’inversion si deux cycles arêtes-disjoints s’intersectent en v

À nouveau, notons que chacun des deux cycles est impair et que dans chacun d’eux, les sommets différents de v ont exactement 2 voisins, sinon on contredit le premier point.

- * On sait que D admet deux cycles non-orientés différents. En effet, chaque arête appartient à un cycle (parce que $\delta(D) \geq 2$, et que tous les cycles s’intersectent). Si D ne contient qu’un cycle non-orienté, alors D est un cycle non-orienté, ce qui contredit les hypothèses précédentes.

Finalement, considérons deux cycles distincts. Par les points précédents, on sait qu’ils s’intersectent sur au moins une arête, D contient donc un sommet de degré au moins 3 différent de v , appelons le x .

Puisque tous les cycles contiennent v , en prenant trois chemin maximaux issus de x , on trouve trois chemin sommet-disjoints de v à x , car si deux de ces trois chemins se rejoignent sur un sommet, on obtient un cycle qui ne contient pas v (voir Figure 6).



FIGURE 6 –

Au moins deux de ces trois chemins ont les propriétés demandées au premier point (car x a deux arêtes incidentes appartenant à la même partie), ce qui contredit à nouveau le choix de (A_1, A_2) (sur la Figure 6, les chemins du haut et du bas conviennent).

Finalement, on a prouvé par l'absurde que A_2 couvre tous les sommets de D .

□

Remarque. En suivant le schéma de la preuve, on peut construire un algorithme qui calcule une telle partition en temps polynomiale si elle existe.

2.3 D'autres cas polynomiaux

(cycle factor, cycle factor)-arc-partition

Théorème 2.3. Un graphe orienté $D = (V, A)$ admet une *(cycle factor, cycle factor)*-arc-partition si et seulement si D est 2-régulier (au sens orienté, c'est-à-dire tout sommet v a exactement 2 arcs entrants et deux arcs sortants).

Démonstration. Si D admet une *(cycle factor, cycle factor)*-arc-partition (A_1, A_2) , alors tout sommet $v \in V$ vérifie $d_{A_1}^+(v) = d_{A_1}^-(v) = 1$. De même, $d_{A_2}^+(v) = d_{A_2}^-(v) = 1$, et on en déduit que $d_D^+(v) = d_D^-(v) = 2$. Le sommet v étant quelconque, D est 2-régulier.

Réciproquement, si D est 2-régulier, construisons le graphe biparti $B = (V', V'', A')$ où V' et V'' sont des copies de V , et A' contient l'arête $u'v''$ pour tout arc uv de A (voir Figure 7). Le graphe B est 2-régulier (au sens non-orienté, c'est-à-dire tout sommet est incident à deux arêtes), par le théorème de Hall, il contient donc 2 couplages parfaits arête-disjoints. On déduit de ces deux couplages une partition (A_1, A_2) de A , telle que tout sommet $v \in V$ vérifie $d_{A_1}^+(v) = d_{A_1}^-(v) = 1$ et $d_{A_2}^+(v) = d_{A_2}^-(v) = 1$. Ainsi, (A_1, A_2) forme une *(cycle factor, cycle factor)*-arc-partition de D . □

Tester si un graphe est 2-régulier se fait bien sûr en temps polynomiale, on en déduit que le problème de *(cycle factor, cycle factor)*-arc-partition se résout en temps polynomiale.

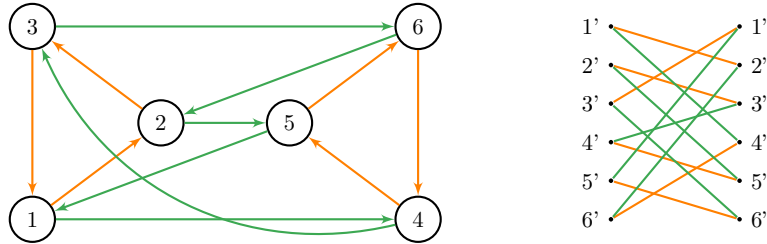


FIGURE 7 – Un graphe 2-régulier et son graphe biparti associé

Réductions vers des problèmes de flots

Théorème 2.4. Le problème de $(\delta^- \geq k, \delta^+ \geq \ell)$ -arc-partition se résout en temps polynomial.

Démonstration. On peut réduire ce problème au problème d'existence d'un (s, t) -flot à valeurs entières (non négatives), où chaque arc peut admettre une borne inférieure sur le flot qui le traverse. Ce problème se résout en temps polynomial.

Soit $D = (V, A)$, on construit V_1, V_2 deux copies de V et on ajoute tous les arcs $v_{i,1}v_{j,2}$ tels que $v_iv_j \in A(D)$ avec une capacité de 1. On ajoute ensuite une source s et tous les arcs $sv_{i,1}$, pour $v_{i,1} \in V_1$, avec une capacité de $d^+(v) - \ell$. Enfin, on ajoute un puits t et tous les arcs $v_{i,2}t$, pour $v_{i,2} \in V_2$, avec une borne inférieure de k (voir Figure 8).

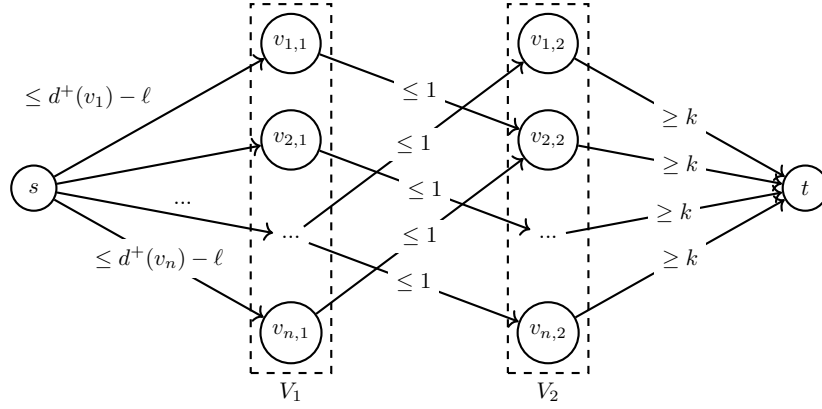


FIGURE 8 – Construction du réseau

Le réseau ainsi construit admet un (s, t) -flot si et seulement si D admet une $(\delta^- \geq k, \delta^+ \geq \ell)$ -arc-partition :

- * Si D admet un (s, t) -flot, posons A_1 l'ensemble des arcs v_iv_j de D , tels que $v_{i,1}v_{j,2}$ soit utilisé par un tel (s, t) -flot. Posons ensuite $A_2 = A \setminus A_1$ les autres arcs de D . Alors pour tout $v_i \in V$, puisque $v_{i,2}$ a un flot sortant supérieur à k , la loi de conservation du flot impose qu'au moins k arcs entrant en $v_{i,2}$ soient utilisés par le flot. Ainsi, $d_{A_1}^-(v_i) \geq k$. De même, $v_{i,1}$ a un flot entrant inférieur à $d^+(v_i) - \ell$, donc la conservation du flot impose qu'au moins ℓ arcs sortants en $v_{i,1}$ ne soient pas utilisés par le flot. Ainsi, $d_{A_2}^+(v_i) \geq \ell$. Finalement, (A_1, A_2) forme bien une $(\delta^- \geq k, \delta^+ \geq \ell)$ -arc-partition de D .
- * Réciproquement, si D admet une $(\delta^- \geq k, \delta^+ \geq \ell)$ -arc-partition (A_1, A_2) , on construit un (s, t) -flot en faisant passer le flot dans les arcs correspondant aux arcs de A_1 . Le flot se conserve car, pour tout $v_i \in V$, $v_{i,2}$ a un flot entrant $\geq k$ (par définition de A_1) et $(v_{i,1})$ a un flot sortant qui ne dépasse pas $d^+(v_i) - \ell$ car A_2 contient au moins ℓ arcs sortants de v_i .

□

Remarque. Dans la preuve, on aurait pu spécifier $d^-(v)$ et $d^+(v)$ pour tout sommet v . Ainsi, si on a k et l sont des fonctions de V dans \mathbb{N} , on peut tester en temps polynomial si $D = (V, A)$ admet une arc-partition (A_1, A_2) telle que pour tout sommet v , $d_{A_1}^+(v) \geq k(v)$ et $d_{A_2}^+(v) \geq l(v)$.

Théorème 2.5. Le problème de $(\delta^+ \geq k, \text{acyclique couvrant})$ -arc-partition se résout en temps polynomial.

Démonstration. Soit $D = (V, A)$ un graphe orienté. Si $\delta^+(D) < k$, D est clairement une instance négative du problème. Sinon, si $\delta^+(D) \geq k$, on considère \mathcal{N} le réseau associé à D construit dans la preuve du Théorème 2.4, en modifiant les capacités :

- Les arcs entre V_1 et V_2 conservent une capacité de 1,
- Les arcs de la forme $sv_{i,1}$ ont une capacité de k ,
- Les arcs de la forme $v_{i,2}t$ ont une capacité de $d^-(v_i) - 1$ si $d^+(v_i) = k$, et de $d^-(v_i)$ sinon.

Alors \mathcal{N} admet un (s, t) -flot de valeur kn si et seulement si D admet une $(\delta^+ \geq k, \text{acyclique couvrant})$ -arc-partition :

- * Si \mathcal{N} admet un tel (s, t) -flot, on met dans A_1 tous les arcs de A correspondant à un arc de \mathcal{N} où du flot circule. On pose ensuite $A_2 = A \setminus A_1$. Puisque \mathcal{N} a un flot de valeur kn , et que les arcs de la forme $sv_{i,1}$ ont une capacité de k , alors de tout sommet $v_{i,1}$ il sort k unités de flots. On a donc bien $\delta_{A_1}^+ \geq k$. La capacité choisie pour les arcs de la forme $v_{i,2}t$ imposent que A_2 soit couvrant (si $d^+(v_i) = k$, un arc entrant en v_i n'est pas emprunté par le flot, si $d^+(v_i) > k$, un arc sortant n'est pas emprunté par le flot). À priori, A_2 n'est pas acyclique, mais tant que A_2 contient un cycle, on peut choisir un de ses arcs et le passer dans A_1 . Avec cette opération, A_2 reste couvrant, et on finit par obtenir (A_1, A_2) la partition voulue.
- * Réciproquement, si D admet (A_1, A_2) une telle partition, tant qu'un sommet a strictement plus de k arcs sortants dans A_1 , on passe l'un de ces arcs dans A_2 . À la fin, tous les sommets ont exactement k arcs sortants dans A_1 . On construit alors un (s, t) -flot en faisant passer une unité de flot dans les arcs correspondant aux arcs de A_1 , puis en faisant passer k unités de flot dans les arcs $sv_{i,1}$. On complète ensuite avec le flot dans les arcs tv_i pour qu'il se conserve. C'est possible car si $d^+(v_i) = k$, alors v_i a au moins un arc entrant dans A_2 , donc le flot sortant en $v_{i,2}$ ne dépassera pas $d^-(v_i) - 1$. Si $d^+(v_i) > k$, alors $v_{i,2}t$ a une capacité de $d^-(v_i)$, qui ne sera donc pas dépassée.

□

Remarque. Dans [12], les auteurs donnent une réduction du problème $(\leq k \text{ arcs, équilibré})$ -arc-partition vers un problème de flots. Ils en déduisent un algorithme de complexité $O(nm \log(n) \log(k))$.

3 Quelques problèmes NP-complets

On présente ici certaines réductions de problèmes connus NP-complet vers des problèmes de partitions d'arcs. Lors des preuves de NP-complétude, on ne montre jamais que les problèmes de partitions d'arcs sont dans la classe NP. Néanmoins, pour toute propriété \mathcal{P} parmi les 15 considérées, on peut vérifier en temps polynomial si un graphe orienté vérifie \mathcal{P} . Ainsi, si (A_1, A_2) est une solution proposée pour un problème de $(\mathcal{P}_1, \mathcal{P}_2)$ -arc-partition sur une instance $D = (V, A)$, on peut vérifier en temps polynomial si (V, A_1) vérifie la propriété \mathcal{P}_1 et si (V, A_2) vérifie la propriété \mathcal{P}_2 . On peut donc vérifier en temps polynomial si (A_1, A_2) est une solution correcte ou non. Ainsi, tous les problèmes de partition d'arcs considérés sont dans la classe NP.

