# Algorithmic and structural results on directed cycles in dense digraphs

Jocelyn Thiebaut

# THÈSE POUR OBTENIR LE GRADE DE DOCTEUR DE L'UNIVERSITÉ DE MONTPELLIER

**En Informatique**

**École Doctorale : I2S**
**Unité de recherche : AIGCo**

# *Algorithmic and Structural Results on Directed Cycles in Dense Digraphs*


**Présentée par Jocelyn THIEBAUT**
**Le 19 Novembre 2019**

**Sous la direction de Stéphane BESSY**


**Devant le jury composé de :**

| | | | |
|---|---|---|---|
| Stéphane BESSY | MCF | Université de Montpellier | Directeur de Thèse |
| Pierre CHARBIT | MCF | Université Paris 7 | Rapporteur |
| Anders YEO | Pr. | University of Southern Denmark | Rapporteur |
| Vincent BERRY | Pr. | Université de Montpellier | Président du jury |
| Eun Jung KIM | CR. | Université Paris-Dauphine | Examinatrice |
| Frédéric HAVET | DR. | Université Nice Sophia-Antipolis | Examinateur |
| Marin BOUGERET | MCF | Université de Montpellier | Invité |

UNIVERSITÉ DE MONTPELLIER

# Contents

# Introduction en Français

The following chapter is a French translation of Chapter 1.

Ce chapitre a pour but de donner quelques intuitions sur les bases de la Théorie des Graphes par le spectre de la thématique de cette thèse : les cycles orientés. Pour cela, nous donnons ici quelques exemples ludiques indépendants les uns des autres.

## À la recherche de petits cycles disjoints

Dans la cours de récréation, six élèves (Amandine, Bruno, Clément, Danielle, Emma et Félix) s'apprêtent à déguster leur goûter. Amandine a trois abricots, Bruno a un paquet de biscuits, Clément une barre chocolatée, Danielle quelques dattes, Emma un éclair à la vanille et, enfin, Félix a des figues. Cependant, leur plat respectif ne leur convient pas tous. Par exemple, Amandine n'aime pas tellement les abricots, et préférerait les biscuits de Bruno. Le hasard fait que Bruno, à l'inverse, préfère les abricots aux biscuits : les deux élèves peuvent échanger leur goûter.

Malheureusement, cette situation ne se produit pas tout le temps : Clément aimerait bien avoir les dattes de Danielle, mais celle-ci n'est pas intéressée par son chocolat. Pourtant, la situation est-elle nécessairement bloquée ? Après le refus de Danielle, Clément interroge ses camarades, et leur demande quels seraient les goûters qui pourraient les intéresser. Ils obtiennent le tableau suivant :

| | Abricots | Biscuits | Chocolat | Dattes | Éclair | Figues |
|---|---|---|---|---|---|---|
| Amandine | | ✓ | | | | |
| Bruno | ✓ | | | ✓ | | |
| Clément | | | | ✓ | | |
| Danielle | ✓ | | | ✓ | | ✓ |
| Emma | ✓ | ✓ | | | | ✓ |
| Félix | ✓ | | ✓ | | | ✓ |

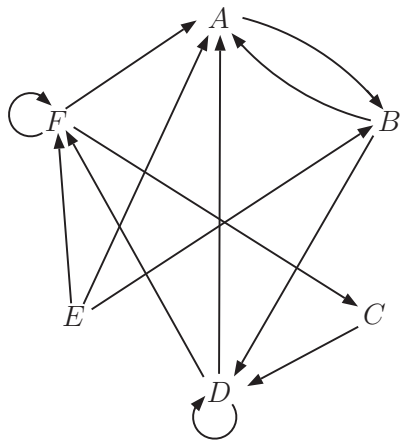
**Table 1** – Tableau résumant les goûts des différents élèves.

Par exemple, d'après le tableau, Félix aime les abricots, le chocolat et ses propres figues. Amandine, quant à elle, n'aime que les biscuits de Bruno.

Une autre façon de représenter ce tableau, plus visuelle, est de placer un point par personne, et de relier deux points par une flèche si l'un des deux veut le goûter de l'autre en fixant une convention pour l'orientation de la flèche (par

exemple ici, on aurait une flèche d'Emma vers Bruno, car Emma est *intéressée* par les biscuits de Bruno[1], mais il n'y a pas de flèche de Bruno vers Emma car l'inverse n'est pas vrai).

Une fois dessiné, et en simplifiant le nom des élèves par leur initiale, on obtient le schéma suivant :




En mathématiques et en informatique, on appelle ce genre de dessins un *graphe*, et plus précisément ici, un *graphe orienté* (ou encore *digraphe*). Les "points" sont appelés *sommets*, et les "flèches" sont appelées *arcs*.

On peut remarquer que, dans le graphe précédent, on peut avoir un arc qui va d'un sommet vers lui-même (c'est notamment le cas pour les sommets $D$ et $F$). Dans ce cas, cela signifie juste que les personnes correspondantes à ces sommets (Danielle et Félix) aiment leur propre goûter. On observe également que le fait que Amandine et Bruno aient la possibilité de faire un échange correspond dans le schéma à une séquence d'arcs *orientés dans le bon sens*[2] et qui revient au même sommet.

Grâce à cette dernière observation, on peut se convaincre qu'à chaque fois qu'il y aura cette succession d'arcs *orientés dans le bon sens* bouclant sur au moins deux sommets on pourra faire un échange entre les personnes correspondantes. On appelle une telle structure un *cycle orienté*. Par exemple, on a un cycle orienté entre les sommets $A$, $B$ et $D$ (on a bien les arcs de $A$ vers $B$, de $B$ vers $D$, et de $D$ vers $A$) ; on note ce cycle $(A, B, D)$. Remarquons que le cas mentionné plus tôt—où un arc va d'un sommet vers lui-même—est un cas particulier où le cycle contient un unique sommet. Dans ce cas, on parle de *boucle*.

Étant tous bienveillants, les élèves souhaitent satisfaire un maximum de monde. Par satisfaire, on entend que le plus possible d'entre eux aient quelque chose à manger qui leur plaisent. En terme de graphe, cela signifie que l'on doit trouver

[1] Ce choix d'orientation est arbitraire, on aurait pu décider d'orienter la flèche dans l'autre sens. Il faut juste que ce choix d'orientation soit respecté tout le temps lorsqu'on dessine le schéma.

[2] Par "orientés dans le bon sens", on entend ici qu'il ne faut pas qu'il y ait deux arcs consécutifs dans cette séquence qui pointent vers le même sommet. Par exemple, si l'on considère les sommets $E$, $F$ et $A$, on a certes un arc de $E$ vers $F$ et de $F$ vers $A$, mais l'arc entre $A$ et $E$ est orienté dans le mauvais sens.

des cycles afin de favoriser les échanges. Cependant, certains cycles ne peuvent pas être considérés en même temps. Par exemple, on ne peut pas avoir en même temps le cycle $(A, B, D)$ et le cycle $(C, D, F)$. En effet, dans le premier cycle, Danielle est censée donner ses dattes à Amandine, et dans le second Danielle doit donner ses dattes à Félix. Autrement dit, Danielle donne ses dattes deux fois : c'est impossible (sans triche).

Ainsi, il faut trouver un ensemble de cycles qui ne s'intersectent pas deux à deux, c'est-à-dire qui ne partagent pas de sommet en commun. De tels cycles sont dits *sommet-disjoints*. On souhaite également qu'un maximum de sommets soit touché par ces cycles car, hormis le cas des boucles, la plupart des personnes ont besoin d'un échange pour être satisfaits.

Par exemple, la collection de cycle $\{(A, B, D), (F)\}$ est bien sommet-disjointe, et touche quatre sommets. Elle satisfait donc quatre élèves. Une autre solution pourrait être la collection $\{(A, B), (D), (F)\}$ satisfaisant également quatre élèves. Cependant, il existe une meilleure option : la collection $\{(A, B), (C, D, F)\}$ satisfait une personne de plus. Peut-on faire mieux ? Emma est la seule qui n'est pas satisfaite après les échanges correspondant à la dernière collection de cycles. Malheureusement pour elle, on peut s'apercevoir qu'aucun arc ne *va vers* le sommet $E$[3]. En d'autres termes, personne n'est intéressé par l'éclair à la vanille d'Emma. Elle ne pourra donc jamais avoir un échange la satisfaisant. La collection $\{(A, B), (C, D, F)\}$ est donc une solution de taille maximum. Les élèves peuvent alors effectuer leurs échanges, puis savourer leur goûter[4] avant de retourner en cours.

Essayer de trouver un maximum de cycles sommets-disjoints (appelé *packing de cycles*) est un problème fondamental en théorie des graphes et central dans le cadre cette thèse. Avant de voir les aspects théoriques de ce problème, correspondant au corps de cette thèse, penchons nous sur d'autres exemples applicatifs du *packing* de cycles orientés.

Hormis ce petit exemple précédent anecdotique, une application concrète du *packing* de cycles correspond à l'optimisation des dons de reins [10, 26]. En effet, un patient ayant besoin d'une greffe de rein a parfois un proche prêt à lui en donner un. Cependant, des problèmes d'incompatibilité immunologique empêchent souvent le don. Heureusement, il est parfois possible de trouver un ou plusieurs autres couples de patients/donneurs avec lesquels un échange est possible. Formellement, on définit un digraphe où les sommets correspondent au *couple* patient/donneur, et on a un arc entre le couple $A$ et le couple $B$ si le *donneur* de $A$ est compatible avec le *patient* de $B$[5]. Dès lors, de la même façon que pour le cas des goûters, un cycle correspond à un *échange* de rein entre les couples patients/donneurs. Par exemple, si on a le cycle $(A, B, C)$, cela signifie que le donneur de $A$ peut donner son rein au patient de $B$, le donneur de $B$ peut donner son rein au patient de $C$, et que le donneur de $C$ peut donner au patient

[3]Dans le Tableau 1, on peut le voir aussi car la colonne "Éclair" est vide.

[4]Devant le désarrois d'Emma, Amandine lui donne un des biscuits qu'elle vient d'échanger, Bruno offre à Emma un de ses abricot, et Félix lui propose une de ses figues.

[5]En réalité, on ajoute généralement des poids sur les arcs afin de représenter la compatibilité entre le donneur de $A$ et le patient de $B$.

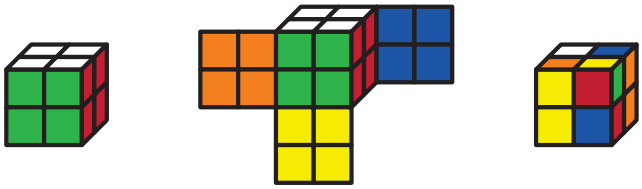


**Figure 1** – De gauche à droite : une vue d'un Rubik's Cube dans son état $\mathcal{S}$, la même vue où sont également représentées les couleurs des faces non-visibles, et une vue d'un Rubik's Cube dans un état mélangé.

de $A$. Au final, tous les patients ont reçu un rein, même s'il ne vient pas *stricto sensus* de son proche.

Cependant, une telle solution a des limites. Puisqu'il n'est pas possible légalement d'établir un contrat pour un organe, toutes les opérations doivent être faites en simultanées. Remarquons de plus qu'un cycle avec 3 sommets correspond en réalité à 6 opérations chirurgicales (pour chaque couple patient/donneur, il y a une opération pour prélever le rein du donneur, et une opération pour la greffe). Plus généralement, un cycle avec $k$ sommets nécessite $2k$ opérations. Ainsi, il est difficile d'un point de vue logistique de chercher des cycles trop grands. On cherche donc un *packing* de "petits" cycles.

## Un grand cycle unique

Joy a récemment reçu un Rubik's Cube $2 \times 2 \times 2$. Pour être exact, c'est en réalité *sa grande sœur* qui a eu un Rubik's Cube pour son anniversaire. Pour rappel, il s'agit d'un casse-tête en forme de cube, composé lui-même de huit cubes plus petits. Les faces du cubes peuvent tourner indépendamment les unes des autres grâce à un axe central. À l'état initial, que nous appellerons $\mathcal{S}$, tous les carrés d'une même face ont la même couleur, et il y a une couleur différente pour chaque face. Avec ces mouvements de rotations que permet l'axe central, les couleurs "se mélangent" et le but du casse-tête est de pouvoir remettre le cube dans son état $\mathcal{S}$. Voir la Figure 1 pour un exemple de différentes vues de Rubik's Cubes.

Ce jeu, Joy le comprend très vite, demande à la fois de l'intelligence et de la dextérité. Malheureusement, à force de rester sur son ordinateur, Joy a développé un syndrome du canal carpien, ce qui l'empêche de faire pivoter les faces dans le sens trigonométrique... Il comprend tout de même que, malgré cet handicap, il possède encore six possibilités pour modifier son cube à chaque fois, c'est-à-dire une pour chaque face[6]. Rapidement, il décide de dessiner le schéma correspondant à la Figure 2.

Ce dessin regroupe deux choses :

- au centre, une vue du cube dans son état $\mathcal{S}$ (les lignes de couleur à côté des arêtes du cube correspondent à la couleur de la face non-visible adjacente) ;

[6]Ici, on considère qu'on ne peut faire des rotations que de 90°, mais techniquement on pourrait faire des rotations de 180°, 270° (voire 360°), mais cela revient à faire deux, trois, (voire quatre) fois de suite le même mouvement.

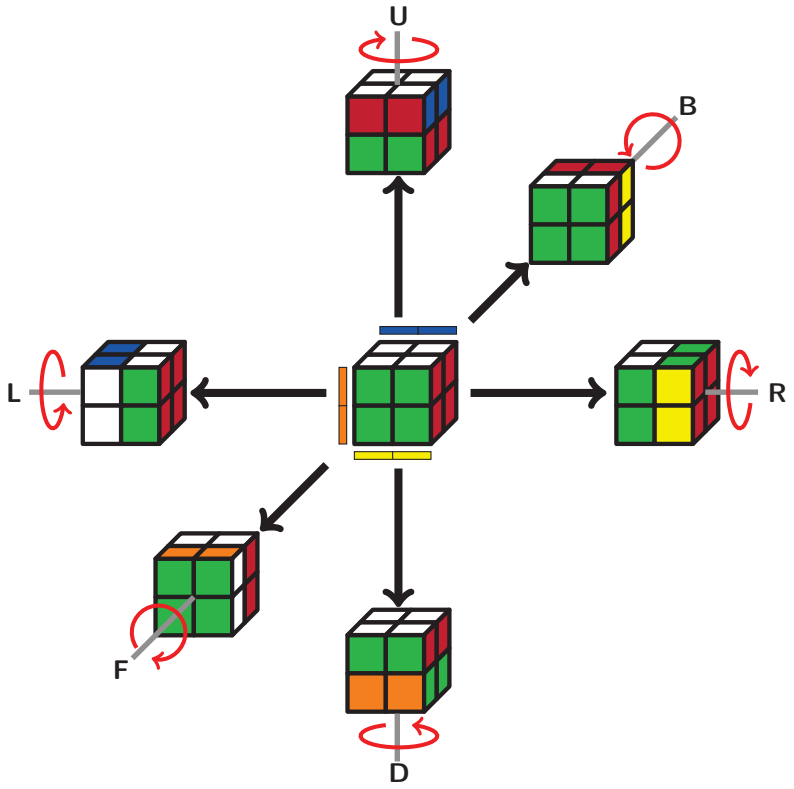



**Figure 2** – Digraphe des états atteignables depuis l'état $\mathcal{S}$ avec les six mouvements **U**, **B**, **R**, **D**, **F** et **L**.

- tout autour, une vue du cube obtenu après chaque rotation possible à partir de l'état $\mathcal{S}$. En partant du cube le plus en haut et en tournant dans le sens horaire on a effectué depuis l'état du centre les mouvements suivants[7] : tourner la face du haut (**U**), la face de derrière (**B**), la face de droite (**R**), la face du bas (**D**), la face avant (**F**), la face de gauche (**L**).

Joy se rend ensuite compte qu'il pourrait répéter ce processus : chacun des nouveaux états possède lui aussi six états qui sont atteignables à partir de lui en effectuant un des mouvements possibles décrits précédemment. Une fois de plus, il s'agit d'un graphe orienté. Ici, les sommets correspondent aux différents états du Rubik's Cube, et on a un arc entre deux états si l'on peut passer du premier au second en effectuant, à partir du premier, une des six rotations de face **U**, **B**, **R**, **D**, **F**, ou **L**.

Rappelons que le but du casse-tête est, étant donné un cube mélangé, de le remettre dans la position où les faces sont homogènes (l'état $\mathcal{S}$). Joy sait qu'il existe des méthodes pour résoudre ce problème "facilement". Malheureusement, il sait qu'il n'est jamais très doué pour appliquer à la lettre ce genre de techniques et préférerait développer sa propre astuce, quitte à être moins efficace. De plus,

[7] Rappelons qu'il s'agit à chaque fois du sens horaire défini lorsqu'on a la face en rotation à l'avant. Le sens de rotation est donné dans la figure par les flèches rouges, et la face qui tourne est celle qui est indiquée par le trait gris.

comme mentionné précédemment, il ne s'agit pas *stricto sensus* de son Rubik's Cube, mais celui de sa sœur, et elle lui a répété au moins un million de fois qu'≪il n'a pas intérêt à toucher ses affaires ≫... Autrement dit, une fois qu'il aura *malgré tout* touché le Rubik's Cube, Joy devra absolument le remettre dans le même état que celui dans lequel il l'a trouvé, afin d'éviter un drame cataclysmique...

Avec la modélisation décrite précédemment, on peut s'interroger sur la *signification* d'un cycle dans un tel graphe. Plus tôt, dans le graphe des goûters, nous avions vus qu'un cycle correspondait à une succession d'échanges entre plusieurs personnes afin de les satisfaire. Ici, puisque les arcs représentent un moyen de passer d'un état de cube à un autre, un cycle correspond à un retour à l'état auquel on était avant le début de la séquence de mouvement. Ainsi, il suffit de trouver pour chacun d'entre eux une succession d'arc (de mouvements) afin d'arriver vers l'état $\mathcal{S}$ (résolvant le problème du Rubik's Cube), et une autre succession de mouvement replaçant le Rubik's Cube dans son état initial (résolvant le "problème" de la sœur) : un solution pour Joy serait de chercher un cycle passant au moins par ces deux sommets.

Cependant, cela nécessiterait potentiellement de se souvenir, pour chaque état, du cycle correspondant pour le résoudre, et Joy souhaiterait n'avoir qu'une chose à retenir. Pour cela, l'astuce est de trouver un cycle dans le graphe des états passant par *tous* les états possibles. Dès lors, à chaque fois que le cube est mélangé, puisque le cycle trouvé passe par tous les états, il passe en particulier par l'état $\mathcal{S}$. Il suffit alors de suivre les arcs du cycle—c'est-à-dire effectuer les mouvements de base correspondant—à partir du sommet correspondant à l'état du Rubik's Cube mélangé jusqu'à arriver à l'état $\mathcal{S}$, puis de finir la séquence du cycle afin de revenir à l'état initial[8].

En pratique, cette solution est évidemment difficilement réalisable. Hormis la contrainte de mémorisation non négligeable (il y a plus de 3 millions d'états possibles[9]), le principal obstacle est qu'un cycle passant par tous les sommets d'un graphe n'existe pas toujours. Par exemple, si l'on considère à nouveau le graphe des goûters de l'exemple précédent, on constate qu'un tel cycle n'existe pas. Heureusement pour Joy, il semblerait qu'un tel cycle puisse être bel et bien trouvé pour ce graphe. En bon théoricien, satisfait qu'une solution existe, Joy abandonne son Rubik's Cube sans jamais le résoudre réellement (au plus grand plaisir de sa sœur).

[8]En réalité, il n'est pas nécessaire de se "placer sur le cycle". À cause de la structure particulière du digraphe des états du Rubik's Cube qui est très symétrique, on peut s'apercevoir que cette succession de mouvements peut en fait être exécutée à partir de n'importe quel autre sommet, c'est-à-dire de n'importe quel autre état initial : on obtient alors un autre cycle passant par tous les sommets, et donc en particulier par l'état $\mathcal{S}$. En d'autres termes, cette *unique succession* de mouvements résoudra *toutes* les configurations de Rubik's Cube... avant de le remettre dans son état initial.

[9]Il y a 8! permutations possibles des cubes composant le Rubik's Cube, avec $3^7$ orientations possibles à chaque fois. L'orientation globale du Rubik's Cube n'ayant pas d'importance, il faut diviser le résultat par $4 \times 6$. Au total, on obtient 3674160 états possibles.

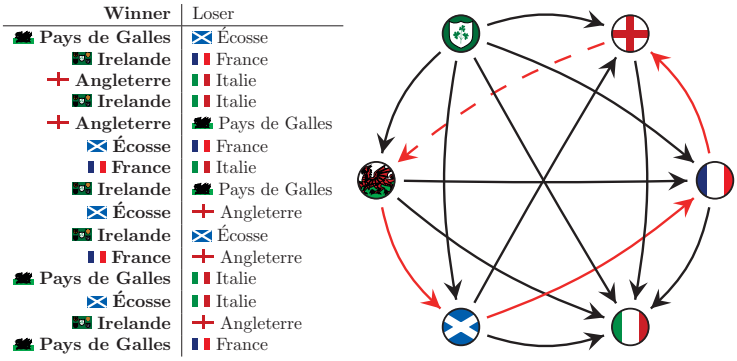



**Figure 3** – Résultats du Tournoi des Six Nations 2018 de rugby. On met un arc du pays $A$ vers le pays $B$ si $A$ a remporté son match contre $B$. On peut remarquer qu'il y a plusieurs cycles, dont celui représenté en rouge. Enfin, observons que l'on peut couper tous les cycles en retirant l'arc en pointillé.

## Les classes de digraphes

Il est assez rare de pouvoir prouver des théorèmes importants sur tous les digraphes. De ce fait, il est courant en informatique théorique de se restreindre à l'étude de classes spécifiques. Ainsi, en considérant une famille de digraphes particulière, il est généralement possible de bénéficier de la "structure" inhérente à celle-ci afin de prouver des résultats plus puissants.

Dans cette thèse, nous présentons des résultats dans des classes de digraphes *denses*, c'est-à-dire ceux qui possèdent un grand nombre d'arcs[10]. En particulier, nous nous intéressons aux *tournois*. Il s'agit d'une classe extrêmement riche de digraphes, et sans aucun doute celle qui a été le plus étudiée, notamment pour les systèmes de vote ou de classement. Formellement, les tournois sont les digraphes où il y a exactement un arc entre chaque paire de sommets distincts.

Une façon simple d'appréhender la notion de tournoi est de s'imaginer une compétition où chaque candidat est amené à affronter exactement une fois chacun de ses concurrents. On peut alors représenter le résultat de la compétition à l'aide d'un digraphe : on définit un sommet par joueur, et on met un arc du joueur $u$ vers le joueur $v$ si $u$ a remporté sa partie contre $v$. Comme mentionné précédemment, on a ainsi un digraphe avec exactement un arc entre chaque paire de sommets distincts : c'est un tournoi.

Une fois la compétition terminée, on souhaite établir le classement final des joueurs. Pour un tel classement, il est raisonnable *a priori* d'espérer que si le joueur $u$ est mieux placé dans le classement que $v$, alors nécessairement $u$ a remporté sa partie contre $v$ ; le classement étant censé représenter un ordre de prédominance entre les joueurs, donc si $u$ est classé *avant* $v$, $u$ est censé être

[10] Il existe plusieurs définitions pour la notion de densité dans les digraphes, nous utilisons ici une version simplifiée, c'est-à-dire lorsque le nombre d'arcs est en $\Omega(n^2)$, où $n$ est le nombre de sommets.

*meilleur* que $v$. Malheureusement, les classements vérifiant cette propriété, dits *transitifs*, existent rarement. En effet, en pratique il y a souvent des résultats paradoxaux. Considérons par exemple le cas où le joueur $u$ gagne contre le joueur $v$, ayant lui-même a gagné contre $w$, et que $w$ a remporté sa partie contre $u$. On constate qu'il est impossible de pouvoir dire dans ce cas qui de $u$, $v$, ou $w$ est le meilleur[11], et encore moins de faire un classement. On remarque également qu'une telle situation correspond à un cycle orienté $(u, v, w)$ de taille trois dans le tournoi considéré. Lorsque le cycle est de taille trois, on parle de *triangle*, mais un cycle de n'importe quelle taille aboutirait au même résultat paradoxal. La Figure 3 représente les résultats du Tournoi des Six Nations 2018. Dans cet exemple, on peut remarquer que l'Irlande (respectivement l'Italie) a remporté (respectivement perdu) toutes ses rencontres. Dès lors, il est naturel que l'Irlande et l'Italie soient à la première et dernière place du classement, respectivement. À l'inverse, on peut trouver plusieurs cycles dans ce tournoi, à savoir (Angleterre, Pays de Galles, Écosse, France), (Angleterre, Pays de Galles, France) et (Angleterre, Pays de Galles, Écosse). En d'autres termes, le classement entre ces pays est plus compliqué.

Il existe plusieurs moyens de résoudre cette problématique. L'un d'entre eux consiste à enlever un ou plusieurs arcs du digraphe afin de "casser" tous les cycles. On appelle un tel sous-ensemble d'arcs un *feedback arc set*. Dans l'exemple de la Figure 3, il est possible de casser tous les cycles en enlevant l'arc (Angleterre, Pays de Galles)[12]. Dans les faits, puisque retirer un arc correspond à ignorer le résultat d'une partie, on souhaite trouver le plus petit *feedback arc set*. Savoir calculer efficacement un tel sous-ensemble d'arcs est un problème fondamental en informatique et qui a été énormément étudié, notamment dans la classe des tournois. Il s'agit également d'un problème ayant un lien très fort avec plusieurs résultats de cette thèse.

## Des cycles dans les digraphes denses

Grâce aux petits exemples ludiques précédents, nous pouvons faire plusieurs observations. Premièrement, les (di)graphes sont un bon outil afin de représenter diverses relations binaires entre des éléments. Généralement, on utilise des digraphes pour représenter les relations qui ne sont pas symétriques. Ces modélisations sont donc très nombreuses. On peut, par exemple, donner des exemples évidents tels que les réseaux routiers, les réseaux sociaux, ou encore les télécommunications, mais il existe également de multiples domaines d'applications qui sont moins immédiats, comme la chimie, la physique, la génétique, l'économie, la médecine, la finance, le commerce, l'ingénierie...

Puisque les modélisations sont très nombreuses, la seconde observation que l'on peut faire est que la structure du graphe issue d'une d'entre elles n'a pas forcément le même intérêt. Si l'on se concentre notamment sur le concept de

[11] La situation est exactement la même dans le jeu Pierre-Feuille-Ciseau, où aucune des trois options n'est plus forte que l'autre.

[12] De façon informelle, on peut considérer que le Pays de Galles n'était pas "censé" perdre son match contre l'Angleterre.

cycle dans un graphe, nous avons vu que sa signification n'était pas la même dans les exemples précédents. Pourtant, la structure en elle-même, *la notion de cycle*, est identique dans les deux cas. Ainsi, on a été amené à chercher une collection de *petits* cycles dans l'exemple des greffes de rein, un *unique* grand cycle dans pour le cas du Rubik's Cube, et on a cherché à *casser* les cycles dans le dernier exemple.

Enfin, de nombreux problèmes de graphes peuvent être vus comme de petits problèmes amusants à résoudre. Il s'agit souvent de petites énigmes mathématiques visuelles et même si, certes, peu de gens sont amenés à trouver un algorithme paramétré de complexité $O^*(4^k k^5 k!)$ pour Feedback Arc Set [115], le concept de *feedbac arc set*, quant à lui, est compréhensible par un grand nombre, car très *visuel*.

## Aperçu des résultats et organisation de la thèse

**Définitions** Avant toutes choses, le Chapitre 2 contient les principales notions et définitions utilisées lors de cette thèse.

**État de l'art** Le Chapitre 3 présente un état de l'art autour des cycles dans les digraphes. Nous y développons notamment les résultats et conjectures structuraux, ainsi que les résultats algorithmiques sur le *packing* de cycles dans les digraphes, et notamment sur son dual (d'un point de vue programmation linéaire), à savoir les *feedback sets*.

***Packing* de triangles sommets-disjoints dans les tournois** Le Chapitre 4 s'intéresse au *packing* de triangles sommets-disjoints dans la classe des tournois. En particulier, nous montrons que ce problème est $\mathsf{NP}$-complet, et qu'il n'admet pas de schéma d'approximation polynomial, sauf si $\mathsf{P} = \mathsf{NP}$.

Du point de vue de la complexité paramétrée, nous donnons un noyau en $O(m)$ sommets, où $m$ est la taille d'un *feedback arc set* donné. Enfin, nous prouvons également que le problème du *packing* de triangles sommets-disjoints dans les tournois ne peut pas admettre de noyau de taille $O(k^{2-\varepsilon})$, sauf si $\mathsf{NP} \subseteq \mathsf{coNP/poly}$.

Ce chapitre correspond à une partie des travaux effectués avec Stéphane Bessy et Marin Bougeret et publiés à *ESA 2017* [21]. Une version journal est en préparation.

***Packing* de cycles et triangles arc-disjoints dans les tournois** Dans le Chapitre 5, nous nous penchons sur la version *arc-disjointe* du problème précédent. Non mentionné jusque là, le *packing* de cycles arc-disjoints n'en demeure pas moins un problème important en théorie des graphes. Deux cycles sont arc-disjoints s'ils n'ont aucun arc en commun ; en revanche, ils peuvent s'intersecter sur un ou plusieurs sommets.

Dans ce chapitre, nous montrons que les problème de *packing* de cycles et de triangles arcs-disjoints demeurent $\mathsf{NP}$-complets dans les tournois. Nous donnons également un noyau linéaire en nombre de sommets avec le paramètre naturel et un algorithme $\mathsf{FPT}$ en $2^{O(k)} \cdot \mathrm{poly}(n)$ pour le *packing* de triangles dans les tournois.

Ce chapitre correspond aux travaux effectués en commun avec Stéphane Bessy et Marin Bougeret [22] et à une partie des résultats qui ont été présentés à *MFCS 2019* [20]. Une version journal est en préparation.

***Packing* de cycles disjoints dans les tournois *sparses*** Dans le Chapitre 6, nous revenons dans ce chapitre sur les versions sommets et arcs-disjointes du *packing* de cycles des chapitres 4 et 5. Cependant, nous considérons cette fois-ci une sous-classe de tournois ayant un *feedback arc set* qui est un couplage. De tels tournois sont dits *sparses*.

Dans ce chapitre, nous donnons quelques résultats positifs sur cette sous-classe de tournois. Nous montrons qu'il existe un algorithme $(1 + \frac{6}{c-1})$-approché

pour le problème du *packing* de triangles sommet-disjoints dans le cas où le *minspan* est au moins $c$. Nous donnons également un noyau linéaire en nombre de sommets pour les tournois *sparses*. Enfin, nous montrons que contrairement au cas sommet-disjoint, le *packing* de triangles arc-disjoints dans les tournois *sparses* peut être résolu en temps polynomial.

Ces résultats sont issus d'une collaboration avec Stéphane Bessy et Marin Bougeret et ont été présentés à *ESA 2017* [21] et à *MFCS 2019* [22].

**Cycles complémentaires de toute taille dans les bipartis complets réguliers** Dans le Chapitre 7, nous nous intéressons à un problème plus structurel dans les bipartis complets réguliers. Nous montrons en effet que les digraphes $D$ de cette classe possèdent pour tout $p$ pair avec $4 \leq p \leq |V(D)| - 4$ un cycle $C$ de taille $p$ tel que $D \setminus V(C)$ est hamiltonien, sauf si $D$ est isomorphe à un digraphe précis. Ceci répond positivement à une conjecture de Zhang *et al.* [157] datant de 1994.

Ce travail a été effectué avec Stéphane Bessy, et les résultats ont été présentés à *EuroComb 2017* [25]. Une version journal à été soumise.

# Chapter 1

# Introduction

The purpose of this chapter is to give some insights into the foundations of Graph Theory through the spectrum of the theme of this thesis: oriented cycles. For this reason, we give here some playful examples that are independent of each other.

### Looking for small disjoint cycles

In the playground, six pupils (Amandine, Bruno, Clement, Danielle, Emma and Félix) are about to enjoy their snacks. Amandine has three apricots, Bruno has a packet of biscuits, Clement a chocolate bar, Danielle a few dates, Emma an eclair with vanilla and, finally, Félix has some figs. However, not all of them are satisfied with their respective dishes. For example, Amandine doesn't like apricots that much, and would prefer Bruno's biscuits. By chance, Bruno, on the other hand, prefers apricots to biscuits: the two pupils can exchange their snacks.

Unfortunately, this situation does not happen all the time: Clement would like to have Danielle's dates, but she is not interested in her chocolate. Yet, is the situation necessarily blocked? After Danielle's refusal, Clement asked her classmates what snacks they might be interested in. They get the following table :

| | Apricot | Biscuits | Chocolate | Dates | Eclair | Figs |
|---|---|---|---|---|---|---|
| Amandine | | ✓ | | | | |
| Bruno | ✓ | | | ✓ | | |
| Clement | | | | ✓ | | |
| Danielle | ✓ | | | ✓ | | ✓ |
| Emma | ✓ | ✓ | | | | ✓ |
| Félix | ✓ | | ✓ | | | ✓ |

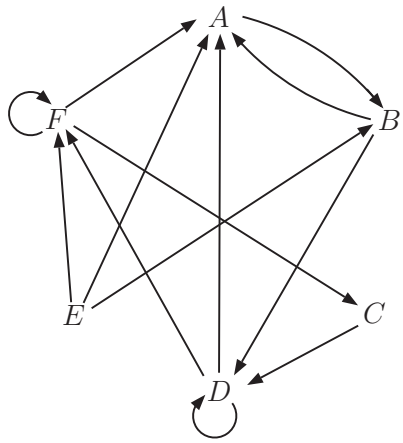
**Table 1.1** – Table summarizing the tastes of the different pupils.

For example, according to the table, Felix likes apricots, chocolate and his own figs. Amandine, on the other hand, only likes Bruno's biscuits.

Another way (and more visual way) to represent this table is to place one point per person, and to connect two points by an arrow if one of them wants to eat the other's snack, by fixing a convention for the orientation of the arrow (for

example, here, we would have an arrow from Emma to Bruno, because Emma is *interested* in Bruno's biscuits[1], but there is no arrow from Bruno to Emma because the opposite is not true).

Once drawn, and by simplifying the names of the pupils by their initial, we obtain the following diagram :




In mathematics and computer science, this type of drawing is called a *graph*, and more precisely here, a *oriented graph* (or *digraph*). The "points" are called *vertices*, and the "arrows" are called *arcs*.

We can notice that, in the previous graph, we can have an arc that goes from one vertex to itself (this is particularly the case for the vertices $D$ and $F$). In this case, it just means that the people corresponding to these vertices (Danielle and Félix) like their own snacks. We also observe that the fact that Amandine and Bruno have the possibility to make an exchange corresponds in the schema to a sequence of arcs *oriented in the right direction*[2] and coming back to the first vertex.

Thanks to this last observation, we can be convinced that each time there will be this succession of arcs *oriented in the right direction* on at least two vertices, we will be able to make an exchange between the corresponding people. Such a structure is called an *oriented cycle*. For example, we have a cycle oriented between the vertices $A$, $B$ and $D$ (we have the arcs from $A$ to $B$, from $B$ to $D$, and from $D$ to $A$); we note this cycle $(A, B, D)$. Note that the case mentioned earlier—where an arc goes from a vertex to itself—is a special case where the cycle contains a single vertex. In this case, it is called a *loop*.

Being all kind, the pupils want to satisfy as many people as possible. By satisfaction, we mean that as many of them as possible have something to eat that they like. In terms of graphs, this means that cycles must be found in order to encourage exchanges. However, some cycles cannot be considered at the

[1]This orientation choice is arbitrary, we could have chosen the other direction for the arrow. You just have to respect this choice of orientation all the time when you draw the diagram.

[2]By "oriented in the right direction", we mean here that there are not two consecutive arcs in this sequence that point to the same vertex. For example, if we consider the vertices $E$, $F$ and $A$, we do have an arc from $E$ to $F$ and from $F$ to $A$, but the arc between $A$ and $E$ is pointing in the wrong direction.

same time. For example, we cannot have both the cycle $(A, B, D)$ and the cycle $(C, D, F)$ simultaneously. Indeed, in the first cycle, Danielle is supposed to give her dates to Amandine, and in the second one, Danielle must give her dates to Félix. In other words, Danielle gives her dates twice: it's impossible (without cheating)

Thus, we must find a set of cycles that do not pairwise intersect, that is, that do not share a common vertex. Such cycles are called *vertex-disjoint*. We also want as many vertices as possible to be touched by these cycles because, except for loops, most people need an exchange to be satisfied.

For example, the collection of cycle $\{(A, B, D), (F)\}$ is indeed vertex-disjoint, and touches four vertices. It therefore satisfies four pupils. Another solution could be the $\{(A, B), (D), (F)\}$ collection, which also satisfies four pupils. However, there is a better option: the collection $\{(A, B), (C, D, F)\}$ satisfies one more person. Can we do better? Emma is the only one who is not satisfied after the exchanges corresponding to the last collection of cycles. Unfortunately for her, we can see that no arc goes *to* $E$ (only *from* $E$)[3]. In other words, no one is interested in Emma's eclair. She will therefore never be able to have a satisfying exchange. The collection $\{(A, B), (C, D, F)\}$ is therefore a maximum-sized solution. The pupils can then make their exchanges, and enjoy their new snacks[4] before going back to school.

Trying to find as many vertex-disjoint cycles as possible (called *cycle packing*) is a fundamental problem in graph theory and central in the context of this thesis. Before seeing the theoretical aspects of this problem, corresponding to the core of this thesis, let's look at other application examples of cycle packing.

Aside from this small anecdotal example, a concrete application of cycle packing corresponds to the optimization of kidney donation. [10, 26]. Indeed, a patient in need of a kidney transplant may have a relative who is willing to give him a kidney transplant. However, immunological incompatibility problems often prevent the donation. Fortunately, it is sometimes possible to find one or more other patient/donor pairs with whom an exchange is possible. Formally, we define a digraph where the vertices correspond to the patient/donor *pair*, and we have an arc between the pair $A$ and the pair $B$ if the *donor* of $A$ is compatible with the *patient* of $B$[5]. Therefore, in the same way as for snacks, a cycle corresponds to a *exchange* of kidneys between patient/donor couples. For example, if we have the cycle $(A, B, C)$, it means that the $A$ donor can give her kidney to the $B$ patient, the $B$ donor can give his kidney to the $C$ patient, and the $C$ donor can give the $A$ patient. In the end, all patients received a kidney, even if it didn't come from his relative.

However, such a solution has its limitations. Since it is not legally possible to establish a contract for an organ, all operations must be done simultaneously. It should also be noted that a cycle with 3 vertices actually corresponds to 6 surgical

[3] In the Table 1.1, we can also see it because the "Eclair" column is empty.

[4] In front of Emma's distress, Amandine gives her one of the biscuits she has just exchanged, Bruno offers Emma one of his apricots, and Félix offers her one of his figs.

[5] In reality, weights are usually added to the arcs to represent the compatibility between the donor of $A$ and the patient of $B$.

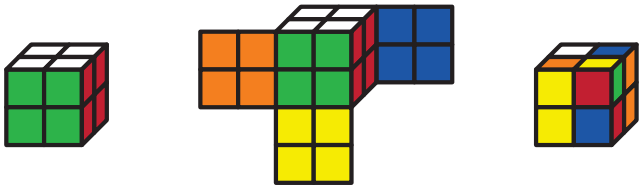


**Figure 1.1** – From left to right: a view of a Rubik's Cube in its state $\mathcal{S}$, the same view where the colors of unseen faces are also represented, and a view of a Rubik's Cube in a scrambled state.

operations (for each patient/donor couple, there is one operation to remove the kidney from the donor, and one operation for the transplant). More generally, a cycle with $k$ vertices requires $2k$ operations. Thus, it is logistically difficult to look for cycles that are too large. So we're looking for a *packing* of "small" cycles.

## A large unique cycle

Joy recently received a Rubik's Cube $2 \times 2 \times 2 \times 2$. To be exact, it was actually *her big sister* who got a Rubik's Cube for her birthday. As a reminder, this is a cube-shaped puzzle, itself made up of eight smaller cubes. The faces of the cubes can be rotated independently of each other thanks to a central axis. In the initial state, which we will call $\mathcal{S}$, all the squares on the same face have the same color, and there is a different color for each face. With these rotational movements allowed by the central axis, the colors "mix" and the purpose of the puzzle is to be able to restore the cube to its state $\mathcal{S}$. See Figure 1.1 for an example of several views of Rubik's Cubes.

Joy quickly understood that this game required both intelligence and dexterity. Unfortunately, by staying on her computer, Joy developed a carpal tunnel syndrome, which prevents her from rotating the faces in a trigonometric direction... She still understands that, despite this handicap, she still has six possibilities to modify her cube each time, that is to say, one for each face[6]. Quickly, she decides to draw the diagram corresponding to the Figure 1.2.

This figure contains two things:

- in the center, a view of the cube in its state $\mathcal{S}$ (the color lines next to the edges of the cube correspond to the color of the adjacent non-visible face);
- all around, a view of the cube obtained after each possible rotation from the state $\mathcal{S}$. Starting from the topmost cube and turning clockwise, we made (each time starting from $\mathcal{S}$) the following movements[7]: turn the upper face

[6]Here, we consider that we can only rotate by 90°, but technically we could rotate by 180°, 270° (or even 360°), but that amounts to doing two, three, (or even four) times the same movement in a row

[7]Remember that each time it is the clockwise direction defined when we have the rotating face at the front. The direction of rotation is given in the figure by the red arrows, and the rotating face is the one indicated by the gray line

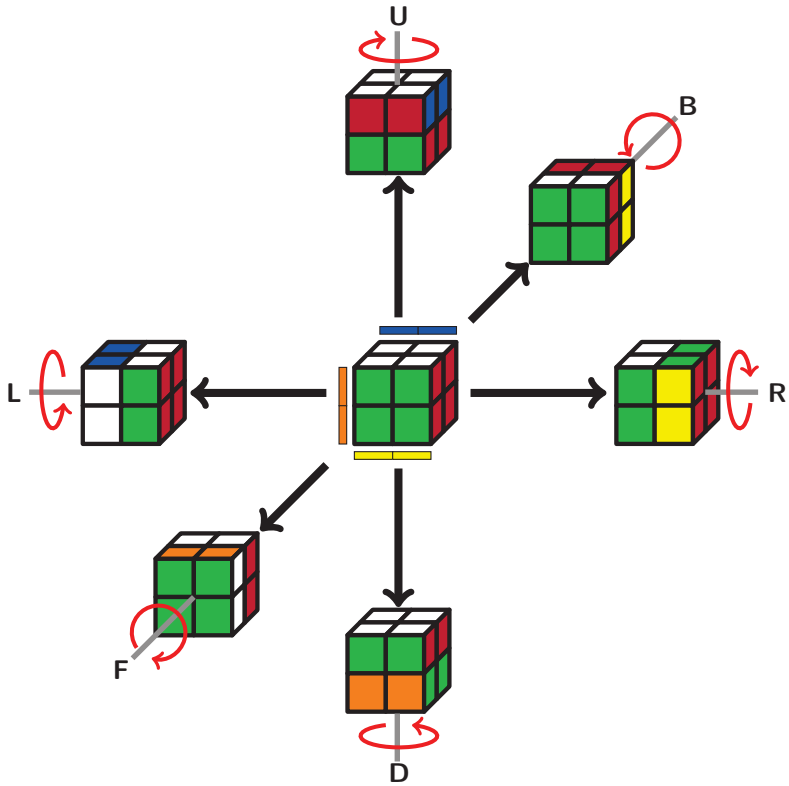



**Figure 1.2** – Digraph of the achievable states from $\mathcal{S}$ with the six movements **U**, **B**, **R**, **D**, **F**, and **L**.

(**U**), the back face (**B**), the right face (**R**), the down face (**D**), the front face (**F**), the left face (**L**).

Joy then realizes that he could repeat this process : each of the new states also has six additional states that can be reached from it by making one of the possible movements described above. Once again, this is an oriented graph. Here, the vertices correspond to the different states of the Rubik's Cube, and we have an arc between two states if we can move from the first to the second by making, from the first, one of the six face rotations **U**, **B**, **R**, **D**, **F**, or **L**.

Remember that the purpose of the puzzle is, given a scrambled cube, to put it back in the position where the faces are homogeneous (the state $\mathcal{S}$). Joy knows that there are methods to solve this problem "easily". Unfortunately, she knows that she is never very good at applying these kinds of techniques and would rather develop her own trick, even if it means being less effective. Moreover, as mentioned above, it is not strictly speaking her Rubik's Cube, but her sister's, and she has repeated to her at least a million times that ≪she better not touch her stuff ever again≫... In other words, once she will touch the Rubik's Cube (despite the threatening), Joy will absolutely have to put it back in the *exact* same state as the one she found it in, in order to avoid a cataclysmic disaster...

With the modeling described above, we can question the *meaning* of a cycle in such a digraph. Earlier, in the snack graph, we saw that a cycle corresponded to

a succession of exchanges between several people in order to satisfy them. Here, since arcs represent a way to move from one cube state to another, a cycle is a return to the state we were in before the beginning of the movement sequence. Thus, it is enough to find for each of them a succession of arcs (movements) in order to reach the state $\mathcal{S}$ (solving the Rubik's Cube problem), and another succession of movements restoring the Rubik's Cube in its initial state (solving the "sister problem"): a solution for Joy would be to look for a cycle going at least by these two vertices.

However, this would potentially require remembering, for each state, the corresponding cycle to solve it, and Joy would like to have only one thing to remember. To do this, the trick is to find a cycle in the graph of states going through *all* possible states. Therefore, each time the cube is scrambled, since the found cycle passes through all states, it passes in particular through the state $\mathcal{S}$. It is then sufficient to "follow" the arcs of the cycle—*i.e.* perform the corresponding basic movements—from the vertex corresponding to the state of the scrambled Rubik's Cube until reaching the state $\mathcal{S}$, then to finish the sequence of the cycle in order to return to the initial state[8].

In practice, this solution is obviously difficult to realize. Apart from the significant memorization constraint (there are more than 3 million possible states[9]), the main obstacle is that a cycle going through all the vertices of a graph does not always exist. For example, if we look again at the snack graph in the previous example, we see that such a cycle does not exist. Fortunately for Joy, it seems that such a cycle can be well found for this graph. As a good theorist, satisfied that a solution exists, Joy abandons her Rubik's Cube without ever really solving it (to her sister's great pleasure).

## The classes of digraphs

It is quite rare to be able to prove important theorems on all digraphs. As a result, it is common in theoretical computing to limit the study to specific classes. Thus, by considering a particular family of digraphs, it is generally possible to benefit from the inherent "structure" of the family in order to prove more powerful results.

In this thesis, we mainly present results in classes of *dense* digraphs, *i.e.* those with a large number of arcs[10]. In particular, we are interested in the *tournaments*. This is an extremely rich class of digraphs, and undoubtedly the one that has

[8]In reality, it is not necessary to " place the scrambled state on the cycle ". Because of the particular structure of the Rubik's Cube state digraph, which is very symmetrical, we can see that this succession of movements can actually be executed from any other vertex, that is, from any other initial state: we then obtain another cycle passing through all the vertices, and therefore in particular through the state $\mathcal{S}$. In other words, this *unique sequence* of movements will solve *all* Rubik's Cube configurations... before restoring it to its original state.

[9]There are 8! possible permutations of the cubes in the Rubik's Cube, with $3^7$ possible orientations each time. Since the overall orientation of the Rubik's Cube is not important, one has to divide the result by $4\times6$. In total, we obtain 3674160 possible states.

[10]There are several definitions for the notion of density in digraphs, we use here a simplified version, *i.e.* when the number of arcs is in $\Omega(n^2)$, where $n$ is the number of vertices.

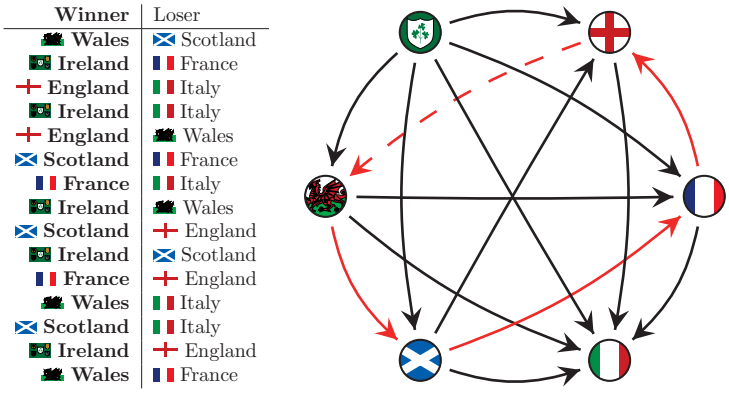



**Figure 1.3** – Result of the 2018 Six Nations Championship of rugby. We put an arc from a country $A$ to a country $B$ if $A$ won its match against $B$. Note that we have several cycles, including the cycle depicted in red. Note also that cutting the dashed arc (England, Wales), makes the tournament acyclic.

been most studied, especially for voting or rating systems. Formally, tournaments are digraphs where there is exactly one arc between each pair of distinct vertices.

A simple way to understand the notion of a tournament is to imagine a competition where each candidate has to face each of their competitors exactly once. The result of the competition can then be represented using a digraph: a vertex is defined per player, and a arc is placed from the player $u$ to the player $v$ if $u$ has won her game against $v$. As mentioned above, we have a digraph with exactly one arc between each pair of distinct vertices: it is a tournament.

Once the competition is over, we want to establish the final ranking of the players. For such a ranking, it is reasonable to expect that if the $u$ player is better placed in the ranking than $v$, then necessarily $u$ has won his game against $v$; the ranking being supposed to represent an order of predominance between players, so if $u$ is ranked *before* $v$, $u$ is supposed to be *better* than $v$. Unfortunately, rankings verifying this property, called *transitive*, rarely exist. Indeed, in practice there are often paradoxical results. Consider for example the case where the $u$ player wins against the $v$ player, having himself won against $w$, and $w$ won his game against $u$. We can see that it is impossible to say in this case which of $u$, $v$, or $w$ is the best[11], and even less possible to make a ranking. We also note that such a situation corresponds to an oriented cycle $(u, v, w)$ of size three in the considered tournament. When the cycle is size three, we talk about *triangle*, but a cycle of any size would lead to the same paradoxical result. See Figure 1.3 which depicts the results of the 2018 Six Nations Championship of rugby. In this example, notice that Ireland (resp. Italy) won all its matches (resp. lost all its matches); it is easy to rank Ireland and Italy to the first and last place,

[11]The situation is exactly the same in the Rock–Paper–Scissors game, where none of the three options is stronger than the other.

respectively. Conversely, we can find several cycles in the tournament, that is (England, Wales, Scotland, France), (England, Wales, France) and (England, Wales, Scotland). In other words, the ranking between these countries is therefore more intricate.

There are several ways to solve this problem of par. One of them consists in removing one or more arcs from the digraph in order to "break" all cycles. Such a subset of arcs is called a *feedback arc set*. In fact, since removing an arc ignores the result of a game, we want to find the smallest *feedback arc set*. In the example depicted in Figure 1.3, we can break all the cycles by removing the arc (England, Wales)[12]. Being able to calculate such a subset of arcs efficiently is a fundamental problem in computing and has been studied extensively, especially in the tournament class. It is also a problem with a very strong link to several of the results of this thesis.

## The structure of cycles in dense digraphs

Thanks to the small playful examples mentioned above, we can make several observations. First, (di)graphs are a good tool to represent various binary relationships between elements. Generally, digraphs are used to represent relationships that are not symmetrical. These models are therefore very numerous. For example, obvious examples can be given such as road networks, social networks, or telecommunications, but there are also multiple areas of application that are less immediate, such as chemistry, physics, genetics, economics, medicine, finance, commerce, engineering....

Since there are many different models, the second observation we can make is that the structure of the graph resulting from one of them does not necessarily have the same interest. If we focus in particular on the concept of the cycle in a graph, we have seen that its meaning is not the same in the previous examples. However, the structure itself, *the notion of cycle*, is identical in both cases. Thus, we were led to look for a collection of *small* cycles in the kidney transplantation example, a *unique* large cycle in the Rubik's Cube one, and we tried to *break down* the cycles in the last example.

Finally, many graph problems can be seen as small and fun problems to solve. These are often small visual mathematical riddles and even if, certainly, few people are led to find a parameterized algorithm of complexity $O^*(4^k k^5 k!)$ for Feedback Arc Set [115], the concept of *feedbac arc set*, as for it, is understandable by a large number, because very *visible*.

[12]Informally, one could understand that Wales was not "supposed" to lose against England.

## Overview of the results and organization of the thesis

**Definitions** First of all, the Chapter 2 contains the main concepts and definitions used in this thesis.

**State of the art** The Chapter 3 presents the state-of-the-art about cycles in digraphs. We develop in particular the structural results and conjectures, as well as the algorithmic results on the packing of cycles in digraphs, and in particular on its dual (from a linear programming point of view), namely the *feedback sets*.

**Packing of vertex-disjoint triangles in tournaments** The Chapter 4 is involved in the packing of vertex-disjoint triangles in tournaments. In particular, we show that this problem is NP-complete, and that it does not admit a polynomial approximation scheme, unless P = NP.

From the parameterized complexity point of view, we give a kernel in $O(m)$ vertices, where $m$ is the size of a given feedback arc set. Finally, we also prove that the problem of the packing of vertex-disjoint triangles in tournaments cannot admit a kernel size of $O(k^{2-\varepsilon})$, unless $NP \subseteq \mathsf{coNP/poly}$.

This chapter corresponds to some of the work done with Stéphane Bessy and Marin Bougeret and published at *ESA 2017* [21]. A journal version is in preparation.

**Packing of arc-disjoint triangles and arc-disjoint cycles in tournaments** In Chapter 5, we focus on the arc-disjoint version of the previous problem. Not mentionned so far, the arc-disjoint packing of cycles is also a very important problem in Graph Theory. Two cycles are said to be arc-disjoint if they do not share the same arc; however, they can intersect on one or more vertices.

In this chapter, we prove that the packing of arc-disjoint cycles and the packing of arc-disjoint triangles are NP-complete in tournaments. We also give a kernel with a linear number of vertices with the natural parameter, and an FPT-algorithm in $2^{O(k)} \cdot \text{poly}(n)$ running time for the packing of arc-disjoint triangles.

This chapter corresponds to the work done jointly with Stéphane Bessy and Marin Bougeret [22] and part of the results that were accepted at *MFCS 2019* [20]. A journal version is in preparation.

**Packing of cycles in sparse tournaments** In Chapter 6, we consider one more time both the vertex and arc-disjoint packing of cycles of chapters 4 and 5. However, we consider there a subclass of tournaments admitting a feedback arc set which is a matching. Such tournaments are called *sparse*.

In this chapter, we give some positive results for this sub-class of tournaments. We show that there is an $(1 + \frac{6}{c-1})$-approximation algorithm for the packing of vertex-disjoint triangles in sparse tournaments with minspan at least $c$. We also give a kernel with a linear number of vertices for this problem. Finally, we show that unlike the vertex-disjoint version, the packing of arc-disjoint cycles (and triangles) can be solved in polynomial-time.

These results are the fruit of a collaboration with Stéphane Bessy and Marin Bougeret and were presented at *ESA 2017* [21] and *MFCS 2019* [22].

**Complementary cycles of any length in regular bipartite tournaments** In Chapter 7, we focus on a structural problem in regular bipartite tournaments. Indeed, we show that for every digraph $D$ in this class and any $p$ even with $4 \leq p \leq |V(D)| - 4$, $D$ has a cycle $C$ of size $p$ such that $D \setminus V(C)$ has an Hamilton cycle, unless $D$ is isomorphic to a special digraph. This answer positively to a conjecture of Zhang *et al.* [157] of 1994.

This work was done with Stéphane Bessy, and the results were presented at *EuroComb 2017* [25]. A journal version is submitted.

# Chapter 2

# Definitions

In this chapter, we give define the preliminaries needed for the thesis. One can find at page 140 an index containing all the vocabulary and notations used.

## 2.1 Mathematical Standard Preliminaries

The set of the integers is denoted by $\mathbb{N}$, and the set of the non-negative integer by $\mathbb{N}^+$. Given two integers $i$ and $j$ with $i \leq j$, $[\![i, j]\!]$ stands for the set $\{k \in \mathbb{N} \colon i \leq k \leq j\}$.

Given two functions $f$ and $g$ from $\mathbb{N}$ to $\mathbb{N}$, and $n \in \mathbb{N}$, we say that $f(n)$ is $O(g(n))$ (written $f(n) \in O(g(n))$) if there exists two integers $c$ and $n_0$ such that $f(n) \leq c \cdot g(n)$ for all $n > n_0$. Similarly, we say that $f(n)$ is $o(g(n))$ (written $f(n) \in o(g(n))$) if for every $\varepsilon > 0$, there exists an integer $n_0$ such that $f(n) \leq \varepsilon \cdot g(n)$ for all $n > n_0$.

## 2.2 Directed Graphs

### 2.2.1 Basic Notions

A *directed graph* $D$ (or *digraph* for short) is an ordered pair $(V(D), A(D))$ where $V(D)$ denotes a set of elements, called *vertices*, and $A(D)$ a set of ordered pair of $V(D)$, called *arcs*. Thus, we say that $V(D)$ is the *vertex set* and $A(D)$ is the *arc set* of $D$. The order of a directed graph is the number of its vertices, that is $|V(D)|$, usually denoted by $n(D)$ or $n$ for short.

Given an arc $a = (u, v)$, we say that $u$ is the *tail* and $v$ is the *head* of $a$; these two vertices are said to be *adjacent*. We denote by $t(a)$ and $h(a)$ the vertices $u$ and $v$, respectively. Furthermore, we denote by $V(a)$ the set $\{u, v\}$. A digraph is *simple* if we cannot have two arcs with the same tail and head (*parallel arcs*) or an arc with the its head equals to its tail (*loop*).

Moreover, we say that $v$ is an *out-neighbor of* $u$, while $u$ is an *in-neighbor of* $v$. Given a vertex $u$ of $V(D)$, the *out-neighborhood of* $u$, denoted by $N_D^+(u)$, is given by the set $\{v \in V(D) \colon (u, v) \in A(D)\}$. Symmetrically, the *in-neighborhood of* $u$, denoted by $N_D^-(u)$, is the set $\{v \in V(D) \colon (v, u) \in A(D)\}$. The *out-degree* $d_D^+(u)$ of a vertex $u$ is the size of its out-neighborhood, that is $|N_D^+(u)|$ and its *in-degree*

$d_D^-(u)$ is the size of its in-neighborhood. The *maximum out-degree* of a digraph $D$, denoted $\Delta^+(D)$, is given by $\max\{d_D^+(v)\colon v \in V(D)\}$ while the *maximum in-degree* of $D$, denoted $\Delta^-(D)$, is $\max\{d_D^-(v)\colon v \in V(D)\}$. Symmetrically, we define the *minimum out-degree* $\delta^+(D)$ and *minimum in-degree* $\delta^-(D)$ of $D$ by $\min\{d_D^+(v)\colon v \in V(D)\}$ and $\min\{d_D^-(v)\colon v \in V(D)\}$ respectively. Additionally, if $\delta^+(D) = \delta^-(D) = \Delta^-(D) = \Delta^+(D)$ we say that a digraph is *regular*; in other words, the vertices have their out-degree and in-degree all equal to the same value, say $k$ (in that case, we say that $D$ is *$k$-regular*).

Given two subsets of vertices $X$ and $Y$, if there is all the arcs from $X$ to $Y$, then we say that $X$ *dominates* $Y$. Let $D$ be a loopless digraph of order $n$. An *ordering* $\sigma(D)$ of its vertices is a $n$-tuple (ordered list of size $n$). If $\sigma(D)$ is an ordering given by $(v_1, \ldots, v_n)$, then an arc $(v_i, v_j) \in A(D)$ is a *forward arc* of $\sigma(D)$ when $i < j$, for $i, j \in [\![1, n]\!]$. Otherwise, $(v_i, v_j)$ is a *backward arc* of $\sigma(D)$.Furthermore, a *topological ordering* of a digraph $D$ is an ordering $\sigma(D)$ such that there is no backward arc. Notice that such ordering may not exist for some digraphs.

The *complement* of a digraph $D$, denoted $\overline{D}$, corresponds to the digraph with vertex set $V(D)$ and arc set $\{(u, v)\colon (u, v) \notin A(D)\}$. A digraph $D'$ is a *subdigraph* of $D$ if $V(D') \subseteq V(D)$, $A(D') \subseteq A(D)$ and every arc of $A(D')$ has its head and tail in $V(D')$. Moreover, if we have $V(D') = V(D)$, then $D'$ is a *spanning* subdigraph. Let $X$ be a subset of vertices of $D$, the *induced subdigraph* $D[X]$ is given by $V(D[X]) = X$ and $A(D[X]) = \{(u, v) \in A(D)\colon u, v \in X\}$. Alternatively, if $A'$ is a subset of arcs of $D$, then the *arc-induced subdigraph* $D[A']$ is given by $A(D[A']) = A'$ and $V(D[A']) = \{u \in V(D)\colon \exists v \in V(D)$ such that $(u, v) \in A'$ or $(v, u) \in A'\}$. Given a subset $X$ of vertices, the subdigraph $D \setminus X$ is obtained by removing from $V(D)$ the vertices in $X$, and removing in $A(D)$ the arcs with a tail or a head in $X$. In other words, $D \setminus X = D[V(D) \setminus X]$. Similarly, if $A'$ is a subset of arcs, $D - A'$ is the subdigraph with vertex set $V(D)$ and arc set $A(D) - A'$.

We say that two digraphs $D_1$ and $D_2$ are *isomorphic* if there exists a bijection $\varphi : V(D_1) \to V(D_2)$ such that, for every ordered pair $x, y$ of vertices in $D_1$, $(x, y)$ is an arc of $D_1$ if and only if $(\varphi(x), \varphi(y))$ is an arc of $D_2$.

In the previous notations, we often omit the name of the corresponding graph as long as it does not lead to ambiguity; for example, we might write $N^+(u)$ instead of $N_D^+(u)$ for the out-neighborhood of $u$ or simply $V$ instead of $V(D)$.

### 2.2.2 Some Structures

A subset of vertices $X$ is an *independent set* (or *stable set*) if there is no arc joining any two vertices of $X$.

Given a digraph $D$ and a set $\{v_1, \ldots, v_t\}$ of $t$ disjoint vertices of $D$, we say that $v_1, \ldots, v_t$ is a *directed path* of $D$ if $(v_i, v_{i+1}) \in A(D)$ for any $i \in [\![1, t-1]\!]$. The *length of a path* is the number of its vertices, that is $t - 1$ in the previous definition. Moreover, given two vertices $u$ and $v$, a *$u, v$-path* denotes a *directed path* starting from $u$ and finishing on $v$. The *distance from $u$ to $v$* is the length of a shortest $u, v$-path. The *size of a path* is its number of vertices.

A digraph $D$ is *strongly connected* (or simply *strong*) if, for any pair $u$ and $v$ of distinct vertices, there exists a $u, v$-path. A *strong component* of $D$ is a maximal subset of vertices $X$ such that $D[X]$ is strong. If a digraph $D$ is not strong, it is well-known that $D$ can be partition into $t$ strong subdigraphs $D_1, \ldots, D_t$ with $t > 1$. For $i \in [\![1, t]\!]$, the subdigraph $D_i$ is an *initial* (respectively *terminal*) *strong component* if there is no arcs $(u, v)$ of $D$ with $u$ a vertex of $D - V(D_1)$ and $v$ a vertex of $D_1$ (respectively $v$ a vertex of $D - V(D_1)$ and $u$ a vertex of $D_1$). Otherwise, if $D_i$ is neither an initial nor a terminal strong component, it is an *intermediate strong component*.

Furthermore, $D$ is said to be *$k$-strongly connected* if it contains at least $k + 1$ vertices, but does *not* contain a set $X$ of at most $k - 1$ vertices such that $D \setminus X$ is not strong.

Given a digraph $D$ and a $v_1, v_t$-path of size $t$, if in addition we have the arc $(v_t, v_1) \in A(D)$, then $(v_1, \ldots, v_t)$ is a *directed cycle* of size $t$, also called a $t$-cycle. A 2-cycle (that is two vertices $u$ and $v$ such that $(u, v)$ and $(v, u)$ are arcs of the digraph) is called a *digon*, and a 3-cycle a *triangle*. Given a cycle $C$ , we denote by $V(C)$ the set of its vertices, and $A(C)$ the arcs of the cycle. Moreover, a directed path (respectively directed cycle) which passes through each vertices of a digraph exactly once is a *Hamilton path* (respectively *Hamilton cycle*). In the following, since we mainly focus on digraphs, we will simply write *cycle* and *path* instead of directed cycle and directed path, respectively.

The cycles $C$ and $C'$ are *vertex-disjoint cycles* (or simply *disjoint cycles*) if they do not share common vertices. Given a digraph $D$, a *packing of vertex-disjoint cycles* $\mathcal{C}$ of $D$ is a set $\{C_i \colon 1 \leq i \leq |\mathcal{C}|\}$ where each $C_i$ is a cycle and such that $V(C_i) \cap V(C_j) = \emptyset$ for every $i \neq j$. Furthermore, a *packing of vertex-disjoint triangles* is a packing of vertex-disjoint cycles containing only triangles. Alternatively, two cycles are *arc-disjoint cycles* if they do not have an arc in common. Given a digraph $D$, a *packing of arc-disjoint cycles* $\mathcal{C}$ of $D$ is a set $\{C_i \colon 1 \leq i \leq |\mathcal{C}|\}$ where each $C_i$ is a cycle and such that $A(C_i) \cap A(C_j) = \emptyset$ for every $i \neq j$. Furthermore, a *packing of arc-disjoint triangles* is a packing of arc-disjoint cycles containing only triangles. Notice that two disjoint cycles are necessarily arc-disjoint cycles, but the opposite is not true. We denote by $\nu_0(D)$ (resp. $\nu_1(D)$) the maximum number of vertex-disjoint cycles (resp. arc-disjoint cycles) of a digraph $D$.

A *cycle factor* of a digraph is a partition of its vertex set whose parts are pairwise vertex-disjoint cycles. For some positive integer $k$, a *$k$-cycle factor* is a cycle factor of $k$ vertex-disjoint cycles; it can alternatively be considered as a partition of the digraph into $k$ Hamilton subdigraphs. In particular, a 1-cycle factor is an Hamilton cycle.

When a digraph does not contain any cycle, we say that it is *acyclic*. Given a digraph $D$, a subset of vertices $X$ is a *feedback vertex set* if the subdigraph $D \setminus X$ is acyclic. The size of a minimum feedback vertex set of $D$ is denoted by $\mathrm{fvs}(D)$. A *feedback arc set* of $D$ is the arc version of the previous definition, that is a subset of arcs $A'$ such that $D - A'$ is acyclic. The size of a minimum feedback arc set of $D$ is denoted by $\mathrm{fas}(D)$.In the following, we say that a feedback arc set $A'$ is a matching if there is no two arcs of $A'$ sharing a common endpoint.

### 2.2.3 Some Classes

A digraph such that there is exactly one arc between every pair of distinct vertices is a *tournament*. If we have at least one arc between each pair of distinct vertices (that is digons are allowed), then it is a *semi-complete digraph*. Finally, if there is a digon for each pair of distinct vertices, it is a *complete digraph*. Notice that, for a positive integer $n$, all the complete digraphs of size $n$ are pairwise isomorphic while two tournaments of size $n$ may not be isomorphic.

A tournament $T$ can alternatively be defined by an *ordering* $\sigma(T)$ of its vertices and a set of *backward arcs* $\overleftarrow{A}_\sigma(T)$ regarding this ordering. For shorthand sake, the backward arc set might be denoted by $\overleftarrow{A}(T)$ whenever the considered ordering is not ambiguous. The ordered pair $(\sigma(T), \overleftarrow{A}(T))$ is a *linear representation* of the tournament $T$. Additionally, $\{(i,j), 1 \leq i < j \leq n\colon\ (j,i) \notin \overleftarrow{A}(T)\}$ are the *forward arcs* of the ordering. One can notice that $\overleftarrow{A}(T)$ is a feedback arc set of $T$. Furthermore, we say that a tournament is *sparse* if it admits a feedback arc set which is a matching. Notice that if a tournament $T$ is sparse with regards to the feedback arc set $A'$, then computing in polynomial time the topological ordering of $T - A'$ gives the ordering $\sigma(T)$ where $\overleftarrow{A}_\sigma(T)$ is a matching.

A digraph $D$ is *bipartite* if its vertex set can be partitioned into two independent sets. Thus, it is easy to see that a bipartite digraph cannot contain cycles with an odd size. A *bipartite tournament* $B$ is a digraph with independent sets $X$ and $Y$ and with exactly one arc for each pair of vertices $\{x, y\}$ with $x \in X$ and $y \in Y$.

For a digraph $D$, the *line digraph* $L(D)$ of $D$ is given by $V(L(D)) = A(D)$ and $A(L(D)) = \{(a,b)\colon\ a, b \in A(D), h(a) = t(b)\}$.

## 2.3 Some Undirected Graph Notions

In this thesis, we mainly deal with directed graphs. However, we need some undirected graph notions too. Therefore, we give here some basis definitions.

An *undirected graph* $G$ (or simply *graph*) is an ordered pair $(V(G), E(G))$ where $V(G)$ denotes a set of *vertices*, and $E(G)$ a set of *unordered* pair of $V(G)$, called *edges*. Thus, we say that $V(G)$ is the *vertex set* and $E(G)$ is the *edge set* of $G$. The order of a graph is the number of its vertices, that is $|V(G)|$ and also denoted by $n(G)$ or simply $n$ for short.

The two vertices $\{u, v\}$ of an edge $e$ are said to be *adjacent*. The vertices $u$ and $v$ are also said to be the *extremities* (or *endpoints*) of $e$.

Given a vertex $u$ of $V(G)$, the *neighborhood of* $u$, denoted by $N_G(u)$, is given by the set $\{v \in V(G)\colon\ \{u, v\} \in E(G)\}$. The *degree* $d_G(u)$ of a vertex $u$ is the size of its neighborhood, that is $|N_G(u)|$. Similarly, we extend this notation to sets: if $X$ is a set of vertices, $N_G(X)$ corresponds to the set $\bigcup_{v \in X} N_G(v)$.

A *matching* is a set of edges with no common endpoints. The *complement* of a graph $G$, denoted by $\overline{G}$, corresponds to the graph with vertex set $V(G)$ and edge set $\{\{u, v\}\colon\ \{u, v\} \notin E(G)\}$. A graph $G'$ is a *subgraph* of $G$ if $V(G') \subseteq V(G)$, $E(G') \subseteq E(G)$ and every edge of $E(G')$ has its extremity in $V(G')$. Moreover,

if we have $V(G') = V(G)$, then $G'$ is a *spanning* subgraph. Let $X$ be a subset of vertices of $G$, the *induced subgraph* $G[X]$ is given by $V(G[X]) = X$ and $E(G[X]) = \{\{u, v\} \in E(G) \colon u, v \in X\}$. Alternatively, if $E'$ is a subset of edges of $G$, then the *edge-induced subgraph* $G[E']$ is given by $E(G[E']) = E'$ and $V(G[E']) = \{u \in V(G) \colon \exists v \in V(G) \text{ such that } \{u, v\} \in E'\}$. Given a subset $X$ of vertices, the subgraph $G \setminus X$ is obtained by removing from $V(G)$ the vertices in $X$, and removing in $E(G)$ the edges with at least an extremity in $X$. In other words, $G \setminus X = G[V(G) \setminus X]$. Similarly, if $E'$ is a subset of edge, $G - E'$ is the subgraph with vertex set $V(G)$ and edge set $E(G) - E'$.

In the previous notations, we often omit the name of the corresponding graph, as long as it does not lead to ambiguity; for example, we might write $N(u)$ instead of $N_G(u)$ to denote the neighborhood of a vertex $u$ or simply $V$ instead of $V(G)$.

Given a graph $G$ and a set $\{v_1, \ldots, v_t\}$ of $t$ disjoint vertices of $G$, we say that $v_1, \ldots, v_t$ is a *undirected path* if $\{v_i, v_{i+1}\} \in E(G)$ for any $i \in [\![1, t-1]\!]$. The *length* of a path is the number of its edges, that is $t - 1$ in the previous definition and the *size* its number of vertices. Moreover, given two vertices $u$ and $v$, a *$u, v$-undirected path* denotes a *undirected path* starting from $u$ and finishing on $v$. The *distance from $u$ to $v$* is a length of the shortest $u, v$-undirected path.

Given a graph $G$ and a $v_1, v_t$-undirected path of size $t$, if in addition we have the edge $\{v_t, v_1\} \in E(G)$, then $v_1, \ldots, v_t$ is an *undirected cycle* of size $t$.

A *connected component* of $G$ is a maximal subdigraph $G'$ by inclusion such that for every pair of vertices $u$ and $v$ of $V(G')$ there exists a $u, v$-undirected path in $G$. If the graph $G$ itself is a connected component, then $G$ is *connected*. Otherwise, we can see that $G$ can be partition into $t$ connected component.

When a graph does not contain any cycle, we say that it is *acyclic*. An acyclic graph is also called a *forest*. Furthermore, if a forest is connected, then it is a *tree*. The *complete graph* $K_n$ is the graph of order $n$ with exactly one edge between each pair of vertices.

## 2.4 Complexity Zoo

### 2.4.1 The P vs NP Problem

The purpose of this subsection is to give to the reader some notions about the classical complexity classes, namely P and NP. We refer the reader to the seminal book of Garey and Johnson [70] for the formal definitions of these classes, using languages and Turing machines.

**Problems and Algorithms** An *algorithmic problem* is a question raised about combinatorial objects (graphs, words, ...) called *instances*. We can see it as subset of all the possible instances (the positive instances). A *decision problem* $\Pi$ is a yes-or-no question such that, given an instance $I$, answers "yes" if $I \in \Pi$ (we say that $I$ is a "yes"-instance of $\Pi$), and "no" otherwise (we say that $I$ is a "no"-instance of $\Pi$). We say that an *algorithm* $A$ *solves* the decision problem $\Pi$ if $A$ is a set of instructions such that, for any instance $I$, returns "yes" (or,

alternatively, true) if and only if $I$ is a "yes"-instance of $\Pi$. If such an algorithm exists, we say that $\Pi$ is *decidable*[1]. In other words, $\Pi$ can be seen as the *set* of the "yes"-instances, while $A$ is a way to know if any given instance *is in* $\Pi$.

For example, we can define the problem $\Pi_{even}$ which asks whether the natural integer in input is even or no. Then, $\Pi_{even}$ is the set $\{2k\colon k \in \mathbb{N}\}$ and an algorithm solving $\Pi_{even}$ is the instruction `return (k%2 == 0)`[2].

$\Pi_{even}$
**Input:** $n$ a natural number.
**Output:** Is $n$ even?

Of course, the decision problems do not capture all the question we need to answer in everyday life. Its main limitation comes from the fact that it needs to be answered by "yes" or "no". Therefore, we can define optimization problems for which we have to find the *best solution* among a set of feasible solutions. More formally, an optimization problem $\Pi_o$ is a tuple $(\mathcal{I}, F, c, goal)$, where: $\mathcal{I}$ is a set of instances; given an instance $I$ in $\mathcal{I}$, $F(I)$ is the set of feasible solutions; given an instance $I$ in $\mathcal{I}$ and a feasible solution $s$ of $F(I)$, $c(I, s)$ represents the *cost* of $s$ (usually a positive real); *goal* is the goal function, and is either min (in that case, we say that $\Pi_o$ is a minimization problem) or max (in that case, we say that $\Pi_o$ is a maximization problem).
Then, a solution of $\Pi_o$ given an instance $I$ is to find a feasible solution $s$ such that $c(I, s) = goal\{c(I, s')\colon\ s' \in F(I)\}$. In other word, $s$ is one of the *best* solutions of $\Pi_o$. Then, similarly as before, we say that an algorithm $A$ solves the optimization problem $\Pi_o$ if $A$ is a set of instructions such that, for any instance $I$, returns one of the best solution of $\Pi_o$.

For example, let $\Pi_{maxdist}$ be the problem that, given a graph $G$ and two vertices $u$ and $v$ of $G$ computes the longest $u, v$-undirected path.

$\Pi_{maxdist}$
**Input:** A graph $G$ and two vertices $u$ and $v$ of $G$.
**Output:** The longest $u, v$-undirected path.

Thus, for the problem $\Pi_{maxdist}$, $\mathcal{I}$ is the set of all the graphs and pairs of vertices. Now, given an instance $I \in \mathcal{I}$ with $I = (G, u, v)$, an element of $S(I)$ can be seen as an undirected paths between the two vertices $u$ and $v$, say $(x_1, \ldots, x_t)$ for some $t$ and where the $x_i$'s are vertices of the graph $G$. Therefore, the cost of the solution $(x_1, \ldots, x_t)$ in the instance $I$ is given by the function $(I, (x_1, \ldots, x_t)) \mapsto t - 1$. The goal is to find a solution in $S(I)$ with maximum cost. An algorithm $A_{maxdist}$ solving the problem $\Pi_{maxdist}$ is thus a set of instructions—not described here—that returns a longest $u, v$-undirected path in $G$.

[1]Note that its existence is not mandatory; some problems can only admit algorithm that can say if $\Pi(x) = yes$, but not if $\Pi(x) = no$ (the problem is said to be *semidecidable*). If, there is no algorithm that can say if a given input *is* or *is not* in $\Pi$, then the problem is said to be *undecidable*.

[2]This instruction returns true if and if $k = 0 \pmod 2$.

**On how difficult a problem is** In computer science, one of the many ways to define the difficulty of an algorithm is to describe the amount of time it would take in the worst-case scenario. The time is given by counting the number of elementary operations performed by the algorithm[3]. This time is expressed depending of the size of the input using usually the big O notation defined in section 2.1. Therefore, given an input of size $n$, we say that an algorithm $A$ runs in $O(f(n))$ time if it takes at most $O(f(n))$ elementary operations to compute a solution. If in addition $f$ is a polynomial function, then we say that $A$ runs in polynomial time (or $A$ is a polynomial-time algorithm). Similarly, if a decision problem $\Pi$ can be solved by an algorithm $A$ which runs in polynomial time, we say that the problem $\Pi$ is a *polynomial problem.*

The set of the decision problems that can be solved in polynomial time by a (deterministic[4]) algorithm is denoted by $\mathsf{P}$. Such problems play a fundamental role in computer science, since they correspond to the problems that can be "solved easily in everyday life". Unfortunately, many of the useful problems are unlikely to be polynomial; thus we need to define a new class of problems.

The class $\mathsf{NP}$ is the set of decision problems $\Pi$ for which, given a "yes"-instance $I$ of $\Pi$, there exist a polynomial algorithm, called a *certificate*, which can check if $I \in \Pi$. In other words, the certificate *do not necessarily solves* the problem $\Pi$, but it can "quickly" guarantee that a solution is a "yes"-instance.

We can easily notice that all the problems in $\mathsf{P}$ are in $\mathsf{NP}$, since the polynomial algorithm solving any problem in $\mathsf{P}$ is also a certificate for $\mathsf{NP}$. Unfortunately, it is unknown if $\mathsf{NP} \subseteq \mathsf{P}$, which would lead to $\mathsf{P} = \mathsf{NP}$. If such an equality holds, it would imply that it as easy to solve a problem than to check the solution[5]. Therefore, most of the researchers consider that $\mathsf{P} \neq \mathsf{NP}$.

**$\mathsf{NP}$-hard and $\mathsf{NP}$-complete problems** Given two decision problems $\Pi$ and $\Pi'$, we say that $\Pi$ is *polynomially reducible* (or admits a *Karp reduction*) to $\Pi'$ if there is a polynomial algorithm $A$ that transforms any instance $I$ of $\Pi$ into an instance $A(I)$ of $\Pi'$, and such that the second instance has the same answer as the first one. In other words, this reduction proves that it is *at least harder* to solve $\Pi'$ than to solve $\Pi$, since if we find a polynomial algorithm $B$ to solve $\Pi'$, then we could find a polynomial algorithm to solve $\Pi$ by applying the algorithms $A$ and $B$ in the row.

A decision problem $\Pi$ is $\mathsf{NP}$-*hard* if *all* the problems in $\mathsf{NP}$ can be polynomially reduced to $\Pi$. This definition can be extended to optimization problems too. Notice that a problem $\Pi$ can be $\mathsf{NP}$-hard but not in $\mathsf{NP}$. If, in addition, the decision problem is in $\mathsf{NP}$, then we say that the problem is $\mathsf{NP}$-*complete*.

[3] We assume that such operations takes a constant amount of time.

[4] "Deterministic" means that the same input will give the same answer. We omit the formal definitions here.

[5] Anyone who ever tried to solve a Sudoku puzzle has probably already felt that it is way easier to *check* if their solution is good than to *solve* the puzzle.

### 2.4.2 Some Other Attack Plans...

In order to find solutions to the NP-hard problems, researchers developed new tools to "attack" these hard problems in more efficient ways.

#### 2.4.2.1 Approximation

One of these ways is to develop approximation algorithms. The main idea is to drop the optimality of the solution in exchange of a faster running time algorithm. See the survey of Papadimitriou and Yannakakis [127] for a more details on approximation algorithms.

Formally, let $A$ be an algorithm of a maximization (resp. minimization) problem $\Pi$. For $\rho \geq 1$, we say that $A$ is a *$\rho$-approximation* of $\Pi$ if and only if, for any instance $I$ of $\Pi$, $A(I) \geq \mathrm{opt}(I)/\rho$ (resp. $A(I) \leq \rho \cdot \mathrm{opt}(I)$) where $A(I)$ is the value of the solution returned by $A$ and $\mathrm{opt}(I)$ the value of a optimal solution of $I$. In other words, a $\rho$-approximation is an algorithm that compute a solution at most $\rho$ times worst that the best possible solution.

Now, we can define the *complexity class* APX as the problems $\Pi$ for which there exists a constant $\rho \geq 1$ such that $\Pi$ admits a $\rho$-approximation algorithm running in polynomial time. In other words, a problem for which it would need an exponential running time to compute an optimal solution, if this problem is in APX, it implies we can find a solution in polynomial time which would be *at least* $\rho$ times worst than the optimal solution.

Similarly, we can define the *complexity class* PTAS as the problems $\Pi$ for which for any $\varepsilon > 0$, there exists a polynomial $(1+\varepsilon)$-approximation of $\Pi$. In that case, it means that we can be as close as possible of the optimal solution; of course, the closer we want to be, the longer the running time will be.

There are multiple ways to prove the approximation class of a problem, one of them is to provide an $L$-reduction from an optimization problem to another one. We give here a formal definition.

**Definition 1** ( *L-reduction* ). *Let $\Pi$ and $\Pi'$ be two optimization (maximization or minimization) problems. We say that $\Pi$ L-reduces to $\Pi'$ if there are two polynomial-time algorithms $f$ and $g$, and two positive constants $\alpha$ and $\beta$ such that:*

*(i) for each instance $I$ of $\Pi$, the algorithm $f$ produces an instance $f(I)$ of $\Pi'$ such that the optima of $I$ (denoted by $\mathrm{opt}_\Pi(I)$) and $f(I)$ (denoted by $\mathrm{opt}_{\Pi'}(f(I))$) satisfy $\mathrm{opt}_{\Pi'}(f(I)) \leq \alpha \cdot \mathrm{opt}_\Pi(I)$,*

*(ii) given any solution $s$ of $f(I)$ of with cost $c_{\Pi'}(s)$, algorithm $g$ produces a solution $g(s)$ of $I$ with cost $c_\Pi(g(s))$ such that we have $|c_\Pi(g(s)) - \mathrm{opt}_\Pi(I)| \leq \beta \cdot |c_{\Pi'}(s) - \mathrm{opt}_{\Pi'}(I')|$.*

If a problem $\Pi$ admits an L-reduction from every problem in APX to $\Pi$, then $\Pi$ is said to be APX-*hard*[6]

[6]Formally, the set of the APX-hard problems are the problems which admit an other kind of reduction, called PTAS-reduction (and not detailed here), from every problem in APX to them. However, a L-reduction implies a PTAS-reduction.

**Theorem 1** (Folklore)**.** *No* APX*-hard problem has a* PTAS*, unless* P = NP*.*

#### 2.4.2.2 Parameterized complexity

Another way in order to solve NP-hard problems has been developed within the framework of parameterized complexity. We give here the main ideas of this scope and its basic definitions, we refer the reader to the survey of Downey and Fellows [59] for more details on parameterized complexity which includes all the formalism using languages.

Under the—likely—assumption that $\mathsf{P} \neq \mathsf{NP}$, we mentioned earlier that it would imply that some NP problems require exponential running time to solve when we measure the complexity with the size of the input. The purpose of parameterized complexity is to contain the superpolynomial running time growth not according to the size of the input anymore, but according to a—if possible small—parameter of the input. Thus, when it is possible and the parameter is small enough, one can expect to have efficient algorithms to solve the problem, called FPT-algorithms.

Therefore, in the framework of parameterized complexity, we have to consider a problem *with* a parameter (which can be the size of the solution, the maximum degree of the graph, etc...). Formally, a *parameterized problem* is a language $L \subseteq \{0,1\}^* \times \mathbb{N}$. For an instance $I = (x,k) \in \{0,1\}^* \times \mathbb{N}$, the integer $k$ is called the *parameter*.

A parameterized problem is *fixed-parameter tractable* (or FPT for short) if there exists an algorithm $A$, a computable function $f$, and a constant $c$ such that given an instance $I = (x,k)$, $A$ (called an FPT*-algorithm*) correctly decides whether $I \in L$ in time bounded by $f(k) \cdot \mathrm{poly}(n)$, where $n$ denotes the size of $I$. We also denote this running time by $O^*(f(k))$, that is we didn't considered the polynomial in $n$ factor.

Unfortunately, not all the NP-hard problems can admit an FPT-algorithm. Therefore, there exists an infinite collection of complexity to refine the difficulty classes of the problems, namely $\mathsf{W}[0], \mathsf{W}[1], \mathsf{W}[2], \ldots$, called the W-hierarchy. The formal definition of such a hierarchy is omitted here, but we have $\mathsf{FPT} = \mathsf{W}[0]$, and $\mathsf{W}[i] \subseteq \mathsf{W}[j]$ for all $i \leq j$. The question if $\mathsf{W}[i] = \mathsf{W}[j]$ for some $i$ and $j$ is an outstanding open question.

Another major field of parameterized complexity is to develop kernelization algorithms. The idea is to find efficient algorithms to reduce the size of the instance so that it only depends on the size of the parameter $k$. Formally, given a computable function $g$, a *kernelization algorithm* (or simply a *kernel*) for a parameterized problem $L$ of *size* $g$ is an algorithm $A$ that given any instance $I = (x,k)$ of $L$, runs in polynomial time and returns an *equivalent* instance $I' = (x',k')$ with $|I'| + k' \leq g(k)$.

It is well-known that the existence of an FPT algorithm is equivalent to the existence of a kernel (whose size may be exponential), implying that problems admitting a polynomial kernel form a natural subclass of FPT problems.

**Theorem 2** (Folklore)**.** *A problem is* FPT *if and only if it admits a kernelization algorithm and is decidable.*

*Proof.* First, let us consider the case where a problem $\Pi$ is $\mathsf{FPT}$. We can assume that there is at least one instance, called $(x_{yes}, k_{yes})$ that is in $\Pi$, and at least one instance, called $(x_{no}, k_{no})$ that is not in $\Pi$. Indeed, if it is not the case, we could easily create a kernelization algorithm by returning a trivial instance. Now, let $A$ be the algorithm that decides if any instance $(x, k)$ is in $\Pi$ in running time $O(f(k) \cdot |x|^c)$, for some computable function $f$, and some constant $c$. Let $A'$ be the algorithm that, given an input $(x, k)$, runs the previous algorithm $A$ on the same input for at most $|x|^{c+1}$ steps. If the algorithm $A$ ends before this amount of steps, return $(x_{yes}, k_{yes})$ or $(x_{no}, k_{no})$ depending if $(x, k)$ is in $\Pi$ or not. Otherwise, that is if the algorithm $A$ does not terminate before $|x|^{c+1}$ steps, then simply return $(x, k)$. It guarantees us that $O(f(k) \cdot |x|^c) > |x|^{c+1}$, proving that the size of the kernel is at most $O(f(x))$, and so computable since $f$ is. The algorithm $A'$ is a kernelization algorithm.

Now, given a decidable problem $\Pi$ which admits a kernelization algorithm $A$ of size $f$, and an input $(x, k)$ of $\Pi$, we can first use the algorithm $A$ to reduce the instance $(x, k)$ into an equivalent instance $(x', k')$ of size at most $f(k)$ in polynomial time. Then, we can use the algorithm that decides $\Pi$ to solve the kernel $(x', k')$ in running time time $O(g(f(k))$, for some computable function $g$. The total running time is $O(g(f(k)) \cdot \operatorname{poly} |x|)$, proving that $\Pi$ is $\mathsf{FPT}$.

□

# Chapter 3

# State of the Art

Given a field of scientific research, the important conjectures of this field give both a picture of the state of research at the present time, and foreshadow its future. Bondy [31] gave six main criteria to define a beautiful conjecture: simplicity (how many basic concepts needed to understand the conjecture); surprise; generality (the conjecture does not only apply on a restricted amount of object); centrality (being a part of the current research); longevity; fecundity (the proof of this conjecture could lead to new results or new proof techniques).

Bondy gives also in this article a list (and ratings according to his criteria) of "beautiful conjectures" in Graph Theory. In this chapter, we give a brief overview of some of the conjectures and results linked to cycles in digraphs[1]. We start by giving structural conjectures and their related results, and then we give the state of the art of algorithmic results about cycles and feedback sets in digraphs.

See [14, 32, 71, 72, 73] for more general conjectures and results about Graph Theory.

## 3.1 Structural Conjectures and Results

### 3.1.1 Vertex-disjoint Cycles in Digraphs

It is natural for a digraph with a large enough minimum out-degree to ask for particular structures in it, such as disjoint cycles for instance. First, Thomassen [143] proved in 1983 that there exists a function $f$ such that a digraph with minimum out-degree $f(\ell)$ contains at least $\ell$ vertex-disjoint cycles. In the same article, he proved that we have $f(\ell) \leq (\ell + 1)!$ for every integer $\ell$. Obviously, the goal is to reduce as much as possible this upper bound. In 1996, Alon [5] proved that the function $f$ is linear. In addition, using probabilistic method, he proved that $f(\ell) \leq 64\ell$. By considering the complete digraph on $2\ell - 1$ vertices, we can notice that this function is at least $2\ell - 1$. Bermond and Thomassen [18] conjectured that this bound is tight.

[1]Not having the claiming to rate them, some of the conjectures detailed below are also mentioned—and therefore rated—in the article of Bondy [31]. We leave the freedom to curious readers to look at themselves its ratings.

**Conjecture 1** (Bermond and Thomassen [18]). *For every positive integer $\ell \leq 1$, every digraph with minimum out-degree at least $2\ell - 1$ contains $\ell$ vertex-disjoint cycles.*

This conjecture is trivial for $\ell = 1$, and Thomassen [143] proved it for $\ell = 2$. A proof for $\ell = 3$ has been provided by Lichiardopol *et al.* [112]. Recently, a simpler proof of this latter case has been provided by Bai and Manoussakis [13].

In the specific class of tournaments, the problem has been proved by Bessy *et al.* [24].