3.1 Réductions depuis le problème du cycle hamiltonien sur les graphes orientés 2-réguliers

Théorème 3.1. Soit $D = (V, A)$ un graphe orienté 2-régulier sur n sommets. Les propositions suivantes sont équivalentes :

0. D est hamiltonien,
1. D admet une (*fort*, $\delta^+ \geq 1$)-arc-partition,
2. D admet une (*eulérien*, $\delta^+ \geq 1$)-arc-partition,
3. D admet une (*fort*, $\geq n$ arcs)-arc-partition,
4. D admet une (*eulérien*, $\geq n$ arcs)-arc-partition,
5. D admet une (*connexe*, *cycle factor*)-arc-partition,
6. D admet une (*fort*, *cycle factor*)-arc-partition,
7. D admet une (*out-branching*, *cycle factor*)-arc-partition,
8. D admet une (*eulérien*, *cycle factor*)-arc-partition,
9. D admet une (*cycle*, *cycle factor*)-arc-partition,
10. D admet une (*cycle*, $\leq n$ arcs)-arc-partition,
11. il existe un arc $a \in A$ tel que $D - \{a\}$ admet une (*acyclique*, *cycle factor*)-arc-partition,
12. il existe un arc $a \in A$ tel que $D - \{a\}$ admet une (*acyclique couvrant*, *cycle factor*)-arc-partition,
13. il existe un arc $a \in A$ tel que $D - \{a\}$ admet une (*acyclique couvrant*, *équilibré*)-arc-partition,

Démonstration. Soit D un graphe orienté 2-régulier. Si D admet un cycle hamiltonien C , alors $D - A(C)$ est un cycle factor (il reste exactement un arc entrant et sortant en chaque sommet). Donc $(A(C), A \setminus A(C))$ forme la partition voulue pour les propositions 1 à 10. Pour les propositions 11 à 13, il suffit de retirer un arc a quelconque de C pour obtenir la partition voulue.

Réciproquement, si D admet (A_1, A_2) une (*fort*, $\delta^+ \geq 1$)-arc-partition, alors A_2 contient au moins n arcs, et donc A_1 forme une structure fortement connexe de D sur au plus n arcs :

A_1 forme donc nécessairement un cycle hamiltonien de D . Cet argument fonctionne pour les propositions 1 à 4.

Si D admet (A_1, A_2) une (*connexe, cycle factor*)-arc-partition alors D est hamiltonien. En effet, D étant 2-régulier, si on retire A_2 à D , alors il reste exactement un cycle factor (car il reste un arc entrant et un arc sortant en chaque sommet). Ce cycle factor étant connexe, c'est exactement un cycle hamiltonien. Cet argument fonctionne pour les propositions 5 à 9.

Si D admet (A_1, A_2) une (*cycle, $\leq n$ arcs*)-arc-partition, alors A_1 forme un cycle de taille au moins n , c'est-à-dire un cycle hamiltonien de D .

S'il existe $a \in A$ tel que $D - \{a\}$ admet (A_1, A_2) une (*acyclique, cycle factor*)-arc-partition, alors $A \setminus A_2 = A_1 \cup \{a\}$ forme un cycle factor de D . Ce cycle factor ne contient qu'un cycle car en retirant l'arc a , on obtient une structure acyclique : $A_1 \cup \{a\}$ forme donc un cycle hamiltonien de D . Cet argument fonctionne pour les propositions 11 et 12.

Enfin, s'il existe $a \in A$ tel que $D - \{a\}$ admet (A_1, A_2) une (*acyclique couvrant, équilibré*)-arc-partition, A_2 étant équilibré et D étant 2-régulier, on sait que $A_1 \cup \{a\}$ est équilibré également. Ainsi, $A_1 \cup \{a\}$ est l'union arc-disjointe d'un ensemble de cycles, mais qui s'intersectent tous sur $\{a\}$ car A_1 est acyclique : $A_1 \cup \{a\}$ forme donc un seul cycle. Puisque A_1 est couvrant, ce cycle est un cycle hamiltonien de D . \square

Le problème du cycle hamiltonien étant connu NP-complet sur les graphes orientés 2-réguliers [5], les différents problèmes de partition d'arcs associés aux propositions 1 à 13 sont NP-complets.

Remarque. On a montré en particulier, avec la proposition 4, que le problème de (*eulérien, $\geq k$ arcs*)-arc-partition est NP-complet. Néanmoins, [12] propose un algorithme FPT (paramétré par k) pour ce problème.

Théorème 3.2. Soit $D = (V, A)$ un graphe orienté 2-régulier. Considérons $B = (V', V'', A')$ le graphe orienté biparti où V' et V'' sont deux copies de V , et où A' contient tous les arcs $u'u''$ (pour $u \in V$) et $u''v'$ (pour $uv \in A$) (voir Figure 9). Alors les propositions suivantes sont équivalentes :

1. D admet un cycle hamiltonien,
2. B admet un cycle hamiltonien,
3. B admet une (*eulérien, acyclique*)-arc-partition,
4. B admet une (*fort, acyclique couvrant*)-arc-partition,

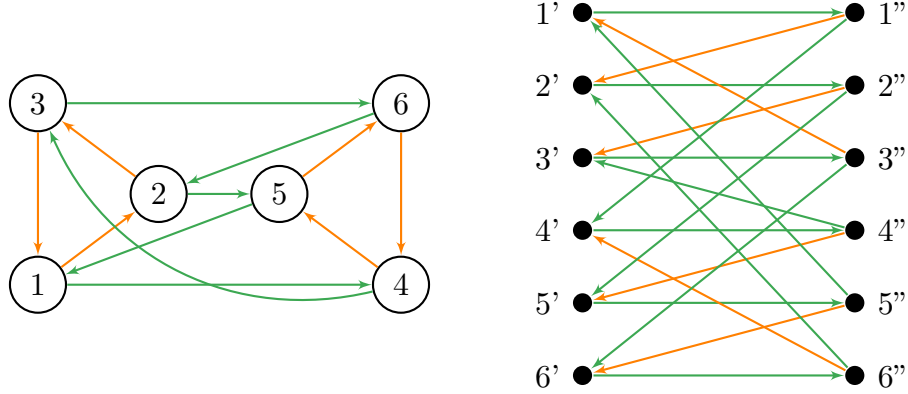


FIGURE 9 – Un graphe 2-régulier et sa construction associée

Démonstration. On va prouver l'équivalence par implications successives.

- * $1 \Rightarrow 2$: Si D est hamiltonien, soit $C = v_1 v_2 \dots v_k$ un cycle hamiltonien de D . Alors $v'_1 v''_1 v'_2 v''_2 \dots v'_k v''_k$ forme un cycle hamiltonien de B .
- * $2 \Rightarrow 3$: Soit C un cycle hamiltonien de B . Puisque tous les sommets de V' ont un seul arc sortant, et que tous les sommets de V'' ont un seul arc entrant, C emprunte tous les arcs de la forme $u'u''$. Ainsi, $B - A(C)$ est bien acyclique, car en retirant les arcs de C , les sommets de V' sont des puits, et les sommets de V'' sont des sources. Finalement, $(A(C), A' \setminus A(C))$ forme une *(eulérien, acyclique)*-arc-partition de B .
- * $3 \Rightarrow 4$: Soit (A_1, A_2) une *(eulérien, acyclique)*-arc-partition de B . Puisque A_1 est équilibré, les sommets de B sont tous incidents à exactement un arc de A_2 (un arc entrant pour les sommets de V' , un arc sortant pour les sommets de V''). Ainsi, A_2 est couvrant et A_1 est fortement connexe, donc (A_1, A_2) forme également une *(fort, acyclique couvrant)*-arc-partition de B .
- * $4 \Rightarrow 1$: Soit (A_1, A_2) une *(fort, acyclique couvrant)*-arc-partition de B . Puisque $(V(B), A_1)$ est fortement connexe, chaque sommet de B a au moins un arc entrant et un arc sortant dans A_1 . Puisque tous les sommets de B sont incidents à exactement 3 arcs et que A_2 est couvrant, les sommets de B ont exactement un arc entrant et un arc sortant dans A_1 . Puisque A_1 est fortement connexe, A_1 forme un cycle hamiltonien C de B . Les sommets de V' ayant un seul arc sortant, et les sommets de V'' un seul arc entrant, C est de la forme $C = v'_1 v''_1 v'_2 v''_2 \dots v'_k v''_k$. On peut alors considérer $v_1 v_2 \dots v_k$, qui forme un cycle hamiltonien de D .

□

À nouveau, le problème du cycle hamiltonien étant connu NP-complet sur les graphes orientés 2-réguliers, on en déduit que les deux problèmes *(eulérien, acyclique)*-arc-partition et *(fort, acyclique couvrant)*-arc-partition sont NP-complets.

Remarque. De la même façon, on peut montrer que D admet un chemin hamiltonien si et seulement si B admet une *(out-branching, acyclique couvrant)*-arc-partition.

Théorème 3.3. Soit $D = (V, A)$ un graphe orienté 2-régulier. Les propositions suivantes sont équivalentes :

0. D admet deux cycles hamiltoniens arc-disjoints,
1. D admet une $(\textit{fort}, \textit{fort})$ -arc-partition,
2. D admet une $(\textit{eulérien}, \textit{fort})$ -arc-partition,
3. D admet une $(\textit{eulérien}, \textit{eulérien})$ -arc-partition,
4. D admet une $(\textit{eulérien}, \textit{connexe})$ -arc-partition,
5. D admet une $(\textit{eulérien}, \textit{out-branching})$ -arc-partition,
6. D admet une $(\textit{cycle}, \textit{cycle})$ -arc-partition,
7. Il existe un arc $a \in A$ tel que $D - \{a\}$ admet une $(\textit{cycle}, \textit{acyclique})$ -arc-partition,
8. Il existe un arc $a \in A$ tel que $D - \{a\}$ admet une $(\textit{cycle}, \textit{acyclique couvrant})$ -arc-partition,
9. Il existe un arc $a \in A$ tel que $D - \{a\}$ admet une $(\textit{eulérien}, \textit{acyclique couvrant})$ -arc-partition,
10. Il existe un arc $a \in A$ tel que $D - \{a\}$ admet une $(\textit{fort}, \textit{out-branching})$ -arc-partition,

Démonstration. Si D admet deux cycles hamiltoniens arc-disjoints C_1 et C_2 , alors $(A(C_1), A(C_2))$ forme la partition voulue pour les propositions 1 à 6. En choisissant $a \in A(C_2)$ quelconque, $(A(C_1), A(C_2) \setminus \{a\})$ forme la partition voulue pour les propositions 7 à 10.

Réciproquement, si D admet (A_1, A_2) une $(\textit{fort}, \textit{fort})$ -arc-partition, A_1 et A_2 contiennent exactement n arcs chacun et forment chacun un cycle hamiltonien. Le graphe D contient donc bien deux cycles hamiltoniens arc-disjoints. Cet argument fonctionne pour les propositions 1 à 3.

Si D admet (A_1, A_2) une $(\textit{eulérien}, \textit{connexe})$ -arc-partition, alors A_2 est eulérien. En effet, D étant 2-régulier et A_1 étant équilibré, A_2 est équilibré et connexe par hypothèse. Donc, (A_1, A_2) forme une $(\textit{fort}, \textit{fort})$ -arc-partition de D , et donc D admet deux cycles hamiltoniens arc-disjoints, par le point précédent. Le même argument fonctionne pour la proposition 5.

Si D admet (C_1, C_2) une $(\textit{cycle}, \textit{cycle})$ -arc-partition, alors C_1 et C_2 forment deux cycles hamiltoniens arc-disjoints car chacun doit contenir exactement n arcs.