**Theorem 3** (Bessy *et al.* [24]). *For every positive integer $\ell$, every tournament with minimum out-degree at least $2\ell - 1$ contains $\ell$ vertex-disjoint cycles.*

However, the problem of finding cycles in digraphs is significantly harder than in the undirected case. A hunch of this is to note that the previous conjecture becomes easy in undirected graphs. Indeed, it is trivially true for $\ell = 1$, and one can prove by induction that a graph $G$ with minimum degree at least $3\ell - 1$ contains $\ell$ disjoint vertex-cycles. To do so, let $c$ be cycle with minimum length in $G$. Now, consider the graph $G'$ induced by the vertices $V(G) \setminus V(c)$. If $c$ is a triangle, then the minimum degree of $G'$ guarantees that we can apply induction on $G'$ and thus find $\ell - 1$ vertex-disjoint cycles in $G'$; adding $c$ yields the result. Otherwise, if $c$ is not a triangle, one can easily see that every vertex of $G'$ is adjacent to at most three vertices of $c$ (otherwise, it contradicts the minimality of $c$). Therefore, we can also apply induction on $G'$, and thus find $\ell$ vertex-disjoint cycles.

Furthermore, by considering the complete graph of order $3\ell - 1$, notice that the bound of $3\ell - 1$ is optimal.

### 3.1.2 The Hamilton Decomposition of Tournaments

One of the most classic theorems about the cycles in tournaments is Camion's Theorem [38].

**Theorem 4** (Camion [38]). *A tournament is strong if and only if it is has a Hamilton cycle.*

This theorem has been widely improved since then, for instance by Moon [123] a couple of years later proving that strong tournaments are vertex-pancyclic (for every vertex of the digraph, there exist cycles of all possible length greater than 2 through it). But it shows the early interest about the structure of cycles in dense classes of digraphs. In addition, one can also mention that Walecki [9, 34, 116] proved in 1890 that the edges of the complete (undirected) graph with an odd number of vertices can be partitioned into edge-disjoint Hamilton cycles. We call such a partition a Hamilton decomposition of the graph.

**Theorem 5** (Walecki [9, 34, 116]). *The complete graph $K_n$ has a Hamilton decomposition if and only if $n$ is odd.*

In the directed case, a *Hamilton decomposition* is a partition of the arcs of a digraph into arc-disjoint Hamilton cycles. However, the directed analog of the

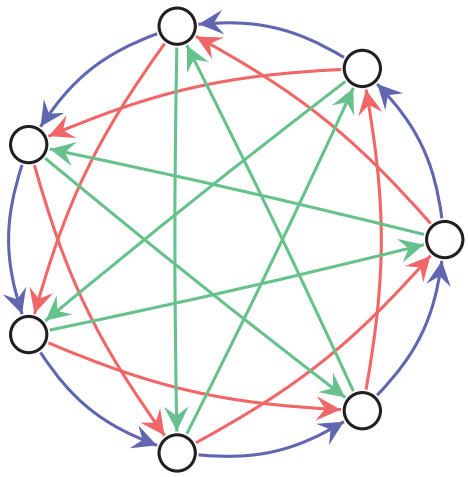


**Figure 3.1** – A 3-regular tournament with one Hamilton decomposition. Each set of arc of the same color forms a Hamilton cycle of the digraph.

previous theorem is a long standing conjecture, formulated by Kelly [122] in 1968.

**Conjecture 2** (Kelly [122])**.** *Every regular tournament has a Hamilton decomposition.*

See Figure 3.1 for an example of a Hamilton decomposition of a 3-regular tournament.

Tillson [145] characterized the complete digraphs admitting a Hamilton decomposition.

**Theorem 6** (Tilson [145])**.** *The complete digraph of order $n$ has a Hamilton decomposition if and only if $n \notin \{4, 6\}$.*

The Kelly's conjecture has been verified for $n \leq 9$ by Alspach [18]. In order to prove it, some results explored the structure of regular tournaments. For instance, one can mention Jackson [89] who proved that for every regular tournament on at least 5 vertices contains a Hamilton path and a Hamilton cycle which are arc-disjoint. Zhang [156] improved the previous result proving we can in fact find two arc-disjoint Hamilton cycles in a regular tournament of order is at least 5. Similarly, Wang and Cai [150] proved that every regular tournament on at least 15 vertices has three arc-disjoint Hamilton cycles.

Conjecture 2 has been studied by Thomassen [142], and Häggkvist [78]. Recently, Kühn and Osthus [107] have proved that the conjecture is verified for large enough tournaments. In fact, they prove a stronger result, since it can be extended to dense digraphs.

**Theorem 7** (Kühn and Osthus [107])**.** *For every $\varepsilon > 0$, there exists $n_0$ such that every $k$-regular oriented graph $G$ on $n \geq n_0$ vertices with $k \geq 3n/8 + \varepsilon n$ has a Hamilton decomposition. In particular, there exists $n_0$ such that every regular tournament on $n \geq n_0$ vertices has a Hamilton decomposition.*

Finally, Jackson [89] proposed the analogous conjecture for bipartite tournaments.

**Conjecture 3** (Jackson [89])**.** *Every regular bipartite tournament has a Hamilton decomposition.*

Note that some results around the Kelly's conjecture tried to consider almost regular tournaments by Thomassen [142]. However, one can easily see that there are almost regular bipartite tournaments without any Hamilton cycle: consider the bipartite tournament $B$ given by $V(B) = (X_1 \cup X_2) \bigcup (Y_1 \cup Y_2)$ with $|X_1| = |X_2| = n/4$, $|Y_1| = n/4-1$, and $|Y_2| = n/4+1$. If we set the arcs of $B$ such that $X_1$ dominates $Y_2$, $Y_2$ dominates $X_2$, $X_2$ dominates $Y_1$, and $Y_1$ dominates $X_1$, we can easily prove that for every vertex $v$, $|d^+(v) - d^-(v)| \leq 2$. However, we cannot find a Hamilton cycle of $B$ since the size of the in-neighborhood of $Y_2$ is smaller than its own size.

Thomassen [142] conjectured in 1982 that for each integer $k \leq 2$, there exists an integer $g(k)$ such that every $g(k)$-strongly connected tournament has $k$ arc-disjoint Hamilton cycles. He also proved that $g(2) > 2$ and that the function $g$ is not linear. Recently, Patel et. al [128] almost proved this conjecture. They prove that $g$ is at most in $O(k^2 \log^2 k)$, and give a family of $O(k^2)$-strongly connected tournaments without $k$ arc-disjoint Hamilton cycles.

**Theorem 8** (Patel et. al [128])**.** *For each integer $k \leq 2$, there exists $c > 0$ such that every $ck^2 \log^2 k$-strongly connected tournament has $k$ arc-disjoint Hamilton cycles. Furthermore, there exists a tournament $(k-1)^2/4$-strongly connected which does not have $k$ arc-disjoint Hamilton cycles.*

### 3.1.3 Small Cycles in Digraphs

Caccetta and Häggkvist [35] conjectured that the length of the smallest cycle of a digraph cannot be too large if the minimum out-degree the digraph is also large. This following conjecture, known as the Caccetta-Häggkvist Conjecture, is one of the most fundamental structural conjecture about cycles in digraphs.

**Conjecture 4** (Caccetta and Häggkvist [35])**.** *Every simple digraph of order $n$ with minimum out-degree at least $k$ has a cycle with length at most $\lceil n/k \rceil$.*

The statement of this conjecture was proved for different values of $k$. Namely, Caccetta and Häggkvist [35] gave the proof for $k = 2$. In 1987, Hamidoune [81] provide a proof for $k = 3$. Then, Hoàng and Reed [86] proved the cases where $k = 4$ and $k = 5$. Finally, Shen [138] proved the conjecture for $k \leq \sqrt{n/2}$.

**Theorem 9** (Shen [138])**.** *Every simple digraph of order $n$ with minimum out-degree at least $k$ and such that $n \geq 2k^2 - 3k + 1$ has a cycle with length at most $\lceil n/k \rceil$.*

Some approximate results exist to this conjecture; one of these approaches is to allow an additive constant to the statement of the conjecture. Formally, it

means that every simple digraph of order $n$ with minimum out-degree at least $k$ has a cycle with length at most $\lceil n/k \rceil + c$ for a small constant $c$. This simpler version was proved for $c = 2500$ by Chvátal and Szemerédi [44], then for $c = 304$ by Nishimura [125], and finally for $c = 73$ by Shen [139].

**Theorem 10** (Shen [139])**.** *Every simple digraph of order $n$ with minimum out-degree at least $k$ has a cycle with length at most $\lceil n/k \rceil + 73$.*

The case where $k = n/3$ has been widely studied, since it would imply that the digraph has a triangle. In this particular case, researchers focused on the multiplicative constant approximate version. Namely, let $\gamma$ be the smallest constant such that every simple digraph of order $n$ with minimum out-degree at least $\gamma n$ has a triangle.

It is easy to show that $\gamma \geq 1/3$. To do so, consider the digraph $D$ given by $V(D) = \{v_1, \ldots, v_{3k+1}\}$ and where $N^+(v_i) = \{v_{i+j \pmod{3k+1}} \colon 1 \leq j \leq k\}$. Let $c$ be a cycle of $D$. Without loss of generality, we may assume that the first vertex of $c$ is $v_1$. Then by construction, the next vertex of $c$ is $v_{1+j_1}$, for some $j_1 \in [\![1, k]\!]$. Similarly, the third vertex of $c$ is $v_{1+j_1+j_2}$ for some $j_2 \in [\![1, k]\!]$. We cannot have an arc from $v_{1+j_1+j_2}$ to $v_1$ since we have $3k + 1$ vertices. So $c$ has at least four vertices while $D$ has a minimum out-degree $k$ and $3k + 1$ vertices, implying $\gamma \geq 1/3$. Note that Figure 3.1 where we remove the green arcs corresponds to our previous example for $k = 2$.

If Conjecture 4 is true, it would imply that $\gamma = 1/3$. It has been proven by Caccetta and Häggkvist [35] that $\gamma \leq (3 - \sqrt{5})/2 \leq 0.382$. Then, Bondy [30] proved that $\gamma \leq (2\sqrt{6} - 3)/5 \leq 0.3798$. Shen [137] relaxed it to $\gamma \leq 3 - \sqrt{7} \leq 0.35425$. Finally, Hladký *et al.* [85] proved that $\gamma \leq 0.3465$.

**Theorem 11** (Hladký *et al.* [85] )**.** *Every simple digraph of order $n$ with minimum out-degree at least $0.3465n$ has a triangle.*

Notice that it is also unknown if a simple digraph with both minimum out-degree *and in-degree* at least $n/3$ has a triangle. Therefore, one can try to find the smallest constant $\beta$ such that every simple digraph of order $n$ with minimum out-degree and in-degree at least $\beta n$ has a triangle. It is easy to see that $\gamma \geq \beta \geq 1/3$, and it is conjectured that $\gamma = \beta = 1/3$. de Graaf *et al.* [50] originally proved that $\beta \leq 0.3487$. Next, Shen [137] showed that $\beta \leq 0.3477$. Then, Hamburger *et al.* [80] proved that $\beta \leq 0.34564$. The best known value so far is due to Lichiardopol [111] with $\beta \leq 0.343545$.

**Theorem 12** (Lichiardopol [111])**.** *Every simple digraph of order $n$ with minimum out-degree and minimum in-degree at least $0.343545n$ has a triangle.*

This special case where we consider both in-degrees and out-degrees is also related to the well-known Seymour's Second Neighborhood Conjecture[2] [51]. In the following, we define the second neighborhood of a vertex $v$, denoted by $N^{++}(v)$, as the set $(\bigcup_{x \in N^+(v)} N^+(x)) \setminus N^+(v)$, that is the set of the vertices at distance *exactly* 2 from $v$.

[2]The Seymour's Second Neighborhood Conjecture is not strictly in the topic of this thesis, but it deserves to be mentioned in view of its importance in the general field of directed graphs.

**Conjecture 5** (Seymour [51]). *Every digon-free digraph has a vertex $v$ such that $|N^+(v)| \leq |N^{++}(v)|$.*

Indeed, we can easily prove that this conjecture implies that every simple digraph with both minimum out-degree and in-degree at least $n/3$ has a triangle. To do so, assume that we can find a vertex $v$ with $|N^+(v)| \leq |N^{++}(v)|$, then as we also both minimum out-degree and in-degree at least $n/3$ , we get $|N^{++}(v)| \geq n/3$. As the digraph is digon-free, we have $N^-(v) \cap N^+(v) = \emptyset$. Moreover, since $|N^{++}(v)| + |N^+(v)| + |N^-(v)| + 1 > n$, it implies $N^-(v) \cap N^{++}(v) \neq \emptyset$: there is a triangle.

The Second Neighborhood Conjecture was proved in the case of tournaments by Fisher [66] using probabilistic methods. Later, Havet and Thomassé [82] provided a constructive proof for tournaments. It was also verified for general digraphs with minimum out-degree at most 6 by Kaneko and Locke [94]. A weaker version was proved by Chen *et al.* [41]; they showed that every simple digraph has a vertex $v$ such that $|N^+(v)| \leq \eta|N^{++}(v)|$, where $\eta$ is the real polynomial root of the equation $2x^3 + x^2 - 1 = 0$ (that is $\eta \approx 0.6573$). Finally, Fidler and Yuster [65] proved the Second Neighborhood Conjecture for several classes of dense orientations. In the same paper [65], they also reduced the conjecture to an asymptotic version of it to a finite case using probabilistic methods.

**Theorem 13** (Fidler and Yuster [65]). *For every $\varepsilon > 0$, there is a finite set $F(\varepsilon)$ of digon-free digraphs such that if the Second Neighborhood Conjecture holds for all elements of $F(\varepsilon)$, then every simple digraph with $n$ vertices has a vertex $v$ with $|N^{++}(v)| \geq (1-\varepsilon)|N^+(v)| - \varepsilon \cdot n$.*

### 3.1.4 Almost Vertex-disjoint Cycles

In the paper [86] proving Caccetta-Häggkvist Conjecture for $k \leq 5$, Hoàng and Reed mentioned that the technique they used to prove the result are unlikely to work for larger $k$. Therefore, they propose in this article to focus on how the cycles in the digraph are connected to each other as a new approach of the problem. Thus, they proposed the following conjecture:

**Conjecture 6** (Hoàng and Reed [86]). *Every directed graph with minimum out-degree at least $k$ has a sequence of $k$ cycles $C_1, \ldots, C_k$ such that for each $i$ with $1 \leq i \leq k$, we have $|V(C_i) \cap \left(\bigcup_{0<j<i} V(C_j)\right)| \leq 1$.*

We can easily show that this conjecture implies the Caccetta-Häggkvist Conjecture. Indeed, we have $|\bigcup_{1\leq i\leq k} V(C_i)| \leq n$ since the sequence is a subdigraph. As each cycle $C_i$ can intersect on at most one vertices among the cycles $C_1, \ldots, C_{i-1}$, we have $|\bigcup_{1\leq i\leq k} V(C_i)| \geq k(g-1)+1$, where $g$ is the size of the smallest cycle in the sequence. It follows that $k(g-1) < n$, and thus $g \leq \lceil n/k \rceil$ proving Conjecture 4.

This conjecture has been proved for $k = 2$ by Thomassen [144], and for $k = 3$ by Welhan [151]. Finally, note that Havet *et al.* [83] proved that the conjecture holds for tournaments.

Given a collection of sets $S_1, \ldots, S_p$, the intersection graph of $\{V(S_i)\}_{i=1}^p$ is the undirected graph $G$ where we create one vertex $v_i$ for each $S_i$, and we connect

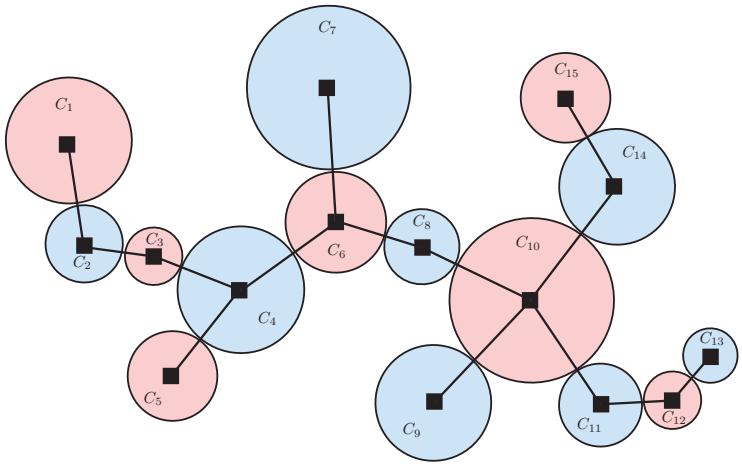



**Figure 3.2** – Example for $k = 13$, and where the intersection graph of the sequence of cycles $C_1, \ldots, C_{15}$ forms a tree. The square nodes are the vertices of the intersection graph. The color of the cycles corresponds to the partition of the sequence. Here, the blue cycles would be a collection of vertex disjoint cycles satisfying the Bermond-Thomassen Conjecture.

$v_i$ and $v_j$ by an edge if their corresponding sets have a common element. More formally, we have $E(G) = \{\{v_i, v_j\} \colon S_i \cap S_j \neq \emptyset\}$. Notice that the intersection graph of a sequence satisfying the Hoàng-Reed Conjecture may contains cycles. For example, three cycles intersecting on the same vertex would lead to a triangle in the intersection graph. Therefore, Welhan [151] considered a stronger version of the Hoàng-Reed Conjecture where the intersection graph of the sequence forms a forest.

**Conjecture 7** (Welhan [151])**.** *Every directed graph with minimum out-degree at least $k$ has a sequence of $k$ cycles $C_1, \ldots, C_k$ such that $|V(C_i) \cap \big(\bigcup_{0<j<i} V(C_j)\big)| \leq 1$ for each $i$ with $1 \leq i \leq k$ and the intersection graph of $\{V(C_i)\}_{i=1}^{k}$ is a forest.*

Note that this stronger version of the conjecture would imply the Bermond-Thomassen Conjecture. Indeed, if such a sequence of cycles exists, then since the intersection graph is a forest, it is also bipartite thus one part is of size at least $\lceil k/2 \rceil$, proving Conjecture 1. See Figure 3.2 for an example of this observation.

## 3.2 Algorithmic Results about Cycles, and Motivations

In this section, we deal with some of the algorithmic results about cycles in digraphs. Namely, we mainly focus on the packing of cycles since it plays a fundamental role in this thesis.

Formally, given a digraph $D$ and a positive integer $k$, the Vertex-Disjoint Cycle Packing (resp. Arc-Disjoint Cycle Packing) problem is to determine whether $D$ has $k$ vertex-disjoint (resp. arc-disjoint) cycles.

In addition, we also consider the optimization versions of these problems, where we want to find the maximum-sized packing of vertex/arc-disjoint cycles. We refer to them by Max CP (vertex-disjoint), and Max ACP (arc-disjoint).

Packing disjoint cycles is a fundamental problem in Graph Theory and Algorithm Design with applications in several areas, and we give in the following subsections some insight about the motivations of its study.

### 3.2.1 The Cycles as a Measure of the Complexity of a Digraph

When considering algorithmic cycle problems in directed graphs, let us first mention that Directed Disjoint Cycle Packing is NP-complete (see [14]). One can also mention the Hamilton Cycle problem which has been widely studied. But a subsequent part of these articles focused of finding necessary/sufficiency condition to guarantee the Hamilton property (see for example the surveys [14, 18, 77, 92, 102]).

Another related example of algorithmic problem about cycles in digraphs is the Longest Directed Cycle problem. The best known polynomial time algorithms for directed graphs essentially find such structures of logarithmic length. More precisely, Alon *et al.* [8] a cycle of length exactly $c \log n$ for any constant $c$, if one exists. We can also mention Gabow and Nie [68] that provide a polynomial algorithm to find a cycle of length $\log n / \log \log n$, if one exists. Then, Björklund *et al.* [27] for any $\varepsilon > 0$, there is no $n^{1-\varepsilon}$-approximation algorithm unless $\mathsf{P} = \mathsf{NP}$, even if the input digraph has constant bounded out-degree and contains a Hamilton cycle.From the parameterized point of view, the Long Directed Cycle problem—asking whether the input digraph contains a cycle of length *at least* $k$—has also been studied; one can for example mention the result of Fomin *et al.* [67], which provide a $6.75^{k+o(k)}$ poly running-time algorithm to solve Long Directed Cycle.

Numerous algorithmic problems become significantly easier when we consider acyclic digraphs. Consider for instance Hamiltonian Path problem: in general digraphs, finding a spanning path of the digraph is an NP-hard problem, while a topological order of an acyclic digraph immediately leads to a linear-time algorithm. This points out that the number of disjoint cycles can be considered as a measure of the complexity of a digraph.

Surprisingly, the packing of cycles in digraphs did not received the same attention than in the undirected case. Therefore, one of the interest of this thesis is also to meet the expectation for algorithmic results about packing cycles in (dense) digraph.

### 3.2.2 The Packing of Cycles as the Dual of the Feedback Set Problems

Many problems in computer science dealing with packing sub-structures in a (di)graph can alternatively be reconsidered from the transversal sub-structure point of view, called *dual*. In the case of vertex-disjoint packing of cycle, its transversal version would be to find the smallest subset of vertices that touches every cycles; it corresponds to find the smallest feedback vertex set, and we

call this problem We can also write these two problems using integer linear programming as follows (note that the same duality holds between Max ACP and Min FAS):

$$\begin{array}{lll} \text{maximize} \sum_{C \in \mathcal{C}} x_C & \textit{Packing cycles version} - \text{Max CP} & \text{(primal)} \\ \text{subject to} \sum_{C:\ v \in C} x_C \leq 1 & \text{for every vertex } v \in V(T) & \\ \text{with } x_C \in [\![0,1]\!] & \text{for every cycle } C \in T & \end{array}$$

$$\begin{array}{lll} \text{minimize} \sum_{v \in V(T)} x_v & \textit{Transversal cycle version} - \text{Min FVS} & \text{(dual)} \\ \text{subject to} \sum_{v:\ v \in C} x_v \geq 1 & \text{for every cycle } C \in T & \\ \text{with } x_v \in [\![0,1]\!] & \text{for every vertex } v \in V(T) & \end{array}$$

This duality leads to a crucial inequality between $\nu_0(D)$ (resp. $\nu_1(D)$) and $\text{fvs}(D)$ (resp. $\text{fas}(D)$), namely:

$$\nu_0(D) \leq \text{fvs}(D) \text{ and } \nu_1(D) \leq \text{fas}(D) \tag{3.1}$$

Min FVS and Min FAS are very fundamental problems, and so have been widely studied. They lead to many applications in numerous fields, such as circuit design [92], operating systems and more precisely deadlock prevention [113, 136], artificial intelligence [16, 52, 53], and VLSI chip design [76, 106, 109], voting theory [49], machine learning [45] or search engine ranking [60]. See the survey of Festa [64] for a more extensive list of applications.

These two problems have been both studied on different classes of digraphs. Among them, we focus here on some of the results of FVS and FAS on tournaments and bipartite tournaments as they play an important role in this thesis.

#### 3.2.2.1 Feedback Vertex Set

**On General Digraph** In the vertex-version of the problem, it is easy to prove that it is NP-complete. It is one of the first 21 problems to be proven as NP-complete by the seminal paper of Karp [96] in 1972.

**Theorem 14** (Karp [96])**.** Feedback Vertex Set *is* NP*-complete.*

From the approximability point of view, Kann [95] proved that FVS is APX-hard. The best known approximation factor is due to Even *et al.* [62] providing a $O(\log n \log\log n)$-approximation algorithm.

It has been a long and important open question to find a fixed parameterized algorithm for FVS in general digraph. Chen [42] gave for the first time an FPT algorithm with running time $O(4^k k^4 k! nm)$. Recently, Lokshtanov *et al.* [115] improved the running time to $O(4^k k^5 k!(n+m))$. Note that even if it is the first

algorithm with a linear dependency on the input size, the parameterized part of the running time is still very big; therefore, in their paper [115], the authors mention the existence of a $2^{O(k)}n^{O(1)}$ running time algorithm as an outstanding problem. The existence of a polynomial kernel for FVS is also open, and is another very important question for future work. Note that is this open even for planar digraphs.

An analogous result of the Erdős-Pósa Theorem [61][3] was conjectured by Younger [154]. It is trivial for $k \leq 1$, and McCuaig [119] proved it for $k = 2$. Seymour [135] proved a fractional version (not defined here) of the conjecture. Finally, Reed *et al.* [131] showed the following theorem, proving the conjecture.

**Theorem 15** (Reed *et al.* [131])**.** *There exists a function $g$ such that every digraph $D$ and every $k \geq 0$, $D$ contains $k$ vertex-disjoint directed cycles or has a minimum feedback vertex set at most $g(k)$.*

However, the function $g$ provided by the authors is highly exponential. Compared to the undirected version or to the fractional version of the problem [135] (where the function $g_{frac}$ is in $O(k \log k \log \log k)$), it is unlikely that this function $g$ would be close to the best possible. Nevertheless, notice that it allows us to bound the size of the feedback vertex set using the inequality (3.1). Thus, we have $\nu_0(D) \leq \text{fvs}(D) \leq g(\nu_0(D))$.

**On Tournaments** When restricted to tournaments, the problem is called Feedback Vertex Set in Tournament (FVST for short). However, it is still NP-complete in this class of digraphs.

**Theorem 16** (Speckenmeyer [141], Band-Jensen and Thomassen [15])**.** Feedback Vertex Set in Tournament *is* NP*-complete.*

By noticing that any tournament contains a cycle if and only if it has a triangle, the 3-approximation of Feedback Vertex Set in Tournament is obvious. Cai *et al.* [36] provide a 5/2-approximation of the problem, followed by Mnich *et al.* [121] who gave a 7/3-approximation algorithm. Recently, a randomized 2-approximation has been described by Lokshtanov *et al.* [114]. Notice that due to the reduction from Vertex Cover preserving the ratio, this approximation factor would the best possible in polynomial time if the *Unique Game Conjecture* is true[4].

The problem has been obviously studied with the parameterized framework. First, one can mention the $O(2^k \cdot n^2(\log \log n + k))$ running time FPT algorithm by Dom *et al.* [57]. Kumar and Lokshtanov [105] gave an algorithm for FVST with running time $O(1.618^k + n^{O(1)})$. Using kernels of $d$-Hitting Set [1], Dom *et al.* [57] gave a kernel for FVST with $O(k^2)$ vertices and $O(k^3)$ arcs. Recently, Le *et al.* [108] provided a kernel with $O(k^{3/2})$ vertices using expansion lemmas.

[3]This seminal result states that every (undirected) graph $G$ and every $k \geq 0$, $G$ contains $k$ vertex-disjoint (undirected) cycles or has a minimum feedback vertex set at most $O(k \log k)$.

[4]The conjecture postulates that the problem of determining the approximate value of a unique game (which is a type of game, not detailed here) is NP-hard. See for example [99] for more details about this conjecture.

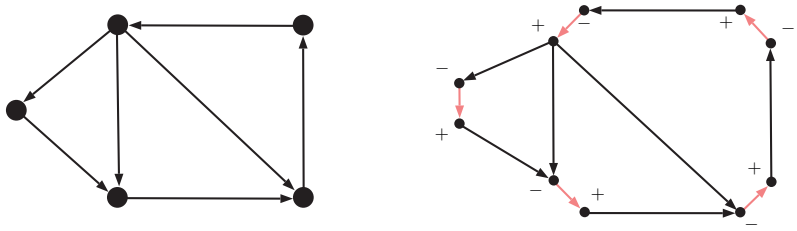



**Figure 3.3** – The right digraph is obtained from the left one using the reduction from FVS to FAS.

**On Bipartite Tournaments** When restricted to bipartite tournaments, the problem is called Feedback Vertex Set in Bipartite Tournament (or simply FVSBT for short). However, Cai *et al.* [37] proved it is still NP-complete in this class of digraphs.

**Theorem 17** (Cai *et al.* [37])**.** Feedback Vertex Set in Bipartite Tournament *is* NP*-complete.*

In the same article, Cai *et al.* [37] provide a 7/2-approximation algorithm. Then, Sasatte [132] improved this result into a 3-approximation algorithm. So far, the best ratio is 2, and given by van Zuylen [147]. Furthermore, the author proved that this ratio is the best one using linear programs.

From the parameterized point of view, Sasatte [133] gave an algorithm with running time $O(3^k \cdot n^2 + n^3)$. Later, Hsiao [87] provided an algorithm with running time $O(2^k \cdot n^6)$. Finally, using similar techniques than for tournaments [105], Kumar and Lokshtanov [104] gave an algorithm for FVSBT with running time $O(1.6181^k + n^{O(1)})$. For the kernel side, thanks to kernels of $d$-Hitting Set [1], Dom *et al.* [57] showed a kernel with $O(k^3)$ vertices and $O(k^4)$ arcs.

#### 3.2.2.2 Feedback Arc Set

**On general Digraph** Note that, on general graphs, FAS and FVS are closely tied. First, it is easy to see that for every digraph $D$, we have $\mathrm{fvs}(D) \leq \mathrm{fas}(D)$. Indeed, given a feedback arc set $F$ of $D$, the subset defined by $\{v\colon \exists a \in F, v \in V(a)\}$ is trivially a feedback vertex set of $D$ since deleting $v$ also delete the arcs $v$ belongs to. Due to this close link to FVS, it is natural that FAS is also NP-complete.

**Theorem 18** (Karp [96])**.** Feedback Arc Set *is* NP*-complete.*

*Proof.* In his article [96], Karp proved the NP-completeness of FAS by reducing from Vertex Cover. We give here an alternate proof of reduction from FVS.

Let $D$ be a digraph given by its vertex set $V(D)$ and its arc set $A(D)$. We can construct in linear time the digraph $D'$ where $V(D') = \{v^-, v^+\colon v \in V(D)\}$, and $A(D') = \{(u^+, v^-)\colon (u, v) \in A(D)\} \cup \{(v^-, v^+)\colon v \in V(D)\}$. In other words, we split every vertex $v$ of $D$ into two vertices $v^-$ and $v^+$ linked with the arc $(v^-, v^+)$, and we put all the in-going arcs and all the out-going arcs of $v$ to $v^-$ and $v^+$, respectively. See Figure 3.4 for an example of construction. Then, given

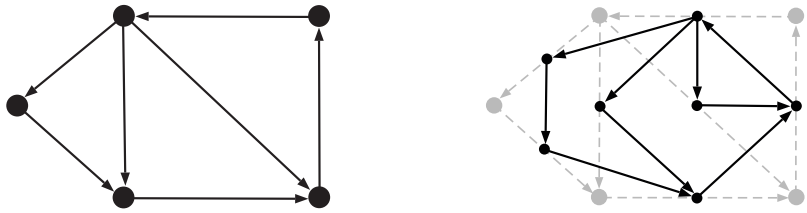


**Figure 3.4** – The right digraph is obtained from the left one using the reduction from FAS to FVS. The gray arcs and vertices in the digraph on the right are just here to depict how we connect the black vertices are connected in the new digraph.

a minimum subset of vertices $X$ such that $D \setminus X$ is acyclic, it is easy to notice that the subset of arc $\{(x^-, x^+)\colon x \in X\}$ cut all the cycles in $D'$, and so is a feedback arc set of same size than $X$.

In the other direction, given a feedback arc set $X'$ of $D'$, assume there is an arc $(u^+, v^-)$ in $X'$, with $u \neq v$. Then we can construct a new solution $X''$ with $X'' = (X' - \{(u^+, v^-)\}) \cup (u^-, u^+)$. Since we have $d^-(u^+) = 1$ by construction, all the cycles through $u^+$ are specifically through the arc $(u^-, u^+)$. So $X''$ is also a feedback arc set of $D'$. By iterating this argument, we can assume that all the arcs of $X'$ are of the form $(x^-, x^+)$. We can now easily see that the set of vertices $\{x\colon (x^-, x^+) \in X'\}$ cut all the cycles of $D$ and has the same size than $X'$. $\square$

We can make two small observations about this previous linear reduction from FVS to FAS. First, this reduction does not work on specific some classes of digraphs; for example, the NP-hardness of FVS in some class does not guarantee the NP-hardness of FAS in this class. Second, it only allows us to extend negative result from FVS to FAS, such that NP-hardness or inapproximability; the reduction in the other direction —that is from FAS to FVS —is mandatory to extend the positive results. Fortunately, such a reduction exists.

Given a digraph $D$, we construct in quadratic time its line digraph $L(D)$ (see Figure 3.4 for an example of construction). By definition of the line digraph, there is a cycles in $D$ if and only if there is a cycle in $L(D)$. Therefore, the arcs of a feedback arc set $X$ of $D$ corresponds exactly to a feedback vertex set in $L(D)$, and the reciprocal statement holds too. This reduction leads to the same approximation factor and FPT running time[5] than for FVS.

**On Tournaments** When considered on tournaments, the problem is called Feedback Arc Set in Tournament (FAST for short). As mentioned before, the specific case of tournaments and its complexity is not implied by reductions from FVS on the general digraphs. As an example, unlike the general case, it has been a quite long standing open question to know if the problem stayed hard in tournaments. Conjectured NP-complete in 1992 by Bang-Jensen and Thomassen [15], Ailon *et al.* [3] were the first ones to its NP-hardness under

[5]Since this is a quadratic reduction, the polynomial factor in $n$ is modified but not the function depending to the parameter.

| | Problems | Approximation factor | FPT running time | Kernel size |
|---|---|---|---|---|
| FVS | General digraphs | $O(\log n \log\log n)$ [62] | $O^*(4^k k^5 k!)$ [115] | **open** |
| | Tournaments | 2 (randomized) [114]<br>7/3 (deterministic) [121] | $O^*(1.618^k)$ [105] | $O(k^{3/2})$ [108] |
| | Bip. tournaments | 2 [147] | $O^*(1.6181^k)$ [104] | $O(k^4)$ [1, 57] |
| FAS | General digraphs | $O(\log n \log\log n)$ [62, 135] | $O^*(4^k k^5 k!)$ [115] | **open** |
| | Tournaments | $1+\varepsilon$ [98] | $O^*(2^{O(\sqrt{k})})$ [63, 97] | $(2+\varepsilon)k$ [23] |
| | Bip. tournaments | 4 [147] | $O^*(3.373^k)$ [57] | $O(k^2)$ [152] |

**Table 3.1** – Overview of the different (approximation and parameterized) results of the feedback sets problems, with some of its variations on specific classes. In this Table, $n$ is the number of vertices in the input graph, $k$ is the natural parameter, and $\varepsilon$ an arbitrary small positive real.

randomized reductions. Since then, it was proved with deterministic reductions multiple times by Alon [6], by Conitzer [46], by Charbit *et al.* [39], and by Ailon *et al.* [3].

**Theorem 19** (Alon [6], Conitzer [46], Charbit *et al.* [39], Ailon *et al.* [3])**.** Feedback Arc Set in Tournament *is* NP*-complete.*

From an approximability point of view, van Zuylen and Williamson [148] gave a 2-approximation of FAST. Fortunately, unlike on the general case, there is a PTAS algorithm for FAST. This result was proved by Kenyon-Mathieu and Schudy [98].

Raman and Saurabh [130] provided a parameterized algorithm which runs in time $O(2.415^k \cdot k^{4.752} + n^{O(1)})$. Then, Alon *et al.* [7] showed an algorithm running in $2^{O(\sqrt{k}\log^2 k)} + n^{O(1)}$ time, being the first to answer a question of Guo *et al.* [75] whether there exists an algorithm for FAST in running time $O(2^k \cdot n^{O(1)})$. Note that except bidimensional problems [55, 56], no other parameterized problems admitted a subexponential running time before this one. Since then, Feige [63] as well as Karpinski and Schudy [97] independently provide a $2^{O(\sqrt{k})} + n^{O(1)}$ running time algorithm.

**On Bipartite Tournaments** When restricted to bipartite tournaments, the problem is called Feedback Arc Set in Bipartite Tournament (FASBT for short). Unfortunately, Guo *et al.* [74] proved it is still NP-complete.

**Theorem 20** (Guo *et al.* [74])**.** Feedback Arc Set in Bipartite Tournament *is* NP*-complete.*

Using a direct application of [148], van Zuylen [147] gave a 4-approximation algorithm for FASBT. In 2010, Dom *et al.* [57] give a fixed parameterized algorithm with running time $O(3.373^k \cdot n^6)$. From the kernelization side, Misra *et al.* [120] gave a $O(k^3)$ kernel, improved recently by Xiao and Guo [152] providing a quadratic kernel.

# Chapter 4

# Vertex-disjoint Packing of Triangles in Tournaments

This chapter corresponds to joint work with Stéphane Bessy and Marin Bougeret presented to *ESA 2017* [21]. We focus here on the Vertex-Disjoint Cycle Packing restricted to tournaments. More precisely, we give both classical and parameterized complexity results, as well as well as approximation results.

## 4.1 Introduction and Preliminaries

### 4.1.1 General problems and Related Work

Given a tournament $T$, a natural problem is to ask whether $T$ contains $k$ vertex-disjoint cycles. The very first observation one can make is to notice that $T$ contains a cycle if and only if it contains a triangle. Indeed, let $c$ be an arbitrary cycle of $T$. If $c$ is a triangle we are done, so we may consider that $c$ has at least four vertices, and let denote by $v_1, \ldots, v_t$ its vertices. Now consider two non-successive vertices, say $v_1$ and $v_3$ without loss of generality. If $(v_3, v_1) \in A(T)$, then $(v_1, v_2, v_3)$ is a triangle proving the claim. So we have $(v_1, v_3) \in A(T)$. In that case, $(v_1, v_3, \ldots, v_t)$ is a cycle of $T$ strictly smaller than $c$; by repeating the process, $T$ necessarily contains a triangle.

This simple observation shows that if we want to consider a packing of vertex-disjoint cycles in a tournament, it is enough to look for vertex-disjoint triangles in it. Therefore, we can define the Maximum Disjoint Triangle Packing in Tournament problem (or simply Max DTT) as follows:

Maximum Disjoint Triangle Packing in Tournament (Max DTT)
**Input:** A tournament $T$.
**Result:** A collection $\triangle$ of vertex-disjoint triangles in $T$.
**Optimization:** Maximize $|\triangle|$.

We also consider the decision problem associated to this problem, namely Disjoint Triangle Packing in Tournament (or simply DTT) where we want to know whether the tournament contains at least $k$ vertex-disjoint triangles.

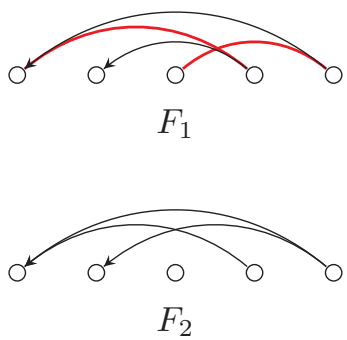



**Figure 4.1** – The forbidden tournaments $F_1$ and $F_2$. The red edges of $F_1$ can be oriented in any direction. All the non-depicted arcs are forward arcs.

Recall that a tournament is sparse if it admits a feedback arc set which is a matching. We define Disjoint Triangle Packing in Sparse Tournament (resp. Maximum Disjoint Triangle Packing in Sparse Tournament) the problem DTT (resp. Max DTT) when restricted to sparse tournaments. For sake of brevity, we may use DTS and Max DTS to refer to Disjoint Triangle Packing in Sparse Tournament and Maximum Disjoint Triangle Packing in Sparse Tournament, respectively.

Given a packing of vertex-disjoint triangle $\triangle$, if we have in addition $V(\triangle) = V(T)$, then $\triangle$ is said to be a *perfect vertex-disjoint packing of triangles* of $T$. Therefore, we consider the following strengthen version of DTT:

Perfect Packing Disjoint-Triangle in Tournament (Perfect-DTT)
**Input:** A tournament $T$.
**Output:** A perfect vertex-disjoint packing of triangles in $T$.

Cai *et al.* [37] proved that DTT is polynomial when the tournament does not contain a family of forbidden sub-tournaments depicted in Figure 4.1.

Alternatively, the DTT problem can be seen as a special case of 3-Set Packing. For the recall, given a universe $\mathcal{U}$ of elements, a family $\mathcal{S}$ of subsets of $\mathcal{U}$ each containing three elements of $\mathcal{U}$, and an integer $k$, the 3-Set Packing problem asks whether there are $k$ subsets of $\mathcal{S}$ that are pairwise disjoint. Formally, we have:

3-Set Packing
**Input:** An universe $\mathcal{U}$, a family $\mathcal{S} = \{\mathcal{S}_1, \ldots, \mathcal{S}_m\}$ of subsets of $\mathcal{U}$ such that for every $j \in [\![1, m]\!]$, $|\mathcal{S}_j| = 3$, and an integer $k$.
**Output:** A subfamily $\mathcal{F} \subseteq \mathcal{S}$ of at least $k$ sets such that for every $f$ and $f'$ distinct sets of $\mathcal{F}$, we have $f \cap f' = \emptyset$.

Therefore, given a tournament $T$, we can construct in polynomial time an instance of 3-Set Packing where the universe are the vertices of $T$ and the family $\mathcal{S}$ contains all the triangles of $T$. One can easily see that the solution of the 3-Set Packing leads to a vertex-disjoint triangle packing of $T$.

Using this trivial reduction, we can extend all the best algorithms for 3-Set

Packing to DTT, such as a $(4/3+\epsilon)$-approximation-approximation algorithm by Cygan [47] and a $O^*(3.52^{3k})$ parameterized algorithm (with the natural parameter) by Wang and Feng [149].

Furthermore, notice that a kernel with $O(k^{1.5})$ vertices was recently shown by Le *et al.* [108] for DTT using interesting variants and generalizations of the expansion lemma applied on several implicit 3-Set Packing problems.

### 4.1.2 Preliminaries on Weak $d$-Compositions

Among the wide literature on polynomial kernelization, there exist several kind of compositions that allows to show lower-bounds for the size of a kernel. Started with the work of Bodlaender [28], these tools have been extended by several authors, such as Dom *et al.* [58], Chen *et al.* [43], or Bodlaender *et al.* [29].

We only recall here the notion of weak $d$-composition introduced by used Hermelin and Wu [84].

**Definition 2** (*weak $d$-composition*, Hermelin and Wu [84])**.** *Let $d \geq 2$ be a constant, and let $L_1, L_2 \subseteq \{0,1\}^* \times \mathbb{N}$ be two parameterized problems. A weak $d$-composition from $L_1$ to $L_2$ is an algorithm $A$ that on input $(x_1,k),\ldots,(x_t,k) \in \{0,1\}^* \times \mathbb{N}$, outputs an instance $(y,k') \in \{0,1\}^* \times \mathbb{N}$ such that:*

*(i) the algorithm $A$ runs in co-nondeterministic polynomial time with respect to $\sum_{i\in[\![1,t]\!]}(|x_i|+k)$,*

*(ii) $(y,k')$ is a "yes"-instance of $L_2$ if and only if there is an $i \in [\![1,t]\!]$ such that $(x_i,k)$ is a "yes"-instance of $L_i$,*

*(iii) $k' \leq t^{1/d}k^{O(1)}$.*

The main difference with other compositions as the cross-composition [29] is that the output parameter $k'$ may also depend of the number of instances $t$. This allows to give polynomial lower-bounds for the kernel size while cross-compositions show that there is no polynomial kernels.

**Theorem 21** (Hermelin and Wu [84])**.** *Let $d \geq 2$ be a constant, and let $L_1, L_2 \subseteq \{0,1\}^* \times \mathbb{N}$ be two parameterized problems such that the unparameterized version of $L_1$ is $\mathsf{NP}$-hard and there is a weak $d$-composition from $L_1$ to $L_2$. Then $L_2$ has no kernel of size $O(k^{d-\varepsilon})$ for all $\varepsilon > 0$ unless $\mathsf{NP} \subseteq \mathsf{coNP/poly}$.*

Weak $d$-compositions allowed several almost tight lower-bounds. One can mention for example the lower-bound of $O(k^{d-\varepsilon})$ for $d$-Set-Packing and $O(k^{d-4-\varepsilon})$ for $K_d$-packing proved by Hermelin and Wu [84]. These lower-bounds were improved by Dell and Marx [54] to $O(k^{d-\varepsilon})$ for Perfect $d$-Set-Packing, $O(k^{d-1-\varepsilon})$ for $K_d$-packing, and leading to $O(k^{2-\varepsilon})$ for Perfect $K_3$-Packing. Notice that negative results for the "perfect" versions of the problems (where the parameter $k = \frac{n}{d}$, with $d$ the number of vertices in the packed structure) are stronger than for the classical version, where $k$ is arbitrary. Kernel lower bounds for these "perfect" versions is sometimes referred as *sparsification lower bounds*.

### 4.1.3 Specific Notations

In the following, we mostly consider a tournament $T$ of order $n$ with its linear representation $(\sigma(T), \overleftarrow{A}(T))$, where $\sigma(T) = (v_1, \ldots, v_n)$. For a backward arc $a$ with endpoints $(v_j, v_i)$ with $i < j$, the *span* of $a$, denoted by $\mathrm{s}(a)$, is the (possibly empty) set of vertices $\{v_k \colon i < k < j\}$. Moreover, the *span value* of $a$ is the size of $\mathrm{s}(a)$, that is $j - i - 1$. In the following, the *minimum span* (resp. *maximum span*) of a tournament $T$ according its ordering $\sigma$ is the minimum (resp. maximum) of the span values of its backward arcs and is denoted $\mathrm{minspan}(\sigma(T))$ (resp. $\mathrm{maxspan}(\sigma(T))$).

Let $T_1$ and $T_2$ be two tournaments of order $n$ and $n'$, and defined by their respective linear representation $((u_1, \ldots, u_n), \overleftarrow{A}(T_1))$ and $((v_1, \ldots, v_{n'}), \overleftarrow{A}(T_2))$. We define the *concatenation* of $T_1$ and $T_2$, written $T_1 \oplus T_2$, as the new tournament $T$ given by the linear representation $(\sigma(T), \overleftarrow{A}(T))$ with $\sigma(T) = (u_1, \ldots, u_n, v_1, \ldots, v_{n'})$ and $\overleftarrow{A}(T) = \overleftarrow{A}(T_1) \cup \overleftarrow{A}(T_2)$. Informally, we add the vertices of $T_2$ after the vertices of $T_1$, and we put all the forward arcs from $T_1$ to $T_2$. Note that if, for any $i \in [\![1, 2]\!]$, $\triangle_i$ is a vertex-disjoint packing of triangles of $T_i$, then $\triangle_1 \cup \triangle_2$ is a vertex-disjoint packing of triangles of $T_1 \oplus T_2$.

### 4.1.4 Our Contributions and Organization of the Chapter

In this chapter, our objective is to study the complexity of DTT, the approximability of Max DTT and the kernelization of DTT. DTT is already known to be NP-complete [117]. However, it is surprising that that many of these results hold even when the tournament is has a very simple structure as sparse tournaments. See Chapter 6 for more details and results on tournaments.

On the approximation side, a natural question is a possible improvement of the $\frac{4}{3} + \varepsilon$-approximation of Cygan [47] implied by 3-Set Packing. We show in section 4.2 that, unlike FAST, Max DTT does not admit a PTAS unless $\mathsf{P} = \mathsf{NP}$, even if the tournament is sparse.

Concerning kernelization, we complete the panorama of sparsification lower-bounds of Jansen and Pieterse [90], by proving in section 4.3 that Perfect-DTT do not admit a kernel of (total bit) size $O(n^{2-\varepsilon})$, unless $\mathsf{NP} \subseteq \mathsf{coNP/poly}$ Theorem 27). This implies that DTT does not admit a kernel of (total bit) size $O(k^{2-\varepsilon})$ unless $\mathsf{NP} \subseteq \mathsf{coNP/poly}$. We also prove that DTT admits a kernel of $O(m)$ vertices, where $m$ is the size of a given feedback arc set of the instance

## 4.2 The NP-completeness and APX-hardness of Packing Vertex-disjoint Triangles in Tournaments

We prove in this section the APX-hardness of Max DTS by providing an L-reduction from the Max 2-SAT(3) problem, which is known to be APX-hard [11, 17]. Recall that an instance $\mathcal{F}$ of Max 2-SAT(3) is defined by a set of $n$ variables $\{x_1, \ldots, x_n\}$ and a set of $m$ clauses $\{c_1, \ldots, c_m\}$, where each variable appears in *at most* three clauses, and each clause has exactly two variables. Then, the goal is to find an affectation $a$ of the variable in $\{\text{true}, \text{false}\}$ which satisfies the

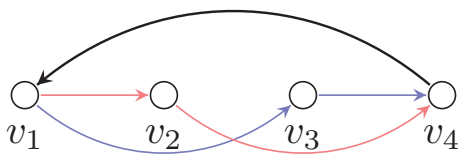



**Figure 4.2** – A piece digraph $\Pi$ with the two possible triangles $\delta^2_\Pi$ (in red) and $\delta^3_\Pi$ (in blue). All the non-depicted arcs are forward arcs.

maximum number of clauses.

In the following, we also assume that each variable occurs exactly twice positively and once negatively. Indeed, if a variable appears only positively (resp. negatively), we can set it to true (resp. false) which satisfies all the clause it belongs. Otherwise, if the variable $x$ appears twice negatively and once positively, let replace it by $\overline{x}$.

### 4.2.1 Definition of the Reduction

We start by defining the reduction $f$ which, given an instance $\mathcal{F}$ of Max 2-SAT(3), returns an instance $T_{f(\mathcal{F})}$ of Max DTS. To do so, we first define the *piece digraph* we will extensively use to our reduction as the four vertices tournament with ordering $(v_1, v_2, v_3, v_4)$ and with the only backward arcs $(v_4, v_1)$. It is easy to see that there are only two triangles in this tournament: the triangle with $v_2$ and the one with $v_3$: if $\Pi$ is the name of the piece digraph, we denote these triangles by $\delta^2_\Pi$ and $\delta^3_\Pi$, respectively. The Figure 4.2 is an example of this digraph.

**The clause tournament** For each clause $c_j$ of $\mathcal{F}$, we create the *clause gadget* $K_j$ with the ordering $\sigma(K_j) = (\theta_j, d^1_j, c^1_j, c^2_j, d^2_j)$ and $\overleftarrow{A}(K_j) = \{(d^2_j, d^1_j)\}$. In other words, this clause gadget is the vertex $\theta_j$ concatenated with a piece digraph.

As we did before, we define our *clause tournament* $K$ as the concatenation of the clause gadgets, that is $K = K_1 \oplus \cdots \oplus K_m$. One can easily notice that such a tournament is sparse and has $5m$ vertices.

**The variable tournament** Similarly, for each variable $x_i$ of $\mathcal{F}$, we create the *variable gadget* $X_i$, a tournament with 35 vertices as the concatenation of eight pieces and three regular vertices. Formally, let $L_i, L'_i, \overline{L_i}, \overline{L'_i}, A_i, B_i, A'_i$ and $B'_i$ be piece digraphs. We define their vertices and their backward arc in a straightforward way, that is for example $V(\overline{L'_i}) = \{\overline{\ell'^1_i}, \overline{\ell'^2_i}, \overline{\ell'^3_i}, \overline{\ell'^4_i}\}$ with only backward arc $(\overline{\ell'^4_i}, \overline{\ell'^1_i})$. We denote by $\beta_i, \beta'_i$ and $\alpha_i$ the three other regular vertices.

Thus, the tournament $X_i$ is given by the concatenation: $X_i = L_i \oplus L'_i \oplus \overline{L_i} \oplus \overline{L'_i} \oplus \{\beta_i\} \oplus \{\beta'_i\} \oplus A_i \oplus B_i \oplus \{\alpha_i\} \oplus A'_i \oplus B'_i$. Lastly, we add to $\overleftarrow{A}(X_i)$ the arc set $\{e^1_i, e^2_i, e^3_i, e^4_i, (a'^2_i, a^2_i), (b'^2_i, b^2_i)\}$, where $e^1_i = (a^3_i, \ell^3_i)$, $e^2_i = (a'^3_i, \ell'^3_i)$, $e^3_i = (b^3_i, \overline{\ell^3_i})$, and $e^4_i = (b'^3_i, \overline{\ell'^3_i})$. The Figure 4.3 is a representation of a variable gadget.

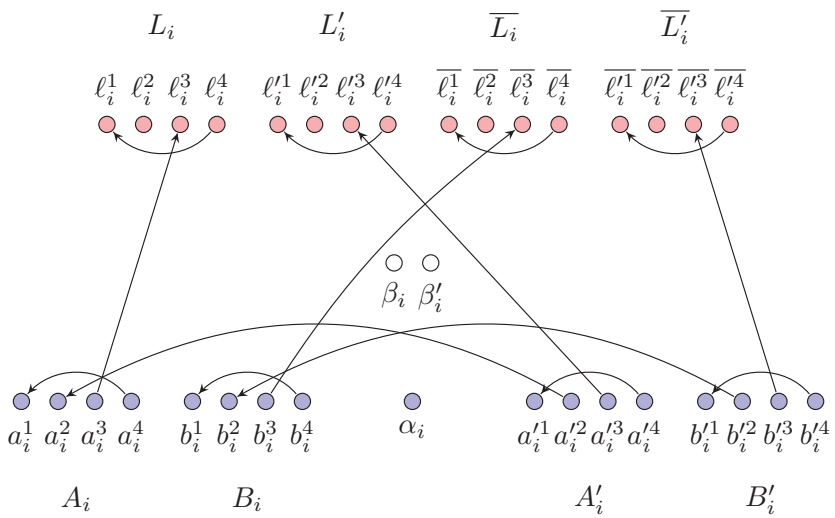



**Figure 4.3** – The variable gadget $X_i$ where only the backward arcs are depicted. In this figure, the ordering of the vertices is read first from top to bottom then from left to right. The red vertices are the vertices of $Lft_i$ and the blue vertices correspond to $Rgt_i$.

We can now create the *variable tournament* $X$ defined by the concatenation of all the variable gadgets $X_i$, that is $X = X_1 \oplus \cdots \oplus X_n$. One more time, one can notice that this tournament has $35n$ vertices and is sparse.

**Our final tournament** We now finish describing the tournament $T_{f(\mathcal{F})}$ of our reduction $f$. Let start by considering the tournament obtained by the concatenation $X \oplus K$. We now add the backward arcs to this tournament, in order to encode the instance $\mathcal{F}$. Let consider that the clause $c_j$ is the disjunction $l_{i_1} \vee l_{i_2}$, where $l_{i_1}$ and $l_{i_2}$ are positive or negative occurrences of the variables $x_{i_1}$ and $x_{i_2}$, respectively. For each $p \in [\![1, 2]\!]$, if $l_{i_p}$ is negative, we add the backward arc $(c_j^p, \overline{\ell_{i_p}^2})$. On contrary, when $l_{i_p}$ is positive, if it is the first occurrence of the variable $x_{i_p}$, we add the backward arc $(c_j^p, \ell_{i_p}^2)$. Otherwise, if it is the second (and last) occurrence, we add the backward arc $(c_j^p, \ell_{i_p}'^2)$. The tournament obtained is $T_{f(\mathcal{F})}$. As we chose an instance of Max 2-SAT(3), we can notice that $\overleftarrow{A}(T_{f(\mathcal{F})})$ is a matching, proving that $T_{f(\mathcal{F})}$ is a sparse tournament.

Notice that vertices of $\overline{L'_i}$ are never linked to the clauses gadget with backward arcs. However, we need this set to keep the variable gadget symmetric so that setting the variable $x_i$ to true or false leads to the same number of triangles inside the variable gadget $X_i$.

**Observation 22.** *The tournament $T_{f(\mathcal{F})}$ is sparse allowing us to prove negative results on sparse tournaments. However, we can easily derive a simpler non-sparse tournament by replacing each piece digraph of $T_{f(\mathcal{F})}$ by a vertex. An example of the new variable gadget with 11 vertices is depicted in Figure 4.5. An analogous—but simpler—proof follows.*

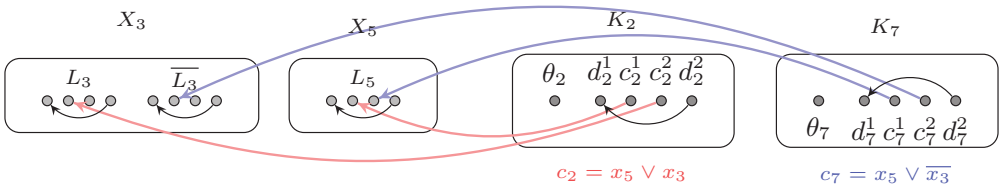



**Figure 4.4** – Example showing how a clause gadget is attached to variable gadgets with the two clauses $c_2 = x_5 \vee x_3$ (in red) and $c_7 = x_5 \vee \overline{x_3}$ (in blue). For sake of brevity, we only depict the piece digraphs of the variable gadgets connected to the clause gadgets and only the backward arcs.

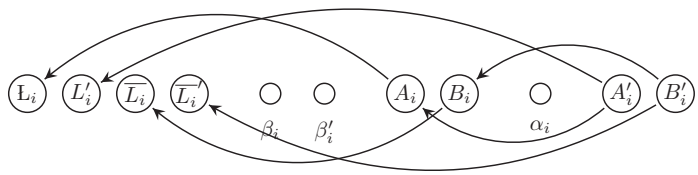



**Figure 4.5** – Variable gadget for the non sparse case.

Recall that if $\Pi$ is a piece digraph with ordering $(v_1, v_2, v_3, v_4)$, $\delta^2_\Pi$ and $\delta^3_\Pi$ denote the triangles $(v_1, v_2, v_4)$ and $(v_1, v_3, v_4)$, respectively. Then, for any $i \in [\![1, n]\!]$, we define the two different sets of vertex disjoint variable inner triangle packing of $X_i$ by:

- $P_i = \{\delta^3_{L_i}, \delta^3_{L'_i}, \delta^2_{\overline{L_i}}, \delta^2_{\overline{L'_i}}, (\overline{\ell_i^3}, \beta_i, b_i^3), (\overline{\ell_i'^3}, \beta'_i, b_i'^3), \delta^3_{A_i}, \delta^2_{B_i}, \delta^3_{A'_i}, \delta^2_{B'_i}, (a_i^2, \alpha_i, a_i'^2)\}$,
- $\overline{P_i} = \{\delta^2_{L_i}, \delta^2_{L'_i}, \delta^3_{\overline{L_i}}, \delta^3_{\overline{L'_i}}, (\ell_i^3, \beta_i, a_i^3), (\ell_i'^3, \beta'_i, a_i'^3), \delta^2_{A_i}, \delta^3_{B_i}, \delta^2_{A'_i}, \delta^3_{B'_i}, (b_i^2, \alpha_i, b_i'^2)\}$.

Notice that $P_i$ and $\overline{P_i}$ both contain 11 triangles that use all the vertices of $X_i$ except two vertices, namely $\{\ell_i^2, \ell_i'^2\}$ and $\{\overline{\ell_i^2}, \overline{\ell_i'^2}\}$, respectively. The two

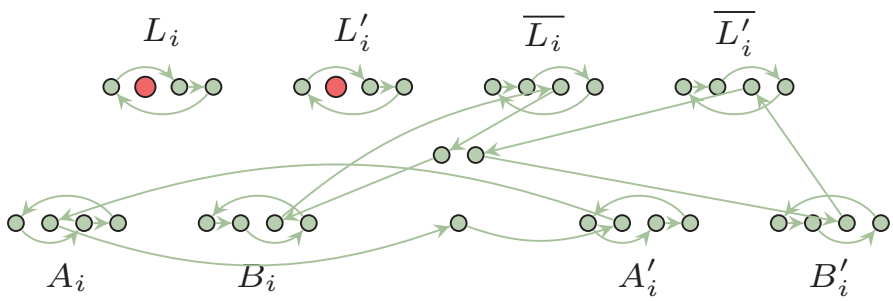



**Figure 4.6** – In green, the vertex-disjoint triangle packing $P_i$. All the vertices of the variable gadget $X_i$ are covered except $\ell_i^2$ and $\ell_i'^2$ (in red).

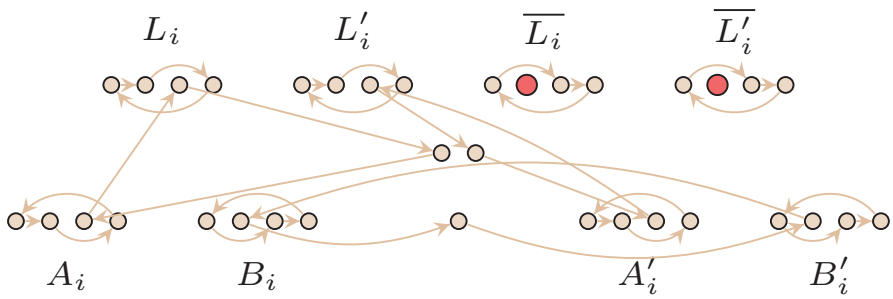



**Figure 4.7** – In brown, the vertex-disjoint triangle packing $\overline{P_i}$. All the vertices of the variable gadget $X_i$ are covered except $\overline{\ell_i^2}$ and $\overline{\ell_i'^2}$ (in red).

Figure 4.6 and Figure 4.7 represent the two vertex-disjoint triangle packings $P_i$ and $\overline{P_i}$, respectively.

Similarly, for any $j \in [\![1, m]\!]$, we define the set of clause-inner triangle packing of $K_j$, by:

- $Q_j^1 = \{(d_j^1, c_j^1, d_j^2)\}$,
- $Q_j^2 = \{(d_j^1, c_j^2, d_j^2)\}$.

Informally, setting variable $x_i$ to true corresponds to create the 11 triangles of $P_i$ in $X_i$. It leaves vertices $\{\ell_i^2, \ell_i'^2\}$ available, which are precisely the vertices that may be connected to the clauses. This allows to create other triangles with these vertices and vertices of the clauses the $x_i$ belongs: we satisfied these clauses. Conversely, setting it to false corresponds to create the 11 triangles of $\overline{P_i}$. From the clause gadget point of view, satisfying a clause $c_j$ using its $p^{th}$ literal (represented by a vertex $v \in V(X)$) corresponds to create triangle in $Q_j^{3-p}$ as it leaves $c_j^p$ available to create the triangle $(v, \theta_j, c_j^p)$. Our objective (in Lemma 22.3) is to prove that satisfying $k$ clauses is equivalent to find $11n+m+k$ vertex-disjoint triangles in $T_{f(\mathcal{F})}$.

### 4.2.2 Restructuring Lemmas

Given a packing of vertex-disjoint triangles $\triangle$, we define for any $i \in [\![1, n]\!]$ and $j \in [\![1, m]\!]$ the three following "shapes" of triangles:

- $\triangle_{X_i} = \{(v_1, v_2, v_3) \in \triangle\colon\ v_1, v_2, v_3 \in V(X_i)\}$,
- $\triangle_{K_j} = \{(v_1, v_2, v_3) \in \triangle\colon\ v_1, v_2, v_3 \in V(K_j)\}$,
- $\triangle_O = \{(v_1, v_2, v_3) \in \triangle\colon\ \exists j \in [\![1, m]\!], \exists p \in [\![1, 2]\!]\ v_2 = \theta_j \text{ and } v_3 = c_j^p\}$.

Furthermore, we say that a triangle is *variable-inner* if it is in $\triangle_{X_i}$, *clause-inner* if it is in $\triangle_{K_j}$, and is *outer* if it is in $\triangle_O$. In the following, we also define $\triangle_X$ and $\triangle_K$ as the set of all the variable-inner and clause-inner triangles, respectively.

Notice first that an outer triangle $(v_1, v_2, v_3)$ has $v_1$ in $V(X)$ by construction. Secondly, one can easily notice that a triangle $(v_1, v_2, v_3)$ of $T_{f(\mathcal{F})}$ can have none of these previous "shapes"; in that case, we necessarily have $v_3 = c_j^p$ for some $j \in [\![1, m]\!]$ and $p \in [\![1, 2]\!]$, and $v_2 \neq \theta_j$—but $v_2$ could be either in $V(X)$ or $V(K)$. This observation shows that $\triangle_X$, $\triangle_K$ and $\triangle_O$ does not necessarily form a partition of the triangles of $\triangle$. However, we will show in the next lemmas how to restructure $\triangle$ in order to get such a property.

**Lemma 22.1.** *For any packing of vertex-disjoint triangles $\triangle$, we can compute in polynomial time a solution $\triangle'$ such that $|\triangle'| \geq |\triangle|$ and for all $j \in [\![1, m]\!]$, there exists $p \in [\![1, 2]\!]$ such that $\triangle'_{K_j} = Q_j^p$ and:*

*(i) either $\triangle'$ does not use any other vertex of the clause gadget $K_j$, that is $V(\triangle') \cap V(K_j) = V(Q_j^p)$,*

*(ii) or $\triangle'$ contains an outer triangle $(v, \theta_j, c_j^{3-p})$ with $v \in V(X)$, implying $V(\triangle') \cap V(K_j) = V(K_j)$.*

*Proof.* Consider a packing of vertex-disjoint triangles $\triangle$ of $T$ that does not satisfy the statement of the lemma. We say that an index $j \in [\![1, m]\!]$ is *nice* if there exists $p \in [\![1, 2]\!]$ such that $\triangle'_{K_j} = Q_j^p$ and either $V(\triangle') \cap V(K_j) = V(Q_j^p)$, or $\triangle'$ contains an outer triangle $(v, \theta_j, c_j^{3-p})$ with $v \in V(X)$. In other words, the index $j$ is nice if it satisfies the property of the lemma.

Let us restructure $\triangle$ in order to increase the number of nice indices. To do so, consider the largest $j \in [\![1, m]\!]$ which is *not* nice. Notice that since there is only one backward arc in $K_j$, we have $|\triangle_{K_j}| \leq 1$. Let us then consider the two possible values of $|\triangle_{K_j}|$.

If $|\triangle_{K_j}| = 1$, let $\delta$ be the only triangle of $\triangle_{K_j}$, and let $v_0$ be the vertex between $c_j^1$ and $c_j^2$ not used by $\delta$.

Let us first consider the subcase where there is no triangle of $\triangle$ having $v_0$ as a vertex, then let us prove that $\theta_j \notin V(\triangle)$. Indeed, by contradiction let $\delta'$ be the triangle of $\triangle$ with $\theta_j$ as one of its vertices. By construction, since $\theta_j$ is neither the head nor the head of a backward arc, we have $\delta' = (v, \theta_j, c_{j'}^{p'})$ for some $j' > j$, which means that $j'$ is nice. However, $\triangle$ cannot contain an outer triangle $(v, \theta_{j'}, c_{j'}^p)$ for some $v \in V(X)$ since $c_{j'}^1$ and $c_{j'}^2$ are already used in one $Q_j^p$ and the other in $\delta'$, a contradiction. It means that if $v_0 \notin V(\triangle)$, then $\theta_j \notin V(\triangle)$. This proves that the property (i) is verified and since $|\triangle_{K_j}| = 1$, it contradicts the fact that $j$ is not nice.

Otherwise, that is if $v_0 \in V(\triangle)$, then let $\delta'$ the triangle of $\triangle$ which contains $v_0$. By construction, we cannot have $v_0$ as the first vertex of $\delta'$, since there is no backward of $T_{f(\mathcal{F})}$ with $v_0$ as its head. Suppose $v_0$ is the second vertex of $\delta'$, it implies that the last vertex is $c_{j'}^p$, for some $p \in [\![1, 2]\!]$ and some $j' > j$ which is nice by assumption. However, the triangle $\delta'$ proves that the property (i) of the lemma is not satisfied. Moreover, both $c_{j'}^1$ and $c_{j'}^2$ are used in the two triangles, so the property (ii) is not satisfied neither, contradicting the fact that $j'$ is nice, and so the maximality of $j$: a contradiction. Then, we have $\delta' = (u, v, v_0)$ for some vertices $u$ and $v$. With the same reasoning that before, we prove that $\theta_j \notin V(\triangle)$.

Then, we can replace $\delta'$ by the triangle $(u, \theta_j, v_0)$, which concludes the case where $|\triangle_{K_j}| = 1$.

If $|\triangle_{K_j}| = 0$. Notice first that by maximality of $j$, the vertex $d_j^2$ cannot be in a triangle of $\triangle$ as it could only be used in a triangle $(v, d_j^2, c_{j'}^p)$ with $j' > j$. Let $Z$ be the set $V(\triangle) \cap \{c_j^1, c_j^2\}$. If $|Z| = 0$, then by maximality of $j$, we get $d_j^1 \notin V(\triangle)$ and $\theta_j \notin V(\triangle)$, and thus we add to $\triangle$ the triangle $(d_j^1, c_j^1, d_j^2)$; we can now suppose that $|Z| \neq 0$.

Otherwise, if $|Z| = 1$, let $c_j^p$ be the vertex in $Z$, and $\delta$ the triangle of $\triangle$ such that $c_j^p \in V(\delta)$. By maximality of $j$, we necessarily have $\delta = (u, v, c_j^p)$ for some vertices $u$ and $v$. If $v \neq \theta_j$ then by maximality of $j$ we have $\theta_j \notin V(\triangle)$, and thus we can swap $v$ and $\theta_j$ in $\delta$; we can now suppose that $\theta_j \in V(\delta)$. Before this swap, we could have had $v = d_j^1$, but now by maximality of $j$ we know that $d_j^1$ is unused. This implies that $d_j^1 \notin V(\triangle)$, and we add $(d_j^1, c_j^{3-p}, d_j^2)$ to $\triangle$, so we have $|Z| = 2$.

In this case, if there exists a triangle $\delta$ of $\triangle$ with $Z \subseteq V(\delta)$, then we necessarily have $\delta = (u, c_j^1, c_j^2)$. Using the same arguments as above, we get that $\{\theta_j, d_1^j\} \cap V(\triangle) = \emptyset$, and thus we swap $c_j^1$ by $\theta_j$ in $\delta$ and add the triangle $(d_j^1, c_j^1, d_j^2)$ to $\triangle$. Otherwise, let $\delta_1$ and $\delta_2$ be the two triangles of $\triangle$ such that $c_j^1 \in V(\delta_1)$ and $c_j^2 \in V(\delta_2)$ . This implies that $\delta_1 = (u_1, v_1, c_j^1)$ and $\delta_2 = (u_2, v_2, c_j^2)$, for some vertices $u_1$, $u_2$, $v_1$ and $v_2$. If $\theta_j \notin (V(\delta_1) \cup V(\delta_2))$ then $\theta_j \notin V(\triangle)$, and we swap $v_1$ with $\theta_j$. Therefore, now we can suppose that $\theta_j$ is in $\delta_x$, for some $x \in [\![1, 2]\!]$. Then, if the vertex $d_j^1$ is not in the triangle $\delta_{3-x}$ (that is the triangle where $\theta_j$ is not), then $d_j^1$ is not in any triangle of $\triangle$, and thus we swap $v_{3-x}$ with $d_j^1$. We can now assume that $d_j^1 \in V(\delta_{3-x})$, so we remove $\delta_{3-x}$ from $\triangle$, and add the triangle $(d_j^1, c_j^{3-x}, d_j^2)$ instead. □

The previous lemma proves that we can restructure a vertex-disjoint triangle packing $\triangle$ of $T_{f(\mathcal{F})}$ into a partition defined by the parts $\triangle_X$, $\triangle_K$ and $\triangle_O$.

**Corollary 1.** *For any vertex-disjoint triangle packing $\triangle$, we can compute in polynomial time another packing $\triangle'$ such that $|\triangle'| \geq |\triangle|$, and $\triangle'$ only contains outer, variable-inner, and clause-inner triangles.*

*Proof.* In the solution $\triangle'$ of Lemma 22.1, given any triangle $\delta$ in $\triangle'$, either $V(\delta)$ intersects $V(K_j)$ for some $j \in [\![1, m]\!]$ and then $\delta$ is an outer or a clause-inner triangle, or $V(\delta) \subseteq V(X_i)$ for some $i \in [\![1, n]\!]$ as there is no backward arc $(u, v)$ with $u \in V(X_{i_1})$ and $v \in V(X_{i_2})$ with $i_1 \neq i_2$, proving that $\delta$ is variable-inner. □

Recall that we want $P_i$ and $\overline{P_i}$ to correspond to set the variable $x_i$ to true and false, respectively. Therefore, we want to prove that we can restructure variable-inner triangles into $P_i$ or $\overline{P_i}$. To do so, we have to focus of the structures of the triangles in $X_i$. Thus, let define $Lft_i$ and $Rgt_i$ as the subsets of vertices $V(L_i) \cup V(L'_i) \cup V(\overline{L_i}) \cup V(\overline{L'_i})$ and $V(A_i) \cup V(B_i) \cup \{\alpha_i\} \cup V(A'_i) \cup V(B'_i)$, respectively. Furthermore, given any vertex-disjoint triangle packing $\triangle$ satisfying Lemma 22.1 and any $i \in [\![1, n]\!]$, we define the following sets:

- $\triangle_{Lft_i} = \{\delta \in \triangle_{X_i} \colon V(\delta) \subseteq Lft_i\}$,
- $\triangle_{Rgt_i} = \{\delta \in \triangle_{X_i} \colon V(\delta) \subseteq Rgt_i\}$,

- $\triangle_{Lft_iRgt_i} = \{\delta \in \triangle_{X_i} : V(\delta) \cap Lft_i \neq \emptyset \text{ and } V(\delta) \cap Rgt_i \neq \emptyset\}$,
- $\triangle_{O_i} = \{\delta \in \triangle_O : V(\delta) \cap V(X_i) \neq \emptyset\}$.

Observe that the three sets define a partition of $\triangle_{X_i}$, and that triangles of $\triangle_{Lft_i}$ have all their vertices included in the same piece digraph. We also define $i^{th}$ signature of $\triangle$, $\mu_i(\triangle)$, which correspond of how the triangles using vertices of $X_i$ are distributed; thus we set $\mu_i(\triangle) = (|\triangle_{Lft_i}|, |\triangle_{Lft_iRgt_i}|, |\triangle_{Rgt_i}|, |\triangle_{O_i}|)$. For example, notice that for any packing of triangle $\triangle'$ such that $\triangle'_{X_i} = P_i$, we have $\mu_i(\triangle') = (4, 2, 5, |\triangle'_{O_i}|)$. Finally, we define by $\nu_i(\triangle)$ the value of $|\triangle_{Lft_i}| + |\triangle_{Lft_iRgt_i}| + |\triangle_{Rgt_i}| + |\triangle_{O_i}|$.

We now give some properties of these sets which will be useful later.

**Claim 22.1.** *Given a vertex-disjoint triangle packing $\triangle$ satisfying Lemma 22.1 and the sets $\triangle_{Lft_i}$, $\triangle_{Lft_iRgt_i}$, $\triangle_{Rgt_i}$, and $\triangle_{O_i}$ as described previously, then for any $i \in [\![1, n]\!]$, we have the following properties:*

($p_1$)**:** $|\triangle_{Lft_i}| \leq 4$, $|\triangle_{Lft_iRgt_i}| \leq 4$, $|\triangle_{Rgt_i}| \leq 5$, *and* $|\triangle_{O_i}| \leq 3$,

($p_2$)**:** *if* $V(\triangle_{Lft_iRgt_i}) \cap Rgt_i$ *contains the at least one of the subsets* $\{a_i^3, b_i^3\}$, $\{a_i'^3, b_i'^3\}$, $\{a_i^3, b_i'^3\}$ *and* $\{a_i'^3, b_i^3\}$, *then* $|\triangle_{Rgt_i}| \leq 4$,

($p_3$)**:** *if* $|\triangle_{Lft_i}| = 4$, *then* $|\triangle_{O_i}| \leq 4 - |\triangle_{Lft_iRgt_i}|$,

($p_4$)**:** *if* $|\triangle_{Lft_i}| = 3$ *and* $|\triangle_{Lft_iRgt_i}| = 4$, *then* $|\triangle_{O_i}| \leq 1$.

*Proof.* We prove in succession the different properties.

The first property ($p_1$) is just a direct consequence of how is defined $T_{f(\mathcal{F})}$: we have $|\triangle_{Lft_i}| \leq 4$ since there is only four backward arcs in $T_{f(\mathcal{F})}[Lft_i]$; the same reasoning proves that $|\triangle_{Lft_iRgt_i}| \leq 4$; the inequality $|\triangle_{Rgt_i}| \leq 5$ holds because $|Rgt_i| = 17$; finally, $|\triangle_{O_i}| \leq 3$ since each variable appears exactly three times, so we have three backward arcs joining $c_j^p$ and $X_i$.

The property ($p_2$) can informally be understood that if we do not make the triangles of $\triangle_{Lft_iRgt_i}$ with the right pair of vertices of $Rgt_i$, we "lose" one triangle. Formally, for any piece digraph $\Pi$ of $T_{f(\mathcal{F})}[Rgt_i]$, let $m_\Pi$ be the only arc of $\Pi$ with one of its endpoint in $\Pi$ and the other one in another piece digraph of $T_{f(\mathcal{F})}[Rgt_i]$; we call such arcs *medium arcs*. One may notice that we have $m_{A_i} = m_{A'_i}$ and $m_{B_i} = m_{B'_i}$. Finally, let $\overleftarrow{A}(\triangle_{Rgt_i})$ be the set of backward arcs used in a triangle of $\triangle_{Rgt_i}$. Observe that $\overleftarrow{A}(\triangle_{Rgt_i})$ contains only small and medium arcs. Recall we are in the case where $V(\triangle_{Lft_iRgt_i}) \cap Rgt_i$ contains the at least one of the subsets $\{a_i^3, b_i^3\}$, $\{a_i'^3, b_i'^3\}$, $\{a_i^3, b_i'^3\}$ and $\{a_i'^3, b_i^3\}$, let $Z$ be such subset. Notice that for any medium arc $m$, there is a vertex $v$ in $Z$ such that the piece digraph $\Pi$ of $v$ also contains an endpoint of $m$. It implies that $\overleftarrow{A}(\triangle_{Rgt_i})$ cannot contain $m_\Pi$. So we have $|\triangle_{Rgt_i}| = |\overleftarrow{A}(\triangle_{Rgt_i})| \leq 4$, proving the second property.

The two last properties come from the fact that for any piece digraph $\Pi$ of $T_{f(\mathcal{F})}[Lft_i]$, we cannot have at the same time $\delta_1 \in \triangle_{O_i}$, $\delta_2 \in \triangle_{Lft_iRgt_i}$ and $\delta_3 \in \triangle_{Lft_i}$ with $V(\delta_i) \cap V(\Pi) \neq \emptyset$.

□

We can now prove that we can restructure variable-inner triangles into $P_i$ or $\overline{P_i}$. This will allow us to get a well-defined assignation for the instance $\mathcal{F}$ of MAX 2-SAT(3).

**Lemma 22.2.** *For any vertex-disjoint triangle packing $\triangle$, we can compute in polynomial time another packing $\triangle'$ such that $|\triangle'| \geq |\triangle|$, $\triangle'$ satisfies Lemma 22.1, and for every $i \in [\![1, n]\!]$, $\triangle'_{X_i} = P_i$ or $\triangle'_{X_i} = \overline{P_i}$.*

*Proof.* Let $\triangle$ be the packing of triangles obtained after applying Lemma 22.1. By Corollary 1, we can partition the triangles of $\triangle$ into $\triangle_X$, $\triangle_K$ and $\triangle_O$. Let us prove that if $\triangle$ does not satisfy the statement of the lemma, we can restructure $\triangle$ to increase the number of indices $i$ such that $\triangle_{X_i} = P_i$ or $\triangle_{X_i} = \overline{P_i}$, while still satisfying Lemma 22.1, which will prove the lemma. In the following, let $i$ be an index such that $\triangle_{X_i} \neq P_i$ and $\triangle_{X_i} \neq \overline{P_i}$.

Our objective is to restructure $\triangle$ into a solution $\triangle'$ with $\triangle' = (\triangle - (\triangle_{X_i} \cup \triangle_{O_i}) \cup (\triangle'_{X_i} \cup \triangle'_{O_i})$. We will define $\triangle'_{X_i}$ and $\triangle'_{O_i}$ in order to have the following properties $(q)$:

$(q_1)$**:** $\triangle'_{X_i} = P_i$ or $\triangle'_{X_i} = \overline{P_i}$,

$(q_2)$**:** $\triangle'_{O_i} \subseteq \triangle_{O_i}$,

$(q_3)$**:** $\nu_i(\triangle') \geq \nu_i(\triangle)$,

$(q_4)$**:** the triangles of $\triangle'_{X_i} \cup \triangle'_{O_i}$ are vertex-disjoint.

Notice that $(q_2)$, $(q_4)$ and our previous observation imply that all triangles of $\triangle'$ are still vertex-disjoint. Indeed, as $\triangle$ satisfies Lemma 22.1, the only triangles of $\triangle$ intersecting $X_i$ are $\triangle_{X_i} \cup \triangle_{O_i}$, and thus replacing them with $\triangle'_{X_i} \cup \triangle'_{O_i}$ satisfying the above property implies that all triangles of $\triangle'$ are vertex-disjoint. Moreover, $\triangle'$ will still satisfy Lemma 22.1 even with $(q_2)$ since removing outer triangles cannot violate property of Lemma 22.1. Finally, $(q_3)$ implies that $|\triangle'| \geq |\triangle|$. Thus, if we can construct in polynomial time $\triangle'_{X_i}$ and $\triangle'_{O_i}$ satisfying these properties, it will be sufficient to prove the lemma.

Recall that if a solution $\triangle'$ satisfies $\triangle'_{X_i} = P_i$ or $\triangle'_{X_i} = \overline{P_i}$, then the $i^{th}$ signature of $\triangle'$ is $\mu_i(\triangle') = (4, 2, 5, |\triangle'_{O_i}|)$, and we have $\nu_i(\triangle') = 11 + |\triangle'_{O_i}|$. Furthermore, by property $(p_3)$ of Claim 22.1, we have $|\triangle'_{O_i}| \in [\![1, 2]\!]$.

In the following we write $(u_1, u_2, u_3, u_4) \leq (v_1, v_2, v_3, v_4)$ if $u_i \leq v_i$, for any $i \in [\![1, 4]\!]$. Let us proceed by case analysis according to the value $|\triangle_{Lft_iRgt_i}|$. Remember that using property $(p_1)$ of Claim 22.1, we have $|\triangle_{Lft_iRgt_i}| \leq 4$.

If $|\triangle_{Lft_iRgt_i}| \leq 1$, then according to $(p_1)$, we get $\mu_i(\triangle) \leq (4, 1, 5, o)$, where $o \in [\![1, 3]\!]$. In this case, we set $\triangle'_{O_i} = \triangle_{O_i} - \{\delta \in \triangle \colon \overline{\ell_i^2} \in V(\delta)\}$ and $\triangle'_{X_i} = P_i$. It is easy to see that these two new packings verify the properties $(q)$: $(q_1)$, $(q_2)$ and $(q_4)$ are obvious. Then, we have $\nu_i(\triangle') \geq \nu_i(\triangle)$ as $\mu_i(\triangle') \geq (4, 2, 5, o - 1)$.

In the case where $|\triangle_{Lft_iRgt_i}| = 2$, then we have $\mu_i(\triangle) = (l, 2, r, o)$, for some $l$, $r$ and $o$. If $l \leq 3$, then $\mu_i(\triangle) \leq (3, 2, 5, o)$ by $(p_1)$ and we set $\triangle'_{O_i} = \triangle_{O_i} - \{\delta \in \triangle \colon \overline{\ell_i^2} \in V(\delta)\}$ and $\triangle'_{X_i} = P_i$. This satisfies $(q)$ with the same arguments than previously. Let us now turn to case where $l = 4$. Let $\triangle_{Lft_iRgt_i} =$

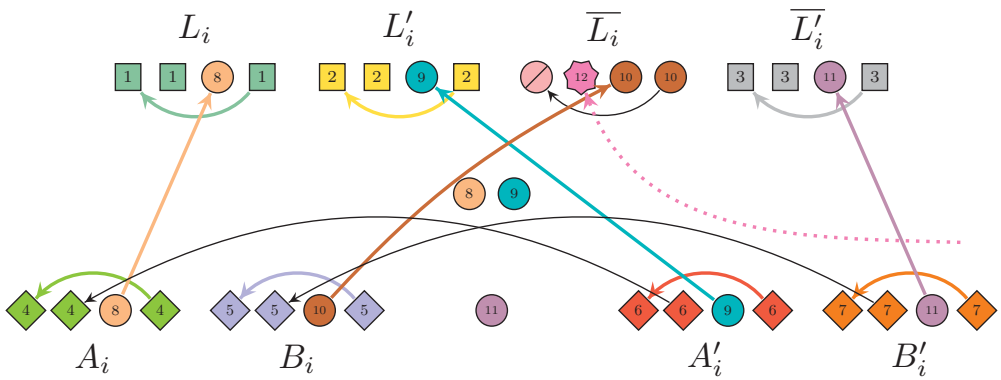



**Figure 4.8** – Example showing a "bad shaped" solution where $\mu_i(\triangle) = (3, 4, 4, 1)$. Here, we have $\triangle_{Lft_i} = \{\delta_1, \delta_2, \delta_3\}$ (the square-shaped vertices), $\triangle_{Rgt_i} = \{\delta_4, \delta_5, \delta_6, \delta_7\}$ (the diamond-shaped vertices), $\triangle_{Lft_iRgt_i} = \{\delta_8, \delta_9, \delta_{10}, \delta_{11}\}$ (the circle-shaped vertices), and $\triangle_{O_i} = \{\delta_{12}\}$ (the star-shaped vertex). The three vertices of a triangle $\delta_i$ are labeled $i$ and share the same color, its corresponding backward is also depicted with the same color. The triangle $\delta_{12}$ has its other vertices outside the variable gadget.

$\{\delta_1, \delta_2\}$. Let us first suppose that the triangles of $\triangle_{Lft_iRgt_i}$ contain $\{e_i^p, e_i^q\}$ with $\{e_i^p, e_i^q\} \in \{\{e_i^1, e_i^3\}, \{e_i^1, e_i^4\}, \{e_i^2, e_i^3\}, \{e_i^2, e_i^4\}\}$. Recall that the arcs $e_i^p$ are the backward arcs from $Rgt_i$ to $Lft_i$. In this case, one can notice that by property $(p_2)$, we have $r \leq 4$, implying $\mu_i(\triangle) \leq (4, 2, 4, o)$. Then, we set we set $\triangle'_{O_i} = \triangle_{O_i} - \{\delta \in \triangle \colon \overline{\ell_i^2} \in V(\delta)\}$ and $\triangle'_{X_i} = P_i$ which satisfies $(q)$ too since $\nu_i(\triangle') \geq \nu_i(\triangle)$ as $\mu_i(\triangle') \geq (4, 2, 5, o-1)$. Otherwise, if the triangles of $\triangle_{Lft_iRgt_i}$ do not contain $\{e_i^i, e_i^j\}$ as described above, then we have $\mu_i(\triangle) \leq (4, 2, 5, o)$, where $o \in [\![1, 2]\!]$ by $(p_3)$. Let us first suppose that $\delta_1$ contains $e_i^1$ and $\delta_2$ contains $e_i^2$. In that case, it implies that the only piece digraph of $Lft_i$ with a vertex in a triangle of $\triangle_{O_i}$ is $\overline{L_i}$. Thus, we set $\triangle'_{O_i} = \triangle_{O_i}$ and $\triangle'_{X_i} = \overline{P_i}$. Finally, in the case where $\delta_1$ contains $e_i^3$ and $\delta_2$ contains $e_i^4$, then the pieces digraphs of $Lft_i$ with a vertex in a triangle of $\triangle_{O_i}$ are in $\{L_i, L'_i\}$. Therefore, we set $\triangle'_{O_i} = \triangle_{O_i}$ and $\triangle'_{X_i} = P_i$. In both cases, these packings verify the properties $(q)$, as in particular $\mu_i(\triangle') = (4, 2, 5, o)$.

If $|\triangle_{Lft_iRgt_i}| = 3$, then the statement of property $(p_2)$ is verified, that is $\mu_i(\triangle) \leq (l, 3, 4, o)$, for some $l$ and $o$. If we also have $l \leq 3$, the sets $\triangle'_{O_i} = \triangle_{O_i} - \{\delta \in \triangle \colon \overline{\ell_i^2} \in V(\delta)\}$ and $\triangle'_{X_i} = P_i$ verify the properties $(q)$; in particular, we have $\nu_i(\triangle') \geq \nu_i(\triangle)$ as $g_i(\triangle') \geq (4, 2, 5, o-1)$. Otherwise, if $l = 4$ then $o \leq 1$ by property $(p_3)$. Thus, we can define $\triangle'_{X_i} = P_i$ if $V(\triangle_{O_i}) \cap (V(L_i) \cup V(L'_i)) \neq \emptyset$, and we define $\triangle'_{X_i} = \overline{P_i}$, otherwise. Furthermore, we set $\triangle'_{O_i} = \triangle_{O_i}$. These sets satisfy the properties $(q)$, as in particular we have $\mu_i(\triangle') = (4, 2, 5, o)$.

Finally, consider the case where $|\triangle_{Lft_iRgt_i}| = 4$, then we have $\mu_i(\triangle) = (l, 4, r, o)$, for some $l$, $r$ and $o$. If we also have $l = 4$, then $o = 0$ by property $(p_3)$, and we have $r \leq 3$ as $l + 4 + r \leq \frac{|V(X_i)|}{3}$. Thus, we set $\triangle'_{O_i} = \triangle_{O_i} = \emptyset$ and $\triangle'_{X_i} = \overline{P_i}$ (which is arbitrary in this case), and we have properties $(q)$ as $\mu_i(\triangle') \geq$

$(4, 2, 5, 0)$. If $l = 3$ (an example of this case is depicted Figure 4.8), then $r \leq 4$ by property $(p_2)$, and $o \leq 1$ by property $(p_4)$. It implies that $\mu_i(\triangle) \leq (3, 4, 4, o)$. Thus, we can define $\triangle'_{X_i} = P_i$ if $V(\triangle_{O_i}) \cap (V(L_i) \cup V(L'_i)) \neq \emptyset$, and we define $\triangle'_{X_i} = \overline{P_i}$, otherwise. Furthermore, we set $\triangle'_{O_i} = \triangle_{O_i}$. These packings satisfy $(q)$ as in particular $\mu_i(\triangle') = (4, 2, 5, o)$. Finally, if $l \leq 2$ then $\mu_i(\triangle) \leq (2, 4, 4, o)$ by property $(p_2)$. In this case, we set $\triangle'_{O_i} = \triangle_{O_i} - \{\delta \in \triangle \colon \overline{\ell_i^2} \in V(\delta)\}$ and $\triangle'_{X_i} = P_i$, which verify the properties $(q)$: we have $\nu_i(\triangle') \geq \nu_i(\triangle)$ as $\mu_i(\triangle') \geq (4, 2, 5, o - 1)$.

Therefore, we prove that we can in polynomial time increase the number of index satisfying the statement of the lemma. By iterating this process until all the indices are good, the lemma is proved. □

### 4.2.3 Proof of the $L$-Reduction

We are now ready to prove the main lemma, and also the main theorem of this section.

**Lemma 22.3.** *Let $\mathcal{F}$ be an instance of* Max 2-SAT(3)*. There exists an assignment of $\mathcal{F}$ satisfying at least $k$ clauses if and only if there exists a vertex-disjoint triangle packing $\triangle$ of $T_{f(\mathcal{F})}$ with $|\triangle| \geq 11n + m + k$, where $n$ and $m$ are respectively the number of variables and clauses in $\mathcal{F}$. Moreover, this assignment can be computed from $\triangle$ in polynomial time.*

*Proof.* Let $a$ be an assignment of $\mathcal{F}$ satisfying $k$ clauses. For any $i \in [\![1, n]\!]$, we define the packings $A_i$ by $A_i = P_i$ if $a(x_i) =$ true, and $A_i = \overline{P_i}$ otherwise. Then, we start constructing our solution $\triangle$ by adding the set $\cup_{i \in [\![1,n]\!]} A_i$; we added $11n$ triangles. Now, let $\{c_{j_q} \colon q \in [\![1, k]\!]\}$ be the $k$ clauses satisfied by the assignation $a$. Furthermore, for any $q \in [\![1, k]\!]$, we denote by $i_q$ the index of a literal satisfying $c_{j_q}$, and let $p_q \in [\![1, 2]\!]$ the index such that the vertex $c_{j_q}^{p_q}$ is the tail of a backward arc pointing to the vertex which corresponds to this literal. We also define $Z_q$ as the set $\{\ell_{i_q}^2, \ell'^2_{i_q}\}$ if the variable $x_{i_q}$ is positive in $c_{j_q}$, or the set $\{\overline{\ell_{i_q}^2}\}$ otherwise.