S'il existe un arc $a \in A$ tel que $D - \{a\}$ admet (A_1, A_2) une $(\textit{cycle}, \textit{acyclique})$ -arc-partition, alors $D - A_1$ est équilibré. C'est donc l'union d'un ensemble de cycles arc-disjoints. Mais puisque $D - (A_1 \cup a)$ est acyclique, $D - A_1$ est donc un seul cycle. Finalement, $(A_1, A_2 \cup a)$ forme une $(\textit{cycle}, \textit{cycle})$ -arc-partition de D . Par le point précédent, D admet donc deux cycles hamiltoniens arc-disjoints. Le même argument fonctionne pour la proposition 8.

S'il existe un arc $a \in A$ tel que $D - \{a\}$ admet (A_1, A_2) une $(\textit{eulérien}, \textit{acyclique couvrant})$ -arc-partition, comme précédemment on sait que $D - A_1$ forme un seul cycle. Puisque A_2 est couvrant, c'est un cycle hamiltonien, et $(A_1, A_2 \cup \{a\})$ forment deux cycles hamiltoniens de D .

Enfin, s'il existe un arc a tel que $D' = D - \{a\}$ admet (A_1, A_2) une $(\textit{fort}, \textit{out-branching})$ -arc-partition, alors A_1 contient exactement n arcs et A_2 contient exactement $n - 1$ arcs (car D' contient exactement $2n - 1$ arcs). Donc A_1 forme un cycle hamiltonien de D (car A_1 est fortement connexe sur exactement n arcs). $D - A_1$ forme donc un cycle factor, et ce cycle

factor est connexe car A_2 contient une arborescence sortante : $D - A_1$ forme donc un cycle hamiltonien de D . Au final, A_1 et $A_2 \cup \{a\}$ forment deux cycles hamiltoniens arc-disjoints de D .

□

Le problème consistant à décider si un graphe orienté 2-régulier admet deux cycles hamiltoniens arc-disjoints a été démontré NP-complet par J. Bang-Jensen et A. Yeo dans [7]. Les différents problèmes de partition d'arcs associés aux propositions 1 à 10 sont donc NP-complets.

3.2 Les problèmes de (*acyclique, autre*)-arc-partition

Problèmes connus

Quelques problèmes de partitions d'arcs en une partie acyclique sont déjà connus NP-complet. Parmi eux, le problème de $(\leq k \text{ arcs}, \text{acyclique})$ -arc-partition correspond exactement au problème de décision de FEEDBACK-ARC-SET, qui consiste à décider s'il existe un ensemble d'au plus k arcs qui intersecte tous les cycles du graphes. Ce problème est connu NP-complet [5].

On peut facilement le réduire au problème de $(\leq k \text{ arcs}, \text{acyclique couvrant})$ -arc-partition : soit D une instance du problème $(\leq k \text{ arcs}, \text{acyclique})$ -arc-partition et D' obtenu à partir de D en ajoutant un sommet v et tous les arcs vu (où u est un sommet de D). Clairement, puisque v est une source adjacente à tous les autres sommets, D admet une $(\leq k \text{ arcs}, \text{acyclique})$ -arc-partition si et seulement si D' admet une $(\leq k \text{ arcs}, \text{acyclique couvrant})$ -arc-partition.

Remarque. Le problème de FEEDBACK-ARC-SET admet un algorithme FPT paramétré par k . Plus précisément, [13] propose un algorithme en complexité $O((1.48k)^k n^{O(1)})$.

De même, le problème de $(\text{biparti}, \text{acyclique})$ -arc-partition est mieux connu comme un problème de coloration de graphes orientés. À partir d'un graphe orienté $D = (V, A)$, une coloration de V est dite propre si chaque couleur induit un graphe acyclique sur D . Le nombre dichromatique de D , noté $\vec{\chi}(D)$, correspond au plus petit nombre de couleurs nécessaires pour colorer proprement V . Alors D admet une $(\text{biparti}, \text{acyclique})$ -arc-partition si et seulement $\vec{\chi}(D) \leq 2$ (voir Figure 10).

En effet, si D admet (A_1, A_2) une telle partition, alors (V, A_1) est un graphe biparti, notons $V = (V_1, V_2)$ cette bipartition de V . Le graphe induit par V_i sur D est acyclique car ses arcs sont inclus dans A_2 . On peut donc 2-colorer D proprement en attribuant la couleur i aux sommets de V_i .

Réciproquement, si $c : V \rightarrow \{0, 1\}$ est une 2-coloration propre de D , on peut fixer :

$$\begin{aligned} A_1 &= \{uv \in A : c(u) = c(v)\} \\ A_2 &= \{uv \in A : c(u) \neq c(v)\} \end{aligned}$$

Clairement, (V, A_2) est biparti, de bipartition (V_1, V_2) où V_i est l'ensemble des sommets de couleur i par c . De plus, (V, A_1) est acyclique car (V_1, A_1) et (V_2, A_1) sont acycliques (par définition de c), et A_1 ne contient aucun arc entre V_1 et V_2 .



FIGURE 10 – Un exemple de graphe orienté 2-colorable et sa $(biparti, acyclique)$ -arc-partition associée

Le problème qui consiste à déterminer un graphe orienté est 2-colorable est connu NP-complet (voir [14]). On en déduit que le problème de $(biparti, acyclique)$ -arc-partition l'est également. Comme précédemment, le problème de $(biparti, acyclique\ couvrant)$ -arc-partition est également NP-complet, il suffit d'ajouter une source dominante pour obtenir une réduction depuis le problème de $(biparti, acyclique)$ -arc-partition.

$(connexe, acyclique\ couvrant)$ -arc-partition

Soit G un graphe non-orienté cubique (chaque sommet a exactement trois voisins), et soit D une orientation quelconque de G . Alors G admet un chemin hamiltonien si et seulement si D admet une $(connexe, acyclique\ couvrant)$ -arc-partition.

En effet, si G admet P un chemin hamiltonien, considérons ce même chemin (non orienté) dans D , et posons $(A_1, A_2) = (A(P), A \setminus A(P))$. Tant que A_2 contient un cycle, on peut prendre un arc de ce cycle et le passer dans A_1 . Au final, (A_1, A_2) forme une $(connexe, acyclique\ couvrant)$ -arc-partition de D .

Réciproquement, si D admet une telle partition (A_1, A_2) , A_1 contient un chemin hamiltonien de G . En effet, G étant cubique et A_2 étant couvrant, $d_{A_1}(v) \leq 2$ pour tout v . C'est donc une union de chemin et de cycles (non orientés) sommet-disjoints connexe : c'est un chemin ou un cycle hamiltonien. Dans les deux cas, A_1 contient un chemin hamiltonien de G . On en déduit que le problème $(connexe, acyclique\ couvrant)$ -arc-partition est NP-complet.

3.3 Les problèmes de $(biparti, autre)$ -arc-partition

$(biparti, biparti)$ -arc-partition

Théorème 3.4. Un graphe orienté D admet une $(biparti, biparti)$ -arc-partition si et seulement si son graphe sous-jacent est 4-colorable (voir Figure 11).

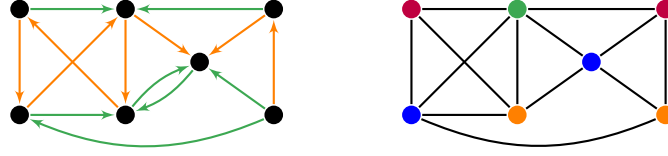


FIGURE 11 – Une graphe orienté admettant une $(biparti, biparti)$ -arc-partition et une 4-coloration des sommets de son graphe sous-jacent

Démonstration. Si D admet une $(biparti, biparti)$ -arc-partition (A_1, A_2) , notons (V'_i, V''_i) la bipartition de V induite par A_i sur D ($i \in \{1, 2\}$). Soit G le graphe sous-jacent de D , alors on peut 4-colorer G de la manière suivante :

Pour tout sommet $v \in V$,

- (i) si $v \in V'_1 \cap V'_2$, on attribue la couleur 1 à v ,
- (ii) sinon, si $v \in V'_1 \cap V''_2$, on attribue la couleur 2 à v ,
- (iii) sinon, si $v \in V''_1 \cap V'_2$, on attribue la couleur 3 à v ,
- (iv) sinon, alors $v \in V''_1 \cap V''_2$, et on attribue la couleur 4 à v .

Réciproquement, si G est 4-colorable, soit alors $\rho : V \rightarrow \{1, 2, 3, 4\}$ une telle coloration. En posant :

- (i) $V'_1 = \{v \in V : \rho(v) \in \{1, 2\}\}$, et $V''_1 = \{v \in V : \rho(v) \in \{3, 4\}\}$,
- (ii) $V'_2 = \{v \in V : \rho(v) \in \{1, 3\}\}$, et $V''_2 = \{v \in V : \rho(v) \in \{2, 4\}\}$,

On peut alors construire deux graphes bipartis $B_1 = (V'_1, V''_1, A_1)$, $B_2 = (V'_2, V''_2, A_2)$, où A_1 contient tous les arcs de la forme $\{1, 3\}, \{1, 4\}, \{2, 3\}, \{2, 4\}$ et A_2 les arcs de la forme $\{1, 2\}, \{3, 4\}$.

(A_1, A_2) forme alors une $(biparti, biparti)$ -arc-partition de D .

□

Le problème de 4-coloration étant NP-complet, le problème de $(biparti, biparti)$ -arc-partition l'est aussi.

$(biparti, \leq k \text{ arcs})$ -arc-partition

Ce problème est équivalent au problème de la coupe maximum. En effet, D admet une coupe de taille au moins ℓ si et seulement si D admet une $(biparti, \leq k \text{ arcs})$ -arc-partition, où $k = m - \ell$.

Le problème de coupe maximum étant connu NP-complet, le problème de $(biparti, \leq k \text{ arcs})$ -arc-partition l'est également.

$(biparti, cycle \text{ factor})$ -arc-partition

Théorème 3.5. Le problème de $(biparti, cycle \text{ factor})$ -arc-partition est NP-complet.

Démonstration. On va réduire le problème 3-SAT à ce problème de partition d'arcs.

Soit $W[u, v, p, q]$ le graphe orienté composés des sommets $\{u, v, y_1, \dots, y_{3p}, z_1, \dots, z_{3q}\}$, des arcs des deux chemins orientés $uy_1, \dots, y_{3p}v$ et $uz_1, \dots, z_{3q}v$, de l'arc vu , de tous les arcs $y_{3i}y_{3i-2}$ ($i \in \{1 \dots p\}$) et des arcs $z_{3i}z_{3i-2}$ ($i \in \{1 \dots q\}$) (voir Figure 12). Ce graphe sera associé aux variables. Remarquons que si l'on souhaite trouver une *(biparti, cycle factor)*-arc-partition de $W[u, v, p, q]$, alors :

- L'arc vu appartient nécessairement à la partie cycle factor, car c'est le seul arc sortant de v et le seul arc entrant en u ,
- Les deux arcs uy_1 et uz_1 appartiennent à des parties différentes,
- Tous les arcs de la forme $y_{3i}y_{3i+1}$ (respectivement $z_{3i}z_{3i+1}$) appartiennent à la même partie que uy_1 (respectivement uz_1). C'est le cas également pour l'arc $y_{3p}v$ (respectivement $z_{3q}v$).

Dans la suite, on identifiera y_{3p+1} et z_{3q+1} avec v , et de la même façon, on identifiera y_0 et z_0 avec u .

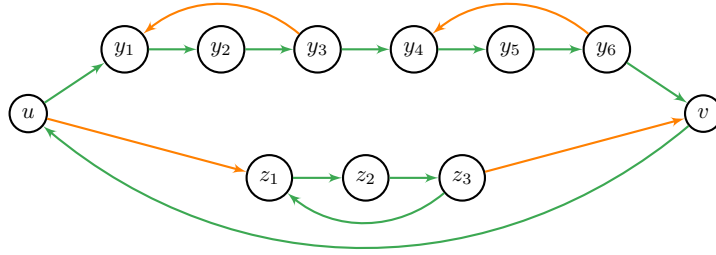


FIGURE 12 – Une des deux *(biparti, cycle factor)*-arc-partitions possibles de $W[u, v, 2, 1]$

Soit ensuite $W'[r, s, t, a_1, a_2, a_3]$ le graphe orienté de la Figure 13 (r, s et t sont des sommets, a_1, a_2 et a_3 sont des arcs). Ce graphe sera associé aux clauses.