For any $j \in [\![1, m]\!]$, if $j = j_p$ for some $p \in [\![1, k]\!]$—meaning that $c_j$ is a satisfied clause, we add to $\triangle$ the triangle $Q_j^{3-p_q}$. Otherwise, if $c_j$ is not satisfied by $a$, we arbitrarily add the triangle $Q_j^1$. This corresponds to $m$ new triangles in $\triangle$.

Finally, for any $q \in [\![1, k]\!]$ we add to $\triangle$ the triangle $\delta_q = (y_q, \theta_{j_q}, c_{j_q}^{p_q})$, where $y_q \in Z_q$ and picked such that $y_q$ is not already used in another triangle. Notice that such a vertex $y_p$ always exists as triangles of $A_i$ do not intersect $Z_q$ (by definition of the $A_i$), and as there are at most two clauses that are true due to positive literal, and one clause that is true due to a negative literal. Thus, $\triangle$ has $11n + m + k$ vertex disjoint triangles.

Conversely, let $\triangle$ a vertex-disjoint triangle packing of $T_{f(\mathcal{F})}$ with $|\triangle| \geq 11n + m + k$. By Lemma 22.2, we can construct in polynomial time a solution $\triangle'$ from $\triangle$ such that: $|\triangle'| \geq |\triangle|$; $\triangle'$ only contains outer, variable or clause-inner triangles; for any $j \in [\![1, m]\!]$, there exists $p \in [\![1, 2]\!]$ such that $\triangle'_{K_j} = Q_j^p$; for each $i \in [\![1, n]\!]$, $\triangle'_{X_i} = P_i$ or $\triangle'_{X_i} = \overline{P_i}$.

This implies that the $k' \geq k$ remaining triangles must be outer ones. Let $\{\delta_q \colon q \in [\![1, k']\!]\}$ be these $k'$ outer triangles, with $\delta_q = (y_q, \theta_{j_q}, c_{j_q}^{p_q})$. Let us define the following assignation $a$: for each $i \in [\![1, n]\!]$, we set $x_i$ to true if $\triangle'_{X_i} = P_i$, and false otherwise. By construction of $T_{f(\mathcal{F})}$, the assignation $a$ satisfies at least clauses $\{c_{j_q} \colon q \in [\![1, k']\!]\}$, that is $k' \geq k$ clauses in total. □

Let us now check that Lemma 22.3 leads to $L$-reduction from Max 2-SAT(3) to DTT.

**Theorem 23.** *The problem* Maximum Disjoint Triangle Packing in Tournament *is* APX*-hard, and thus does not admit a* PTAS*, unless* P = NP*. This result also holds when the tournament is sparse.*

*Proof.* Let $\mathrm{opt}_1$ and $\mathrm{opt}_2$ be the optimal value of $\mathcal{F}$ and $T_{f(\mathcal{F})}$, respectively. Notice that Lemma 22.3 implies that $\mathrm{opt}_2 = \mathrm{opt}_1 + 11n + m$, where $n$ and $m$ are respectively the number of variables and clauses in $\mathcal{F}$. It is known that $\mathrm{opt}_1 \geq \frac{3}{4}m$. As each variable has at least one positive and one negative occurrence, we also have $n \leq m$. Thus, we get $\mathrm{opt}_2 = \mathrm{opt}_1 + 11n + m \leq \alpha \cdot \mathrm{opt}_1$, for an appropriate constant $\alpha$.

Now, given a solution $\triangle$ of $T_{f(\mathcal{F})}$, according to Lemma 22.3 we can construct in polynomial time an assignment of $\mathcal{F}$ which satisfies $k$ clauses, and such that $k \geq |\triangle| - 11n - m$. Thus, we have $\mathrm{opt}_1 - k \leq \mathrm{opt}_1 + 11n + m - |\triangle| = \mathrm{opt}_2 - |\triangle|$. □

Notice that the reduction of Theorem 23 does not imply the NP-hardness of Perfect-DTT, as there remain some unused vertices. However, it is straightforward to adapt the reduction by adding backward arcs whose heads (resp. tails) are before (resp. after) the tournament $T_{f(\mathcal{F})}$ to consume the remaining vertices. This leads to the following result.

**Theorem 24.** Perfect Packing Disjoint-Triangle in Tournament *is* NP*-hard. This result also holds when the tournament is sparse.*

*Proof.* Let $(\mathcal{F}, k)$ be an instance of the decision problem of Max 2-SAT(3), and let $T_{f(\mathcal{F})}$ be the tournament defined in our previous reduction. Let $N$ be the total number of vertices in $T_{f(\mathcal{F})}$, that is $35n + 5m$, and $x^*$ be the number of vertices used in a solution obtained with Lemma 22.3, that is $x^* = 33n + 3m + 3k$. Finally, let $n'$ be the number of remaining vertices. Thus, we have $n' = N - x^*$.

We now construct a new tournament $T'_{f(\mathcal{F})}$ by adding $2n'$ vertices in $T_{f(\mathcal{F})}$ as follows: for $i \in [\![1, 2]\!]$, we define the tournament $R_i$ as the transitive tournament of $n'$ vertices, that is $\sigma(R_i) = r_i^1, \ldots, r_i^{n'}$ and $\overleftarrow{A}(R_i) = \emptyset$. Then, we concatenate these two tournaments to $T_{f(\mathcal{F})}$ such that $T'_{f(\mathcal{F})} = R_1 \oplus T_{f(\mathcal{F})} \oplus R_2$. We refer the reader to Figure 4.9 to see an example of this construction. Then, we add to $\overleftarrow{A}(T'_{f(\mathcal{F})})$ the set of arcs $\{(r_2^p, r_1^p) \colon p \in [\![1, n']\!]\}$, and we call such arcs *perfect arcs*. Furthermore, we say that a triangle $\delta = (u, v, w)$ is perfect if $u \in V(R_1)$, $v \in V(T_{f(\mathcal{F})})$, and $w \in V(R_2)$. Let us now prove that there are at least $k$ clauses satisfiable in $\mathcal{F}$ if and only if there exists a perfect packing of vertex-disjoint triangles in $T'_{f(\mathcal{F})}$.

Given an assignment satisfying $k$ clauses, we define a solution $\triangle$ with $V(\triangle) \subseteq V(T_{f(\mathcal{F})})$ as it has been described in Lemma 22.3; roughly, we have the triangles of

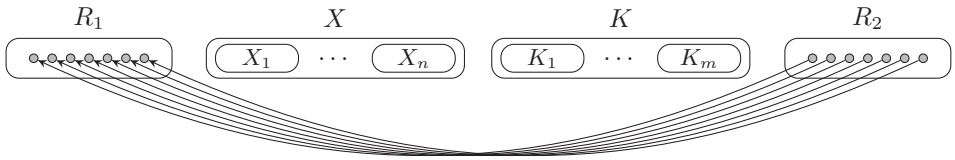



**Figure 4.9** – Instance with the perfect arcs going along from $R_2$ to $R_1$. Each of these arcs will consume one vertex of the previous tournament in order to create a perfect triangle.

$P_i$ or $\overline{P_i}$ for each $i \in [\![1, n]\!]$, a triangle $Q_j^x$ for each $j \in [\![1, m]\!]$, and an outer triangle $t_q$ with $q \in [\![1, k]\!]$ for each satisfied clause. Then, we have $|\triangle| = 11n + m + k$. This implies that $|V(T_{f(\mathcal{F})}) - V(\triangle)| = n'$, and thus we use $n'$ remaining vertices of $V(T_{f(\mathcal{F})})$ by adding $n'$ perfect triangles to the solution $\triangle$.

Conversely, let $\triangle'$ be a perfect packing of vertex-disjoint triangles $T'_{f(\mathcal{F})}$. We define $\triangle$ as the triangles of $\triangle'$ with all its vertices in $T_{f(\mathcal{F})}$, that is $\{\delta \in \triangle' \colon V(\delta) \subseteq V(T_{f(\mathcal{F})})\}$. Furthermore, let $X$ be the set of remaining vertices, $X = V(T_{f(\mathcal{F})}) - V(\triangle)$. As $\triangle'$ is a perfect packing of $T'_{f(\mathcal{F})}$, the vertices of $X$ must be used by $|X|$ perfect triangles of $\triangle'$, implying $|X| \leq n'$ and $|\triangle| \geq 11n + m + k$. As $\triangle$ is set of vertex disjoint triangles of $T_{f(\mathcal{F})}$ of size at least $11n + m + k$, by Lemma 22.3 this implies that at least $k$ clauses are satisfiable in $\mathcal{F}$. □

To establish the kernel lower bound of subsection 4.3.2, we also need the NP-hardness of Perfect-DTT where instances have a slightly simpler structure (to the price of losing the property that there exists a feedback arc set which is a matching).

**Theorem 25.** Perfect Packing Disjoint-Triangle in Tournament *remains* NP*-hard even restricted to tournament $T$ admitting the following linear representation:*

- *$T$ is the concatenation of two tournaments $X$ and $K$, where:*
  - *$K = K_1 \oplus \cdots \oplus K_m$ for some $m$, where for each $j \in [\![1, m]\!]$ we have $\sigma(K_j) = (\theta_j, d_j)$,*
  - *$X = R_1 \oplus X_1 \oplus \cdots \oplus X_n \oplus R_2$, where each $X_i$ has is a copy of the variable gadget of section 4.2, $R_i = \{r_i^p \colon p \in [\![1, n']\!]\}$ where $n' = 2n - m$, and in addition $\overleftarrow{A}(X)$ also contains $\{(r_2^p, r_1^p) \colon p \in [\![1, n']\!]\}$.*
- *for any $a \in \overleftarrow{A}(T)$, $V(a) \cap V(K) \neq \emptyset$ implies $a = (v, d_j)$ for $v \in X$ (there are no backward arc included in $K$, and none of the $\theta_j$ is endpoint of a backward arc).*

*Proof.* We adapt the reduction of section 4.2, reducing now from Max 3-SAT(3) instead of Max 2-SAT(3). We first recall the definition of Max 3-SAT(3) which is known to be APX-hard [11].

An instance of Max 3-SAT(3) $\mathcal{F}$ is a set of $m$ clauses $\{c_1, \ldots, c_m\}$ over $n$ variables $\{x_1, \ldots, x_n\}$ and such that each clause has three variables, and each

variables appears *at most* three times. The objective is to find an assignation of the variable satisfying the maximum number of clauses.

For each variable $x_i$ with $i \in [\![1, n]\!]$, we create a tournament $X_i$ exactly as described in section 4.2, and we define the tournament $X$ as the concatenation $X_1 \oplus \cdots \oplus X_n$. Similarly, for each clause $c_j$, we create a tournament $K_j$ with $V(K_j) = (\theta_j, d_j)$. We also define the two tournaments $R_1$ and $R_2$ such that $\sigma(R_i) = (r_i^1, \ldots, r_i^{n'})$, with $n' = 2n - m$. We can now define the tournament $K$ as the concatenation $R_1 \oplus X_1 \oplus \cdots \oplus X_n \oplus R_2$, and we add the set $R$ of backward arcs given by $\{(r_2^p, r_1^p) \colon p \in [\![1, n']\!]\}$, called the perfect arcs.

Let us now define our final tournament $T$ as the concatenation $X \oplus K$. We also add to $\overleftarrow{A}(T)$ the following backward arcs from $V(K)$ to $V(X)$ (again, we follow the construction of section 4.2, but now each $d_j$ will be the tail of three backward arcs). If $c_j$ is a clause of $\mathcal{F}$ given by $l_{i_1} \vee l_{i_2} \vee l_{i_3}$, then we add the backward arcs $(d_j, v_{i_1})$, $(d_j, v_{i_2})$, and $(d_j, v_{i_3})$ where $v_{i_p}$ is the vertex in $V(X_{i_p})$ corresponding to $l_{i_p}$; more precisely, if $l_{i_p}$ is a negative occurrence of variable $x_{i_p}$ we choose $v_{i_p} = \overline{\ell_{i_c}^2}$. Otherwise, we choose $v_{i_p} \in \{\ell_{i_c}^2, \ell_{i_c}'^2\}$, in such a way that there exists a unique arc $a \in \overleftarrow{A}(T)$ such that $h(a) = v$. This is always possible as each variable has at most two positive occurrences and one negative.

Notice that the tournament $T$ satisfies the structure defined in the statement of the theorem. We define an outer and variable-inner triangle as in subsection 4.2.2. One can notice that there are no more clause-inner triangle with this construction. Finally, we say that a triangle $(u, v, w)$ is perfect if $u \in V(R_1)$, $v \in V(T_{f(\mathcal{F})})$, and $w \in V(R_2)$. Let us prove that there is an assignment satisfying the $m$ clauses of $\mathcal{F}$ if and only if $T$ has a perfect packing of vertex-disjoint triangles.

Given an assignment satisfying the $m$ clauses, we define a solution $\triangle$ containing only outer, variable-inner and perfect triangles. The variable-inner triangle taken are defined as in Lemma 22.3 (that is triangles of $P_i$ or $\overline{P_i}$ for each $i \in [\![1, n]\!]$). Now, for each clause $c_j$ satisfied by a literal $l_{i_p}$ we create an outer triangle $(v_{i_p}, \theta_j, d_j)$. It remains now $2n - m = n'$ vertices of $X$, that we use by adding $n'$ perfect triangles to $\triangle$. It concludes this direction of the proof.

For the necessary condition, let $\triangle$ be a perfect packing of $T$. Notice that the restructuring lemmas of subsection 4.2.2 do not directly usable because we now have the perfect arcs. However, we can adapt in a straightforward way the arguments of these lemmas, using the fact that $\triangle$ is even a perfect packing of vertex-disjoint triangles.

Given a solution $\triangle$, we define the sets $\triangle_{X_i}$ and $\triangle_O$ as we did in subsection 4.2.2. Furthermore, let $\triangle_P$ be the set of perfect triangles of $\triangle$. Again, we do not claim (at this point) that $\triangle$ does not contain other shapes of triangles. Given any perfect packing $\triangle$ of $T$, we can prove the following properties.

- We have $|\triangle_O| = m$. Indeed, the only way to use $\theta_j$ is to create an outer triangle $(v, \theta_j, d_j)$. This implies that triangles of $\triangle_O$ consume exactly $m$ disjoint vertices in $X$.

- For any $i \in [\![1, n]\!]$, we have $|\triangle_{X_i}| = 11$. First, as $|V(X_i)| = 35$ we have $|X_i| \leq 11$. Second, let $x$ be the number of vertices of $X$ used in $\triangle$ (as $\triangle$ is a

perfect packing we know that $x = |V(X)| = 35n$). The only triangles of $\triangle$ that can use a vertex of $X$ are the perfect, the outer, and the variable-inner, using respectively one, one and three vertices of $X$ per triangle. This implies that $x \leq 3(\sum_{i \in [\![1,n]\!]} |\triangle_{X_i}|) + m + n'$, proving that $|X_i| \geq 11$ for any $i$, yielding the equality.

Let us now consider the tournament $T'$ obtained from $T$ by removing the vertices of $R_1$ and $R_2$. Furthermore, let $\triangle'$ be the set of triangles $\{\delta \in \triangle \colon V(\delta) \subseteq V(T')\}$. We adapt in a straightforward way the notion of variable-inner and outer triangle in $T'$. Observe that the variable-inner and outer triangles of $\triangle$ and $\triangle'$ are the same, and thus are both denoted respectively by $\triangle_{X_i}$ and by $\triangle_{O_i}$. In particular, $\triangle'$ still contains $m$ outer triangle of $T'$.

Now we simply apply proof of Lemma 22.2 on $\triangle'$. More precisely, Lemma 22.2 restructures $\triangle'$ into a solution $\triangle''$ with $\triangle'' = (\triangle' - (\triangle_{X_i} \cup \triangle_{O_i}) \cup (\triangle''_{X_i} \cup \triangle''_{O_i}))$, where $\triangle''_{X_i}$ and $\triangle''_{O_i}$ satisfy properties $(q)$. In particular, as $|\triangle_{X_i}| = |\triangle''_{X_i}| = 11$, $(q_3)$ implies that $|\triangle''_{O_i}| \geq |\triangle_{O_i}|$, and thus that $|\triangle''_{O_i}| = m$. Thus, $\triangle''$ satisfies $\triangle_{X_i} = P_i$ or $\triangle_{X_i} = \overline{P_i}$ for any $i$, and has $m$ outer triangles. We can now define as in Lemma 22.3 from $\triangle''$ an assignment satisfying the $m$ clauses.

□

## 4.3 Kernelization Results

In all this section we consider the decision problem DTT parameterized by the size of the solution. Thus, an instance is a pair $(T, k)$ for a natural integer $k$, and we say that $(T, k)$ is a "yes"-instance if and only if there exists a vertex-disjoint packing of size $k$ in $T$.

### 4.3.1 A Linear Kernel with the Minimum Feedback Arc Set as Parameter for DTT

As mentioned before, the problem DTT can be seen as a special case of 3-Set Packing. Indeed, given an instance $(T, k)$ of DTT, we create a new instance of 3-Set Packing by creating a subset for each triangle of $T$. It allows us to extend the algorithms solving 3-Set Packing to DTT. However, it does not necessarily work for kernelization algorithms. Indeed, it is not guarantee that we can transform the kernel of 3-Set Packing into a *feasible* instance of DTT. In other words, the kernel returned may have destroyed the special structure of the instance of 3-Set Packing, where subsets correspond to triangles.

Hopefully, since the quadratic kernel of 3-Set Packing by Abu-Khzam [2] only removes vertices, the result is induced instance of the input, and thus can be interpreted as a subtournament of $T$ with $O(k^2)$ vertices.

As expected, we show in the following theorem that it is also possible to have kernel bounded by the number of backward arcs.

**Theorem 26.** Disjoint Triangle Packing in Tournament *parameterized with the size of the solution admits a polynomial kernel with $O(m)$ vertices, where $m$ is the number of arcs in a given feedback arc set of the input.*

*Proof.* Let $I$ be an instance of the decision problem associated to DTT, and denote by $T$ the tournament in the input, and $k$ the parameter. Recall given a tournament $T$ with feedback arc set $A'$, then computing in polynomial time the topological ordering of $T - A'$ gives the ordering $\sigma(T)$ where $\overleftarrow{A}_\sigma(T)$ is $A'$. The general idea of this kernel is to remove useless vertices which are not endpoints of backward arcs.

We create the undirected bipartite graph $B$, such that $V(B) = V^* \cup \overleftarrow{A}(T)$ where $V^*$ is the set of vertices which are not endpoint of any backward arc of $\overleftarrow{A}(T)$. Then, given $a \in \overleftarrow{A}(T)$ and $v \in V^*$, we have the edge $\{a, v\}$ in $E(B)$ if $(h(a), v, t(a))$ is a triangle in $T$. By Hall's theorem, we can in polynomial time partition the sets $V^*$ and $\overleftarrow{A}(T)$ into $V^* = B_0 \cup B_1 \cup B_2$ and $\overleftarrow{A}(T) = A_1 \cup A_2$ such that $N(A_2) \setminus B_1 = B_2$, $|B_2| \leq |A_2|$, there is a perfect matching $M'$ between vertices of $A_1$ and $B_1$, and $B_0$ are the remaining vertices, that is $V^* - (B_1 \cup B_2)$.

Since $V^*$, $B_0$, $B_1$ and $B_2$ are vertices of $B$ which also correspond to vertices in $T$, we will not make the distinction between these sets of vertices and their corresponding sets in the tournament. For example, a vertex in $B_0$ will also could be considered as a vertex in $T$. In the same way, we will consider elements of $A_1$ and $A_2$ as vertices when we deal with the bipartite, but as arcs with the tournaments $T$.

We define the tournament $T'$ defined by removing from $T$ the vertices of $B_0$, that is $T \setminus B_0$. We now create the new instance $I'$ by the couple $(T', k)$. We point out that this definition of $T'$ is similar to the final step of the quadratic kernel of 3-Set Packing by Abu-Khzam [2]. Now, it is easy to observe that $|V(T')| = |V(T)| - |V^*| + |B_1| + |B_2| \leq 2m + |A_1| + |A_2| \leq 3m$, implying the desired bound of the number of vertices of the kernel.

It remains to show that $I$ and $I'$ are equivalent. Let us prove that there exists a solution $\triangle$ of $T$ with $|\triangle| \geq k$ if and only if there exists a solution $\triangle'$ of $T'$ with $|\triangle'| \geq k$. Observe that the converse direction is obvious, since $T'$ is a subtournament of $T$. Let us now prove the sufficiency direction. Let $\triangle$ be a solution of $T$ with $|\triangle| \geq k$. Let partition $\triangle$ into $\triangle_0 \cup \triangle_1$, where $\triangle_0 = \{(h(a), v, t(a)) \in \triangle \colon v \in V^*, a \in \overleftarrow{A}(T)\}$, and $\triangle_1 = \triangle \setminus \triangle_0$. Observe that we have $V(\triangle_1) \cap V^* = \emptyset$. Now, let $\triangle_0^1$ and $\triangle_0^2$ be the sets of vertex-disjoint triangles $\{(h(a), v, t(a)) \in \triangle_0 \colon a \in A_1\}$ and $\{(h(a), v, t(a)) \in \triangle_0 \colon a \in A_2\}$, respectively; such sets form a partition of $\triangle_0$.

For any $a \in A_1$, we define $M'(a)$ as the vertex of $B_1$ associated to $a$ by the perfect matching $M'$. In the same way, we associate to any triangle $\delta = (h(a), v, t(a))$ the triangle $M'(\delta)$, where $M'(\delta) = (h(a), M'(a), t(a))$ if $\delta \in \triangle_0^1$, and we set $M'(\delta) = \delta$ if $\delta \in \triangle_1 \cup \triangle_0^2$. Let $M'(\triangle_0^1)$ be the set $\{M'(\delta) \colon \delta \in \triangle_0^1\}$, and let $\triangle'$ be the union $\triangle_1 \cup \triangle_0^2 \cup M'(\triangle_0^1)$.

Let us now check that the packing $\triangle'$ of triangles is vertex-disjoint. Let $M'(\delta)$ be a triangle of $M'(\triangle_0^1)$ such that $M'(\delta) = (h(a), M'(a), t(a))$. Observe that $h(a)$ and $t(a)$ cannot belong to any other triangle $M'(\delta')$ of $\triangle'$, as for any $M'(\delta') \in \triangle'$, the vertices of $M'(\delta')$ which are not endpoint of backward arcs are the same than those in $\delta'$. It remains to prove that $M'(a)$ does not belong to any other triangles in $\triangle'$. For any triangle $M'(\delta')$ in $\triangle_1$, as $V(M'(\delta)) \cap V^* = \emptyset$, then we have $M'(a) \notin V(M'(\delta'))$. Now, if $M'(\delta') = (h(a'), v, t(a'))$ is a triangle of $\triangle_0^2$,

then by definition $a' \in A_2$ and $v \in B_2$ as $N(A_2) \subseteq B_2$. It proves that $v \neq M'(a)$. Finally, for any distinct triangle $(h(a'), M'(a'), t(a'))$ of $M'(S_0^1)$, we know that $M'(a) \neq M'(a')$ as $M'$ is a matching. □

### 4.3.2 No (Generalized) Kernel in $\mathcal{O}(k^{2-\epsilon})$

For sake of clarity, let us rename the special case of Perfect-DTT described in Theorem 25 by Perfect-DTT*.

In this subsection, we provide an weak 2-composition from Perfect-DTT* to Perfect-DTT parameterized by the total number of vertices. Informally, it means that we will construct a big instance of Perfect-DTT by concatenating multiples instances of Perfect-DTT* in such a way that this constructed instance is positive if and only if at least one of the instances of Perfect-DTT* is positive. To do that, we first describe a gadget which will allow us to "pick" one instance among the multiples possible instance of Perfect-DTT*. We call this digraph an *instance selector*.

#### 4.3.2.1 Definition of the instance selector

**Lemma 26.1.** *For any power of two $\gamma$ and positive integer $\omega$ we can construct in polynomial time (in $\gamma$ and $\omega$) a tournament $P_{\omega,\gamma}$ such that:*

- $|V(P_{\omega,\gamma})| \in O(\omega\gamma)$,
- *there exists $\gamma$ subsets $D_i$ called the distinguished set of vertices, where $D_i$ contains $\omega$ vertices, and such that:*
  - *(i) the $D_i$ have pairwise empty intersection,*
  - *(ii) for any $i \in [\![0, \gamma-1]\!]$, there exists a packing $\triangle$ of triangles of $P_{\omega,\gamma}$ such that $V(P_{\omega,\gamma}) - V(\triangle) = D_i$ (using this packing of $P_{\omega,\gamma}$ corresponds to select instance $i$)*
  - *(iii) for any packing $\triangle$ of triangles of $P_{\omega,\gamma}$ with $|V(\triangle)| = |V(P_{\omega,\gamma})| - \omega$, there exists $i \in [\![1, \gamma]\!]$ such that $V(P_{\omega,\gamma}) - V(\triangle) \subseteq D_i$.*

*Proof.* Let us first describe vertices of $P_{\omega,\gamma}$. For any $i \in [\![0, \gamma-1]\!]$, let $D_i$ be the vertices $\{d_i^j \colon j \in [\![1, \omega]\!]\}$, and let $D$ be the union of the distinguished sets, that is $D = \cup_{i \in [\![0,\gamma-1]\!]} D_i$. So, the $(i)$ in the statement of the lemma is trivially true.

Since $\gamma$ is a power of two, let $\gamma'$ be the integer such that $\gamma = 2^{\gamma'}$. For any $\ell \in [\![0, \gamma'-1]\!]$, let $Y^\ell$ be the set of vertices $\{y_k^\ell \colon k \in [\![1, \omega\gamma/2^\ell + 2]\!]\}$ corresponding the vertices of *level* $\ell$. Similarly, let $Y$ be $\cup_{\ell \in [\![0,\gamma'-1]\!]} Y^\ell$. Finally, we add a set $\alpha$ containing one dummy vertex $\alpha^\ell$ for each $\ell \in [\![0, \gamma'-1]\!]$. We can now set the vertices of $P_{\omega,\gamma}$ such that $V(P_{\omega,\gamma}) = D \cup Y \cup \alpha$. Observe that $|V(P_{\omega,\gamma})| = \omega\gamma + \sum_{\ell \in [\![0,\gamma'-1]\!]} (|Y^\ell| + 1) \in O(\omega\gamma)$.

Let us now describe $\sigma(P_{\omega,\gamma})$ and $\overleftarrow{A}(P_{\omega,\gamma})$ recursively. Let $P^0_{\omega,\gamma}$ be the tournament such that:

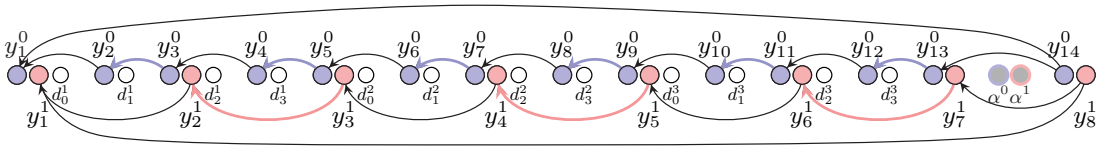


**Figure 4.10** – An example of the instance selector where $\omega = 3$ and $\gamma = 4$ (that is $\gamma' = 2$). The blue vertices correspond to $Y^0$ and the red ones to $Y^1$. The blue backward arcs are the arcs of $Z_P^{0,1}$ and the red ones are the arcs from $Z_P^{1,1}$.

- $\sigma(P^0_{\omega,\gamma}) = \big(\ [y^0_1,\ d^1_0,\ y^0_2,\ d^1_1,\ \ldots,\ y^0_\gamma,\ d^1_{\gamma-1}],\ [y^0_{\gamma+1},\ d^2_0,\ \ldots,\ y^0_{2\gamma},\ d^2_{\gamma-1}],\ \ldots,$ $[y^0_{(\omega-1)\gamma+1},\ d^\omega_0,\ \ldots,\ y^0_{\omega\gamma},\ d^\omega_{\gamma-1}],\ [y^0_{\omega\gamma+1}, \alpha^1, y^0_{\omega\gamma+2}]\ \big)$[1],
- $\overleftarrow{A}(P^0_{\omega,\gamma}) = Z^0_P$, where $Z^0_P = A^0_P \cup A'^0_P$ with $A^0_P = \{(y^0_{k+1}, y^0_k)\colon\ k \in [\![1, |Y^0| - 2]\!]\}$, and $A'^0_P = \{(y^0_{|Y^0|}, y^0_{|Y^0|-1}), (y^0_{|Y^0|}, y^0_1)\}$.

Now, given a tournament $P^\ell_{\omega,\gamma}$ with $\ell \in [\![0, \gamma' - 2]\!]$, we construct the tournament $P^{\ell+1}_{\omega,\gamma}$ such that the vertices of $P^{\ell+1}_{\omega,\gamma}$ are those of $P^\ell_{\omega,\gamma}$ to which are added the set $Y^{\ell+1}$. For $k \in [\![1, |Y^{\ell+1}| - 2]\!]$, we add the vertex $y^{\ell+1}_k$ of $Y^{\ell+1}$ just after the vertex $y^\ell_{2k-1}$ in the ordering of $P^{\ell+1}_{\omega,\gamma}$, and for $i \in [\![0, 1]\!]$ we add vertex $y^{\ell+1}_{|Y^{\ell+1}|-i}$ just after $y^\ell_{|Y^\ell|-i}$. In other words, we add a vertex after every other vertex in the previous level. Then, we add the vertex $\alpha^{\ell+1}$ just after the vertex $\alpha^\ell$.

The backward arcs of $P^{\ell+1}_{\omega,\gamma}$ are defined by: $\overleftarrow{A}(P^{\ell+1}_{\omega,\gamma}) = \overleftarrow{A}(P^\ell_{\omega,\gamma}) \cup Z^{\ell+1}_P$, where $Z^{\ell+1}_P = A^{\ell+1}_P \cup A'^{\ell+1}_P$ are called the *arcs of level* $\ell + 1$, and such that $A^{\ell+1}_P = \{(y^{\ell+1}_{k+1}, y^{\ell+1}_k)\colon\ k \in [\![1, |Y^{\ell+1}|-2]\!]\}$ and $A'^{\ell+1}_P = \{(y^{\ell+1}_{|Y^{\ell+1}|}, y^{\ell+1}_{|Y^{\ell+1}|-1}), (y^{\ell+1}_{|Y^{\ell+1}|}, y^{\ell+1}_1)\}$.

We can now define our gadget tournament $P_{\omega,\gamma}$ as the tournament corresponding to $P^{\gamma'-1}_{\omega,\gamma}$. We refer the reader to Figure 4.10 where an example of the gadget is depicted, where $\omega = 3$ and $\gamma = 4$.

In all the following, given $i \in [\![0, \gamma - 1]\!]$, we call *the last $x$ bits* (respectively the $x^{th}$ bit) of $i$ its $x$ right most (respectively the $x^{th}$, starting from the right) bit(s) in the binary representation of $i$; notice that we start the counting starting with the $0^{th}$ bit, which is the right most one in the binary representation. Notice that the vertices of $D$ are not endpoints of any backward arcs of $P_{\omega,\gamma}$. Furthermore, for any $\ell \in [\![0, \gamma' - 1]\!]$, the endpoint of the arcs of level $\ell$ are exactly $Y^\ell$ (in other words, we have $V(Z^\ell_P) = Y^\ell$), and the arcs of $Z^\ell_P$ induce an undirected cycle on $Y^\ell$ of even size.

Let us now state the following observations:

($r_1$): For any $a \in A^\ell_P$ with $\ell \geq 1$, the span of $a$ contains $2^\ell$ consecutive vertices of $D$. More precisely $\mathrm{s}(a) \cap D = \{d^j_i, \ldots, d^j_{i+2^\ell-1}\}$ for $j \in [\![1, \omega]\!]$ and $i$ such that its $\ell - 1$ last bits are equal to 0,

($r_2$): There is a unique partition $Z^{\ell,0}_P \cup Z^{\ell,1}_P$ of $Z^\ell_P$ such that:

- $|Z^{\ell,0}_P|$ and $|Z^{\ell,1}_P|$ are both equal to the the size of a maximum matching of backward arcs in $P_{\omega,\gamma}[Y^\ell]$,

[1]In this ordering, the square brackets have no other meaning than to help the reader to better understand the structure of the construction.

- each $Z_P^{\ell,q}$ is a matching (for any $a$ and $a'$ in $Z_P^{\ell,q}$, $V(a) \cap V(a') = \emptyset$),
- for each $q \in [\![0,1]\!]$, $\bigcup_{a \in (Z_P^{\ell,q} - A_P'^{\ell})} \mathrm{s}(a) \cap D$ is the set of all vertices $d_i^j$ of $D$ whose $\ell^{th}$ bit of $i$ is $q$.

Now let us first prove that for any $i \in [\![0, \gamma - 1]\!]$, there exists a packing of vertex-disjoint triangles $\triangle_i$ of $P_{\omega,\gamma}$ such that $V(P_{\omega,\gamma}) - V(\triangle) = D_i$. In the following, let $(b_{\gamma'-1} \dots b_0)$ be the binary representation of $i$ and for any $\ell \in [\![0, \gamma' - 1]\!]$, let $\mu(\ell)$ be the size of a maximum matching of backward arcs in $P_{\omega,\gamma}[Y^\ell]$.

We recursively define the vertex-disjoint packing of triangles $\triangle_i$ as the union of partial packings $\triangle_i^\ell$ with $\ell \in [\![0, \gamma' - 1]\!]$, and constructed by maintaining the invariant the following invariant: for any $\ell$, the remaining vertices of $D$ when we have defined $\cup_{z \in [\![0,\ell]\!]} \triangle_i^z$ are the vertices of $D$ with their last bits equal to $(b_{\ell-1}, \dots, b_0)$. Given a backward arc $a$ of $(Z_P^{\ell,1-b_\ell} \setminus A_P'^{\ell})$, notice by $(r_1)$ and our invariant applied on the previous sets $\triangle_{\ell'}$ with $\ell' < \ell$, there remains exactly one vertex in $\mathrm{s}(a)$ denoted by $x_a$. Therefore, we can define $\triangle_i^\ell$ as the set of the $\mu(\ell) - 1$ triangles given by $\{(h(a), x_a, t(a)) \colon a \in (Z_P^{\ell,1-b_\ell} \setminus A_P'^{\ell})\}$. By observation $(r_2)$, these $\mu(\ell) - 1$ triangles consume all the remaining vertices of $D$ whose $\ell^{th}$ bit is $1 - b_\ell$. We also add a last triangle in $\triangle_\ell^i$ using an arc in $A_P'^{\ell}$; namely, we add to the partial packing $\triangle_\ell^i$ the triangle $(y^\ell_{|Y^0|}, \alpha^\ell, y^\ell_{|Y^0|-1})$ if the bit $b_\ell$ is set to 1, and we add $(y_0^\ell, \alpha^\ell, y^0_{|Y^0|})$, otherwise. Thus, by our invariant, the remaining vertices of $D$ after defining $\triangle_i$ are exactly $D_i$. See figures 4.11, 4.12 and 4.13 for an example of this process. As $\triangle_i$ also uses all the vertices of $\alpha$ and $Y$, we have $V(P_{\omega,\gamma}) - V(\triangle_i) = D_i$. Notice that this definition of $\triangle_i$ shows that $|V(P_{\omega,\gamma})| - \omega = |V(\triangle_i)| = 3 \sum_{\ell \in [\![0,\gamma'-1]\!]} \mu(\ell)$.

We now prove that for any packing $\triangle$ of $P_{\omega,\gamma}$ with $|V(\triangle)| = |V(P_{\omega,\gamma})| - \omega = 3 \sum_{\ell \in [\![0,\gamma'-1]\!]} \mu(\ell)$, there exists $i \in [\![1, \gamma]\!]$ such that $V(P_{\omega,\gamma}) \setminus V(\triangle) \subseteq D_i$. Let $\delta_1, \dots, \delta_\mu$ be the triangles of $\triangle$. For any $\delta_k$ of $\triangle$, we associate one backward arc $a_k$ of $\delta_k$ (if there are two backward arcs in the triangle, we pick one of them arbitrarily). Let $Z$ be the set $\{a_k \colon k \in [\![1, |\triangle|]\!]\}$ and for every $\ell \in [\![0, \gamma' - 1]\!]$, let $Z^\ell$ be the set of the backward arcs which are between two vertices of level $\ell$, that is $\{a_k \in Z \colon V(a_k) \subset Y^\ell\}$. Notice that by construction, the these sets form a partition of $Z$. For any $\ell \in [\![1, \gamma']\!]$, we have $|Z^\ell| \leq \mu(\ell)$ since two arcs of $Z^\ell$ correspond to two different triangles of $\triangle$, implying that $Z^\ell$ is a matching. Furthermore, as $|\triangle| = |Z| = \sum_{\ell \in [\![0,\gamma'-1]\!]} |Z^\ell| = \sum_{\ell \in [\![0,\gamma'-1]\!]} \mu(\ell)$, we get the equality $|Z^\ell| = \mu(\ell)$ for any $\ell \in [\![0, \gamma' - 1]\!]$. This implies that for each $Z^\ell$ there exists $q \in [\![0, 1]\!]$ such that $Z^\ell = Z_P^{\ell,q}$, implying also that $V(Z^\ell) = Y^\ell$, and $V(Z) = Y$.

Let $A^\ell$ be the set $(Z^\ell - A_P'^{\ell})$ and $\triangle^\ell$ the set of triangles of $\triangle$ with an arc in $A^\ell$, that is $\{\delta_k \in \triangle \colon a_k \in A^\ell\}$. We can now prove by induction that all the set $R_\ell$ of remaining vertices $D - V(\cup_{x \in [\![0,\ell]\!]} \triangle^\ell)$ have the same last $\ell$ bits. Notice that since all vertices of $Y$ are already used, and as triangles of $\triangle^\ell$ cannot use a dummy vertex in $\alpha$, all triangles of $\triangle^\ell$ must be of the from $(h(a_k), v, t(a_k))$ with $v \in D$. Furthermore, as $A^\ell = Z_P^{\ell,q} - A_P'^{\ell}$, by $(r_2)$ we know that $\cup_{a \in A^\ell} \mathrm{s}(a) \cap D$ contains all the remaining vertices of $D$, and thus of $R_{\ell-1}$, whose $\ell^{th}$ bit is $q$.

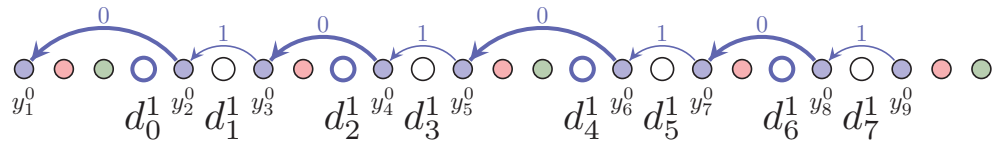



**Figure 4.11** – A part of an instance selector where $\omega = 2$, $\gamma = 8$, and $i = 5 = 101_2$. Here we only depicted in blue the arcs of $Z_P^{0,0}$ (labeled 0) and $Z_P^{0,1}$ (labeled 1). We construct the vertex-disjoint triangles of $\triangle_0$ by considering the arcs of $Z_P^{0,0}$ since the $0^{th}$ bit of $i$ is 1; these are the thick arcs. We can notice that there is exactly one vertex $d_1^j$ in the span of any arcs in $Z_P^{0,0}$ (in blue with center in white), so we can create a triangle with this vertex.

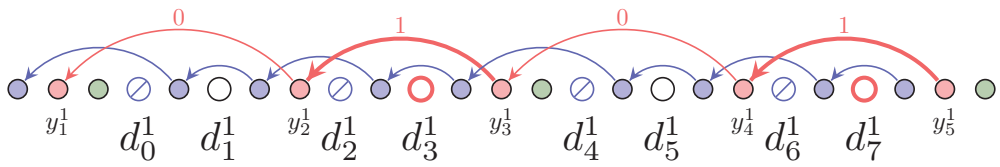



**Figure 4.12** – We are in the same case than in Figure 4.11. Here we also depicted in red the arcs of $Z_P^{1,0}$ (labeled 0) and $Z_P^{1,1}$ (labeled 1). We construct the vertex-disjoint triangles of $\triangle_1$ by considering the arcs of $Z_P^{1,1}$ since the $1^{st}$ bit of $i$ is 0; these are the thick arcs. We can notice that since some vertex are already taken in $\triangle_0$ (the crossed-out blue vertices), there is exactly one vertex $d_1^j$ in the span of any arcs in $Z_P^{1,1}$ (in red with center in white), so we can create a triangle with this vertex.

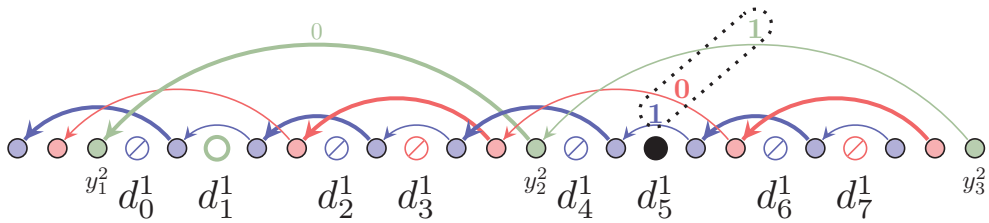



**Figure 4.13** – We are in the same case than in Figure 4.12. Here we also depicted in green the arcs of $Z_P^{2,0}$ (labeled 0) and $Z_P^{2,1}$ (labeled 1). We construct the vertex-disjoint triangles of $\triangle_2$ by considering the arcs of $Z_P^{2,0}$ since the $2^{nd}$ bit of $i$ is 1; these are the thick arcs. We can notice that since some vertex are already taken in $\triangle_0$ (the crossed-out blue vertices) or in $\triangle_1$ (the crossed-out red vertices), there is exactly one vertex $d_1^j$ in the span of any arcs in $Z_P^{2,0}$ (in green with center in white), so we can create a triangle with this vertex. Notice that the only remaining vertex in $D_1$ is exactly $d_1^i$. Observe also that given any $i' \in [\![0, \gamma - 1]\!]$, the labels of the arcs the vertex $d_{i'}^j$ is in give the binary representation of $i'$ (depicted with the dotted shape).

Moreover, by $(r_1)$, we know that for any $a \in A^\ell$ we have $|R_{\ell-1} \cap \mathrm{s}(a)| \leq 1$ because as $a \in A^\ell_P$ we know exactly the structure of $\mathrm{s}(a) \cap D$, and if there remain two vertices in $\mathrm{s}(a) \cap D$ then their last $\ell - 1$ last bits would be different. Thus, as triangles of $\triangle^\ell$ remove one vertex in the span of each arc $a$ in $A^\ell$, they remove all vertices of $R_{\ell-1}$ whose $\ell^{th}$ bit is $q$, implying the desired result.

□

#### 4.3.2.2 Definition of the reduction

Using the instance selector we just defined, we can now describe the construction of the weak composition from Perfect-DTT* to Perfect-DTT.

Given a family $F$ of $t$ instances $\{\mathcal{I}^i \colon i \in [\![1, t]\!]\}$ of Perfect-DTT*, where the instance $\mathcal{I}^i$ correspond to know if there is a vertex-disjoint triangle perfect packing in the tournament $T^i = X^i \oplus K^i$. Furthermore, for any $i \in [\![1, t]\!]$, we have $|V(X^i)| = n$ and $|V(K^i)| = m$. If necessary, we can also copy some of the $t$ instances until that $t$ is a square number and $\sqrt{t}$ is a also a power of two. In the following, let $g$ be $\sqrt{t}$. We rename our family of instances $F$ into $F = \{\mathcal{I}^{(p,q)} \colon p, q \in [\![1, g]\!]\}$, and where $\mathcal{I}^{(p,q)}$ asks if there is a perfect vertex-disjoint packing in the tournament $T^{(p,q)}$ given by the concatenation $X^p \oplus K^q$. Remember that according to Theorem 25, all the $X^p$ are equals, as well as all the $K^q$.

Let us point out that the idea of using a problem on "bipartite" instances to allow encoding $t$ instances on a "meta" bipartite graph $G$ with parts $(A, B)$ (where $A = \{A_i \colon i \in [\![1, \sqrt{t}]\!]\}$, and $B = \{B_j \colon j \in [\![1, \sqrt{t}]\!]\}$) such that each instance $(i, j)$ is encoded in the graph induced by $G[A_i \cup B_j]$ comes from the article of Dell and Marx [54].

We now construct our final tournament $T$. To do so, we start by constructing the first half $T_G$ of our final tournament. It requires to define the following subparts:

- for $p \in [\![1, g]\!]$, $X^p$ is the variable tournament defined in section 4.2, and let $X$ be the concatenation $X^1 \oplus \cdots \oplus X^g$,
- $M_G$ is a tournament of order $(g-1)n$, and such that $\overleftarrow{A}(\tilde{M}_G) = \emptyset$,
- for $p \in [\![1, g]\!]$, $\tilde{X}^p$ is a tournament of order $n$ such that $\overleftarrow{A}(\tilde{X}^p) = \emptyset$, and let $\tilde{X}$ be the concatenation $\tilde{X}^1 \oplus \cdots \oplus \tilde{X}^g$,
- $\tilde{M}_G$ is a tournament of order $n$, and such that $\overleftarrow{A}(\tilde{M}_G) = \emptyset$,
- $P_{n,g}$ is the instance selector defined in Lemma 26.1.

Thus, the tournament $G$ is defined by the concatenation $X \oplus M_G \oplus \tilde{X} \oplus \tilde{M}_G \oplus P_{n,g}$. Furthermore, we add to $\overleftarrow{A}(G)$ all the possible $n^2$ backward arcs going from $\tilde{X}^p$ to $X^p$ and, for every distinguished set $D_p$ of the instance selector $P_{n,g}$ (see definition in Lemma 26.1), we add all the possible $n^2$ backward arcs from $D_p$ to $\tilde{X}^p$. We refer the reader to Figure 4.14 where the upper part represents the tournament $G$.

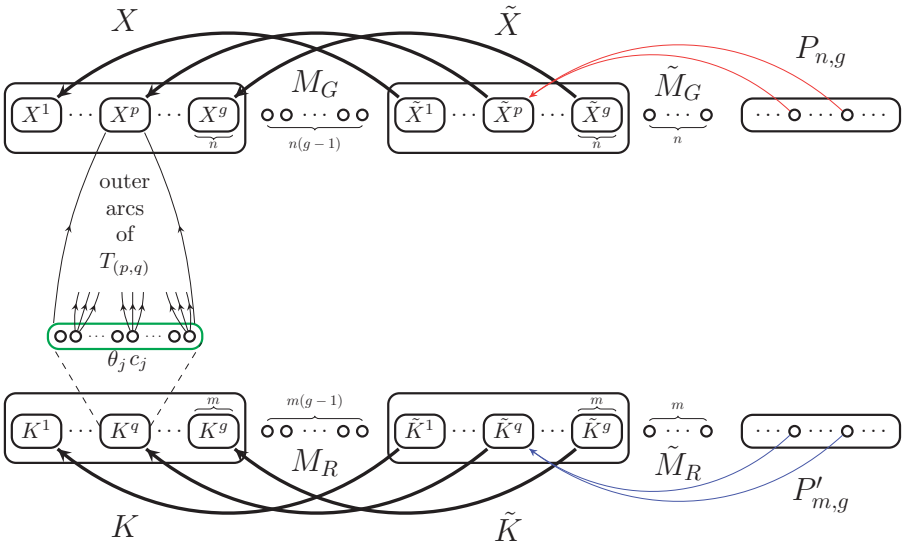



**Figure 4.14** – An example of the weak composition. In this figure, the ordering of the vertices is read from top to bottom, then from left to right, ignoring the green set. We only depict backward arcs. The bold arcs represent the $n^2$ (or $m^2$ possible backward arcs joining the two tournaments. The colored arcs from $P_{n,g}$ (resp. $P'_{m,g}$) means that the tails are the vertices of the prescribed distinguished set. The upper part corresponds to $G$ while the lower one is $R$. The green set corresponds to a "zoom" in the clause tournament $K^q$.

Now, in a symmetric way, let us defined the second half $R$ of our final tournament. To do so, we will need the following subparts:

- for $q \in [\![1, g]\!]$, $K^q$ is the clause tournament defined in section 4.2, and let $K$ be the concatenation $K^1 \oplus \cdots \oplus K^g$,
- $M_R$ is a tournament of order $(g-1)m$, and such that $\overleftarrow{A}(\tilde{M}_R) = \emptyset$,
- for $q \in [\![1, q]\!]$, $\tilde{K}^q$ is a tournament of order $m$ such that $\overleftarrow{A}(\tilde{K}^q) = \emptyset$, and let $\tilde{K}$ be the concatenation $\tilde{K}^1 \oplus \cdots \oplus \tilde{K}^g$,
- $\tilde{M}_R$ is a tournament of order $m$, and such that $\overleftarrow{A}(\tilde{M}_R) = \emptyset$,
- $P'_{m,g}$ is the instance selector defined in Lemma 26.1.

Thus, the tournament $R$ is defined by the concatenation $K \oplus M_R \oplus \tilde{K} \oplus \tilde{M}_R \oplus P'_{m,g}$. Furthermore, we add to $\overleftarrow{A}(R)$ all the possible $m^2$ backward arcs going from $\tilde{K}^q$ to $K^q$ and, for every distinguished set $D'_q$ of the instance selector $P'_{m,g}$ (see definition in Lemma 26.1), we add all the possible $m^2$ backward arcs from $D'_q$ to $\tilde{K}^q$. We refer the reader to Figure 4.14 where the lower part represents the tournament $R$.

Finally, we define can define our final tournament $T$ as the concatenation $G \oplus R$. Let us now add some backward arcs from $R$ to $G$ in order to encode the instances of $F$. For any $p$ and $q$ in $[\![1, g]\!]$, we add the backward arcs from $K^q$ to

$X^p$ such that $T[V(X^p) \cup V(K^q)]$ corresponds to the tournament $T_{(p,q)}$ defined in Theorem 25. Notice that this is possible as for any fixed $p$, all the $T_{(p,q)}$ for some $q \in [\![1, g]\!]$ have the same left part $X^p$, and the same goes for any fixed right part. We refer the reader to Figure 4.14 which represents the different parts of $T$.

#### 4.3.2.3 Restructuring lemmas

As we did at multiple times so far, we start by studying the different types of triangles a set of triangles could have. Formally, given a packing of vertex-disjoint triangles $\triangle$, we define:

- $\triangle_P = \{\delta \in \triangle\colon\ V(\delta) \subseteq V(P_{n,g})\}$,
- $\triangle_{P'} = \{\delta \in \triangle\colon\ V(\delta) \subseteq V(P'_{m,g})\}$,
- $\triangle_{\tilde{M}_G} = \{(u, v, w) \in \triangle\colon\ u \in V(\tilde{X}),\ v \in V(\tilde{M}_G),\ w \in V(P_{n,g})\}$,
- $\triangle_{M_G} = \{(u, v, w) \in \triangle\colon\ u \in V(X),\ v \in V(M_G),\ w \in V(\tilde{X})\}$,
- $\triangle_{\tilde{M}_R} = \{(u, v, w) \in \triangle\colon\ u \in V(\tilde{K}),\ v \in V(\tilde{M}_R),\ w \in V(P'_{m,g})\}$,
- $\triangle_{M_R} = \{(u, v, w) \in \triangle\colon\ u \in V(K),\ v \in V(M_R),\ w \in V(\tilde{K})\}$.

We also define the packings $\triangle_G$, $\triangle_R$ and $\triangle_{GR}$ by $\{\delta \in \triangle\colon\ V(\delta) \subseteq V(G)\}$, $\{\delta \in \triangle\colon\ V(\delta) \subseteq V(R)\}$, and $\{(u, v, w) \in \triangle\colon\ V(\delta) \cap V(G) \neq \emptyset$ and $V(\delta) \cap V(R) \neq \emptyset\}$, respectively. Observe that $\triangle_G$ and $\triangle_R$ contain some of the triangles of the previously defined sets. On the contrary, none of the triangles of the previous defined sets are in $\triangle_{GR}$. Finally, notice that $\triangle_G$, $\triangle_R$ and $\triangle_{GR}$ form a partition of $\triangle$.

**Claim 26.1.** *If there exists a perfect packing of vertex-disjoint triangles $\triangle$ of the tournament $T$, then $|\triangle_{\tilde{M}_R}| = m$ and $|\triangle_{M_R}| = (g-1)m$. This implies that $V(\triangle_{\tilde{M}_R} \cup \triangle_{M_R}) \cap V(\tilde{K}) = V(\tilde{K})$, meaning that the vertices of $\tilde{K}$ are entirely used by $\triangle_{\tilde{M}_R} \cup \triangle_{M_R}$.*

*Proof.* We have $|\triangle_{\tilde{M}_R}| \leq m$ since $\tilde{M}_R$ has $m$ vertices in total. We obtain the equality since the vertices of $\tilde{M}_R$ only lie in the span of backward arcs from $P'_{m,g}$ to $\tilde{K}$, and they are not the head or the tail of any backward arc in $T$. Thus, the only way to use vertices of $\tilde{M}_R$ is to create triangles in $\triangle_{\tilde{M}_R}$, implying $|\triangle_{\tilde{M}_R}| \geq m$.

Using the same kind of arguments we also get $|\triangle_{M_R}| = (g-1)m$. Finally, as $|V(\tilde{K})| = gm$ we get the last part of the claim. □

In the following, let $\triangle^1_{P'}$ be the triangles of $\triangle$ with at least one vertex in $P'_{m,g}$.

**Claim 26.2.** *If there exists a perfect packing of vertex-disjoint triangles $\triangle$ of the tournament $T$, then there exists $q_0 \in [\![1, g]\!]$ such that $V(\tilde{K}) \cap V(\triangle_{\tilde{M}_R}) = V(\tilde{K}^{q_0})$.*

*Proof.* According to Claim 26.1, vertices of $\tilde{K}$ are entirely used by $\triangle_{\tilde{M}_R} \cup \triangle_{M_R}$, the only way to consume vertices of $P'_{m,g}$ is by creating local-triangles in $P'_{m,g}$, or triangles in $\triangle_{\tilde{M}_R}$. In particular, we cannot have a triangle $(u, v, w)$ with $u$ and $v$ in $V(\tilde{K})$ and $w \in V(P'_{m,g})$, nor with $u \in V(\tilde{K})$ and $\{v, w\} \subseteq V(P'_{m,g})$. More formally, we get the partition of $\triangle^1_{P'}$ into $\triangle_{P'} \cup \triangle_{\tilde{M}_R}$. As $\triangle$ is a perfect packing, it uses in particular all the vertices of the instance selector $P'_{m,g}$, and we get $|V(\triangle^1_{P'})| = |V(P'_{m,g})|$. By Claim 26.1, this implies $|V(\triangle_{P'})| = |V(P'_{m,g})| - m$.

Now, using Lemma 26.1, there exists $q_0 \in [\![1, g]\!]$ such that $V(P'_{m,g}) \setminus V(\triangle_{P'}) \subseteq D'_{q_0}$, where $D'_{q_0}$ is a distinguished set of $P'_{m,g}$. As $V(P'_{m,g}) \setminus V(\triangle_{P'})$ are the only remaining vertices that can be used by triangles of $\triangle_{\tilde{M}_R}$, we get that the $m$ triangles of $\triangle_{\tilde{M}_R}$ are of the form $(u, v, w)$ with $u \in V(\tilde{K}^{q_0})$, $v \in V(\tilde{M}_R)$, and $w \in (V(P'_{m,g}) \setminus V(\triangle_{P'}))$, concluding the proof. □

**Claim 26.3.** *If there exists a perfect packing of vertex-disjoint triangles $\triangle$ of the tournament $T$, then there exists $q_0 \in [\![1, g]\!]$ such that $V(\triangle^1_{P'} \cup \triangle_{\tilde{M}_R} \cup \triangle_{M_R}) = V(R) \setminus K^{q_0}$.*

*Proof.* By Claim 26.1 we know that $|\triangle_{M_R}| = (g-1)m$. Now, using Claim 26.2, there exists $q_0 \in [\![1, g]\!]$ such that $V(\tilde{K}) \cap V(\triangle_{\tilde{M}_R}) = V(\tilde{K}^{q_0})$. We get that the $(g-1)m$ triangles of $\triangle_{M_R}$ are of the form $(u, v, w)$ with $u \in V(K) \setminus V(K^{q_0})$, $v \in V(M_R)$, and $w \in V(\tilde{K}) \setminus V(\tilde{K}^{q_0})$. □

We would like to have similar claims to the previous ones on the subtournament $G$. However, observe the backward arcs which encode the different instances of Perfect-DTT* forbid us to trivially use the same arguments as we did. The following lemma will solve this problem, since it states that the triangles of $\triangle_{GR}$ do not use any vertex of $M_G$, $\tilde{X}$, $\tilde{M}_T$ and $P_{n,g}$.

**Lemma 26.2.** *If there exists a perfect packing of vertex-disjoint triangles $\triangle$ of the tournament $T$, then $(V(\triangle_{GR}) \cap V(G)) \subseteq V(X)$.*

*Proof.* By Claim 26.3, there exists an index $q_0$ in $[\![1, g]\!]$ such that $V(\triangle^1_{P'} \cup \triangle_{\tilde{M}_R} \cup \triangle_{M_R}) = V(R) - K^{q_0}$. By Theorem 25, we also know that $K^{q_0}$ is obtained by the concatenation of clause gadgets, namely $K^{q_0} = K^{q_0}_1 \oplus \cdots \oplus K^{q_0}_{m'}$ for some $m'$, and where each ordering $\sigma(K_{(q_0,j)})$ is $(\theta^{q_0}_j, d^{q_0}_j)$. One can also notice that since $|V(K^{q_0})| = m$ by definition, we have $m' = \frac{m}{2}$.

Moreover, for any $p \in [\![1, g]\!]$, the last property of Theorem 25 ensures that for any $a \in \overleftarrow{A}(T^{(p,q_0)})$, $V(a) \cap V(K^{q_0}) \neq \emptyset$ implies $a = (v, d_j)$ for $v \in V(X^p)$. So no arc of $\overleftarrow{A}(T^{(p,q_0)})$—and thus no arc of $T$—is entirely included in $K^{q_0}$. This implies that the vertices in triangles of $\triangle$ cannot cover the vertices of $K^{q_0}$ using triangles $\delta$ with $V(\delta) \subseteq V(K^{q_0})$, and thus that all these vertices must be used by triangles of $\triangle_{GR}$. It implies that $V(\triangle_{GR}) \cap V(R) = K^{q_0}$.

The last property of Theorem 25 also implies that none of the $\theta^q_j$ is a the tail of a backward arc with the given ordering. Thus, by induction for any $j$ from $m'$ to 1, we can prove that the only way for triangles of $\triangle_{GR}$ to use the vertex $\theta^q_j$ is to create a triangle $(v, \theta^q_j, d^q_j)$, for some $v \in V(X)$. □

Using this lemma, we immediately adapt the previous structure claims to the subtournament $G$. Therefore, we get the following claims. In the following, let $\triangle_P^1$ be the triangles of $\triangle$ with at least one vertex in $P_{n,g}$.

**Claim 26.4.** *If there exists a perfect packing of vertex-disjoint triangles $\triangle$ of the tournament $T$, then $|\triangle_{\tilde{M}_G}| = n$ and $|\triangle_{M_R}| = (g-1)n$. This implies that $V(\triangle_{\tilde{M}_G} \cup \triangle_{M_G}) \cap V(\tilde{X}) = V(\tilde{X})$, meaning that the vertices of $\tilde{X}$ are entirely used by $\triangle_{\tilde{M}_G} \cup \triangle_{M_G}$.*

**Claim 26.5.** *If there exists a perfect packing of vertex-disjoint triangles $\triangle$ of the tournament $T$, then there exists $p_0 \in [\![1, g]\!]$ such that $V(\tilde{X}) \cap V(\triangle_{\tilde{M}_G}) = V(\tilde{X}^{p_0})$.*

**Claim 26.6.** *If there exists a perfect packing of vertex-disjoint triangles $\triangle$ of the tournament $T$, then there exists $p_0 \in [\![1, g]\!]$ such that $V(\triangle_P^1 \cup \triangle_{\tilde{M}_G} \cup \triangle_{M_G}) = V(G) - X^{p_0}$.*

According Claim 26.3 and Claim 26.6, we can define the vertex-disjoint triangle packing $\triangle^{(p_0,q_0)}$ given by $\triangle - \big((\triangle_P^1 \cup \triangle_{\tilde{M}_G} \cup \triangle_{M_G}) \cup (\triangle_{P'}^1 \cup \triangle_{\tilde{M}_R} \cup \triangle_{M_R})\big)$. Then, the following final claim is straightforward:

**Claim 26.7.** *If there exists a perfect packing of vertex-disjoint triangles $\triangle$ of the tournament $T$, then there exist $p_0, q_0 \in [\![1, g]\!]$ and $\triangle^{(p_0,q_0)} \subseteq \triangle$ such that $V(\triangle^{(p_0,q_0)}) = V(T^{(p_0,q_0)})$ (or, equivalently, such that $\triangle^{(p_0,q_0)}$ is a perfect packing of vertex-disjoint triangles of $T^{(p_0,q_0)}$).*

#### 4.3.2.4 Proof of the weak composition

**Theorem 27.** *For any $\varepsilon > 0$,* Perfect Packing Disjoint-Triangle in Tournament *parameterized by the total number of vertices $N$ does not admit a polynomial kernelization with size bound $O(N^{2-\varepsilon})$, unless* $\mathsf{NP} \subseteq \mathsf{coNP/poly}$.

*Proof.* Given $t$ instances $\{\mathcal{I}^1, \ldots, \mathcal{I}^t\}$ of Perfect-DTT*, we define an instance $T$ of Perfect-DTT as defined in section 4.3. We recall that $g = \sqrt{t}$, and that for any $i \in [\![1, l]\!]$, $|V(X^i)| = n$ and $|V(K^i)| = m$. Furthermore, let $N$ be the order of $T$. We have by construction

$$N = \big(ng + (g-1)n + ng + n + |V(P_{n,g})|\big) + \big(mg + (g-1)m + mg + m + |V(P'_{m,g})|\big)$$

By Lemma 26.1, the instance selector $P_{\omega,\gamma}$ has $O(\omega\gamma)$ vertices, then it implies that $N \in O(g(n+m)) = O(t^{\frac{1}{2+o(1)}} \times \max_{i \in [\![1,t]\!]}(|\mathcal{I}^i|))$.

Let us now verify that there exists $i \in [\![1, t]\!]$ such that $\mathcal{I}^i$ admits a perfect packing of vertex-disjoint triangles if and only if $T$ admits a perfect packing of vertex-disjoint triangles.

First, assume that there exist $p_0, q_0 \in [\![1, g]\!]$ such that $\mathcal{I}^{(p_0,q_0)}$ admits a perfect packing of vertex-disjoint triangles. By Claim 26.7, there is a packing $\triangle_{P'}$ of $P'_{m,g}$ such that $V(\triangle_{P'}) = V(P'_{m,g}) - D'_{q_0}$. Next, we define a set $\triangle_{\tilde{M}_R}$ of $m$ vertex-disjoint triangles of the form $(u, v, w)$ with $u \in \tilde{X}^{q_0}$, $v \in \tilde{M}_R$, and $w \in D'_{q_0}$. Then, we define a set $\triangle_{M_R}$ of $(g-1)m$ vertex-disjoint triangles of the form $(u, v, w)$ with

$u \in X - L_{q_0}$, $v \in M_R$, and $w \in \tilde{X} - \tilde{X}^{q_0}$. In the same way we define $\triangle_P$, $\triangle_{\tilde{M}_G}$ and $\triangle_{M_G}$. Observe that $V(T) - ((\triangle_{P'} \cup \triangle_{\tilde{M}_R} \cup \triangle_{M_R}) \cup (\triangle_P \cup \triangle_{\tilde{M}_G} \cup \triangle_{M_G})) = X^{p_0} \cup Kq_0$. Therefore, we can complete our packing into a perfect packing of $T$ as $\mathcal{I}^{(p_0,q_0)}$ admits a perfect packing.

Conversely, if there exists a perfect packing $\triangle$ of $T$, then by Claim 26.7 there exists $p_0$ and $q_0$ in $[\![1, g]\!]$, and $\triangle^{(p_0,q_0)} \subseteq \triangle$ such that $V(\triangle^{(p_0,q_0)}) = V(T^{(p_0,q_0)})$, implying that $\mathcal{I}^{(p_0,q_0)}$ admits a perfect packing of vertex-disjoint triangles. □

The previous theorem directly implies a lower-bound on the kernel size for DTT.

**Theorem 28.** *For any $\varepsilon > 0$,* Disjoint Triangle Packing in Tournament *(parameterized by the size $k$ of the solution) does not admit a polynomial kernel with size $O(k^{2-\varepsilon})$, unless* NP $\subseteq$ coNP.

## 4.4 Concluding Remarks

In this chapter, we proved the NP-hardness of Disjoint Triangle Packing in Tournament. Surprisingly, the problem is still hard even if the tournament is sparse. We also proved that there is no $O(k^{2-\varepsilon})$ in total bit size kernel. Recently, Le et al [108] gave a kernel of $O(k^{1.5})$ vertices. Such kernel leads to a $O(k^3)$ total bit size kernel. In the light of these results, the following question could be of great interest:

**Question 1.** *Does* Disjoint Triangle Packing in Tournament *admit a kernel with $O(k)$ vertices?*

In Chapter 6, we answer this question positively for the sparse tournaments. But the generalization in general case might be a difficult problem.

From the approximation point of view, we proved that Disjoint Triangle Packing in Tournament is APX-hard too. However, it would be interesting to find if we can do better than the $(4/3 + \varepsilon)$-approximation-approximation algorithm by Cygan [47] for 3-Set Packing. In order to make progress on this, we give in Chapter 6 an approximation algorithm for sparse tournament with large minspan.

# Chapter 5

# Arc-disjoint Packing of Cycles in Tournaments

This chapter corresponds to joint work with Stéphane Bessy and Marin Bougeret [22], then merged with related work of Krithika *et al.* [101] in order to be presented to *MFCS 2019* [20]. We focus here on the Arc-Disjoint Cycle Packing restricted to tournaments. More precisely, we give both classical and parameterized complexity results.

## 5.1 Introduction and Preliminaries

### 5.1.1 General Problems and Related Work

In this chapter, we consider the alternate version of the cycle packing in tournaments where the cycles must be *arc-disjoint*. Unlike the vertex-disjoint case of Chapter 4, notice that the maximum number of arc-disjoint triangles may be strictly less to the number of arc-disjoint cycles. Pick for example the tournament depicted in Figure 5.1[1], one can easily notice that there exists at most three arc-disjoint cycles, namely two triangles and one 4-cycle. Let us prove that there is no three arc-disjoint triangles in it. Let $\delta_1$, $\delta_2$ and $\delta_3$ be three arc-disjoint triangles. Since the order of this tournament is 5, two arc-disjoint triangles must share exactly one common vertex and this vertex cannot be in the three triangles. Therefore, we have $|V(\delta_1) \cup V(\delta_2) \cup V(\delta_3)| = |V(\delta_1)| + |V(\delta_2)| + |V(\delta_3)| - |V(\delta_1) \cap V(\delta_2)| - |V(\delta_2) \cap V(\delta_3)| - |V(\delta_3) \cap V(\delta_1)| + |V(\delta_1) \cap V(\delta_2) \cap V(\delta_3)|$ leading to $|V(\delta_1) \cup V(\delta_2) \cup V(\delta_3)| = 6 > 5$, a contradiction.

This observation hints that packing arc-disjoint cycles could be significantly harder than packing vertex-disjoint cycles. It also hints that packing arc-disjoint cycles and arc-disjoint triangles in tournaments could be problems of different complexities. This is the starting point of this chapter.

Formally, we can define the Maximum Arc-Disjoint Cycle Packing in Tournament problem (or simply Max ACT) as follows:

[1] One can notice that this example is a non-linear representation of the tournament $F_2$ described in Figure 4.1.

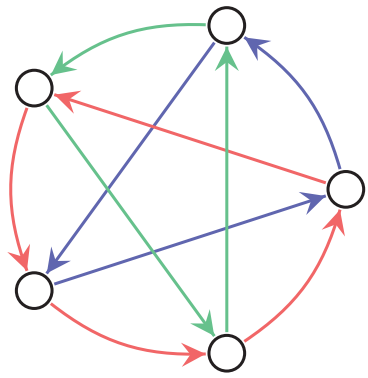


**Figure 5.1** – The tournament $F_2$ with its packing of three arc-disjoint cycles in green, blue and red. Notice that the red cycle is not a triangle.

Maximum Arc-Disjoint Cycle Packing in Tournament (Max ACT)
**Input:** A tournament $T$.
**Result:** A collection $\mathcal{C}$ of arc-disjoint cycles in $T$.
**Optimization:** Maximize $|\mathcal{C}|$.

And, similarly, the triangle-restricted version of the problem, namely Maximum Arc-Disjoint Triangle Packing in Tournament (or simply Max ATT) :

Maximum Arc-Disjoint Triangle Packing in Tournament (Max ATT)
**Input:** A tournament $T$.
**Result:** A collection $\triangle$ of arc-disjoint triangles in $T$.
**Optimization:** Maximize $|\triangle|$.

We also consider the decision problems associated to these problems, that are Arc-Disjoint Cycle Packing in Tournament (or simply ACT) (resp. Arc-Disjoint Triangle Packing in Tournament, or ACT for short), where we want to know whether the tournament contains at least $k$ arc-disjoint cycles (resp. triangles).

From a structural point of view, the problem of partitioning the arc set of a directed graph into a collection of triangles has been studied for regular tournaments by Yuster [155], almost regular tournaments by Akaria and Yuster [4] and complete digraphs by Gadner [69].

Finally concerning FPT-algorithms, Krithika *et al.* [101] recently showed that ACT can be solved in $O^*(2^{O(k \log k)})$ running-time reusing the linear kernelization of Bessy *et al.* [23]. They also provide a linear kernel for ACT.

### 5.1.2 Our Contributions and Organization of the Chapter

In this chapter, our objective is to study the classical complexity of Max ATT and Max ACT as well as the parameterized complexity of ATT.

First, we prove in section 5.2 that Max ATT and Max ACT are NP-hard. As a consequence, we also show that their respective parameterized versions cannot admit algorithms with running time in $O^*(2^{o(\sqrt{k})})$, unless **ETH** fails. Moreover, we prove that deciding if a tournament has an arc-disjoint cycle packing (or arc-disjoint triangle packing) and a feedback arc set of the same size is NP-complete.

From the parameterized point of view, we provide in section 5.3 an FPT-algorithm in $O^*(2^k)$ running time for ATT using color coding , and we show a kernel for this problem with $O(k)$ vertices.

### 5.1.3 Spectific Notations

Recall that we mostly consider a tournament $T$ of order $n$ with its linear representation $(\sigma(T), \overleftarrow{A}(T))$, where $\sigma(T) = (v_1, \ldots, v_n)$. Furthermore, if $T_1$ and $T_2$ are two tournaments of order $n$ and $n'$, and defined by their respective linear representation $((u_1, \ldots, u_n), \overleftarrow{A}(T_1))$ and $((v_1, \ldots, v_{n'}), \overleftarrow{A}(T_2))$ recall that the concatenation of $T_1$ and $T_2$, written $T_1 \oplus T_2$, is the new tournament $T$ given by the linear representation $(\sigma(T), \overleftarrow{A}(T))$ with $\sigma(T) = (u_1, \ldots, u_n, v_1, \ldots, v_{n'})$ and $\overleftarrow{A}(T) = \overleftarrow{A}(T_1) \cup \overleftarrow{A}(T_2)$.

Concerning SAT formulas, given a conjunction $F = c_1 \wedge \ldots \wedge c_p$, a clause $c_j$ of $F$ and a variable $x_i$ of $c_j$, we define the occurrence number of $x_i$ in $c_j$, denoted by $o(x_i, c_j)$, by the value of $|\{j' \in [\![1, j]\!] \colon x_i \text{ is a variable of } c_{j'}\}|$.

## 5.2 NP-hardness of the Different Problems

### 5.2.1 Reduction from Max 3-SAT(3)

In this section, we define the reduction $g$ from Max 3-SAT(3) to Max ATT. Recall that an instance $\mathcal{F}$ of Max 3-SAT(3) is defined by a set of $n$ variables $\{x_1, \ldots, x_n\}$ and a set of $m$ clauses $\{c_1, \ldots, c_m\}$, where each variable appears in at most three clauses, and each clause has *at most* three variables.

As we did in previous chapter, notice that we can consider that each variable occurs at most twice positively and exactly once negatively. In the following, we also suppose that $n \equiv 3 \pmod 6$ as well as $m + 1 \equiv 3 \pmod 6$. Indeed, if it is not the case, if we add a new variable $x'$ and the trivial clause $\overline{x'} \vee x'$. Thus, we can reach $n \equiv 3 \pmod 6$ by adding up to five variables. Similarly, as long as $m + 1 \not\equiv 3 \pmod 6$, we can add six new variables $x'_1, \ldots, x'_6$ (preserving $n \equiv 3 \pmod 6$) and add the six trivial clauses $\overline{x'_i} \vee x'_i$ for $i \in [\![1, 6]\!]$ and the clause $x'_1 \vee x'_2$, that is seven clauses in total. Once again, by repeating this operation up to five times, we can assume that $n \equiv 3 \pmod 6$ and $m + 1 \equiv 3 \pmod 6$. In total, we added at most 35 new variables and 40 new clauses.

Informally, the idea of the reduction is a classical variable/clause gadget construction, where the arcs joining the clause to the variable gadget encode the instance of Max 3-SAT(3). However, the main difficulty is to obtain a tournament *and* to be sure that the arcs joining two variable gadgets (resp. clause gadgets)—the arcs that do not have any "meaning" in the instance $\mathcal{F}$—do not "pollute" the packing of arc-disjoint cycles. The idea is then to guarantee that

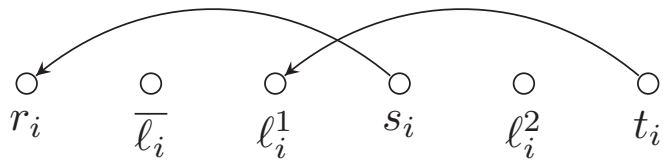



**Figure 5.2** – The variable tournament $X_i$, where we only depict the backward arcs.

we can pack *exactly all* the arcs joining joining two variable gadgets (resp. two clause gadgets) from the beginning, and regardless of the instance $\mathcal{F}$. This is the reason why we need to have $n \equiv 3 \pmod 6$ (resp. $m+1 \equiv 3 \pmod 6$) since the edges of the undirected clique of order $n$ (resp. $m+1$) can be partition into triangles [100] in polynomial time.

Our final tournament $T_{g(\mathcal{F})}$ will be obtained by the concatenation of two subtournaments $X$ and $Q$ we now describe.

**The clause tournament** For each clause of $\mathcal{F}$, we define the *clause gadget* $Q_j$, a tournament of order 3, such that $\sigma_j(Q_j) = (c_j^1, c_j^2, c_j^3)$ and $\overleftarrow{A}(Q_j) = \emptyset$. In addition, we create an extra tournament, *the dummy triangle* $Q_{m+1}$, such that $\sigma_{m+1}(Q_{m+1}) = (d^1, d^2, d^3)$ and $\overleftarrow{A}(Q_{m+1}) = \{(d^3, d^1)\}$. We now define the *clause tournament* $Q$ as the concatenation of all the $Q_j$, that is $Q = Q_1 \oplus \cdots \oplus Q_m \oplus Q_{m+1}$.

We add to this tournament the following backward arcs: since we picked $m+1 \equiv 3 \pmod 6$, we can compute in polynomial time an edge-disjoint (undirected) triangle packing of the clique of order $m+1$ using exactly all the edges [100]; let $\Delta_{K_{m+1}}$ be such a perfect packing, and let $\{v_1, \ldots, v_{m+1}\}$ be an arbitrary enumeration of the vertices of the clique. We now create the tournament $T_{K_{m+1}}$ by providing the following linear representation: we set $\sigma(T_{K_{m+1}}) = (v_1, \ldots, v_{m+1})$ and $\overleftarrow{A}(T_{K_{m+1}}) = \{(v_i, v_k) \colon (v_i, v_j, v_l) \in \Delta_{K_{m+1}} \text{ and } i < j < k\}$. Finally, we now set the backward set of our variable tournament $Q$:

$$\overleftarrow{A}(Q) = \Big( \bigcup_{j=1}^{m+1} \overleftarrow{A}(Q_j) \Big) \cup \{(u, v) \colon u \in Q_i, v \in Q_j \text{ and } (v_j, v_i) \in \overleftarrow{A}(T_{K_{m+1}})\}$$

Informally, we can see the tournament $Q$ as the tournament $T_{K_{m+1}}$ where we "blow up" every $v_j$ into the clause gadget $Q_j$.

**The variable tournament** For each variable of $\mathcal{F}$, we define the *variable gadget* $X_i$, a tournament of order 6, as depicted in Figure 5.2. Formally, we set the following ordering $\sigma_i(X_i) = (r_i, \overline{\ell_i}, \ell_i^1, s_i, \ell_i^2, t_i)$ and the corresponding backward arc set $\overleftarrow{A}(X_i) = \{(s_i, r_i), (t_i, \ell_i^1)\}$. One can notice that this set is a *minimum* feedback arc set of the tournament.

As we just did before, since $n \equiv 3 \pmod 6$, we can use the same operation to add the backwards arcs to $\overleftarrow{A}(X)$ from a perfect packing of undirected triangles of $K_n$.

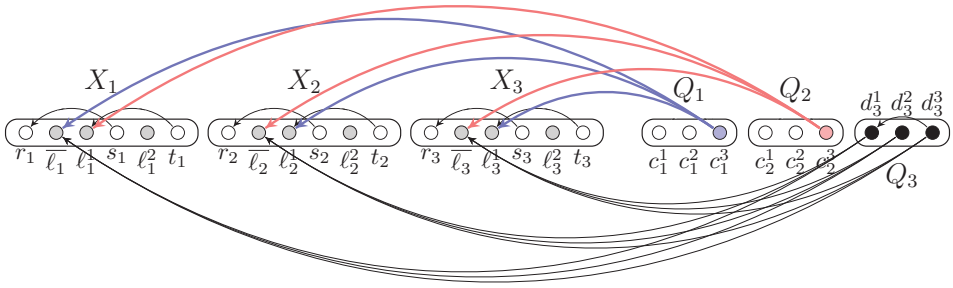



**Figure 5.3** – Example of the tournament obtained with the clauses $c_1 = \overline{x_1} \vee x_2 \vee x_3$ and $c_2 = x_1 \vee \overline{x_2} \vee \overline{x_3}$. We only depicted backward arcs. We omitted the backward arcs from any vertex of $X_3$ to $X_1$ (resp. $Q_3$ to $Q_1$) corresponding to the orientation of $K_3$. The blue arcs (resp. red arcs) correspond to the encoding of the clause $c_1$ (resp. $c_2$).

**The tournament $T_{g(\mathcal{F})}$** We can now describe our final tournament $T_{g(\mathcal{F})}$. Let $\overleftarrow{A}_{enc}(T_{g(\mathcal{F})})$ be the set of backward arcs from $Q$ to $X$. Such arcs will encode the description of the clauses of $\mathcal{F}$. Let $l_i$ be a literal corresponding to the variable $x_i$ appearing in the clause $c_j$. If $l_i$ is negative, then we add in $\overleftarrow{A}_{enc}(T_{g(\mathcal{F})})$ the backward arc $(c_j^2, \overline{\ell_i})$. Otherwise, if $l_i$ is positive and let $k$ be the value of $o(x_i, c_j)$, that is $c_j$ contains the $k^{th}$ occurrence of the variable $x_i$, then we add to $\overleftarrow{A}_{enc}(T_{g(\mathcal{F})})$ the arc $(c_j^2, \ell_i^k)$. Finally, for any $p \in [\![1,3]\!]$ and $i \in [\![1,n]\!]$, we add to $\overleftarrow{A}_{enc}(T_{g(\mathcal{F})})$ the backward arcs $(d^p, \overline{\ell_i})$. Such arcs are called *dummy arcs*. Now, we define $T_{g(\mathcal{F})}$ as the concatenation $X \oplus Q$ and we set its backward arc set $\overleftarrow{A}(T_{g(\mathcal{F})}) = \overleftarrow{A}(X \oplus Q) \cup \overleftarrow{A}_{enc}(T_{g(\mathcal{F})})$.

Finally, one can easily notice that this tournament has $6n + 3(m+1)$ vertices and can be construct in polynomial time. The Figure 5.3[2] is an example of $T_{g(\mathcal{F})}$ for a naive instance of Max 3-SAT(3). This concludes the definition of the reduction $g$.

### 5.2.2 Observations and Lemmas on the Tournament $T_{g(\mathcal{F})}$

First of all, observe that in each variable gadget $X_i$, there are only four possible triangles: let $\delta_i^1$, $\delta_i^2$, $\delta_i^3$ and $\delta_i^4$ be the triangles $(r_i, \overline{\ell_i}, s_i)$, $(r_i, \ell_i^1, s_i)$, $(\ell_i^1, s_i, t_i)$ and $(\ell_i^1, \ell_i^2, t_i)$, respectively. Moreover, notice that among these triangles there are only three maximum arc-disjoint triangle packing of $X_i$, namely $\{\delta_i^1, \delta_i^3\}$, $\{\delta_i^1, \delta_i^4\}$ and $\{\delta_i^2, \delta_i^4\}$. We respectively denote these packings by $\triangle_i^\top$, $\triangle_i^{\top'}$ and $\triangle_i^\perp$ (see Figure 5.4).

Informally, we want to set the variable $x_i$ at true (respectively false) when

[2]Note that in this example, each variable appears only once positively. This is due to the fact that the variables need to appear also once negatively *and* having $n \equiv 3 \pmod 6$ as well as $m + 1 \equiv 3 \pmod 6$; an example with a variable with two positive occurrences would have been to large to be understandable. Despite that, we mention to the reader that if $x_1$ had also appeared positively in another clause $c_j$, the vertex of $X_1$ linked to $Q_j$ would have been $\ell_i^2$.

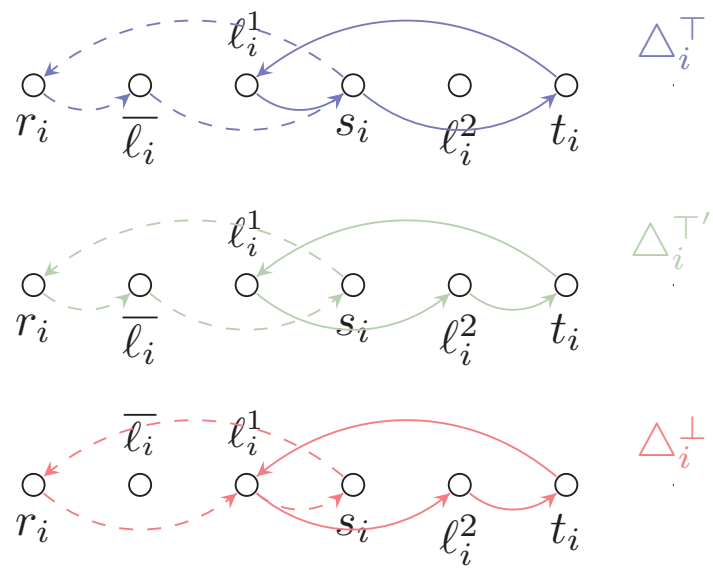



**Figure 5.4** – The three maximum arc-disjoint triangle packings of $X_i$: $\triangle_i^\top$ (in blue), $\triangle_i^{\top\prime}$ (in green) and $\triangle_i^\perp$ (in red).

one of the locally-optimal $\triangle_i^\top$or $\triangle_i^{\top\prime}$ (respectively $\triangle_i^\perp$) is taken in the variable gadget $X_i$ in the global solution.

In addition, we also want the vertex $\overline{\ell_i}$ to have "enough" out-neighbors in $X_i$ in order create triangles with the dummy arcs. To do so, let us define the local out-degree of a vertex. Given an arc-disjoint triangle packing $\triangle$ of $T_{g(\mathcal{F})}$ and a subset $K$ of vertices of $T_{g(\mathcal{F})}$, we define for any $v \in K$ the *$\triangle$-local out-degree* of the vertex $v$, denoted $d^+_{K\setminus\triangle}(v)$, as the remaining out-degree of $v$ in $T_{g(\mathcal{F})}[K]$ when we remove the arcs of the triangles of $\triangle$. More formally, we have $d^+_{K\setminus\triangle}(v) = |\{(v,u)\colon\ u \in K, (v,u) \in A(T_{g(\mathcal{F})}[K]), (v,u) \notin A(\triangle)\}|$.

We can now make the following observations on the local out-degrees of the vertices when we consider the arc-disjoint triangle packings $\triangle_i^\top$, $\triangle_i^{\top\prime}$ and $\triangle_i^\perp$.

**Observation 29.** *Given a gadget variable $X_i$, we have:*

*(i)* $d^+_{X_i\setminus\triangle_i^\top}(\ell_i^1) = d^+_{X_i\setminus\triangle_i^\top}(\ell_i^2) = 1$ *and* $d^+_{X_i\setminus\triangle_i^\top}(\overline{\ell_i}) = 3$,

*(ii)* $d^+_{X_i\setminus\triangle_i^{\top\prime}}(\ell_i^1) = 1$, $d^+_{X_i\setminus\triangle_i^{\top\prime}}(\ell_i^2) = 0$ *and* $d^+_{X_i\setminus\triangle_i^{\top\prime}}(\overline{\ell_i}) = 3$,

*(iii)* $d^+_{X_i\setminus\triangle_i^\perp}(\ell_i^1) = d^+_{X_i\setminus\triangle_i^\perp}(\ell_i^2) = 0$ *and* $d^+_{X_i\setminus\triangle_i^\perp}(\overline{\ell_i}) = 4$,

*(iv) none of* $(\overline{\ell_i}, \ell_i^1)$, $(\overline{\ell_i}, \ell_i^2)$ *and* $(\overline{\ell_i}, t_i)$ *belongs to* $\triangle_i^\top$ *nor* $\triangle_i^\perp$.

Now, given a triangle packing $\triangle$ of $T_{g(\mathcal{F})}$, we partition $\triangle$ into the following sets corresponding to the possible "shapes" a triangle can have (see Figure 5.5):

- $\triangle_{X,X,X} = \{(v_1,v_2,v_3) \in \triangle\colon\ v_1 \in V(X_i), v_2 \in V(X_j), v_3 \in V(X_k)$ with $i < j < k\}$,
- $\triangle_{X,X,Q} = \{(v_1,v_2,v_3) \in \triangle\colon\ v_1 \in V(X_i),\ v_2 \in V(X_j),\ v_3 \in V(Q_k)$ with $i < j\}$,

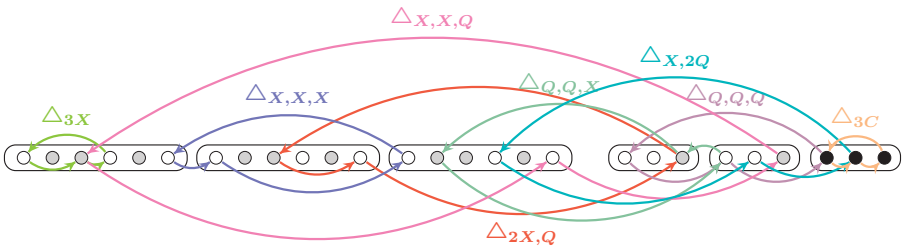



**Figure 5.5** – Example of the different possible shapes of the triangles in $T_{g(\mathcal{F})}$.

- $\triangle_{X,Q,Q} = \{(v_1, v_2, v_3) \in \triangle\colon\ v_1 \in V(X_i),\ v_2 \in V(Q_j),\ v_3 \in V(Q_k) \text{ with } j < k\}$,
- $\triangle_{Q,Q,Q} = \{(v_1, v_2, v_3) \in \triangle\colon\ v_1 \in V(Q_i), v_2 \in V(Q_j), v_3 \in V(Q_k) \text{ with } i < j < k\}$,
- $\triangle_{2X,Q} = \{(v_1, v_2, v_3) \in \triangle\colon\ v_1, v_2 \in V(X_i),\ v_3 \in V(Q_j)\}$,
- $\triangle_{X,2Q} = \{(v_1, v_2, v_3) \in \triangle\colon\ v_1 \in V(X_i),\ v_2, v_3 \in V(Q_j)\}$,
- $\triangle_{3X} = \{(v_1, v_2, v_3) \in \triangle\colon\ v_1, v_2, v_3 \in V(X_i)\}$,
- $\triangle_{3Q} = \{(v_1, v_2, v_3) \in \triangle\colon\ v_1, v_2, v_3 \in V(Q_i)\}$.

Notice that in the previous partition, all the "theoretical shapes" have not been defined: some are impossible by definition of $T_{g(\mathcal{F})}$. For example, there is no triangle with two of its vertices in a variable gadget $X_i$ and its last vertex in another variable gadget $X_j$ since all the arcs between two distinct variable gadgets are oriented in the same direction. Obviously, we have the same observation for clause gadgets.

Let establish some quick lemmas to give upper bounds on the size of the parts of the solution $\triangle$.

**Lemma 29.1.** *There exists an arc-disjoint triangle packing $\triangle^x$ (resp. $\triangle^q$) which use exactly the arcs between distinct variable gadgets (resp. clause gadgets). Therefore, we have $|\triangle_{X,X,X}| \leq 6n(n-1)$ and $|\triangle_{Q,Q,Q}| \leq 3m(m+1)/2$ and these bounds are tight.*

*Proof.* We will prove that from each triangle of the perfect packing arc-disjoint triangles of $T_{K_n}$ we can obtain a new packing of arc-disjoint triangles using all the arcs joining the corresponding variable gadgets. For sake of clarity, we temporally relabel the vertices of $X_i$, $X_j$ and $X_k$ respectively by $\{u_1, \ldots, u_6\}$, $\{v_1, \ldots, v_6\}$ and $\{w_1, \ldots, w_6\}$ and consider the tripartite tournament $K_{6,6,6}$ given by its vertex set $V(K_{6,6,6}) = \{u_i, v_i, w_i\colon\ i \in [\![1, 6]\!]\}$ and its arc set $A(K_{6,6,6}) = \{(u_i, v_j), (v_i, w_j), (w_i, u_j)\colon\ i, j \in [\![1, 6]\!]\}$.

Then it is easy to check that $\{(u_i, v_j, w_{(i+j) \pmod 6})\colon\ i, j \in [\![1, 6]\!]\}$ is a perfect triangle packing of $K_{6,6,6}$ (see Figure 5.6). Now, notice that every triangle of $T_{K_n}$ becomes a $K_{6,6,6}$ in $X$, then we can find a triangle packing $\triangle^x$ which use all the arcs between disjoint variable gadgets, that is with $6n(n-1)$ triangles.