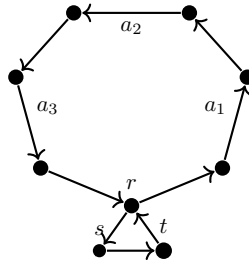


FIGURE 13 – Le graphe $W'[r, s, t, a_1, a_2, a_3]$

Soit alors \mathcal{F} une instance de 3-SAT sur n variables x_1, \dots, x_n et m clauses C_1, \dots, C_m . L'ordre (quelconque) sur les clauses nous donne un ordre sur les occurrences de chaque variable x et

de sa négation \bar{x} . On suppose que chaque variable apparaît au moins une fois sous sa forme positive et sous sa forme négative (quitte à simplifier \mathcal{F} si ce n'est pas le cas).

On associe alors à chaque variable x_i une copie de $W[u_i, v_i, p_i, q_i]$, où u_i et v_i sont deux nouveaux sommets, p_i correspond au nombre d'occurrences de x_i et q_i au nombre d'occurrences de \bar{x}_i .

On associe ensuite à chaque clause C_j une copie de $W'[r_j, s_j, t_j, a_{j,1}, a_{j,2}, a_{j,3}]$ où r_j, s_j and t_j sont trois nouveaux sommets, et on identifie $a_{j,1}, a_{j,2}$ et $a_{j,3}$ de la manière suivante :

Supposons que C_j contienne les variables x_i, x_k, x_l (où x_i, x_k et x_l peuvent être sous leur forme positive ou négative). Si x_i apparaît sous sa forme positive, et qu'il s'agit de la r -ème occurrence positive de x_i , alors on identifie $a_{j,1}$ avec $y_{i,3r}y_{i,3r+1}$. De façon analogue, si x_i apparaît négativement dans C_j et qu'il s'agit de la r -ème occurrence de \bar{x}_i , alors on identifie $a_{j,1}$ avec $z_{i,3r}z_{i,3r+1}$. On identifie $a_{j,2}$ et $a_{j,3}$ de la même façon pour les variables x_k et x_l .

On va maintenant prouver que D' , le graphe obtenu, admet une (*biparti, cycle factor*)-arc-partition si et seulement si \mathcal{F} est satisfiable.

- Si D' admet (A_1, A_2) une (*biparti, cycle factor*)-arc-partition.

Posons Φ l'assignation des variables qui consiste à assigner x_i à vrai si et seulement si $u_i y_{i,1}$ appartient à A_2 .

Soit alors C_j une clause de \mathcal{F} . On sait que chaque arc de la forme $y_{i,3k-2}y_{i,3k-1}$ (respectivement $y_{i,3k-1}y_{i,3k}$) appartient à A_2 car c'est l'unique arc entrant en $y_{i,3k-1}$ (respectivement sortant de $y_{i,3k-1}$). De même pour chaque arc de la forme $z_{i,3k-2}z_{i,3k-1}$ (respectivement $z_{i,3k-1}z_{i,3k}$). On en déduit que, dans le graphe W' associé à C_j , l'arc entre a_1 et a_2 doit appartenir à A_1 , sinon on aurait dans A_2 deux arcs sortant d'un sommet de la forme $y_{i,3k-2}$, et A_2 ne formerait pas un cycle factor. De même, l'arc entre a_2 et a_3 appartient à A_1 .

Ensuite, on sait que les arcs du triangle formé par les sommets r, s et t appartient à A_2 (car ce sont les seuls arcs entrants et sortants en s et t).

On en déduit que dans le graphe W' associé à C_j , on a un cycle impair (de taille 7) dont les seuls arcs pouvant appartenir à A_2 sont $\{a_1, a_2, a_3\}$. Puisque A_1 forme un graphe biparti, au moins un de ces arcs appartient à A_2 . Soit x_i la variable qui correspond à cet arc. Si x_i apparaît positivement dans C_j , alors $y_{i,3r}y_{i,3r+1}$ appartient à A_2 et donc $u_i y_{i,1}$ appartient également à A_2 (grâce aux précédentes remarques sur W), et x_i est donc vrai dans Φ . À l'inverse, si x_i apparaît négativement dans C_j , alors $u_i z_{i,1}$ appartient à A_2 , et donc $u_i y_{i,1}$ appartient à A_1 : x_i est donc faux dans Φ .

On en déduit que \mathcal{F} est satisfiable (car \mathcal{F} est satisfaite par Φ).

- Réciproquement, si \mathcal{F} est satisfiable, soit Φ un modèle de \mathcal{F} .

Pour toute variable x_i , si x_i est vrai dans Φ on pose E_i l'ensemble des arcs du cycle $u y_1 y_2 \dots y_{3p_i} v$ et des triangles de la forme $z_{i,3k-2} z_{i,3k-1} z_{i,3k}$ (pour k de 1 à q_i).

De façon analogue, si x_i est faux dans Φ , E_i désigne l'ensemble des arcs du cycle $u z_1 z_2 \dots z_{3q_i} v$ et des triangles de la forme $y_{i,3k-2} y_{i,3k-1} y_{i,3k}$ (pour k de 1 à p_i).

Pour toute clause C_j , \tilde{E}_j désigne l'ensemble des trois arcs du triangle r_j, s_j, t_j .

On obtient alors la partition voulue en posant A_2 comme l'union de tous les E_i et \tilde{E}_j , et A_1 comme l'ensemble des arcs restants. Alors (A_1, A_2) forme une (*biparti, cycle factor*)-arc-partition de D' : tout sommet de D' a exactement un arc entrant et un arc sortant

dans A_2 , donc $(V(D'), A_2)$ forme bien un cycle factor. Ensuite, les cycles impairs de D' ont tous au moins un arc dans A_2 : les cycles internes à W_i ont tous au moins un arc dans A_2 , le cycle de taille 7 de W'_j admet un arc dans A_2 car C_j est satisfaite par Φ .

□

Remarque. Dans la preuve précédente, le graphe construit D' a un degré maximum exactement 4, ce qui prouve que le problème est NP-complet sur cette classe de graphes, et qu'on ne peut pas espérer un algorithme FPT paramétré par $\Delta(D)$ (sous l'hypothèse $P \neq NP$).

Sur les graphes où le degré maximum est borné par 3, le problème devient polynomial car il revient à trouver un cycle factor. En effet, si un tel graphe admet un cycle factor, en retirant ses arcs on obtient un graphe où le degré maximum est borné par 1, qui est donc évidemment biparti.

(biparti, équilibré)-arc-partition

Théorème 3.6. Le problème de (biparti, équilibré)-arc-partition est NP-complet.

Démonstration. On va réduire le problème 3-SAT à ce problème de partition d'arcs. La réduction est semblable à celle effectuée dans la preuve du théorème 3.5.

Soit $W[u, v, p, q]$ le graphe orienté composé de deux chemins orientés uy_1, \dots, y_pv et uz_1, \dots, z_qv , de trois sommets a, b, c et des arcs va, ab, bu, ca et cb (voir Figure 14).

Remarquons que si on souhaite calculer (A_1, A_2) une (biparti, équilibré)-arc-partition de $W[u, v, p, q]$, les arcs incidents à c appartiennent à A_1 car c est une source. Ensuite, puisque a, b, c forme un triangle, ab doit appartenir à A_2 . Enfin, puisque A_2 doit être équilibré, va et bu appartiennent à A_2 . On peut ensuite fixer les arcs d'un chemin de u à v dans A_1 , et les arcs de l'autre chemin dans A_2 .

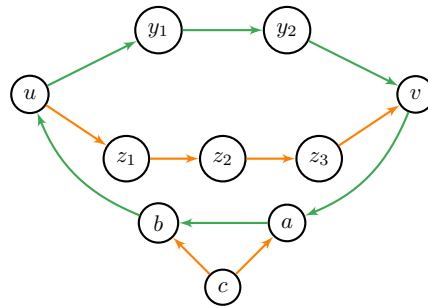
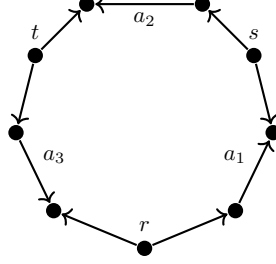


FIGURE 14 – Une des deux (biparti, cycle factor)-arc-partitions possibles de $W[u, v, 2, 3]$

Soit maintenant $W'[r, s, t, a_1, a_2, a_3]$ le graphe orienté de la Figure 15 (r, s et t sont des sommets, a_1, a_2 et a_3 sont des arcs). Remarquons que a_1, a_2 et a_3 sont les seuls arcs pouvant appartenir à la partie biparti (car r, s et t sont des sources).

FIGURE 15 – The clause gadget W'

Soit alors \mathcal{F} une instance de 3-SAT sur n variables x_1, \dots, x_n et m clauses C_1, \dots, C_m . On admet que chaque variable apparaît au moins une fois positivement et négativement. L'ordre (quelconque) sur les clauses nous donne un ordre sur les occurrences de chaque variable x et de sa négation \bar{x} .

On associe à chaque variable x_i une copie de $W[u_i, v_i, p_i, q_i]$ où p_i est le nombre d'occurrences de x_i , et q_i le nombre d'occurrences de \bar{x}_i .

On associe ensuite à chaque clause C_j une copie de $W'[r_j, s_j, t_j, a_{j,1}, a_{j,2}, a_{j,3}]$ où r_j, s_j et t_j sont trois nouveaux sommets, et on identifie $a_{j,1}, a_{j,2}$ et $a_{j,3}$ de la manière suivante :

Supposons que C_j contienne les variables x_i, x_k, x_l (sous leur forme positive ou négative). Si x_i apparaît sous sa forme positive, et qu'il s'agit de la r -ème occurrence positive de x_i , alors on identifie $a_{j,1}$ avec $y_{i,r}y_{i,r+1}$. De façon analogue, si x_i apparaît négativement dans C_j et qu'il s'agit de la r -ème occurrence de \bar{x}_i , alors on identifie $a_{j,1}$ avec $z_{i,r}z_{i,r+1}$. On identifie $a_{j,2}$ et $a_{j,3}$ de la même façon pour les variables x_k et x_l .

On va maintenant prouver que D' , le graphe obtenu, admet une (*biparti, équilibré*)-arc-partition si et seulement si \mathcal{F} est satisfiable.

— Si D' admet (A_1, A_2) une (*biparti, équilibré*)-arc-partition.

Posons Φ l'assignation des variables qui consiste à assigner x_i à vrai si et seulement si $u_i y_{i,1}$ appartient à A_2 (on sait en particulier que si $u_i y_{i,1}$ appartient à A_2 , alors tous les arcs de la forme $y_{i,k} y_{i,k+1}$ appartiennent à A_2 aussi).

Soit alors C_j une clause de \mathcal{F} . Puisque A_1 est biparti, W'_j contient un arc de A_2 . Cet arc est nécessairement a_1, a_2 ou a_3 . Soit x_i la variable associée à cet arc, si x_i apparaît positivement dans C_j , alors $y_{i,r} y_{i,r+1}$ appartient à A_2 et x_i est vrai dans Φ . À l'inverse, si x_i apparaît négativement dans C_j , alors $u_i z_{i,1}$ appartient à A_2 , et donc $u_i y_{i,1}$ appartient à A_1 : x_i est faux dans Φ .

On en déduit que \mathcal{F} est satisfiable (car \mathcal{F} est satisfaite par Φ).

— Réciproquement, si \mathcal{F} est satisfiable, soit Φ un modèle de \mathcal{F} .

Pour toute variable x_i , si x_i est vrai dans Φ on pose E_i l'ensemble des arcs du cycle $u_i, y_{i,1}, y_{i,2}, \dots, y_{i,p_i}, v_i, a_i, b_i$. De façon analogue, si x_i est faux dans Φ , E_i désigne l'ensemble des arcs du cycle $u_i, z_{i,1}, z_{i,2}, \dots, z_{i,p_i}, v_i, a_i, b_i$.

Il suffit alors de poser A_2 comme l'union de tous les E_i et A_1 comme l'ensemble des arcs restants. Alors (A_1, A_2) forme une (*biparti, équilibré*)-arc-partition de D' : tout sommet

de D' a le même nombre d'arcs entrants et d'arcs sortants dans A_2 , donc $(V(D'), A_2)$ est bien équilibré. Ensuite, les cycles (non orientés) impairs de D' ont tous au moins un arc dans A_2 : les cycles (non orientés) internes à W_i ont tous au moins un arc dans A_2 et le cycle (non orienté) de taille 9 de W'_j admet un arc dans A_2 car C_j est satisfaite par Φ .

□

(biparti, cycle)-arc-partition

Ce problème est également NP-complet. On ne donne pas le détail de la preuve, il suffit simplement de considérer la construction effectuée dans la preuve du théorème 3.6, en identifiant tous les sommets v_i et u_{i+1} et en remplaçant toutes les structures autour des sommets a_i, b_i, c_i par une seule structure entre v_n et u_1 (voir Figure 16). Les structures associées aux clauses sont les mêmes, mais cette fois le graphe équilibré est nécessairement un cycle.

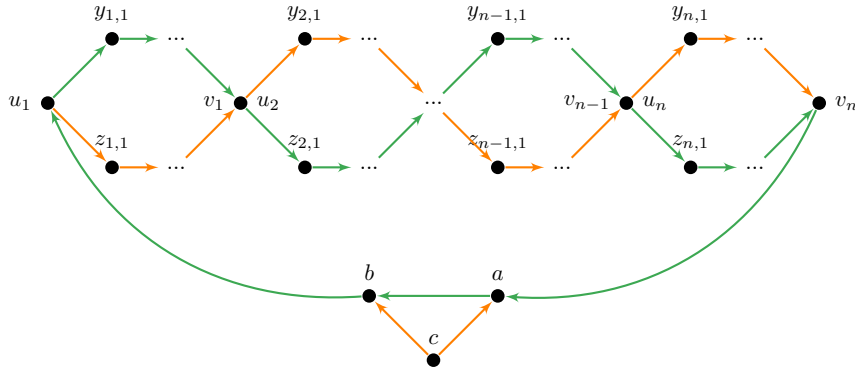


FIGURE 16 – Réduction de 3-SAT vers (biparti, cycle)-arc-partition

(biparti, eulérien)-arc-partition

Théorème 3.7. Le problème (biparti, eulérien)-arc-partition est NP-complet.

Démonstration. On va réduire le problème (biparti, équilibré)-arc-partition au problème (biparti, eulérien)-arc-partition.

Soit $D = (V, A)$ un graphe orienté, instance du problème (biparti, équilibré)-arc-partition. On prend $V = \{x_1, \dots, x_n\}$ un ordre quelconque sur V et on construit $D' = (V', A')$, instance du problème (biparti, eulérien)-arc-partition de la manière suivante :

$$\begin{aligned} V' &= V \cup \{d\} \cup \{z_{i,1}, z_{i,2} : 1 \leq i \leq n\} \\ A' &= A \cup \{dz_{i,1}, z_{i,1}x_i, x_i z_{i,2}, z_{i,2}d : 1 \leq i \leq n\} \end{aligned}$$

On va montrer que D admet une $(biparti, \textit{équilibré})$ -arc-partition si et seulement si D' admet une $(biparti, \textit{eulérien})$ -arc-partition.

Si D admet (A_1, A_2) une $(biparti, \textit{équilibré})$ -arc-partition. Posons :

$$A'_2 = A_2 \cup \{dz_{i,1}, z_{i,1}x_i, x_iz_{i,2}, z_{i,2}d : 1 \leq i \leq n\}$$

Alors A'_2 est équilibré, car A_2 l'est et qu'on passe de A_2 à A'_2 en ajoutant un arc entrant et un arc sortant à chaque sommet, sauf pour d , à qui on ajoute $|V|$ arc entrants et $|V|$ arcs sortants. De plus, (V', A'_2) est fortement connexe car tout sommet a un chemin vers d et un chemin depuis d . Finalement, (V', A'_2) est eulérien, et en posant $A'_1 = A_1$, (A'_1, A'_2) forme une $(biparti, \textit{eulérien})$ -arc-partition de D' .

Réciproquement, si D' admet (A'_1, A'_2) une $(biparti, \textit{eulérien})$ -arc-partition, alors $A'_1 \subseteq A$, car chaque arc de $A' \setminus A$ est l'unique arc sortant ou entrant de $z_{i,1}$ ou de $z_{i,2}$. Puisque (V', A'_2) est eulérien, A'_2 contient nécessairement tous ces arcs. Posons ensuite $A_2 = A'_2 \cap A$, alors A_2 est équilibré car on a retiré exactement un arc entrant et un arc sortant à chaque sommet de V par rapport à A'_2 . Il nous suffit donc de prendre (A'_1, A_2) pour obtenir une $(biparti, \textit{équilibré})$ -arc-partition de D . \square

4 Résultats de complexité sur des classes de graphes particulières

On présente dans cette partie quelques résultats de complexité des problèmes de partition d'arcs sur des classes particulières.

4.1 Graphes acycliques

De nombreux problèmes de partition d'arcs deviennent triviaux sur la classe des graphes orientés acycliques. En effet, un graphe acyclique n'admet aucune (P_1, P_2) -arc-partition pour P_1 appartenant à la liste de propriétés suivantes : *fortement connexe*, *cycle factor*, $\delta^- \geq k$, $\delta^+ \geq k$, *eulérien* et *cycle*.

Dans [15], J. Bang-Jensen, S. Thomassé et A. Yeo proposent un théorème qui concerne les arborescences sortantes de degré borné (et donc, symétriquement, les arborescences entrantes de degré borné) sur les graphes acycliques :

Théorème 4.1 (Bang-Jensen, Thomassé, Yeo). Soit $D = (V, A)$ un graphe orienté acyclique et soit $f : V \rightarrow \mathbb{N}$. Supposons que D admet exactement une source s . Alors D admet une arborescence sortante T enracinée en s qui vérifie :

$$d_T^+(v) \leq f(v) \text{ pour tout } v \in V \quad (1)$$

si et seulement si :

$$\sum_{x \in X^-} f(x) \geq |X| \text{ pour tout } X \subset (V \setminus s)$$

De plus, il existe un algorithme polynomial qui vérifie si un graphe acyclique admet une arborescence sortante qui vérifie (1).

En choisissant spécifiquement f , le théorème 4.1 permet de montrer que certains problèmes de partitions d'arcs peuvent être résolus en temps polynomial sur les graphes acycliques :

- *(out-branching, in-branching)* [15] : On sait que tout graphe orienté acyclique D admet une arborescence entrante (respectivement sortante) si et seulement si D a exactement un puits (respectivement une source).

Supposons donc que D admet exactement une source s et un puits p . Posons :

$$\begin{aligned} f(v) &= d^-(v) - 1 \text{ si } v \neq p \\ f(p) &= d^-(p) \end{aligned}$$

Alors D admet une *(out-branching, in-branching)*-arc-partition si et seulement si D admet une arborescence sortante T tel que $d_T^+(v) \leq f(v)$ pour tout $v \in V$. Par le théorème 4.1, ce problème de partition d'arcs se résout donc en temps polynomial sur la classe des graphes acycliques.

- *(out-branching, acyclic spanning)* : On suppose cette fois que D admet exactement une source s , et posons $f(v) = d^-(v) - 1$. De la même façon, D admet une telle arc-partition si et seulement si D admet une arborescence sortante T tel que $d_T^+(v) \leq f(v)$ pour tout $v \in V$, ce qui peut être vérifié en temps polynomial. Symétriquement, *(in-branching, acyclic spanning)*-arc-partition se résout en temps polynomial sur les graphes acycliques.

4.2 Condition suffisante pour $(\text{fort}, \text{connexe})$ -arc-partition

On présente dans cette partie une condition suffisante pour l'existence d'une $(\text{fort}, \text{connexe})$ -arc-partition. Un sommet universel est un sommet adjacent à tous les autres sommets du graphe.

Théorème 4.2. Tout graphe orienté $D = (V, A)$ planaire, 3-arc-connexe, avec un sommet universel, admet une $(\text{fort}, \text{connexe})$ -arc-partition.

Démonstration. Soit $D = (V, A)$ un graphe planaire, 3-arc-connexe, admettant un sommet universel d . On suppose que $D \setminus d$ est connexe, quitte à traiter séparément les composantes connexes de $D \setminus d$ (si C est une composante connexe de $D \setminus d$, $D[V(C) \cup \{d\}]$ est également planaire et 3-arc-connexe).

Dans un premier temps, remarquons que D admet un sommet (différent de d) avec exactement trois voisins : il est clair que tout sommet a au moins trois voisins car D est 3-arc-connexe. Ensuite, prenons une représentation planaire de D , et on numérote v_1, \dots, v_{n-1} les sommets de $D \setminus d$ dans le sens trigonométrique autour de d . Prenons deux sommets adjacents v_i et v_j où $(j - i) \bmod n > 1$, tel que $(j - i) \bmod n$ soit minimum (v_i et v_j sont le plus proche possible) : voir Figure 17, l'arc $v_1 v_3$ correspond à l'arc choisi.

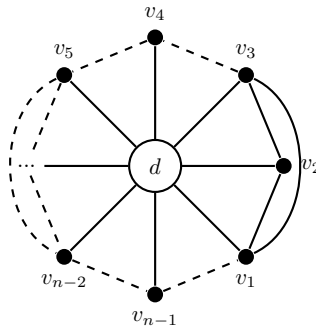


FIGURE 17 – Existence d'un sommet de degré 3

Si deux tels sommets n'existent pas, alors tous les sommets ont exactement 3 voisins. Sinon, tous les sommets entre v_i et v_j (dans le sens où v_i et v_j sont les plus proches) en ont exactement trois (sinon, on peut trouver deux sommets $v_{i'}$ et $v_{j'}$ avec $(j' - i') \bmod n < (j - i) \bmod n$, car D est planaire).

Soit alors x un sommet différent de d avec exactement trois voisins, x admet donc un arc doublement orienté vers chacun de ses trois voisins (car D est 3-arc-connexe). Soit a l'un de ses voisins (différent de d), il existe toujours une $(\text{fort}, \text{connexe})$ -arc-partition de $D[\{d, x, a\}]$ (voir Figure 18).

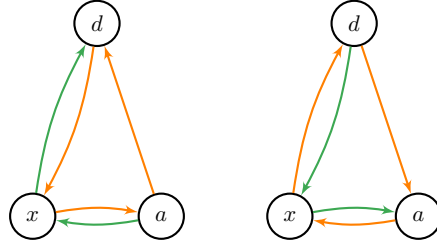


FIGURE 18 – Une $(\text{fort}, \text{connexe})$ -arc-partition de $D[\{d, x, a\}]$, suivant l'orientation de l'arc $\{a, d\}$

On contracte $\{x, a\}$. Dans la suite, x désigne donc un sommet admettant au moins un arc vers d et un arc depuis d utilisé par la partie fortement connexe et au moins un arc utilisé par la partie connexe (voir Figure 19). Le multigraphe D peut également admettre des arcs multiples vers ou depuis x . Remarquons que l'opération de contraction préserve la planarité et l'arc-connexité.



FIGURE 19 – Structure de x après la première contraction

Dans la suite, on applique exhaustivement les règles de réduction suivantes :

- * *Règle 1* : S'il existe dans $D \setminus d$ un circuit passant par x , le contracter. En effet, on peut étendre une partition des arcs en plaçant les arcs du circuit dans la partie fortement connexe et les arcs vers ou depuis d dans la partie connexe (voir Figure 20).