We use the same reasoning to prove that there exists a triangle packing $\triangle^q$ which use all the arcs available in $Q$ between two distinct clause gadget. $\square$

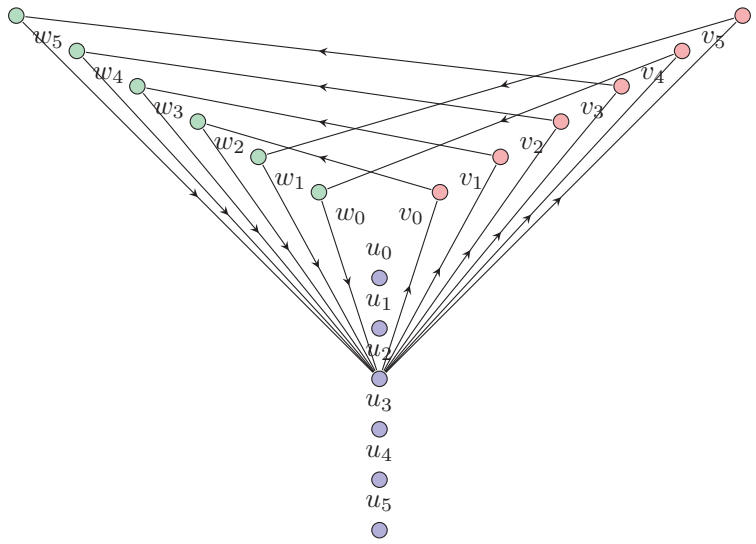



**Figure 5.6** – All the arc-disjoint triangles the vertex $u_2$ belongs to. Notice that it uses all the arcs going in and out $u_2$.

In the following, let $\triangle^x$ and $\triangle^q$ be such arc-disjoint triangle packings for $X$ and $Q$, respectively.

**Claim 29.1.** *For any arc-disjoint triangle packing $\triangle$ of the tournament $T_{g(\mathcal{F})}$, we have the following inequalities:*

*(i)* $|\triangle_{X,X,X}| + |\triangle_{Q,Q,Q}| \leq 6n(n-1) + 3m(m+1)/2$,

*(ii)* $|\triangle_{2X,Q}| + |\triangle_{X,2Q}| + |\triangle_{X,Q,Q}| + |\triangle_{X,X,Q}| \leq |\overleftarrow{A}_{enc}(T_{g(\mathcal{F})})|$,

*(iii)* $|\triangle_{3X}| \leq 2n$,

*(iv)* $|\triangle_{3Q}| \leq 1$.

*Therefore, we have* $|\triangle| \leq 6n(n-1) + 3m(m+1)/2 + 2n + |\overleftarrow{A}_{enc}(T_{g(\mathcal{F})})| + 1$.

*Proof.* Let $\triangle$ be an arc-disjoint triangle packing of $T_{g(\mathcal{F})}$. We successively prove the different inequalities of the claim:

(i) First, inequality $(i)$ is a direct consequence of Lemma 29.1.

(ii) We have $|\triangle_{2X,Q}|+|\triangle_{X,2Q}|+|\triangle_{X,Q,Q}|+|\triangle_{X,X,Q}| \leq |\overleftarrow{A}_{enc}(T_{g(\mathcal{F})})|$ since every triangle of these sets consumes exactly one backward arcs from $Q$ to $X$.

(iii) We have $|\triangle_{3X}| \leq 2n$ since we have at most two arc-disjoint triangles in each variable gadget.

(iv) Finally, we also have $|\triangle_{3Q}| \leq 1$ since the dummy triangle is the only one lying into a clause gadget.

□

In the following, let $k_{g(\mathcal{F})}$ be the value of $6n(n-1) + 3m(m+1)/2 + 2n + |\overleftarrow{A}_{enc}(T_{g(\mathcal{F})})| + 1$. This will correspond to the threshold value for the existence of a satisfiable assignation of $\mathcal{F}$. This is the purpose of our next subsection.

### 5.2.3 NP-hardness of ATT

**Theorem 30.** *The formula $\mathcal{F}$ is satisfiable if and only if there exists an arc-disjoint packing of triangles $\triangle$ of size $k_{g(\mathcal{F})}$ in the tournament $T_{g(\mathcal{F})}$.*

*Proof.* First, let suppose that there exists an assignation $a$ of the variables which satisfies $F$.

We construct a packing $\triangle$ of $T_{g(\mathcal{F})}$ with the desired number of arc-disjoint triangles. We start by picking all the disjoint triangles of $\triangle^x$ and $\triangle^q$ which consume all the arc joining distinct variable gadgets and distinct clause gadgets, respectively. By Lemma 29.1, it corresponds to $6n(n-1)+3m(m+1)/2$ triangles.

Then, for any variable $x_i$ of the formula $\mathcal{F}$ set to true with the assignation $a$, we add in $\triangle$ the triangles $\triangle_i^\top$. Otherwise, we add $\triangle_i^\perp$. Since such triangles lie inside a variable gadget, it is easy to notice that they are arc-disjoint to the triangles we added so far. Thus, we increased by $2n$ the number of triangles in $\triangle$.

Now, we want to consume all the backward arcs from $Q$ to $X$. First, we add in $\triangle$ the triangles $(\overline{\ell_i}, \ell_i^1, d^2)$, $(\overline{\ell_i}, \ell_i^2, d^2)$ and $(\overline{\ell_i}, t_i, d^3)$ which will consumes all the dummy arcs of the tournament. By the item (iv) of Observation 29, we added arc-disjoint triangles to $\triangle$.

All the remaining the backward arcs from $Q$ to $X$ will now be used to encode the satisfiability of the clause. To do so, given a clause $c_j$, let $x_i$ be one variable satisfying $c_j$. If $x_i$ appears positively in $c_j$, let $k$ be the value of $o(x_i, c_j)$, that is it is the $k^{th}$ occurrence of $x_i$. By Observation 29(i), we have $d^+_{X_i \setminus \triangle_i^\top}(\ell_i^1) = d^+_{X_i \setminus \triangle_i^\top}(\ell_i^2) = 1$. So we know that there exists $v \in X_i$ such that the arc $(\ell_i^k, v)$ is available to make the triangle $(\ell_i^k, v, c_j^3)$.

Otherwise, that is if $x_i$ appears negatively in $c_j$, then by Observation 29(iii), we have $d^+_{X_i \setminus \triangle_i^\perp}(\overline{\ell_i}) = 4$. Three of these four available arcs were used previously in the triangles which consume the dummy arcs, then we can still make the triangle $(\overline{\ell_i}, s_i, c_j^3)$.

Now, assume that the clause is of size 3, since the reasoning is the same for clauses of size 2. Let $x_{i_1}$ and $x_{i_2}$ be the two other variables of $c_j$ and denote by $v_{i_1}$ and $v_{i_2}$ the vertices of $X_i$ connected to $c_j^3$ corresponding to $x_{i_1}$ and $x_{i_2}$, respectively. Then, we add the triangles $(v_{i_1}, c_j^1, c_j^3)$ and $(v_{i_2}, c_j^2, c_j^3)$.

Therefore, we used all the backward arc from $Q$ to $X$, and there are no triangles which use two arcs of $\overleftarrow{A}_{enc}(T_{g(\mathcal{F})})$.

Finally, by adding the dummy triangle we obtain an arc-disjoint triangle packing $\triangle$ of size $6n(n-1)+3m(m+1)/2+2n+|\overleftarrow{A}_{enc}(T_{g(\mathcal{F})})|+1$, which concludes this direction of the proof.

Conversely, let $\triangle$ be a triangle packing of $T_{g(\mathcal{F})}$ with $|\triangle| = k_{g(\mathcal{F})}$. In the same way as we already did before, we partition $\triangle$ into the different subsets we previously defined. We have:

$$|\triangle| = |\triangle_{X,X,X}| + |\triangle_{X,X,Q}| + |\triangle_{X,Q,Q}| + |\triangle_{Q,Q,Q}| + |\triangle_{2X,Q}| + |\triangle_{X,2Q}| + |\triangle_{3X}| + |\triangle_{3Q}|$$

Using Claim 29.1, it implies that all the upper bounds described above are tight, that is:

- $|\triangle_{X,X,X}| + |\triangle_{Q,Q,Q}| = 6n(n-1) + 3m(m+1)/2$,
- $|\triangle_{2X,Q}| + |\triangle_{X,2Q}| + |\triangle_{X,Q,Q}| + |\triangle_{X,X,Q}| = |\overleftarrow{A}_{enc}(T_{g(\mathcal{F})})|$,
- $|\triangle_{3X}| = 2n$,
- $|\triangle_{3Q}| = 1$.

As $|\triangle_{3Q}| = 1$, the dummy triangle $Q_{m+1}$ belongs to $\triangle$. Since $|\triangle_{3X}| = 2n$ and we have at most two arc-disjoint triangles in each variable gadget $X_i$, it implies that $\triangle_i$, the collection of triangles of $\triangle$ with all their vertices in $X_i$, is in $\{\triangle_i^\top, \triangle_i^{\top\prime}, \triangle_i^\perp\}$. Let consider the following assignation $a$: for any variable $x_i$, if $\triangle_i = \triangle_i^\perp$, then we set $a(x_i) =$ false and $a(x_i) =$ true otherwise. Let prove that such assignation satisfies the formula $\mathcal{F}$.

To do so, let us first prove that $\triangle_{X,X,Q} = \triangle_{X,Q,Q} = \emptyset$. Informally, triangles of $\triangle_{X,X,Q}$ and $\triangle_{X,Q,Q}$ use arcs joining two variable/clause gadgets, but such arcs would have be used for the triangles of $\triangle^x$ and $\triangle^q$ and so the threshold value could not be reached. So we have $|\triangle_{X,X,X}| < |\triangle^x|$ or $|\triangle_{Q,Q,Q}| < |\triangle^q|$. Moreover, since each triangle of the sets $\triangle_{X,X,Q}, \triangle_{X,Q,Q}, \triangle_{2X,Q}$ and $\triangle_{X,2Q}$ use only one arc of $\overleftarrow{A}_{enc}(T_{g(\mathcal{F})})$, it implies that $|\triangle_{2X,Q}| + |\triangle_{X,2Q}| \leq |\overleftarrow{A}_{enc}(T_{g(\mathcal{F})})| - |\triangle_{X,X,Q}| - |\triangle_{X,Q,Q}|$. Recall we still have $|\triangle_{3X}| \leq 2n$ and $|\triangle_{3Q}| \leq 1$ by construction. Therefore, if $|\triangle_{X,X,Q}| + |\triangle_{X,Q,Q}| \neq 0$, we have:

$$|\triangle| < |\triangle^x| + |\triangle^q| + |\triangle_{X,X,Q}| + |\triangle_{X,Q,Q}| + (|\overleftarrow{A}_{enc}(T_{g(\mathcal{F})})| - |\triangle_{X,X,Q}| - |\triangle_{X,Q,Q}|) + 2n + 1$$

That is $|\triangle| < k_{g(\mathcal{F})}$, which is impossible. So we have $\triangle_{X,X,Q} = \triangle_{X,Q,Q} = \emptyset$. It proves that the backward arcs from $Q$ to $X$ are all used in $\triangle_{2X,Q}$ and $\triangle_{X,2Q}$. So every dummy arc $(d^p, \overline{\ell_i})$ is contained in a triangle of $\triangle$ which uses at least an arc of $X_i$. Therefore, in each $X_i$, we have $d^+_{X_i \setminus \triangle_i}(\overline{\ell_i}) \geq 3$.

Let $c_j$ be a clause of size $p$. It implies that there are $p$ backward arcs coming from the clause gadget to the corresponding variable gadgets, and thus $p$ triangles of $\triangle$ using these arcs. By construction, at most one triangle with two vertices in $Q_j$ and one vertex in a gadget variable can lie in $\triangle_{X,2Q}$. Let $\delta$ be one of the triangle of $\triangle_{2X,Q}$ with one vertex in $Q_j$, and let $X_i$ be the variable gadget containing the two remaining vertices. Furthermore, we call $v$ the vertex of $X_i$ which is also the head of the backward arc from $Q_j$ to $X_i$.

Let consider first the case where the corresponding variable $x_i$ appears negatively in $c_j$. Then, we have $v = \overline{\ell_i}$. Since $\overline{\ell_i}$ must use three out-going arcs to consume the dummy arcs and one out-going arc for the triangle $\delta$, we have $d^+_{X_i \setminus \triangle_i}(\overline{\ell_i}) \geq 4$. By Observation 29, it implies that $\triangle_i = \triangle_i^\perp$, so $a(x_i) =$ false: the clause $c_j$ is satisfied.

Otherwise, if $x_i$ appears positively in $c_j$, then we have $v \in \{\ell_i^1, \ell_i^2\}$. In both cases, since $\delta$ has a second vertex in $X_i$, we have $d^+_{X_i \setminus \triangle_i}(v) > 0$. Thus, using Observation 29 we cannot have $\triangle_i = \triangle_i^\perp$ so the assignment sets $x_i$ to true, which satisfies $c_j$.

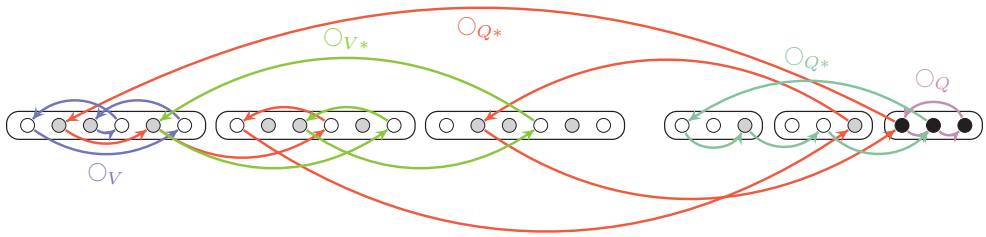



**Figure 5.7** – Example of the different possible shapes of the cycles in $T_{g(\mathcal{F})}$.

This proves that the assignation $a$ satisfies the formula $\mathcal{F}$, and it concludes the proof.

□

As Max 3-SAT(3) is known to be NP-hard [126], Theorem 30 directly implies the following result.

**Theorem 31.** *The problem* Arc-Disjoint Triangle Packing in Tournament *is* NP*-hard.*

### 5.2.4 NP-hardness of ACT

As every cycle in a tournament contains a triangle, it is well known that, for vertex-disjoint packings, any cycle packing of size $k$ implies a triangle packing of same size. This implies that cycle and triangle packings are equivalent in the vertex-disjoint case. However, as we mentioned before, for the arc-disjoint case, this observation is no longer true in the general.

However, the following lemma shows that in the specific case of the tournament constructed by our reduction, this observation holds. This will transfer the previous NP-hardness result to ACT.

Informally, the proof of the NP-hardness of ACT will be similar than the proof we used for the arc-disjoint triangle packing: we will categorize the cycles of a solution into sets according their possible "shapes":

Formally, given a cycle packing $\mathcal{C}$ of $T_{g(\mathcal{F})}$ of size $k_{g(\mathcal{F})}$, we partition it into the following sets (see Figure 5.7):

- $\mathcal{C}_X = \{(v_1, \dots, v_p) \in \mathcal{C} \colon \exists i \in [\![1, n]\!], \forall k \in [\![1, p]\!], v_k \in V(X_i)\}$,
- $\mathcal{C}_Q = \{(v_1, \dots, v_p) \in \mathcal{C} \colon \exists j \in [\![1, m+1]\!], \forall k \in [\![1, p]\!], v_k \in V(Q_j)\}$,
- $\mathcal{C}_{X^*} = \{(v_1, \dots, v_p) \in \mathcal{C} \colon \forall k \in [\![1, p]\!], \exists i \in [\![1, n]\!], v_k \in V(X_i)\} - \mathcal{C}_X$,
- $\mathcal{C}_{Q^*} = \{(v_1, \dots, v_p) \in \mathcal{C} \colon \forall k \in [\![1, p]\!], \exists j \in [\![1, m+1]\!], v_k \in V(Q_j)\} - \mathcal{C}_Q$,
- $\mathcal{C}_{X^*,Q^*} = \mathcal{C} - \big(\mathcal{C}_X \cup \mathcal{C}_Q \cup \mathcal{C}_{X^*} \cup \mathcal{C}_{Q^*}\big)$.

We can now give upper bounds of these sets:

**Claim 31.1.** *For any arc-disjoint cycle packing $\mathcal{C}$ of the tournament $T_{g(\mathcal{F})}$, we have the following inequalities:*

*(i)* $|\mathcal{C}_{X^*}| + |\mathcal{C}_{Q^*}| \leq 6n(n-1) + 3m(m+1)/2$,

*(ii)* $|\mathcal{C}_{X^*,Q^*}| \leq |\overleftarrow{A}_{enc}(T_{g(\mathcal{F})})|$,

*(iii)* $|\mathcal{C}_X| \leq 2n$,

*(iv)* $|\mathcal{C}_Q| \leq 1$.

*Therefore, we have* $|\mathcal{C}| \leq 6n(n-1) + 3m(m+1)/2 + 2n + |\overleftarrow{A}_{enc}(T_{g(\mathcal{F})})| + 1$.

*Proof.* Let $\mathcal{C}$ be an arc-disjoint cycle packing of $T_{g(\mathcal{F})}$. We successively prove the different inequalities of the claim:

(i) First, inequality $(i)$ is a direct consequence of Lemma 29.1.

(ii) Notice that a cycle of $\mathcal{C}_{X^*}$ cannot belong to exactly two distinct variable gadgets since the arcs between them are all in the same direction. Thus, the cycles of $\mathcal{C}_{X^*}$ have at least three vertices which implies $|\mathcal{C}_{X^*}| \leq 6n(n-1)$. We obtain $|\mathcal{C}_{Q^*}| \leq 3m(m+1)/2$ using the same reasoning on $\mathcal{C}_{Q^*}$.

(iii) Recall that $\text{fas}(X_i) = 2$. Thus, we have $|\mathcal{C}_X| \leq 2n$.

(iv) Finally, by construction, we also have $|\mathcal{C}_Q| \leq 1$.

Putting these upper bound together, we obtain that $|\mathcal{C}| \leq 6n(n-1) + 3m(m+1)/2 + 2n + |\overleftarrow{A}_{enc}(T_{g(\mathcal{F})})| + 1 = k_{g(\mathcal{F})}$

□

We now prove that we can restructure every cycles of an arc-disjoint cycle packing of size $k_{g(\mathcal{F})}$ into an arc-disjoint *triangle* packing of the same size.

**Lemma 31.1.** *Given a* Max 3-SAT(3) *instance $\mathcal{F}$, and $T_{g(\mathcal{F})}$ the tournament constructed from $\mathcal{F}$ with the reduction $g$, we have an arc-disjoint triangle packing of $T_{g(\mathcal{F})}$ of size $k_{g(\mathcal{F})}$ if and only if there is an arc-disjoint cycle packing of size $k_{g(\mathcal{F})}$.*

*Proof.* First of all, it is easy to notice than an arc-disjoint triangle packing $\mathcal{C}$ of $T_{g(\mathcal{F})}$ of size $k_{g(\mathcal{F})}$ directly implies an arc-disjoint cycle packing of size $k_{g(\mathcal{F})}$. Then, the only remaining thing to prove is than we can find a arc-disjoint triangle packing of size $k_{g(\mathcal{F})}$ from an arc-disjoint cycle packing $\mathcal{C}$ of same size.

Putting together the inequalities of Claim 31.1, we have $|\mathcal{C}| \leq k_{g(\mathcal{F})}$ which implies that all these inequalities are tight. In particular, cycles of $\mathcal{C}_{X^*}$ and $\mathcal{C}_{Q^*}$ use exactly three arcs that are between distinct variable and clause gadgets, using all these arcs. So we can construct a new cycle packing by replacing the cycles of $\mathcal{C}_{X^*}$ and $\mathcal{C}_{Q^*}$ by the arc-disjoint triangle packings $\triangle^x$ and $\triangle^q$ defined in Lemma 29.1. The new solution has the same size.

The tightness of the upper bound of the size $\mathcal{C}_{X^*,Q^*}$ implies that these cycles use exactly one backward arc of $\overleftarrow{A}_{enc}(T_{g(\mathcal{F})})$. Moreover, since all the arcs joining two distinct variable gadgets and two distinct clause gadgets ares used in $\mathcal{C}_{X^*}$

and $\mathcal{C}_{Q^*}$, it implies that vertices of a cycle of $\mathcal{C}_{X^*,Q^*}$ cannot be in multiples variable/clause gadgets.

Let $c_j$ be a clause of $\mathcal{F}$. Consider first it has two variables, say $x_{i_1}$ and $x_{i_2}$. In the following, we denote by $v_1$ and $v_2$ the vertices corresponding to the variables $x_{i_1}$ and $x_{i_2}$, respectively, and such that they are the head of the backward arcs from $c_j^3$. By the previous remark, these two vertices are both in a cycle of $\mathcal{C}_{X^*,Q^*}$, and they cannot be in the same cycle. Let $C_1$ and $C_2$ be the cycles of $v_1$ and $v_2$, respectively. Then we can restructure them with the triangles $(v_1, c_j^1, c_j^3)$ and $(v_2, c_j^2, c_j^3)$. All the arcs of these triangles were either used by $C_1$ and $C_2$, or free, proving that the new packing is still arc-disjoint.

Then $c_j$ as three variables, say $x_{i_1}$, $x_{i_2}$ and $x_{i_3}$. As we previously did, let $v_1$, $v_2$ and $v_3$ be the vertices corresponding to the variables of $c_j$ such that they are the head of the backward arcs from $c_j^3$. Moreover, let $C_1$, $C_2$ and $C_3$ be the cycles of $v_1$, $v_2$ and $v_3$, respectively. By construction of $Q_j$, there is a vertex, say $v_3$, whose following vertex along $C_3$ is in $X_{i_3}$. Denote by $v$ such vertex. Then, we can restructure $C_1$ and $C_2$ into the triangles $(v_1, c_j^1, c_j^3)$ and $(v_2, c_j^2, c_j^3)$, and replace $C_3$ by $(v_3, v, c_j^3)$. The arc $(v, c_j^3)$ is available since it could have been used only in $C_3$. It proves that we can restructure $\mathcal{C}_{X^*,Q^*}$ into an arc-disjoint packing of triangles of same size.

We now want to prove the analogous property on $\mathcal{C}_X$. We recall that $\mathcal{C}_X[X_i]$ (*i.e.* the cycles of $\mathcal{C}_X$ with all its vertices in $X_i$) has exactly two cycles, and notice that these cycles cannot both have size greater than 3 by construction. First, if these cycles are already triangles, we are done. Then $\mathcal{C}_X[X_i]$ contains a triangle $\delta$ and a cycle $C$ of size greater than 3.

If $C$ contains the backward arc $(s_i, r_i)$, then by construction we have $C = (r_i, \overline{\ell_i}, \ell_i^1, s_i)$ and $\delta = (\ell_i^1, \ell_i^2, t_i)$. Therefore, we can restructure $C$ into the triangle $(r_i, \ell_i^1, s_i)$. The arc $(r_i, \ell_i^1)$ is not contained in $\mathcal{C}$ since the only arcs inside $X_i$ we may have already taken previously are out-going arcs of $\ell_i^1$, $\ell_i^2$ and $\overline{\ell_i}$.

Otherwise, if $C$ contains the backward arc $(t_i, \ell_i^1)$, then by construction we have $C = (\ell_i^1, s_i, \ell_i^2, t_i)$ and $\delta = (r_i, \overline{\ell_i}, s_i)$. In the same way, we can restructure $C$ into $(\ell_i^1, s_i, t_i)$ whose all the arcs are available.

Finally, as $\mathcal{C}_Q$ is trivially a triangle already, $T_{g(\mathcal{F})}$ admits an arc-disjoint triangle packing of size $k_{g(\mathcal{F})}$. □

The previous lemma and Theorem 31 directly imply the following result:

**Theorem 32.** *The problem* Arc-Disjoint Cycle Packing in Tournament *is* NP*-hard.*

Finally, notice that the size of the required packing in Theorem 30 verifies $O((n+m)^2)$. Under the *Exponential Time Hypothesis*[3], 3-SAT cannot be solved in $2^{o(n+m)}$ [48, 88]. Then, using the linear reduction from 3-SAT to Max 3-SAT(3) [146], we also get the following result.

[3]The hypothesis states that be the infimum of the real numbers $s_3$ for which 3-SAT can be algorithmically solved in time $O(2^{s_3 n})$ is strictly positive. If this conjecture is true, it implies that 3-SAT cannot be solved in subexponential running-time.

**Corollary 2.** *The problems* Arc-Disjoint Triangle Packing in Tournament *and* Arc-Disjoint Cycle Packing in Tournament *cannot be solved in* $2^{o(\sqrt{k})}$*, unless the* Exponential Time Hypothesis *fails.*

### 5.2.5 NP-hardness of the Tight Versions of ATT and ACT

In this subsection, we focus on two special cases of ATT and ACT, where the tournament $T$ of the input is given with a linear representation with exactly $\mathrm{fas}(T)$ backward arcs, and the goal is to decide if there is an arc-disjoint packing of $\mathrm{fas}(T)$ triangles/cycles. We call these problems Tight Arc-Disjoint Triangle Packing in Tournament (Tight-ATT, in short) and Tight Arc-Disjoint Cycle Packing in Tournament (Tight-ACT, in short), respectively.

We say that this is the "tight" versions of ATT and ACT because if the minimum feedback arc set of a digraph is of size $k$, then the size of an arc-disjoint packing of triangles/cycles is at most $k$—implying that inequalities are tight. In other words, we provide the tournament with one of its minimum feedback arc set, and the problem is to decide if we can do the "best possible" arc-disjoint packing of triangles/cycles.

To explore the complexity of these two problems, we start by proving the following claim:

**Claim 32.1.** *We can construct in polynomial time an ordering of* $T_{g(\mathcal{F})}$ *with* $k_{g(\mathcal{F})}$ *backward arcs.*

*Proof.* It is easy to see that the linear representation we gave in the definition of the reduction verifies the statement of the claim. Indeed, we oriented the perfect packing of undirected edge-disjoint triangles of $K_n$ such that there is exactly one backward arc per triangle. Then, we replaced each vertex by a variable gadget while preserving the orientation. It leads to an ordering with $36\big(n(n-1)/6\big) = 6n(n-1)$ backward arc joining distinct variable gadgets. The same observation shows that we have $3m(m+1)/2$ backward arcs joining distinct clause gadgets.

In the provided ordering, there are two backward arcs in each $X_i$, one backward arc in $Q_{m+1}$, and $|\overleftarrow{A}_{enc}(T_{g(\mathcal{F})})|$ backward arcs from $Q$ to $X$: it proves there exist an ordering of $T_{g(\mathcal{F})}$ with $k_{g(\mathcal{F})}$ backward arcs. □

The previous claim basically states that $\mathrm{fas}(T_{g(\mathcal{F})}) \leq k_{g(\mathcal{F})}$, since removing the backward arcs of this ordering would make the tournament acyclic. We now prove that $k_{g(\mathcal{F})}$ is the optimal value, that is $\mathrm{fas}(T_{g(\mathcal{F})}) = k_{g(\mathcal{F})}$.

**Lemma 32.1.** *The size of an optimal feedback arc set of* $T_{g(\mathcal{F})}$ *is at most* $k_{g(\mathcal{F})}$.

*Proof.* In the following, we consider that the tournament $T_{g(\mathcal{F})}$ is given with the linear representation defined previously in Claim 32.1. Furthermore, let $F$ be a minimum-sized feedback arc set of $T_{g(\mathcal{F})}$. Recall that, given an arc $a$, $V(a)$ denotes the set of the vertices of $a$, that is $\{h(a), t(a)\}$.

As done previously on the packing of arc-disjoint triangles or cycles, we partition the solution into the following sets.
For any $i \in [\![1, n]\!]$ and $j \in [\![1, m+1]\!]$, we define

- $F_{X_i} = \{a \in F\colon\ V(a) \subseteq V(X_i)\}$,
- $F_{Q_j} = \{a \in F\colon\ V(a) \subseteq V(Q_j)\}$,
- $F_{X_i,Q_j} = \{a \in F\colon\ |V(a) \cap V(X_i)| = |V(a) \cap V(Q_j)| = 1\}$,
- $F_{X_i,X_{i'}} = \{a \in F\colon\ |V(a) \cap V(X_i)| = |V(a) \cap V(X_{i'})| = 1\}$ with $i \neq i'$,
- $F_{Q_j,Q_{j'}} = \{a \in F\colon\ |V(a) \cap V(Q_j)| = |V(a) \cap V(Q_{j'})| = 1\}$ with $j \neq j'$,
- $F_{X,X} = \bigcup_{i,i' \in [\![1,n]\!], i \neq i'} F_{X_i,X_{i'}}$,
- $F_{Q,Q} = \bigcup_{j,j' \in [\![1,m+1]\!], j \neq j'} F_{Q_j,Q_{j'}}$,
- $F_{X,Q_j} = \bigcup_{i \in [\![1,n]\!]} F_{X_i,Q_j}$.

By Lemma 29.1, there is an arc-disjoint triangle packing $\triangle^x$ and $\triangle^q$ using exclusively arcs joining gadgets. Therefore, we have $|F_{X,X}| \geq |\triangle^x|$ and $|F_{Q,Q}| \geq |\triangle^q|$, that is $|F_{X,X}| \geq 6n(n-1)$ and $|F_{Q,Q}| \geq 3m(m+1)/2$. We also have to make $Q_{m+1}$ acyclic, then $|F_{Q_{m+1}}| \geq 1$.

For $j \in [\![1,m]\!]$, let $\lambda(j)$ be the size of the clause $c_j$. By construction, $\lambda(j)$ is also the number of out-going backward arcs of $Q_j$. We denote by $L$ the set $\{j \in [\![1,m]\!]\colon\ |F_{X,Q_j} \cup F_{Q_j}| \geq \lambda(j)\}$, and $S$ the set $[\![1,m]\!] \setminus L$. Informally, $L$ contains the indices of clauses which spend a large amount of their "outer-arcs" in $F$, that is more than the $\lambda(j)$ needed—while the clauses with index in $S$ only use a small part of it.

Let us now prove that for any $j \in S$, we have $|F_{Q_j}| \geq \lambda(j) - 1$. Suppose it is not the case, that is $|F_{Q_j}| < \lambda(j) - 1$. For $p \in [\![1,\lambda(j)]\!]$, let be $\tilde{u}_p$ be the vertices which are the head of a backward arc from $c_j^3$.

If $\lambda(j) = 2$, then $|F_{X,Q_j}| = 0$ or $|F_{X,Q_j}| = 1$. In both cases, there is a vertex $v_1$ with none of its arcs from and to $Q_j$ in $F$; it is easy to find a triangle (for example $(v_1, c_j^1, c_j^3)$), contradicting the fact that $F$ is a feedback arc set of $T_{g(\mathcal{F})}$.

Thus, we have $\lambda(j) = 3$. It implies that $|F_{Q_j}| = 0$ or $|F_{Q_j}| = 1$. Let us consider the latter first. In that case, it means that $|F_{X,Q_j}| = 0$ or $|F_{X,Q_j}| = 1$. In both cases, there is also a vertex $v_1$ with none of its arcs from and to $Q_j$ in $F$; we can find a triangle which does not use the only arc of $F_{Q_j}$, contradicting the fact that $F$ is a feedback arc set.
Therefore, we have $|F_{Q_j}| = 0$, which implies that $|F_{X,Q_j}| \in [\![0,2]\!]$. Once again, in all cases we can find a vertex $v_1$, with none of its arcs from and to $Q_j$ in $F$; the $(v_1, c_j^1, c_j^3)$ has none of its arcs in $F$, contradicting the fact it is a feedback arc set of $T_{g(\mathcal{F})}$.

By definition of $S$, we have $|F_{X,Q_j} \cup F_{Q_j}| \leq \lambda(j) - 1$. Now, since we proved that for any $j \in [\![1,m]\!]$, we have $|F_{Q_j}| \geq \lambda(j) - 1$, it directly implies that $|F_{Q_j}| = \lambda(j) - 1$ and $|F_{X,Q_j}| = 0$.

Let us now prove that $|S| \leq 1$. If it is not the case, let $j_1$ and $j_2$ be two distinct indices in $S$. We also define $a_1$ and $a_2$, two backward arcs such that there exists $i_1$ and $i_2$ with $t(a_1) \in Q_{j_1}$, $h(a_1) \in X_{i_1}$, $t(a_2) \in Q_{j_2}$ and $h(a_2) \in X_{i_2}$. In other words, $h(a_1)$ and $h(a_2)$ are the vertices corresponding to the variables $x_{i_1}$ and $x_{i_2}$ appearing in the clauses $c_{j_1}$ and $c_{j_2}$, respectively. Notice than one can

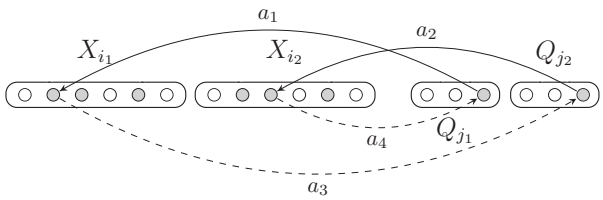



**Figure 5.8** – Example where $i_1 < i_2$. We can close the two backward arcs $a_1$ and $a_2$ with the forward arcs $a_3$ and $a_4$ into a cycle.

have $i_1 = i_2$, but by construction we necessarily have $h(a_1) \neq h(a_2)$. Furthermore, $h(a_1)$ and $h(a_2)$ do not have another backward arc coming from $Q$, except the dummy arcs. It implies that the arcs $a_3 = (h(a_1), t(a_2))$ and $a_4 = (h(a_2), t(a_1))$ are forward arcs of our ordering. Finally, as $|F_{X,Q_{j_1}}| = 0$ and $|F_{X,Q_{j_2}}| = 0$, $(t(a_1), h(a_1), t(a_2), h(a_2), t(a_1))$ is a cycle using no arc of $F$, a contradiction; the set $S$ contains at most one index. See Figure 5.8 for an example of this case.

In the following, let $L'$ be the set $\{i \in [\![1, n]\!] \colon \exists a \in \overleftarrow{A}(T_{g(\mathcal{F})}), \exists j \in S$ with $h(a) \in X_i, t(a) \in Q_j, \}$. As $|S| \leq 1$, notice that if $S = \{j_0\}$, then $|L'| = \lambda(j_0)$, and $L' = \emptyset$ otherwise. We also define $S' = [\![1, n]\!] - L'$. Furthermore, for any $i \in [\![1, n]\!]$, let $\overleftarrow{A}_{X_i,Q_{m+1}}$ be the set of the backward arcs from the dummy triangle to the variable gadget $X_i$; by construction, it means that we have $\overleftarrow{A}_{X_i,Q_{m+1}} = \{(d^p, \overline{\ell_i^i}) \colon p \in [\![1, 3]\!]\}$. Finally, for any $v \in \{\ell_i^1, \ell_i^2, \overline{\ell_i}\}$, we define $A_{v,X_i}$ as the set of out-going arcs from $v$ with its head in $X_i$. Then, we have $|A_{\ell_i^1,X_i}| = 2$, $|A_{\ell_i^2,X_i}| = 1$ and $|A_{\overline{\ell_i},X_i}| = 4$.

Let us now prove that for any $i \in S'$, $|F_{X_i} \cup F_{X_i,Q_{m+1}}| \geq 5$. Notice that $F_{X_i}$ must be a feedback arc set of $X_i$. Then, if $A_{\overline{\ell_i},X_i} \subseteq F_{X_i}$, then as $A_{\overline{\ell_i},X_i}$ is not a feedback arc set of $X_i$, it implies that $F_{X_i}$ must contain another arc, proving that $|F_{X_i}| \geq 5$. Otherwise, notice that each out-neighbor of $\overline{\ell_i}$ in $X_i$, denoted $v$, and each dummy vertex $d^p$ create the triangle $(\overline{\ell_i}, v, d^p)$. Then $\overleftarrow{A}_{X_i,Q_{m+1}} \subseteq F_{X_i,Q_{m+1}}$. Now, as $\mathrm{fas}(X_i) = 2$, we get $|F_{X_i} \cup F_{X_i,Q_{m+1}}| \geq 5$.

Similarly, let us finally prove that for any $i \in L'$, $|F_{X_i} \cup F_{X_i,Q_{m+1}}| \geq 6$. Let us consider the case where $L' \neq \emptyset$, because the other case is a vacuously true. By definition, there is an arc $a \in \overleftarrow{A}(T_{g(\mathcal{F})})$ such that $h(a) \in X_i$ and $t(a) \in Q_{j_0}$, with $j_0$ the unique index of $S$. Notice that $h(a) \in \{\ell_i^1, \ell_i^2, \overline{\ell_i}\}$ by construction. Since, for any out-neighbor of $h(a)$ in $X_i$, denoted $v$, we have the triangle $(h(a), v, c_{j_0}^3)$, then we must have $A_{h(a),X_i} \subseteq F_{X_i}$. We now consider the three possible cases:

If $h(a) = \overline{\ell_i}$, then it is easy to see that we still need at least two other arcs of $X_i$ to remove all the remaining arc-disjoint cycles inside the variable gadget. For example, one can point out the two following arc-disjoint triangles $(r_i, \ell_i^1, s_i)$ and $(\ell_i^1, \ell_i^2, t_i)$. So we get $|F_{X_i}| \geq 6$.

In the case where $h(a) = \ell_i^1$, then if moreover we also have $A_{\overline{\ell_i},X_i} \subseteq F_{X_i}$, it directly implies $|F_{X_i} \cup F_{X_i,Q_{m+1}}| \geq 6$. Otherwise, as previously, we must have $\overleftarrow{A}_{X_i,Q_{m+1}} \subseteq F$. But as $A_{\overline{\ell_i},X_i}$ is not a feedback arc set of $X_i$, then we need at least another arc, implying $|F_{X_i} \cup F_{X_i,Q_{m+1}}| \geq 6$.

Finally, we are in the case where $h(a) = \ell_i^2$. If we also have $A_{\overline{\ell_i},X_i} \subseteq F_{X_i}$, then one can notice that $A_{\ell_i^2 X_i} \cup A_{\overline{\ell_i},X_i}$ is not a feedback arc set of $X_i$—the

triangle $(r_i, \ell_i^1, s_i)$ for example. Then we need at least another arc, implying $|F_{X_i} \cup F_{X_i,Q_{m+1}}| \geq 6$. Otherwise, we have one more time $A_{\overline{\ell_i},X_i} \subseteq F_{X_i}$, but as we can find two arc-disjoint cycles which do not use arcs of $A_{\ell^2 X_i}$, we need at least two additional arcs in $F_{X_i}$. So we get $|F_{X_i} \cup F_{X_i,Q_{m+1}}| \geq 6$.

Putting all the pieces together, we get:

$$\begin{aligned}
|F| &= |F_{X,X}| + |F_{Q,Q}| + |F_{Q_{m+1}}| + \sum_{j\in L}(|F_{X,Q_j} \cup F_{Q_j}|) + \sum_{j\in S}(|F_{X,Q_j} \cup F_{Q_j}|) \\
&\quad + \sum_{i\in S'}(|F_{X_i} \cup F_{X_i,Q_{m+1}}|) + \sum_{i\in L'}(|F_{X_i} \cup F_{X_i,Q_{m+1}}|) \\
&\geq 6n(n-1) + \frac{3m(m+1)}{2} + 1 + \sum_{j\in L}\lambda(j) + \sum_{j\in S}(\lambda(j)-1) + 5|S'| + 6|L'| \\
&\geq 6n(n-1) + \frac{3m(m+1)}{2} + 1 + \sum_{j\in [\![1,m]\!]}\lambda(j) + 5n \\
&= k_{g(\mathcal{F})}
\end{aligned}$$

This concludes the proof of the lemma, that is $\mathrm{fas}(T_{g(\mathcal{F})}) \leq k_{g(\mathcal{F})}$.

□

Then, by combining Claim 32.1 and Lemma 32.1, we get the NP-hardness of Tight-ATT and Tight-ACT.

**Theorem 33.** *The problems* Tight Arc-Disjoint Cycle Packing in Tournament *and* Tight Arc-Disjoint Cycle Packing in Tournament *are* NP*-hard.*

## 5.3 An FPT-algorithm and Linear Kernel for the Arc-disjoint Triangle Packing in Tournaments Problem

In this section, we focus on the parameterized version of the ATT problem, and we provide an FPT-algorithm for it as well as a kernel with a linear number of vertices.

First, using the classical technique of color coding [8] for packing subgraphs of bounded size, we obtain the following result:

**Theorem 34.** *There exists an algorithm with running time $O^*(2^k)$ to solve $k$-*Arc-Disjoint Triangle Packing in Tournament.

*Proof.* Let $(T, k)$ be an instance of $k$-ATT, and denote by $n$ and by $m$ its number of vertices and arcs, respectively. Moreover, we label the arcs of $T$ by $\{a_1, \ldots, a_m\}$.

If $(T, k)$ is a positive instance, then it admits an arc-disjoint triangle packing with $k$ triangles, which thus contains $3k$ arcs. We consider a $3k$-perfect family of hash functions from $\{a_1, \ldots, a_m\}$ to $\{1, \ldots, 3k\}$. In other words, this is a set

of colorings of the arcs of $T$ using $3k$ colors, and such that for every subset of $\{a_1, \ldots, a_m\}$ of size $3k$, there exists one of these colorings that colors the elements of this subset with exactly $3k$ different colors—each color is used once. Schmidt and Siegal [134] explicitly provide such a family of colorings of size $2^{O(k)} \log^2 m$ which can be computed within $O^*(2^{O(k)})$ time.

Now for each coloring, we use dynamic programming to obtain an arc-disjoint triangle packing of size $k$ whose all arcs use different colors. One can easily notice that the instance $(T, k)$ is positive if and only if at least one of the colorings contains such a packing. So, given a coloring $c$, we first compute for every set of three distinct colors $\{i_1, i_2, i_3\}$ if the arcs colored with $i_1$, $i_2$ or $i_3$ induce a triangle using all the three different colors. Then, we recursively repeat this process: for every set $X$ of $3(p+1)$ colors with $p \in [\![1, k-1]\!]$, we check if the arcs colored with the colors of $X$ induce $p+1$ arc-disjoint triangles whose arcs use all the colors of $X$. To do so, we check for every subset $\{i_1, i_2, i_3\}$ of $X$ if there exists a triangle using colors $i_1$, $i_2$ and $i_3$ and a collection of $p$ arc-disjoint triangles whose arcs use all the colors of $X \setminus \{i_1, i_2, i_3\}$. It is clear that $X$ being fixed we can find this collection of triangles in time $O(p^3)$ that is $O(k^3)$.

Finally, we answer the color-version of the problem in time $O^*(k^3 . 2^k) \subseteq O^*(2^k)$. As we have $O^*(2^k)$ different colorings, we obtain the announced running time. □

Moreover, we obtain the following kernelization algorithm.

**Theorem 35.** *$k$-ATT admits a kernel with $O(k)$ vertices.*

*Proof.* Let $(T, k)$ be an instance of $k$-ATT. We first start by greedily computing a maximal packing $X$ of arc-disjoint triangles of $T$. Let $V_X$ and $A_X$ be the set of vertices and arcs induced by the triangles of $X$, respectively. Moreover, let $U$ be the set of remaining vertices of $T$, that is $U = V(T) - V_X$. If $|X| \geq k$, then $(T, k)$ is trivially a positive instance of $k$-ATT. Then, we may assume that $|X| < k$, implying that $|V_X| < 3k$. Moreover, one can easily notice that $T[U]$ is acyclic and that all the triangles of $T$ with its some of its vertices in $V_X$ and the remaining in $U$ have necessarily exactly one vertex in $U$—a triangle with one vertex in $V_X$ and two in $U$ would contradict the maximality of $X$.

Let us construct the following undirected bipartite graph $B$. We set its vertex set $V(B)$ by $A_X \cup U$ and its edge set $E(B)$ by $\{\{a, u\} \colon a \in A_X, u \in U, (h(a), u), (u, t(a)) \in A(T)\}$. Informally, the edges of $B$ can be seen as the triangles of $T$ which are "joining" $V_X$ and $U$. Thus, by defining $M$ a maximum matching of $B$, it corresponds to an arc-disjoint packing of triangles of $T$. In the following, we denote by $A'$ (resp. $U'$) be the vertices of $A_X$ (resp. $U$) covered by $M$. Moreover, we define $\overline{A'} = A' \setminus A_X$ and $\overline{U'} = U' \setminus U$.

We now prove that $(T[V_X \cup U'], k)$ is a linear vertex-kernel of $(T, k)$. Let $\triangle$ be a maximum sized triangle packing that minimizes the number of vertices of $\overline{U'}$ belonging to a triangle of $\triangle$. By previous remarks, we can partition $\triangle$ into $\triangle_X \cup F$ where $\triangle_X$ are the triangles of $\triangle$ included in $T[V_X]$ and $F$ are the triangles of $\triangle$ containing one vertex of $U$ and two vertices of $V_X$. It is clear that $F$ corresponds to a union of vertex-disjoint stars of $B$ with centers in $U$. Denote

by $U[F]$ the vertices of $U$ which belong to a triangle of $F$. If $U[F] \subseteq U'$ then $(T[V_X \cup U'], k)$ is immediately a kernel.

Therefore, let $x_0$ be a vertex in $U[F] \cap \overline{U'}$. Given a subset $S \subseteq U$, we define $N_F(S)$ as the set $\{a \in A_X \colon \exists s \in S \text{ with } (t(a), h(a), s) \in F \text{ and } \{a, s\} \notin M\}$. Furthermore, given a subset $S \subseteq A_X$, we define the set $N_M(S)$ by $\{u \in U \colon \exists a \in A_X \text{ with } \{a, u\} \in M\}$ We build a tree rooted in $x_0$ with edges alternating between $F$ and $M$. To do this, let $H_0 = \{x_0\}$ and construct recursively the sets $H_{i+1}$ such that

$$H_{i+1} = \begin{cases} N_F(H_i) \text{ if } i \text{ is even,} \\ N_M(H_i) \text{ if } i \text{ is odd,} \end{cases}$$

Notice that $H_i \subseteq U$ when $i$ is even and that $H_i \subseteq A_X$ when $i$ is odd. Furthermore, all the $H_i$ are distinct as $F$ is a union of disjoint stars and $M$ a matching in $B$. For $i \geq 1$ we call $L_i$ the set of edges between $H_i$ and $H_{i-1}$. Now we define the graph $G$ such that $V(G) = \bigcup_i L_i$ and $E(G) = \bigcup_i L_i$. As $L_i$ is a matching (if $i$ is even) or a union of vertex-disjoint stars with centers in $H_{i-1}$ (if $i$ is odd), it is clear that $G$ is a tree.