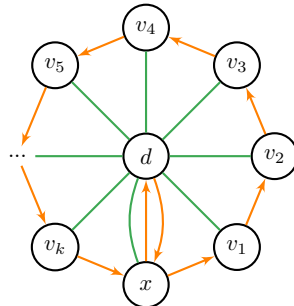


FIGURE 20 – Exemple d'application de la règle 1

* *Règle 2* : S'il existe dans $D \setminus d$ un chemin P de x à u tel que $ud \in A$ et tel qu'au moins une des deux conditions suivantes soit vérifiée :

- $\exists v \in V(P), uv \in A$,
- $\exists v \in (V(P) \cup \{d\}) \setminus \{v_k\}, vu \in A$,

alors, contracter P sur x . En effet, on peut étendre une partition en plaçant les arcs de P et l'arc ud dans la partie fortement connexe, et les arcs dv_i ainsi que l'arc uv (ou vu) dans la partie connexe (voir Figure 21).

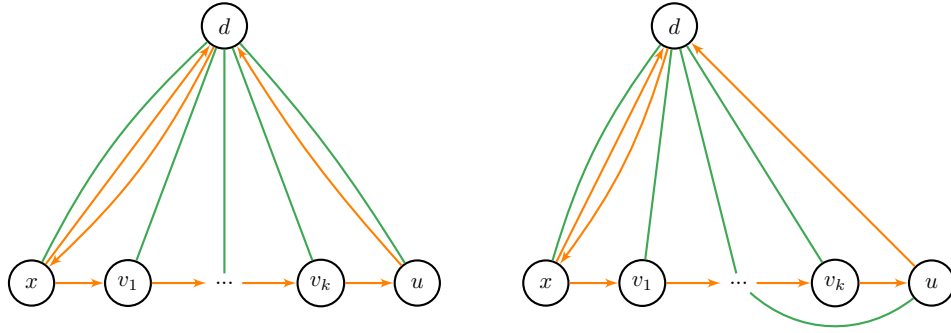


FIGURE 21 – Exemples d'application de la règle 2

* *Règle 3* : Symétriquement, s'il existe dans $D \setminus d$ un chemin $P = uv_1 \dots v_k x$ de u à x tel que $du \in A$ et tel qu'au moins une des deux conditions suivantes soit vérifiée :

- $\exists v \in V(P), vu \in A$,
- $\exists v \in (V(P) \cup \{d\}) \setminus \{v_1\}, uv \in A$,

alors, contracter P sur x .

Montrons par l'absurde qu'après une application exhaustive de ces trois règles, les sommets du graphe sont réduits à $\{x, d\}$.

Supposons qu'aucune des règles précédentes ne soit applicable et qu'il existe au moins 3 sommets dans D . Par hypothèse, $D \setminus d$ est connexe, x admet donc au moins un arc sortant ou un arc entrant.

- Si x admet un arc sortant, prenons $P = xv_1 \dots v_k$ un chemin maximal de $D \setminus d$ partant de x .

Le sommet v_k admet trois voisins sortants. Parmi ces voisins, il ne peut pas y avoir x , sinon la règle 1 serait applicable, ni d , sinon la règle 2 serait applicable. Puisque P est maximal, il existe $v_i \in V(P)$ un voisin sortant de v_k dans P . Puisque les arcs multiples sont tous incidents à x , et que D est planaire, il existe entre v_i et v_k un sommet v_j avec exactement trois voisins (voir Figure 22). Le chemin $P' = xv_1 \dots v_j$ permet d'appliquer la règle 2 : contradiction.

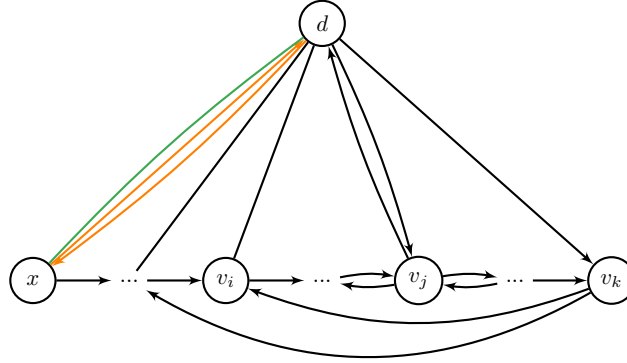


FIGURE 22 – Exemple d'un chemin maximal sortant de x

- Symétriquement, si x admet un arc entrant, on peut construire $P' = v_j \dots v_k x$, un chemin qui permet d'appliquer la règle 3.

Ainsi, puisqu'après une application exhaustive des règles précédentes, les sommets de D sont réduits à $\{d, x\}$, on a construit une *(fort, connexe)*-arc-partition de D . \square

4.3 Condition suffisante pour *(fort,fort)*-arc-partition

On montre dans cette partie que le problème *(fort,fort)*-arc-partition se résout en temps polynomial sur la classe des graphes orientés planaires extérieurs. Un graphe planaire extérieur est un graphe admettant une représentation planaire où tous les sommets appartiennent à la face externe (voir Figure 23).

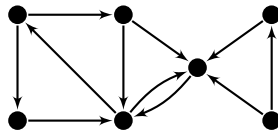


FIGURE 23 – Un exemple de graphe orienté planaire extérieur

Théorème 4.3. Soit D un multigraphe orienté planaire extérieur. Alors D admet une *(fort,fort)*-arc-partition si et seulement si D est 2-arc-connexe.

Ce résultat reste vrai si D est un multi-graphe.

Démonstration. Soit alors $D = (V, A)$ un multigraphe orienté planaire extérieur.

Si D admet (A_1, A_2) une *(fort,fort)*-arc-partition, alors D est 2-arc-connexe : soit $a \in A$, si $a \in A_1$ alors $D - \{a\}$ est fortement connexe par A_2 , sinon $D - \{a\}$ est fortement connexe par A_1 .

Supposons maintenant que D est 2-arc-connexe, on va prouver que D admet une $(fort, fort)$ -arc-partition.

Si D admet un sommet x tel que $D \setminus \{x\}$ ne soit plus connexe, alors pour chaque composante connexe C de $D - \{x\}$, $D[x \cup V(C)]$ est planaire extérieur et 2-arc-strong. On peut donc traiter séparément chacune des composantes connexes de $D \setminus \{x\}$. Supposons donc que D n'admet aucun sommet séparateur. On en déduit en particulier que la face externe de D forme un cycle hamiltonien (non orienté).

On va alors considérer les règles de réduction suivantes. On notera que chacune d'elle est correcte, dans le sens où si le graphe obtenu admet une $(fort, fort)$ -arc-partition, alors le graphe initial en admet une aussi. De plus, chacune de ces règles réduit strictement la taille du graphe.

- * *Règle 1* : Si D contient un cycle C tel que tout arc $uv \in A(C)$ a multiplicité 2 ou tel que pour tout arc $uv \in A(C)$, $vu \in A(C)$, contracter C (voir Figure 24).

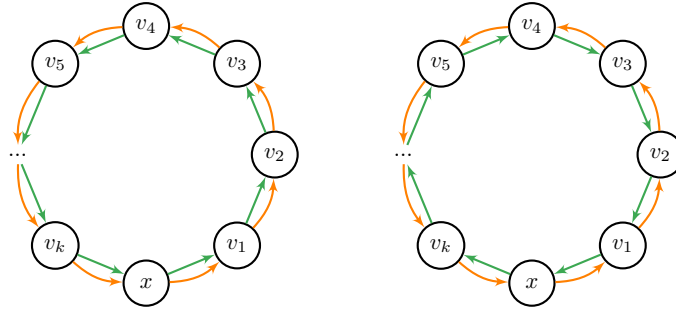


FIGURE 24 – Application de la règle 1

- * *Règle 2* : Si D contient un arc uv tel que $D - \{uv\}$ reste 2-arc-connexe, supprimer uv .
- * *Règle 3* : Si D contient trois sommets u, v, w tel que :
 - $uv, vw \in A$ avec multiplicité 2
 - $wv, vu \in A$ avec multiplicité 1.
 - w et u sont les seuls voisins v

Remplacer D par $D' = (V', A')$ où $V' = V \setminus \{v\}$ et $A' = (A \setminus \{uv, vw, wv, vu\}) \cup \{uw, wu\}$ avec multiplicité 2 sur uw et 1 sur wu (voir Figure 25). Remarquons que cette règle est correcte car si D' admet $(fort, fort)$ -arc-partition, alors il en admet une avec les deux arcs uw dans différentes parties.

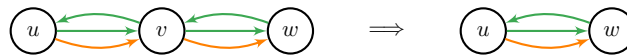


FIGURE 25 – application de la règle 3

- * *Règle 4* : Si D contient un cycle $C = v_1 \dots v_k$, $k \geq 3$, tel que chaque arc $v_i v_{i+1}$ a multiplicité 2 ou tel que chaque arc $v_{i+1} v_i$ appartiennent également à A , et tel que chaque sommet de $\{v_2, \dots, v_{k-1}\}$ a exactement 2 voisins, contracter $\{v_1, \dots, v_{k-1}\}$ (voir figure 26). À nouveau,

si le graphe construit contient une $(fort, fort)$ -arc-partition, alors il en admet une then avec les deux arcs v_1v_k (ou v_kv_1) soient dans différentes parties.

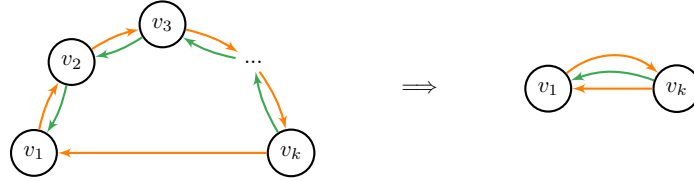


FIGURE 26 – Application de la règle 4

* *Règle 5* : Remplacer chacune des structures suivantes par un arc uv de multiplicité 2 (voir figure 27).

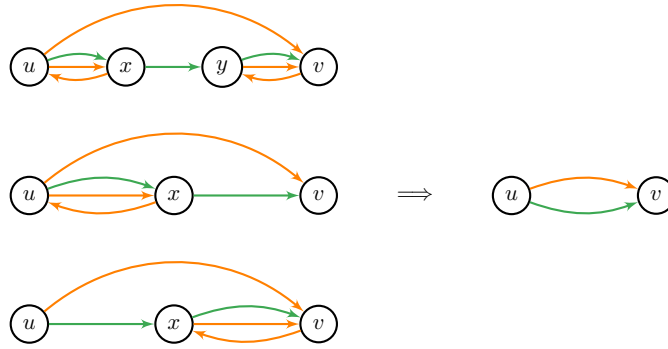


FIGURE 27 – Application de la règle 5

* *Règle 6* : Remplacer chacune des structures suivantes par deux arcs uv et vu (voir figure 28). Remarquons que si uv et vu appartiennent à la même partie d'une $(fort, fort)$ -arc-partition, elle peut quand même être étendue (voir figure 29).

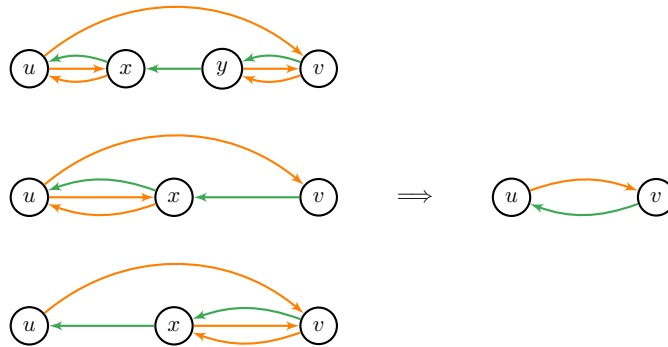


FIGURE 28 – Application de la règle 6 (1/2)

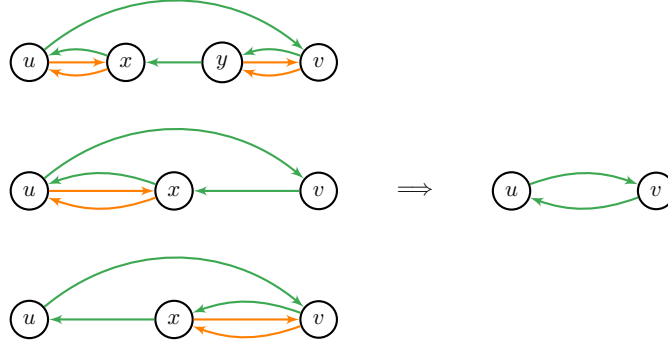


FIGURE 29 – Application de la règle 6 (2/2)

Maintenant, on va prouver par l'absurde que si aucune de ces règles ne peut être appliquée, alors D ne contient qu'un sommet, et donc D admet une $(fort, fort)$ -arc-partition.

Supposons donc qu'aucune de ces règles ne soit applicable, et prenons u et v deux sommets adjacents dans D . Par la règle 2, on sait que uv a multiplicité au plus 2. De plus, si uv a multiplicité 2, alors vu a multiplicité au plus 1 (par la règle 1). Donc, si D contient exactement 2 sommets, D ne peut pas être 2-arc-connexe. On admet donc que D contient au moins 3 sommets, et on considère $C = v_1, \dots, v_n$, le cycle hamiltonien (non orienté) formé par la face externe.

- Si C n'admet aucune corde, on prend v_1, \dots, v_n une numérotation de V le long de C (dans la suite, on travaille modulo n sur les indices de v_i).
 - S'il existe i tel que $v_i v_{i+1} \in A$ avec multiplicité 2, et tel que $v_{i+1} v_i \notin A$, alors pour tout $j \in \{1, \dots, n\}$, $v_j v_{j+1} \in A$ avec multiplicité 2 (en considérant la coupe $\{v_{i+1} \dots v_j\}$, qui doit admettre deux arcs sortants). On peut donc appliquer la règle 1.
 - Sinon s'il existe i tel que $v_i v_{i+1} \in A$ et $v_{i+1} v_i \in A$ avec multiplicité 1 dans chaque sens, alors pour tout $j \in \{1, \dots, n\}$, $v_j v_{j+1} \in A$ et $v_{j+1} v_j \in A$ (à nouveau, en considérant la coupe $\{v_{i+1} \dots v_j\}$).
 - Sinon s'il existe i tel que $v_i v_{i+1} \in A$ avec multiplicité 1 et $v_{i+1} v_i \notin A$, alors pour tout $j \neq i$, $v_j v_{j+1} \in A$ avec multiplicité 2 et $v_{j+1} v_j \in A$ avec multiplicité 1 (en considérant la même coupe). Puisque C contient au moins 3 sommets, la règle 3 peut être appliquée.
 - Sinon, pour tout $i \in \{1, \dots, k\}$, il y a exactement trois arcs entre v_i et v_{i+1} . On peut donc appliquer la règle 1 sur C .
- Supposons maintenant que C admet une corde. On choisit une corde minimale $v_k v_l$ (dans le sens où $C' = v_k, v_{k+1}, \dots, v_l$ n'admet pas de corde). On admet que $l > k$, quitte à numérotter les sommets dans l'autre sens le long de C . Soit alors v_i et v_{i+1} deux voisins le long de C' (avec $k \leq i < l$).
 - Si $v_i v_{i+1} \in A$ avec multiplicité 2 et $v_{i+1} v_i \notin A$, alors pour tout $j \in \{k, \dots, l-1\}$, $v_j v_{j+1} \in A$ avec multiplicité 2 (en considérant les coupes $\{v_{j+1} \dots v_i\}$ et $\{v_{i+1} \dots v_j\}$). On peut donc appliquer la règle 4 sur C' .

- Sinon si $v_i v_{i+1} \in A$ et $v_{i+1} v_i \in A$ avec multiplicité 1, alors pour tout $j \in \{k, \dots, l-1\}$, $v_j v_{j+1} \in A$ and $v_{j+1} v_j \in A$ (en considérant la même coupe). De nouveau, la règle 4 peut être appliquée sur C'
- Sinon si $v_i v_{i+1} \in A$ avec multiplicité 1 et $v_{i+1} v_i \notin A$, alors pour tout $j \in \{k, \dots, l-1\}, j \neq i$, $v_j v_{j+1} \in A$ avec multiplicité 2 et $v_{j+1} v_j \in A$ avec multiplicité 1 (en considérant la même coupe). Si C' contient au moins 5 sommets, la règle 3 peut être appliquée. Sinon, si C' contient au plus 4 sommets, la règle 5 ou la règle 6 peut être appliquée.
- De façon similaire, si $v_{i+1} v_i \in A$ avec multiplicité 1 et $v_i v_{i+1} \notin A$, alors pour tout $j \in \{k, \dots, l-1\}, j \neq i$, $v_{j+1} v_j \in A$ avec multiplicité 2 et $v_j v_{j+1} \in A$ avec multiplicité 1. Si C' contient au moins cinq sommets, la règle 3 est applicable. Sinon, la règle 4 ou la règle 6 est applicable.
- Sinon, i étant quelconque, on peut admettre que pour tout $i \in \{1, \dots, k\}$, il y a exactement 3 arcs entre v_i et v_{i+1} . La règle 4 peut donc être appliquée sur C' .

□

Remarque. En suivant le schéma de la preuve, on peut construire un algorithme polynomial qui calcule une $(\text{fort}, \text{fort})$ -arc-partition à partir d'un graphe planaire extérieur et de sa représentation.

Remarque. Le théorème 4.3 prouve la conjecture 1.1 pour la classe des graphes planaires extérieurs.

5 Conclusion et synthèse

Au fil du stage nous avons pu classé tous les problèmes de partition d'arcs associés aux 120 paires de propriétés choisies. On présente dans les tableaux 1, 2 et 3 l'ensemble des résultats de complexité des problèmes de partition d'arcs étudiés (dans le cas général). On justifie brièvement l'ensemble de ces résultats en annexe A.

	Biparti	Connexe	Fort. Connexe	Acyclique	Acyclique couvr.	Out-Branching	In-Branching
Biparti	NPC	Poly	Poly	NPC	NPC	Poly	Poly
Connexe	×	Poly	NPC	Poly	NPC	NPC	NPC
Fort. Connexe	×	×	NPC	Poly	NPC	NPC	NPC
Acyclique	×	×	×	Poly	Poly	Poly	Poly
Acyclique couvr.	×	×	×	×	Poly	NPC	NPC
Out-Branching	×	×	×	×	×	Poly	NPC
In-branching	×	×	×	×	×	×	Poly

Tableau 1 – Complexité des problèmes de partition d'arcs

	$\delta^- \geq k$	$\delta^+ \geq k$	Cycle Factor	$\leq k$ arcs	$\geq k$ arcs	Équilibré	Eulérien	Cycle
Biparti	Poly	Poly	NPC	NPC	Poly	NPC	NPC	NPC
Connexe	Poly	Poly	NPC	Poly	Poly	Poly	NPC	NPC
Fortement Connexe	NPC	NPC	NPC	Poly	NPC	Poly	NPC	NPC
Acyclique	Poly	Poly	NPC	NPC	Poly	Poly	NPC	NPC
Acyclique couvrant	Poly	Poly	NPC	NPC	Poly	NPC	NPC	NPC
Out-Branching	Poly	NPC	NPC	Poly	Poly	Poly	NPC	NPC
In-branching	NPC	Poly	NPC	Poly	Poly	Poly	NPC	NPC

Tableau 2 – Complexité des problèmes de partition d'arcs

	$\delta^- \geq k$	$\delta^+ \geq k$	Cycle Factor	$\leq k$ arcs	$\geq k$ arcs	Équilibré	Eulérien	Cycle
$\delta^- \geq k$	Poly	Poly	Poly	Poly	Poly	Poly	NPC	Poly
$\delta^+ \geq k$	×	Poly	Poly	Poly	Poly	Poly	NPC	Poly
Cycle Factor	×	×	Poly	Poly	Poly	Poly	NPC	NPC
$\leq k$ arcs	×	×	×	Poly	Poly	Poly	NPC	NPC
$\geq k$ arcs	×	×	×	×	Poly	Poly	NPC	Poly
Équilibré	×	×	×	×	×	Poly	Poly	Poly
Eulérien	×	×	×	×	×	×	NPC	NPC
Cycle	×	×	×	×	×	×	×	NPC

Tableau 3 – Complexité des problèmes de partition d'arcs

Nous avons aussi pu étudier la conjecture 1.1 sur différentes classes de graphes. Durant les dernières semaines, nous nous consacrerons à l'étude de cette conjecture sur l'ensemble des graphes admettant un sommet universel, la conjecture 1.2 ayant d'ailleurs déjà été démontrée sur cette classe de graphes.

Certains problèmes de partition d'arcs restent ouverts. Par exemple, l'article [8] étudie la complexité de certains problèmes de partition d'arcs sur les graphes orientés 2-réguliers. Que peut-on dire sur les graphes orientés 3-réguliers ? Plus généralement, que peut-on dire sur les graphes orientés k -réguliers pour de grandes valeurs de k ? De plus, on pourrait s'intéresser à l'étude des problèmes de partition d'arcs sur des classes de graphes plus larges, comme les graphes planaires. On pourrait également s'intéresser à d'autres propriétés intéressantes, comme le fait de contenir un couplage parfait, le fait d'être biparti et couvrant ou encore le fait de ne pas contenir de cycle orienté impair.

Références

- [1] W. T. Tutte, “On the problem of decomposing a graph into n connected factors,” *Journal of the London Mathematical Society*, vol. 1, no. 1, pp. 221–230, 1961.
- [2] J. Bang-Jensen and A. Yeo, “Decomposing k -arc-strong tournaments into strong spanning subdigraphs,” *Comb.*, vol. 24, no. 3, pp. 331–349, 2004.
- [3] C. Thomassen, “Configurations in graphs of large minimum degree, connectivity, or chromatic number,” in *Proceedings of the third international conference on Combinatorial mathematics*, pp. 402–412, 1989.
- [4] J. Bang-Jensen, “Edge-disjoint in-and out-branchings in tournaments and related path problems,” *Journal of Combinatorial Theory, Series B*, vol. 51, no. 1, pp. 1–23, 1991.
- [5] J. Bang-Jensen and G. Z. Gutin, *Digraphs - Theory, Algorithms and Applications, Second Edition*. Springer Monographs in Mathematics, Springer, 2009.
- [6] J. Edmonds, “Edge-disjoint branchings,” *Combinatorial Algorithms*, 1973.
- [7] J. Bang-Jensen and A. Yeo, “Arc-disjoint spanning sub(di)graphs in digraphs,” *Theor. Comput. Sci.*, vol. 438, pp. 48–54, 2012.
- [8] J. Bang-Jensen and S. Simonsen, “Arc-disjoint paths and trees in 2-regular digraphs,” *Discrete Applied Mathematics*, vol. 161, no. 16, pp. 2724–2730, 2013.
- [9] J. Bang-Jensen, S. Bessy, and A. Yeo, “Non-separating spanning trees and out-branchings in digraphs of independence number 2,” *CoRR*, vol. abs/2007.02834, 2020.
- [10] J. Bang-Jensen and F. Havet, “Finding good 2-partitions of digraphs i. hereditary properties,” *Theoretical Computer Science*, vol. 636, pp. 85–94, 2016.
- [11] J. Bang-Jensen, N. Cohen, and F. Havet, “Finding good 2-partitions of digraphs ii. enumerable properties,” *Theoretical Computer Science*, vol. 640, pp. 1–19, 2016.
- [12] M. Cygan, D. Marx, M. Pilipczuk, M. Pilipczuk, and I. Schlotter, “Parameterized complexity of eulerian deletion problems,” *Algorithmica*, vol. 68, no. 1, pp. 41–61, 2014.
- [13] J. Chen, Y. Liu, and S. Lu, “Directed feedback vertex set problem is fpt,” in *Dagstuhl Seminar Proceedings*, Schloss Dagstuhl-Leibniz-Zentrum für Informatik, 2007.
- [14] D. Bokal, G. Fijavz, M. Juvan, P. M. Kayll, and B. Mohar, “The circular chromatic number of a digraph,” *Journal of Graph Theory*, vol. 46, no. 3, pp. 227–240, 2004.
- [15] J. Bang-Jensen, S. Thomassé, and A. Yeo, “Small degree out-branchings,” *Journal of Graph Theory*, vol. 42, no. 4, pp. 297–307, 2003.

Annexes

A Classification des problèmes de partition d'arcs

On dresse ici une justification rapide de la classification de chacun des 120 problèmes présentés dans la section 5.

- biparti, biparti* : NPC, équivalent à être 4-colorable, voir 3.3.
- biparti, connexe* : Trivial, voir 2.1.
- biparti, fortement connexe* : Trivial, voir 2.1.
- biparti, acyclique* : NPC, revient à déterminer si un graphe orienté est 2-colorable, voir 3.2.
- biparti, acyclique couvrant* NPC, réduction depuis *biparti, acyclique*.
- biparti, out-branching* : Trivial, voir 2.1.
- biparti, in-branching* : Trivial, voir 2.1.
- biparti, $\delta^- \geq k$* : Trivial, voir 2.1.
- biparti, $\delta^+ \geq k$* : Trivial, voir 2.1.
- biparti, cycle factor* : NPC, réduction depuis 3-SAT, voir 3.3.
- biparti, $\delta^+ \leq k$ arcs* : NPC, réduction depuis MAX-CUT, voir 3.3.
- biparti, $\delta^+ \geq k$ arcs* : Trivial, voir 2.1.
- biparti, équilibré* : NPC, réduction depuis 3-SAT, voir 3.3.
- biparti, eulérien* : NPC, réduction depuis (*biparti, équilibré*), voir 3.3.
- biparti, cycle* : NPC, réduction depuis 3-SAT, voir 3.3.
- connexe, connexe* : Poly, peut être résolu avec le théorème de Tutte-Nash-Williams, voir [5].
- connexe, fortement connexe* : NPC, voir [7].
- connexe, acyclique* : Trivial, voir 2.1.
- connexe, acyclique couvrant* : NPC, équivalent à admettre un cycle hamiltonien dans les graphes cubiques, voir 3.2.
- connexe, out-branching* : NPC, voir [7].
- connexe, in-branching* : NPC, voir [7].
- connexe, $\delta^- \geq k$* : Poly, se résout grâce aux matroïdes : soit $D = (V, A)$, on pose M_1 le matroïde graphique de D , et M_2 le matroïde contenant tous les sous-ensemble A' de A qui vérifient $\delta^-(D - A') \geq k$. D admet la partition voulue si et seulement si $M_1 \cap M_2$ contient un élément de taille $n - 1$, ce qui peut être testé en temps polynomial.
- connexe, $\delta^+ \geq k$* : Poly, symétrique de (*connexe, $\delta^- \geq k$*).
- connexe, cycle factor* : NPC, voir 3.1.
- connexe, $\leq k$ arcs* : Trivial, voir 2.1.
- connexe, $\geq k$ arcs* : Poly, cela revient à tester si D est connexe et si $k \geq m - n + 1$.
- connexe, équilibré* : Trivial, voir 2.1.
- connexe, eulérien* : NPC, voir 3.1.
- connexe, cycle* : NPC, voir [7].
- fortement connexe, fortement connexe* : NPC, voir 3.1. Pour la preuve originale, voir [2].
- fortement connexe, acyclique* : Trivial, voir 2.1.
- fortement connexe, acyclique couvrant* : NPC, voir 3.1.
- fortement connexe, out-branching* : NPC, voir 3.1.
- fortement connexe, in-branching* : NPC, voir 3.1.
- fortement connexe, $\delta^- \geq k$* : NPC, voir 3.1.
- fortement connexe, $\delta^+ \geq k$* : NPC, voir 3.1.
- fortement connexe, cycle factor* : NPC, voir 3.1.
- fortement connexe, $\leq k$ arcs* : Trivial, voir 2.1.

fortement connexe, $\geq k$ arcs : NPC, voir 3.1.
fortement connexe, équilibré : Trivial, voir 2.1.
fortement connexe, eulérien : NPC, voir 3.1.
fortement connexe, cycle : NPC, voir [7].
acyclique, acyclique : Poly, voir 2.2.
acyclique, acyclique couvrant : Poly, voir 2.2.
acyclique, out-branching : Trivial, voir 2.1.
acyclique, in-branching : Trivial, voir 2.1.
acyclique, $\delta^- \geq k$: Trivial, voir 2.1.
acyclique, $\delta^+ \geq k$: Trivial, voir 2.1.
acyclique, cycle factor : NPC, voir 3.1.
acyclique, $\leq k$ arcs : NPC, c'est exactement le problème FEEDBACK-ARC-SET, voir 3.2.
acyclique, $\geq k$ arcs : Trivial, voir 2.1.
acyclique, équilibré : Poly, on met tous les arcs dans la partie acyclique, et tant qu'elle contient un cycle, on passe les arcs de ce cycle dans la partie équilibrée.

acyclique, eulérien : NPC, voir 3.1.
acyclique, cycle : NPC, voir 3.1.
acyclique couvrant, acyclique couvrant : Poly, voir 2.2.
acyclique couvrant, out-branching : NPC, voir 3.1.
acyclique couvrant, in-branching : NPC, voir 3.1.
acyclique couvrant, $\delta^- \geq k$: Poly, voir 2.3.
acyclique couvrant, $\delta^+ \geq k$: Poly, symétrique de (*acyclique couvrant, $\delta^- \geq k$*).
acyclique couvrant, cycle factor : NPC, voir 3.1.
acyclique couvrant, $\leq k$ arcs : NPC, réduction depuis (*acyclique, $\leq k$ arcs*) voir 3.2.
acyclique couvrant, $\geq k$ arcs : Poly, cela revient à calculer un ensemble couvrant de taille minimum. Pour cela, on peut calculer un couplage maximum, puis le compléter de façon gloutonne jusqu'à obtenir un ensemble couvrant. Le résultat obtenu est un ensemble couvrant de taille minimum, et on a une telle partition si et seulement s'il est de taille $\leq m - k$.

acyclique couvrant, équilibré : NPC, voir 3.1.
acyclique couvrant, eulérien : NPC, voir 3.1.
acyclique couvrant, cycle : NPC, voir 3.1.
out-branching, out-branching : Poly, peut être résolu avec le théorème d'Edmonds, voir [6].
out-branching, in-branching : NPC, voir [4].
out-branching, $\delta^- \geq k$: Poly, soit $D = (V, A)$, si $\delta^-(D) > k$, cela revient à trouver une arborescence sortante. Sinon s'il existe un sommet r de degré entrant k , et que tous les autres ont degré entrant au moins $k + 1$, cela revient à trouver une arborescence sortante enracinée en r . Dans les autres cas, c'est toujours faux.

out-branching, $\delta^+ \geq k$: NPC, voir 3.1.
out-branching, cycle factor : NPC, voir 3.1.
out-branching, $\leq k$ arcs : Trivial, voir 2.1.
out-branching, $\geq k$ arcs : Poly, cela revient à vérifier si $m \geq n + k - 1$ et si D admet une arborescence sortante.

out-branching, équilibré : Trivial, voir 2.1.
out-branching, eulérien : NPC, voir 3.1.
out-branching, cycle : NPC, on considère la réduction de 3-SAT vers le problème consistant à déterminer si D admet un chemin P de s à t tel que $D - A(P)$ admette une arborescence, proposée dans [7]. En y ajoutant l'arc ts , on obtient une réduction de 3-SAT vers notre problème.

in-branching, in-branching : Poly, peut être résolu avec le théorème d'Edmonds, voir [6].
in-branching, $\delta^- \geq k$: NPC, voir 3.1.

in-branching, $\delta^+ \geq k$: Poly, symétrique de (*out-branching*, $\delta^- \geq k$).
in-branching, *cycle factor* : NPC, voir 3.1.
in-branching, $\leq k$ arcs : Trivial, voir 2.1.
in-branching, $\geq k$ arcs : Poly, symétrique de (*out-branching*, $\geq k$ arcs).
in-branching, équilibré : Trivial, voir 2.1.
in-branching, eulérien : NPC, voir 3.1.
in-branching, *cycle* : NPC, symétrique de (*out-branching*, *cycle*).
 $\delta^- \geq k$, $\delta^- \geq \ell$: Poly, cela revient à vérifier si $\delta^-(D) \geq k + \ell$.
 $\delta^- \geq k$, $\delta^+ \geq \ell$: Poly, voir 2.3.
 $\delta^- \geq k$, *cycle factor* : Poly, si $\delta^-(D) \geq k$ c'est toujours faux, sinon cela revient à tester si le graphe admet un cycle factor.

 $\delta^- \geq k$, $\leq \ell$ arcs : Trivial, voir 2.1.
 $\delta^- \geq k$, $\geq \ell$ arcs : Poly, il suffit de vérifier si la somme des $d^-(v) - k$ est supérieure à ℓ .
 $\delta^- \geq k$, équilibré : Trivial, voir 2.1.
 $\delta^- \geq k$, eulérien : NPC, voir 3.1.
 $\delta^- \geq k$, *cycle* : Poly, cela revient à vérifier si le graphe induit par les sommets de degré entrant au moins k est acyclique.

 $\delta^+ \geq k$, $\delta^+ \geq \ell$: Poly, symétrique de ($\delta^- \geq k$, $\delta^- \geq \ell$).
 $\delta^+ \geq k$, *cycle factor* : Poly, symétrique de ($\delta^- \geq k$, *cycle factor*).
 $\delta^+ \geq k$, $\leq \ell$ arcs : Trivial, voir 2.1.
 $\delta^+ \geq k$, $\geq \ell$ arcs : Poly, symétrique de ($\delta^- \geq k$, $\geq \ell$ arcs).
 $\delta^+ \geq k$, équilibré : Trivial, voir 2.1.
 $\delta^+ \geq k$, eulérien : NPC, voir 3.1.
 $\delta^+ \geq k$, *cycle* : Poly, symétrique de ($\delta^- \geq k$, *cycle*).
cycle factor, *cycle factor* : Poly, voir 2.3.
cycle factor, $\leq k$ arcs : Poly, revient à trouver un cycle factor et vérifier si $m \leq k + n$.
cycle factor, $\geq k$ arcs : Poly, revient à trouver un cycle factor et vérifier si $m \geq k + n$.
cycle factor, équilibré : Poly, revient à décider si D admet un cycle factor et est équilibré.
cycle factor, eulérien : NPC, voir 3.1.
cycle factor, *cycle* : NPC, voir 3.1.
 $\leq k$ arcs, $\leq \ell$ arcs : Poly, revient à décider si $m \leq k + n$.
 $\leq k$ arcs, $\geq \ell$ arcs : Trivial, voir 2.1.
 $\leq k$ arcs, équilibré : Poly, voir [12].
 $\leq k$ arcs, eulérien : NPC, voir [12].
 $\leq k$ arcs, *cycle* : NPC, voir 3.1.
 $\geq k$ arcs, $\geq \ell$ arcs : Poly, revient à décider si $m \geq k + n$.
 $\geq k$ arcs, équilibré : Trivial, voir 2.1.
 $\geq k$ arcs, eulérien : NPC, voir 3.1.
 $\geq k$ arcs, *cycle* : Poly, revient à trouver un cycle de taille minimum.
équilibré, équilibré : Poly, revient à déterminer si D est équilibré.
équilibré, eulérien : Poly, revient à déterminer si D est eulérien.
équilibré, *cycle* : Poly, revient à déterminer si D est équilibré et admet au moins un arc.
eulérien, eulérien : NPC, voir 3.1.
eulérien, *cycle* : NPC : si le graphe donné est eulérien, le problème revient à résoudre le problème de (*fortement connexe*, *cycle*). La réduction depuis 3-SAT vers (*fortement connexe*, *cycle*) proposée dans [7] construit un graphe eulérien, et constitue donc une réduction de 3-SAT vers (*eulérien*, *cycle*).

cycle, *cycle* : NPC, voir 3.1.