For $i$ being odd, every vertex of $H_i$ is incident to an edge of $M$ otherwise $B$ would contain an augmenting path for $M$, a contradiction. So every leaf of $G$ is in $U$ and incident to an edge of $M$ in $G$, and $G$ contains as many edges of $M$ than edges of $F$. Now, for every arc $a \in A_X \cap V(G)$, we replace the triangle of $\triangle$ containing $a$ and corresponding to an edge of $F$ by the triangle $(t(a), h(a), u)$ where $\{a, u\} \in M$ (and $\{a, u\}$ is an edge of $G$).

This operation leads to another collection of arc-disjoint triangles with the same size as $\triangle$, but containing a strictly smaller number of vertices in $\overline{U'}$, yielding a contradiction.

Finally $V_X \cup U'$ can be computed in polynomial time and we also have $|V_X \cup U'| \leq |V_X| + |M| \leq 2|V_X| \leq 6k$; it concludes the prof that the number of vertices in the kernel is $O(k)$. □

## 5.4 Concluding Remarks

In this chapter, we proved the NP-hardness of both Arc-Disjoint Cycle Packing in Tournament and Arc-Disjoint Triangle Packing in Tournament, and their tight versions where the input tournament $T$ is provided with a linear representation whose backward arcs are exactly the minimum feedback arc set of $T$. Notice that these reductions do not produce sparse tournaments (consider for example the vertices of the dummy triangle). In fact, we prove in Chapter 6 that Arc-Disjoint Cycle Packing in Tournament and Arc-Disjoint Triangle Packing in Tournament admit a polynomial-time algorithm in sparse tournaments.

From the parameterized point of view, we provided a kernel of $O(k)$ vertices and a $O^*(2^k)$ running-time FPT-algorithm to solve ATT. Recently, Krithika *et al.* [101] showed that ACT can be solved in $O^*(2^{O(k \log k)})$ running-time and provided a linear kernel for ACT. Therefore, the following question is natural:

**Question 2.** *Do* Arc-Disjoint Triangle Packing in Tournament *or* Arc-Disjoint Cycle Packing in Tournament *admit a* $O^*(2^{\sqrt{k}})$ *running-time algorithm?*

Only few problems are known to admit a $O^*(2^{\sqrt{k}})$ when parameterized by the standard parameter $k$ [129]. Alon *et al.* [7] and Feige [63] proved that the parameterized version of FAST is one of them. To the best of our knowledge, outside bidimensionality theory, no packing problems are known to admit such an FPT running-time algorithms, and maybe ATT (or even ACT) could be a candidate for this and so deserve some attention, especially in the light of the $2^{o(\sqrt{k})}$ lower bound of Corollary 2.

# Chapter 6

# Packing of Cycles in Sparse Tournaments

This chapter corresponds to joint work with Stéphane Bessy and Marin Bougeret presented to *ESA 2017* [21] and to *MFCS 2019* [20, 22]. In chapters 4 and 5, we obtained several negative results on the packing of cycles and/or triangles in tournaments. We focus here on a subclass of tournaments, called sparse, admitting a feedback arc set which is a matching.

## 6.1 Introduction

### 6.1.1 Motivations for Sparse Tournaments

Surprisingly, despite its relatively simple structure, we proved in Theorem 24 that computing a maximum-sized packing of triangles in sparse tournaments is NP-hard. This motivates the interest of the packing of cycles in such tournaments. Another reason to study sparse tournaments is that we can decide in polynomial time if a tournament is sparse. Additionally, if it is the case, we can provide a linear representation where the backward arcs form a matching.

**Lemma 35.1.** *In polynomial time, we can decide if a tournament $T$ is sparse or not, and if so, to give a linear representation $(\sigma(T), \overleftarrow{A}(T))$ where $\overleftarrow{A}(T)$ is a matching.*

*Proof.* If a tournament $T$ is sparse, then let us prove that we can chose the first vertex (or possible vertices) of a linear representation $\sigma(T)$ of $T$ where $\overleftarrow{A}(T)$ is a matching.

If $T$ has a vertex $v$ of such that $d^-(v) = 0$, then necessarily $v$ is the first or the second vertex of the ordering $\sigma(T)$, and we can always suppose that $v$ is the first vertex of $\sigma(T)$. Otherwise, we look at the set of vertices $X$ of $T$ with in-degree 1. As $T$ is a tournament, we have $|X| \leq 3$. First, if $X$ is empty, then $T$ is not a sparse tournament, contradicting our first hypothesis. Now, if $|X| = 1$, then the only element of $X$ must be the first vertex of $\sigma(T)$. If $|X| = 2$ with $X = \{u, v\}$ for some vertices $u$ and $v$ of $V(T)$ and such that $(u, v)$ is an arc of $T$, then $u$ must be the first element of the ordering $\sigma(T)$, and $v$ its second element.

Finally, if $|X| = 3$ with $X = \{u, v, w\}$ for some vertices $u$, $v$, and $w$, then $T[X]$ is necessarily a triangle, and must be placed at the beginning of $\sigma(T)$.

By repeating inductively these arguments, we obtain a polynomial time algorithm that returns the linear representation $\sigma(T)$ of $T$ such that $\overleftarrow{A}(T)$ is a matching if $T$ is sparse, or a certificate that $T$ is not. □

### 6.1.2 Our contributions and Organization of the Chapter

In this chapter, we start by focusing on the vertex-disjoint packing of triangles in sparse tournaments in section 6.2. Namely, we give in subsection 6.2.1 an approximation algorithm for sparse tournaments with large minspan, then we give in subsection 6.2.2 a linear-vertex kernel for DTT, answering positively to Question 1 in the case of sparse tournaments.

Then, we prove section 6.3 that Arc-Disjoint Cycle Packing in Sparse Tournament and Arc-Disjoint Triangle Packing in Sparse Tournament are both polynomial problems.

Finally, we give in section 6.4 some concluding remarks.

## 6.2 Approximation and Kernelization Algorithms for the Vertex-Disjoint Packing of Triangles in Sparse Tournaments Problem

### 6.2.1 An $(1+\frac{6}{c-1})$-approximation Algorithm when Backward Arcs Have Large minspan

In all this subsection, we consider an instance $T$ of Max DTS with a linear representation $(\sigma(T), \overleftarrow{A}(T))$, and such that $\text{minspan}(\sigma(T)) \geq c$, for some constant $c$. The motivation for studying the approximability of this case comes from the problem Max $c$-SAT, for which the approximability becomes easier when $c$ grows: the derandomized uniform assignment provides a $\frac{2^c}{2^c-1}$-approximation algorithm [91]. Somehow, one could claim that Max-$c$-SAT becomes easy to approximate for large $c$ as there more ways to satisfy a given clause.

For tournament admitting an ordering with a minspan at least $c$, the intuition is similar: there are $c-1$ vertices available to create a triangle with a given backward arc. Furthermore, since $\overleftarrow{A}(T)$ form a matching, $(u, v, w)$ is triangle if and only if $(w, u) \in \overleftarrow{A}(T)$ and $u < v < w$. In other words, the $c-1$ vertices are the the only possible vertices one can pick to use the backward arc. Thus, our objective is to find a polynomial approximation algorithm whose ratio tends to the optimal when $c$ increases.

#### 6.2.1.1 The Approximation Algorithm

We now define our approximation algorithm $\Phi$. First, given our input tournament $T$ with its linear representation $(\sigma(T), \overleftarrow{A}(T))$, we first create the undirected bipartite graph $B$. We set $V(B) = V^* \cup \overleftarrow{A}(T)$, where $V^*$ is the set of vertices of

**Algorithm 1** Approximation algorithm for DTS with minspan at least $c$

```
procedure Φ(T)
    V* ← vertices of T which are not endpoint of backward arcs in ←A(T);
    B ← bipartite undirected graph with V(B) ← (V* ∪ ←A(T)) and edge set
            E(B) ← {{v, a}: v ∈ V*, a ∈ ←A(T), (h(a), v, t(a)) triangle of T};
                                                                    ▷ Phase 1
    M ← maximum matching of V*;
    △1 ← vertex-disjoint triangles corresponding to M;
    T2 ← T \ V(△1);
                                                                    ▷ Phase 2
    △2 ← {};
    for all {a1, a2, a3} arcs of ←A(T2) do
        if a1, a2 and a3 can be packed into two triangles δ1 and δ2 then
            △2 ← △2 ∪ {δ1, δ2};
            V(T2) ← V(T2) \ (V(δ1) ∪ V(δ2));
        end if
    end for
    return △Φ := △1 ∪ △2;
end procedure
```

$T$ which are neither head nor tail of a backward arc. Then, given $a \in \overleftarrow{A}(T)$ and $v \in V^*$, we have the edge $\{a, v\}$ if $(h(a), v, t(a))$ is a triangle in $T$.

In Phase 1, $\Phi$ computes a maximum matching $M$ in $B$. For every edge $\{a, u\}$ of $M$, we create the corresponding triangle. Let $\triangle_1$ be the set of the triangles we just created. Notice that since $M$ is a matching, the triangles of $\triangle_1$ are vertex-disjoint.

Let us now turn to Phase 2. Let $T_2$ be the tournament obtained from $T$ by removing all vertices of $V(\triangle_1)$. Let $(\sigma(T_2), \overleftarrow{A}(T_2))$ be the linear representation of $T_2$ obtained by such a operation. We say that three distinct backward edges $\{a_1, a_2, a_3\} \subseteq \overleftarrow{A}(T_2)$ can be packed into triangles $\delta_1$ and $\delta_2$ if $V(\{\delta_1, \delta_2\}) = V(\{a_1, a_2, a_3\})$, and $\delta_1$ and $\delta_2$ are vertex-disjoint. For example, if we have the ordering: $h(a_1) < h(a_2) < t(a_1) < h(a_3) < t(a_2) < t(a_3)$, then $\{a_1, a_2, a_3\}$ can be packed into $(h(a_1), h(a_2), t(a_1))$ and $(h(a_3), t(a_2), t(a_3))$. On contrary, if $h(a_1) < h(a_2) < t(a_2) < h(a_3) < t(a_3) < t(a_1)$, then $\{a_1, a_2, a_3\}$ cannot be packed into two triangles. In Phase 2, while it is possible, $\Phi$ finds a triplet of arcs of $\overleftarrow{A}(T_2)$ that can be packed into triangles, then create the two corresponding triangles, and remove their vertices of $T_2$—proving that the triangles are vertex-disjoint. Let $\triangle_2$ be the set of triangles created in this phase, and let $\triangle_\Phi$ be the vertex-disjoint packing of triangles corresponding to $\triangle_1 \cup \triangle_2$.

Since we constructed the triangles of $\triangle_1$ and $\triangle_2$ by associating backward arcs with vertices or other backward arcs, it is easy to notice that no backward arc has one endpoint in $V(\triangle_\Phi)$ and the other outside $V(\triangle_\Phi)$. So we can partition the backward arcs of $T$ into $\overleftarrow{A}_0 \cup \overleftarrow{A}_1 \cup \overleftarrow{A}_2$, where $\overleftarrow{A}_1$ and $\overleftarrow{A}_2$ are the sets of arcs used in Phase 1 and Phase 2, respectively, and where $\overleftarrow{A}_0$ are the remaining arcs. Let $\overleftarrow{A}_\Phi$ be the set of backward arcs of $\triangle_\Phi$, that is $\overleftarrow{A}_1 \cup \overleftarrow{A}_2$, and for $i \in [\![0, 2]\!]$ let $m_i$ be the size of $\overleftarrow{A}_i$. Furthermore, we denote by $m$ the total number of backward

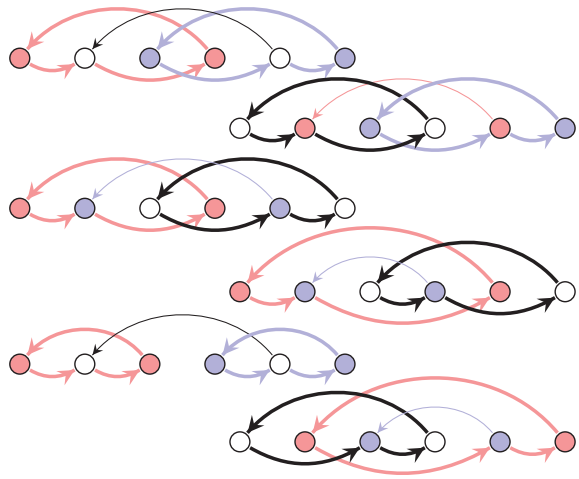


**Figure 6.1** – In each subfigure, the black backward arc correspond to the arc $inc(b_i^0)$ and the colored ones to the arcs with endpoints in $Z$. We give all the possible permutations of the head and tails of these three arcs. In each case, the thick arcs show how the three arcs can be packed into two triangles.

arcs, namely $m_0 + m_1 + m_2$ and $m_\Phi$ the backward arcs (entirely) consumed by $\Phi$, that is $m_1 + m_2$.

#### 6.2.1.2 Proof of the Ratio

To prove the $(1+\frac{6}{c-1})$ desired approximation ratio, we will first prove in Corollary 3 that the algorithm $\Phi$ uses "almost" all the backward arcs ($m_\Phi \geq (1-\varepsilon(c))m$), and in Theorem 36 that the number of triangles made with these arcs is "optimal".

In the following, we refer by $\mathrm{s}'(a)$ of an arc $a$ refers to the vertices in the initial linear representation of $T$—not only the vertices remaining in $T_2$. We also denote by $b_1^0, \ldots, b_{|m_0|}^0$ the backward arcs of $\overleftarrow{A}_0$, that is the remaining backward arcs after the execution of the algorithm $\Phi$. Given $b_i^0 \in \overleftarrow{A}_0$, let $inc(b_i^0)$ be the the backward arc $b_j^0$ of $\overleftarrow{A}_0$ such that $\mathrm{s}'(b_j^0) \subseteq \mathrm{s}'(b_i^0)$ and there is no other backward arc $b_k^0$ such that $\mathrm{s}'(b_k^0) \subseteq \mathrm{s}'(b_j^0)$. In other words, $inc(b_i^0)$ is the "smallest" backward arc not taken by $\Phi$ in the span of $b_i^0$. Notice that we may have $inc(b_i^0) = b_i^0$. Furthermore, let $Z$ be the set of endpoints not taken by any phases of $\Phi$ in the span of $inc(b_i^0)$, that is $V(\overleftarrow{A}_0) \cap \mathrm{s}'(inc(b_i^0))$. We can first easily give an upper bound of the size of $Z$.

**Claim 35.1.** *We have* $|Z| \leq 1$.

*Proof.* Suppose that $|Z| \geq 2$. First, if $u \in V^* \cup Z$, then $(h(inc(b_i^0)), u, t(inc(b_i^0)))$ is a triangle contradicting the maximality of the matching in Phase 1. Therefore, all the vertices of $Z$ are an endpoint of some backward arc. Furthermore cannot have both endpoints in $\mathrm{s}'(inc(b_i^0))$ without contradicting the definition of $inc(b_1^0)$. So there are at least two arcs with exactly one of their endpoints in $Z$. Now, we can notice by considering all the subcases depicted in Figure 6.1 that we always

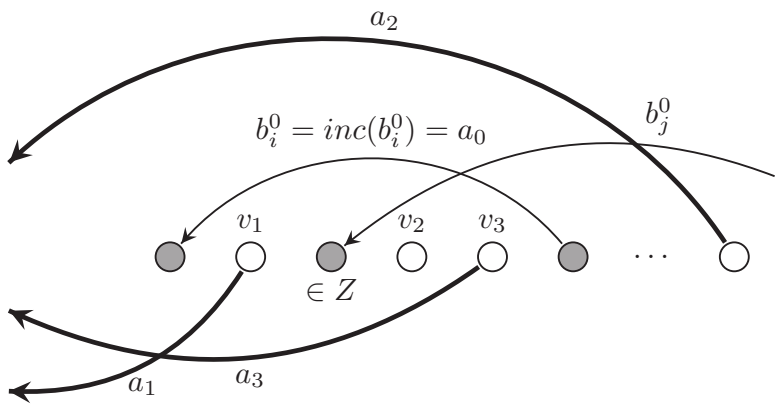



**Figure 6.2** – In this example, the vertices taken by the algorithm $\Phi$ are in white, and the not taken in gray. In this case, we have $\overleftarrow{A}_i = \{a_p \colon p \in [\![1,3]\!]\}$. Indeed, for each $p \in [\![1,3]\!]$, the vertex $v_p$ in the span of $a_0$ (and not in $Z$) brings $a_p$ in $\overleftarrow{A}_i$. In particular, the vertex $v_2$ is in $V^*$ and was used with the backward arc $a_2$ to create a triangle in Phase 1.

can pack these two arcs and $inc(b_i^0)$ into two triangles, contradicting the Phase 2 of $\Phi$. □

We now want to associate for every arc $b_i^0$ a set $\overleftarrow{A}_i$ of backward arcs used by the algorithm during Phase 1 or Phase 2. To do so, if there is a backward arc $a \in \overleftarrow{A}_\Phi$ such that $\mathrm{s}'(a) \subseteq \mathrm{s}'(inc(b_i^0))$, then let set $a_0 = a$, and set $a_0 = inc(b_i^0)$ otherwise. Now, we can associate for any $inc(b_i^0)$ the set of backward arcs $\overleftarrow{A}_i$ as follows: for any vertex $v$ in $\mathrm{s}'(a_0) \setminus Z$, we add to $\overleftarrow{A}_i$ the backward arc of $\overleftarrow{A}_\Phi$ corresponding to the triangle $v$ belongs to. Such backward necessarily exist, since by definition of $Z$, $v$ is a vertex taken in Phase 1 or Phase 2 in the algorithm $\Phi$.

In other words, a backward arc $a$ is added to $\overleftarrow{A}_i$ if:

- either a vertex in the span of backward arc $b_i^0$ (or more precisely $inc(b_i^0)$) created a triangle with $a$ in Phase 1 of $\Phi$,
- or $a$ has one of its endpoints in the span of $inc(b_i^0)$ and was taken by the algorithm in Phase 2.

See Figure 6.2 for an example of the $\overleftarrow{A}_i$ for a given arc.

**Claim 35.2.** *We have* $|\overleftarrow{A}_i| \geq c - 1$.

*Proof.* Since we added exactly one backward arc in $\overleftarrow{A}_i$ for each vertex in $\mathrm{s}'(a_0) \setminus Z$, we have $|\overleftarrow{A}_i| = |\mathrm{s}'(a_0) \setminus Z|$. By hypothesis, we have $\mathrm{minspan}(\sigma(T)) \geq c$, so $\mathrm{s}'(a_0) \geq c$. By Claim 35.1, we have $|Z| \leq 1$ implying $|\overleftarrow{A}_i| \geq c - 1$. □

We now define the bipartite undirected graph $B'$ with vertex set $\overleftarrow{A}_0 \cup \overleftarrow{A}_\Phi$ and with an edge between $b_i^0 \in \overleftarrow{A}_0$ and $a \in \overleftarrow{A}_\Phi$ if $a \in \overleftarrow{A}_i$. Furthermore, let $m'$ be the number of edges of $B'$. We can now prove the following claim:

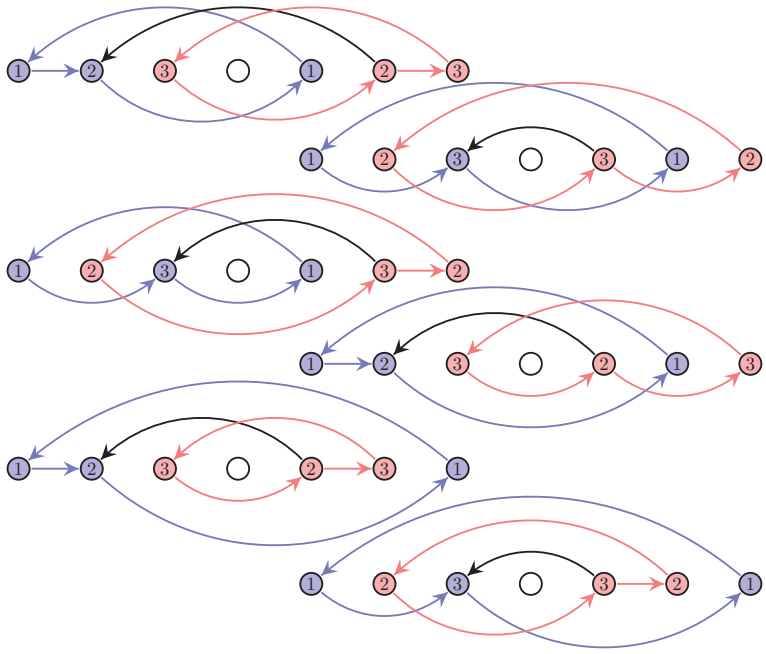


**Figure 6.3** – In each subfigure, the backward arcs connect the vertices with same label. The ordering of the vertices after the black vertex form all the 3! possible permutations of arcs with the black vertex in their spans. In each case, we can pack the three arcs into two triangles.

**Claim 35.3.** *We have* $m_0(c-1) \leq m' \leq 6m_\Phi$.

*Proof.* Given a vertex $v$ in the final vertex-disjoint packing of triangles $\triangle_\Phi$ returned by $\Phi$, we define by $w(v)$ as the cardinal of $\{b_i^0 \in \overleftarrow{A}_0 \colon v \in s'(b_i^0)\}$. Informally, it corresponds to the number of not-taken backward arcs for which $v$ is in the span. Observe that $w(v) \leq 2$ since otherwise any triplet of arcs containing $v$ in their respective spans could be packed into two triangles; the six cases to check corresponding to the 3! possible orderings of the tails of these three arcs are depicted in Figure 6.3.

Given a backward arc $a$ in $\overleftarrow{A}_1$, let $V'(a)$ be the three vertices of the triangle of $\triangle_\Phi$ using the arc $a$. Similarly, for any $a \in \overleftarrow{A}_2$, let $V'(a) = V(a)$. Thus, we have for any $a \in \overleftarrow{A}_\Phi$, $V'(a) \leq 3$.

Furthermore, given a backward arc $a \in \overleftarrow{A}_\Phi$, we define $q(a)$ as the cardinal of $\{\overleftarrow{A}_i \colon a \in \overleftarrow{A}_i\}$. Informally, $q(a)$ corresponds to the number of times the arc $a$ can be in a $\overleftarrow{A}_i$ for some $b_i^0$. Observe that by definition of the $\overleftarrow{A}_i$, a backward arc $a$ of $\overleftarrow{A}_i$ implies that $inc(b_i^0)$ contributes to the value of $w(v)$ for a vertex in $V'(a)$. As in particular $w(v) \leq 2$ for any $v \in V'(a)$, this implies by pigeonhole principle that $q(a) \leq 6$ (notice that this bound is tight as depicted Figure 6.4).

Putting all the pieces together, in our bipartite graph $B'$, the previous reasoning implies that $d_{B'}(a) \leq 6$ and the Claim 35.2 implies that $d_{B'}(b_i^0) \geq c-1$. A double counting leads to the result, that is $|\overleftarrow{A}_0|(c-1) \leq m' \leq 6|\overleftarrow{A}_\Phi|$. $\square$

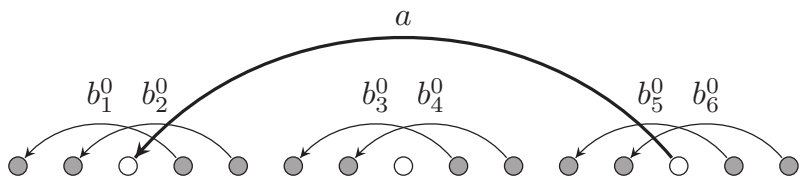



**Figure 6.4** – In this example, the backward arc $a$ is taken in Phase 1 using the vertex in the middle, so we have $V'(a) = 3$ (the three white vertices). Thus, $q(a) = 6$ because $a \in \overleftarrow{A}_i$ for any $i \in [\![1, 6]\!]$. Here, we consider that $inc(b_i^0) = b_i^0$ for any $i \in [\![1, 6]\!]$. We only depict backward arcs.

**Corollary 3.** *For any $c \geq 2$, we have $m \leq (1 + \frac{6}{c-1})m_\Phi$.*

*Proof.* This is a direct consequence of Claim 35.3, since $m_0 = m - m_\Phi$. □

We can now prove the ratio of our approximation algorithm $\Phi$.

**Theorem 36.** *For any $c \geq 2$, $\Phi$ is a polynomial $(1 + \frac{6}{c-1})$-approximation algorithm for* Maximum Disjoint Triangle Packing in Sparse Tournament*, where the given linear representation $(\sigma(T), \overleftarrow{A}(T))$ of the tournament $T$ verifies* $\text{minspan}(\sigma(T)) \geq c$.

*Proof.* Let $\triangle^{\text{opt}}$ be an optimal solution of DTS, where the given linear of the tournament $T$ has a minspan at least $c$. Let $\delta = (u, v, w)$ be a triangle of $\triangle^{\text{opt}}$. As the feedback arc set of the instance is a matching, we know that we have $(w, u) \in \overleftarrow{A}(T)$ since we cannot have a triangle with two backward arcs. Recall that $V^*$ is the set of vertices of $T$ which are not one endpoint of a backward arc. We define for any $i \in [\![1, 3]\!]$ the sets $\triangle_i^{\text{opt}} \subseteq \triangle^{\text{opt}}$ and $\overleftarrow{A}_i^{\text{opt}} \subseteq \overleftarrow{A}(T)$ as follows:

If $v \in V^*$, then we add the triangle $\delta$ to $\triangle_1^{\text{opt}}$, and $(w, u)$ to $\overleftarrow{A}_1^{\text{opt}}$.

Otherwise, let $v'$ be the other endpoint of the unique arc $a$ containing $v$. If $v' \in V(\triangle^{\text{opt}})$, let $\delta'$ be the triangle of $\triangle^{\text{opt}}$ containing $v'$. Because of the sparse property, we know that $v'$ is the second vertex of the triangle $\delta'$, or more formally that $\delta' = (u', v', w')$, for some $(u', w') \in \overleftarrow{A}(T)$. In this case, we add both $\delta$ and $\delta'$ to $\triangle_2^{\text{opt}}$, and $\{(w, u), a, (w', u')\}$ to $\overleftarrow{A}_2^{\text{opt}}$.

Finally, if $v' \notin V(\triangle^{\text{opt}})$, we add $\delta$ to $\triangle_3^{\text{opt}}$ and the arcs $\{(w, u), a\}$ to $\overleftarrow{A}_3^{\text{opt}}$. Notice that the $\{\triangle_i^{opt}\}_{i \in [\![1,3]\!]}$ form a partition of $\triangle^{\text{opt}}$, and that the $\overleftarrow{A}_i^{\text{opt}}$ have pairwise empty intersection, implying $|\overleftarrow{A}_1^{\text{opt}}| + |\overleftarrow{A}_2^{\text{opt}}| + |\overleftarrow{A}_3^{\text{opt}}| \leq m$. Notice also that as triangles of $\triangle_1^{\text{opt}}$ correspond to a matching of size $|\triangle_1^{\text{opt}}|$ in the bipartite graph defined in Phase 1 of algorithm $\Phi$, we have $|\triangle_1^{\text{opt}}| = |\overleftarrow{A}_1^*| \leq |\overleftarrow{A}_1|$, with $\overleftarrow{A}_1$ the set previously defined.

Furthermore, since we added one arc to $\overleftarrow{A}_1^{\text{opt}}$ for each triangle added to $\triangle_1^{\text{opt}}$, they have the same cardinal. Similarly, we get $|\overleftarrow{A}_2^{\text{opt}}| = 2|\triangle_2^{\text{opt}}|/3$ and $|\overleftarrow{A}_3^{\text{opt}}| = |\triangle_3^{\text{opt}}|/2$.

Putting pieces together we get:

$$\begin{aligned}|\triangle^{\text{opt}}| =&|\triangle_1^{\text{opt}}| + |\triangle_2^{\text{opt}}| + |\triangle_3^{\text{opt}}| = |\overleftarrow{A}_1^{\text{opt}}| + \frac{2}{3}|\overleftarrow{A}_2^{\text{opt}}| + \frac{1}{2}|\overleftarrow{A}_3^{\text{opt}}| \\ \leq&|\overleftarrow{A}_1^{\text{opt}}| + \frac{2}{3}(|\overleftarrow{A}_2^{\text{opt}}| + |\overleftarrow{A}_3^{\text{opt}}|) \leq |\overleftarrow{A}_1^{\text{opt}}| + \frac{2}{3}(m - |\overleftarrow{A}_1^{\text{opt}}|) \\ \leq&\frac{1}{3}|\overleftarrow{A}_1| + \frac{2}{3}m\end{aligned}$$

and

$$\begin{aligned}|\triangle_\Phi| =&|\triangle_1| + |\triangle_2| = |\overleftarrow{A}_1| + \frac{2}{3}|\overleftarrow{A}_2| \\ \geq&|\overleftarrow{A}_1| + \frac{2}{3}\big((1 + \frac{6}{c-1})^{-1}m - |\overleftarrow{A}_1|\big) \\ \geq&\frac{1}{3}|\overleftarrow{A}_1| + \frac{2}{3}\big(1 + \frac{6}{c-1}\big)^{-1}m\end{aligned}$$

This implies the desired ratio, and the theorem. □

### 6.2.2 Linear Vertex Kernel for DTT

Using the kernel in $O(m)$ vertices, where $m$ is the number of arcs in a given feedback arc set of the input described in Theorem 26, we can provide a kernel with $O(k)$ vertices for DTT restricted to sparse tournaments.

**Theorem 37.** Disjoint Triangle Packing in Sparse Tournament *parameterized by $k$ the size of the solution admits a polynomial kernel with $O(k)$ vertices.*

*Proof.* Let $I$ be an instance of the decision problem associated to DTS, and denote by $T$ the sparse tournament in the input, $(\sigma(T), \overleftarrow{A}(T))$ its linear representation, and $k$ the parameter. We say that an arc $a$ is a *consecutive backward arc* of $\sigma(T)$ if it is a backward arc $(v', v)$ of $T$ such that $v$ is just before $v'$ in the ordering $\sigma(T)$. If $T$ admits a consecutive backward arc $(v', v)$, then we can exchange $v$ and $v'$ in $\sigma(T)$. The backward arcs of the new linear representation is exactly $\overleftarrow{A}(T) \setminus (v', v)$, and so is still a matching. Hence, we can assume that $T$ does not contain any consecutive backward arc.

Now, if we have not a large number of backward arc, that is if $|\overleftarrow{A}(T)| < 5k$, then by Theorem 26 we directly have a kernel with $O(k)$ vertices. Otherwise, if $|\overleftarrow{A}(T)| \geq 5k$ we will prove that $T$ is a positive instance of DTS. Indeed, in that case we can greedily produce a family of $k$ vertex-disjoint triangles in $T$.

To do so, consider a backward arc $a$ of $\overleftarrow{A}(T)$. As it is not a consecutive backward arc, then there exists a vertex $v$ in the span of $a$, we can select the triangle $(h(a), v, t(a))$ and then remove the vertices $h(a)$, $v$ and $t(a)$ from $T$. Denote by $T'$ the resulting tournament and let $\sigma(T')$ be the order induced from $\sigma(T)$ on $T'$ obtained with such process. It is easy to see that we lose at most two backward arcs in $\sigma(T')$: the arc $a$, and a possible backward arc containing the vertex $v$ as endpoint. By removing the consecutive backward arcs we might have created when we delete the vertices $h(a)$, $v$ and $t(a)$ as previously, we can

assume that $\sigma(T')$ does not contain any consecutive backward arc and satisfies $|\overleftarrow{A}(T')| \geq |\overleftarrow{A}(T)| - 5 \geq 5(k-1)$. Finally, by repeating inductively this process, we obtain the desired family of $k$ vertex-disjoint triangles in $T$, proving that $(T, k)$ is a positive instance of DTS. □

## 6.3 Polynomial Algorithm for the Arc-Disjoint Packing of Triangles in Sparse Tournaments Problem

We now focus on the following optimization problems:

Maximum Arc-Disjoint Triangle Packing in Sparse Tournament (Max ATS)
**Input:** A sparse tournament $T$.
**Result:** A collection $\triangle$ of arc-disjoint triangles of $T$.
**Optimization:** Maximize $|\triangle|$.

and

Maximum Arc-Disjoint Cycle Packing in Sparse Tournament (Max ACS)
**Input:** A sparse tournament $T$.
**Result:** A collection $\mathcal{C}$ of arc-disjoint cycles of $T$.
**Optimization:** Maximize $|\mathcal{C}|$.

Let $T$ be a sparse tournament with regards to the ordering of its vertices $\sigma(T)$, that is the set of its backward arcs $\overleftarrow{A}(T)$ is a matching. As we did previously, we can assume that the backward arcs of $T$ do not contain consecutive vertices in the linear representation, since exchanging their positions would lead to another sparse tournament.

If a vertex $u$ of $T$ is not contained in any backward arc of $T$, then call $A_u$ (resp. $B_u$) the vertices of $T$ which are strictly after (resp. before) $u$ in the ordering $\sigma(T)$. Let $\triangle_u$ be the set of triangles made with a backward arc from $A_u$ to $B_u$ and the vertex $u$. As $T$ is sparse, it is clear that $\triangle_u$ is a set of disjoint triangles.

Moreover, notice that there exists an optimal packing of triangles (resp. cycles) of $T$ which is the union of an optimal packing of triangles (resp. cycles) of $T[A_u]$, one of $T[B_u]$ and $\triangle_u$. Thus, to solve Max ATS or Max ACS on $T$, we can solve the problem on $T[A_u]$ and on $T[B_u]$ and build the optimal solution for $T$. Therefore, we can focus on the case where *every vertex* of $T$ is the endpoint of a backward arc of $\overleftarrow{A}(T)$. We say that such tournaments are *fully sparse*. So we first address the following optimization problem.

ATS for Fully Sparse ($\Pi$)
**Input:** A fully sparse tournament $T$.
**Result:** A collection $\triangle$ of arc-disjoint triangles of $T$.
**Optimization:** Maximize $|\triangle|$.

Now, let order the arcs $e_1, \ldots, e_b$ of $\overleftarrow{A}(T)$ such that, for any $i \in [\![1, b-1]\!]$, we have $h(e_i)$ before $h(e_{i+1})$ in the ordering $\sigma(T)$. Moreover, given a fully sparse tournament $T$ and the previous ordering of its arc, let $G_T$ be the digraph with vertex set $V_T = \{e_i \colon i \in [\![1, b]\!]\}$ and arc set $A_T$ defined such that $(e_i, e_j) \in A_T$ if $(h(e_i), h(e_j), t(e_i))$ or $(h(e_i), t(e_j), t(e_i))$ is a triangle of $T$. In other words, we

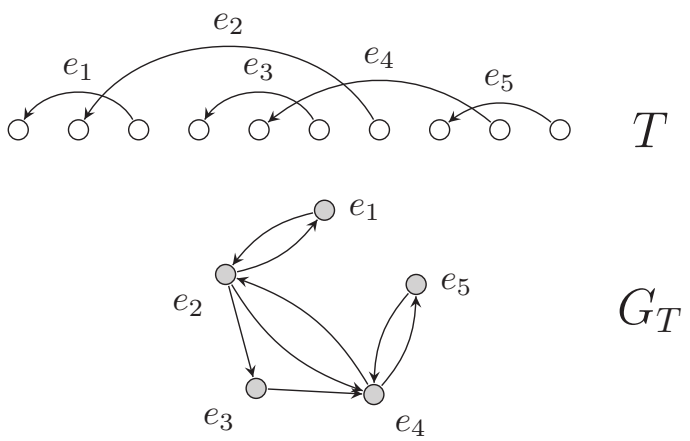



**Figure 6.5** – Example of a fully sparse tournament $T$ (at the top) and its corresponding $G_T$ (at the bottom).

have an arc from $e_i$ to $e_j$ if the backward arc $e_i$ can make a triangle with an endpoint of $e_j$. In addition, such endpoint is denoted by $s_{e_i}(e_j)$. An example of a fully sparse tournament $T$ and its corresponding $G_T$ is depicted in Figure 6.5.

Let $\Pi'$ be the following problem. Recall that a functional digraph is a digraph where all the vertices have out-degree exactly one.

Max Digon-Free Functional Subdigraph ($\Pi'$)
**Input:** A digraph $D$.
**Result:** A subset $X$ of $A(D)$ such that the digraph induced by the arcs of $X$ is a functional digraph and digon-free.
**Optimization:** Maximize $|X|$.

We want to prove that, given a fully sparse tournament $T$, if $X$ an optimal solution of $\Pi'$ with the instance $G_T$ if and only if we can construct in polynomial time an optimal solution of $\Pi$ on the instance $T$.

To do so, let $X$ be a solution (not necessary optimal) of $\Pi'(G_T)$ with the instance $G_T$, and $(e_i, e_j)$ an arc of $X$. We denote by $\Pi((e_i, e_j))$ the triangle $(h(e_i), s_{e_i}(e_j), t(e_i))$. In the same way, we define $\Pi(X)$ as the collection of triangles $\bigcup_{x \in X} \Pi(x)$.

**Claim 37.1.** *Let $X$ be a solution of $\Pi'(G_T)$. The set $X$ is an optimal solution if and only if $\Pi(X)$ is an optimal solution of $\Pi(T)$.*

*Proof.* Let $(e_i, e_j)$ and $(e_k, e_l)$ be two distinct arcs of $X$. We cannot have $e_i = e_k$ as $X$ induces a functional digraph in $G_T$. Without loss of generality, we may assume that $i < k$, that is $h(e_i)$ is before $h(e_k)$ in the ordering. Moreover, we cannot have $t(e_i) = t(e_k)$ without contradicting that $T$ is a sparse tournament. As $h(e_i)$ is before $h(e_k)$, the arc $(h(e_i), s_{e_i}(e_j))$ is not an arc of $\Pi((e_k, e_l))$. Thus, if $\Pi((e_i, e_j))$ and $\Pi((e_k, e_l))$ share a common arc, it means that we have $(s_{e_i}(e_j), t(e_i)) = (h(e_k), s_{e_k}(e_l))$. But, in this case, we have $e_i = e_l$ and $e_j = e_k$, implying $\{(e_i, e_j), (e_k, e_l)\}$ is a digon of $G_T$, which contradict the fact that $X$ is

a solution $\Pi'(G_T)$. So, if $X$ is a solution of $\Pi'(G_T)$, then $\Pi(X)$ is an solution of $\Pi(T)$. Notice that the size of the solution does not change.

On the other hand, if $X$ is a subset of the arcs of $G_T$ such that $\Pi(X)$ is a solution of $\Pi(T)$. We cannot have a vertex $e_i$ of $G_T$ such that $d^+_X(e_i) > 1$, since it would imply that the backward arc $e_i$ of $T$ is covered by at least two triangles of $\Pi(X)$. So $X$ induces a functional subdigraph of $G_T$. As previously the digraph induced by $X$ is also digon-free otherwise we would have two arc-disjoint triangles on only four vertices in $\Pi(X)$, which is impossible. Thus, $X$ is a solution of $\Pi'(G_T)$, and the solution of the same size.

The two problems $\Pi$ and $\Pi'$ being both maximization problems, they have the same optimal solution. □

Now, we show that we can to solve $\Pi'$ in polynomial time. We first focus on the case where $G_T$ is strong.

**Claim 37.2.** *If $G_T$ is strongly connected and has a cycle $C$ of size at least 3 then the solution of $\Pi'(G_T)$ is the number of vertices of $G_T$.*

*Proof.* We construct the arc set $X$ as follows: we start by taking the arcs of $C$. Then, while there is a vertex $u$ of $G_T$ which is not covered by any arcs of $X$, we add to $X$ the arcs of the shortest path from $u$ to any vertex of $X$. By construction, every vertex $u$ of every arcs of $X$ verify $d^+_X(u) = 1$, and $X$ is digon-free. Since $X$ covers every vertex of $G_T$, it is a maximum-sized solution of $\Pi'(G_T)$. □

In the following, we say that a digraph $D$ is a *digoned-tree* if $D$ arises from a non-trivial tree whose each edge is replaced by a digon.

**Claim 37.3.** *If $G_T$ is strongly connected and has only cycles of size 2 then $G_T$ is a digoned-tree.*

*Proof.* Since $G_T$ is strongly connected, then for any arc $(u, v)$ of $G_T$ there exists a path from $v$ to $u$. As $G_T$ only contains cycles of size 2, the only path from $v$ to $u$ is the direct arc $(v, u)$. So every arc of $G_T$ is contained in a digon. Finally, it is clear that the underlying graph of $G_T$ is a tree since $G_T$ would contain a cycle of size more than 2 otherwise. □

**Claim 37.4.** *If $G_T$ is a digoned-tree or if $|V(G_T)| = 1$, then the optimal solution of $\Pi'(G_T)$ has size $|V(G_T)| - 1$.*

*Proof.* The case $|V(G_T)| = 1$ is clear. So assume that $G_T$ is a digoned-tree and let $X$ be a set of arcs of $G_T$ corresponding to an optimal solution of $\Pi'(G_T)$. Then $X$ is acyclic and then has size at most $|V(G_T)| - 1$. Moreover, any in-branching of $G_T$ provides a solution of size $|V(G_T)| - 1$. □

Let us now prove that we can solve in polynomial time $\Pi'$ when $G_T$ is not strong too.

**Lemma 37.1.** *Let $G_T$ be a digraph with $n$ vertices. Denote by $S_1, \ldots, S_p$ terminal strong components of $G_T$ such that for any $i$ with $i \in [\![1, k]\!]$, $S_i$ is a digoned-tree or an isolated vertex and for any $i \in [\![k + 1, p]\!]$, $S_i$ contains a cycle of length at least 3. Then an optimal solution of $\Pi'(G_T)$ has size $n - k$ and we can construct it in polynomial time.*

*Proof.* We can assume that the underlying graph of $G_T$ is connected otherwise we apply the result on every connected components and the disjoint union of the solutions produces an optimal solution on the whole digraph $G_T$.

So assume that $G_T$ is connected but not strong, and let $S$ be a terminal strong component of $G_T$. If $X$ is an optimal solution of $\Pi'(G_T)$ then the restriction of $X$ to the arcs of $G_T[S]$ is an optimal solution of $\Pi'(G_T[S])$. Indeed otherwise we could replace this set of arcs in $X$ by an optimal solution of $\Pi'(G_T[S])$ and obtain a better solution for $\Pi'(G_T)$, a contradiction.

So by Claim 37.2 and Claim 37.4 the set $X$ contains at most $\sum_{i\in[\![1,p]\!]} |S_i| - k$ arcs lying in a terminal component of $G_T$. Now, as every vertex of $G_T \setminus \bigcup_{i\in[\![1,p]\!]} S_i$ is the beginning of at most one arc of $X$, the set $X$ has size at most $n-k$. Conversely, by growing in-branchings in $G_T$ from the union of the optimal solutions of $\Pi'(G_T[S_i])$ for $i \in [\![1,p]\!]$, by Claim 37.2 and Claim 37.4; we obtain a solution of $\Pi'(G_T)$ of size $n-k$ which is then optimal. Moreover, this solution can clearly be built in polynomial time.

□

**Corollary 4.** Maximum Arc-Disjoint Triangle Packing in Sparse Tournament *can be solved in polynomial time.*

Using Claim 37.1 and Lemma 37.1, we now prove that we can solve Max ACS in polynomial time by proving that the size of a maximum cycle packing is equal to the size of a maximum triangle packing in a fully sparse tournament $T$.

**Lemma 37.2.** *In a fully sparse tournament $T$, the size of a maximum cycle packing is equal to the size of a maximum triangle packing.*

*Proof.* First, if $T$ has an optimal triangle packing of size $|\overleftarrow{A}(T)|$ then as $\overleftarrow{A}(T)$ is a feedback arc set of $T$, every optimal cycle packing of $T$ has size $|\overleftarrow{A}(T)|$ and we are done. Otherwise, we build from $T$ the digraph $G_T$ as previously. By Lemma 37.1, $G_T$ has some terminal components $S_1, \ldots, S_k$ which are either a single vertex or induces a digoned-tree and every optimal triangle packing of $T$ has size $|\overleftarrow{A}(T)| - k$. Notice that no $S_i$ can be a single vertex. Indeed, if we have $S_i = \{e\}$ where $e$ is a backward arc of $T$, then it means that no backward of $T$ begins or ends between $h(e)$ and $t(e)$ in $\sigma(T)$. As $T$ is fully sparse, it means that $h(e)$ and $t(e)$ are consecutive in $\sigma(T)$ what we forbid previously.

Now, consider a component $S_i$ which induces a digoned-tree in $G_T$. Let $\sigma_i$ be the order $\sigma(T)$ restricted to the endpoints of the arcs of $T$ corresponding to the vertices of $S_i$. First, suppose that there exist two backward arcs $a$ and $b$ of $T$ such that $a \in S_i$, $b \notin S_i$ and $h(a)$ is before the head or the of $b$ which is before $t(a)$ in $\sigma(T)$. In this case, there is an arc in $G_T$ from $a$ to $b$ contradicting the fact that $S_i$ is a terminal component of $G_T$. So $\sigma_i$ is an interval of the order $\sigma(T)$.

So we can denote $\sigma_i$ by $(u_1, u_2, \ldots, u_\ell)$ and notice that $u_1$ and $u_2$ must be the heads of backward arcs belonging to $S_i$. If $u_3$ is also the head of backward arc of $S_i$, then we obtain that the three corresponding backward arcs form a 3-cycle in $G_T$ contradicting the fact that $S_i$ induces a digoned-tree in $G_T$.

Repeating the same argument, we show that $\ell$ is even and that the backward arcs corresponding to the elements of $S_i$ are exactly $(u_3, u_1)$, $(u_\ell, u_{\ell-2})$ and

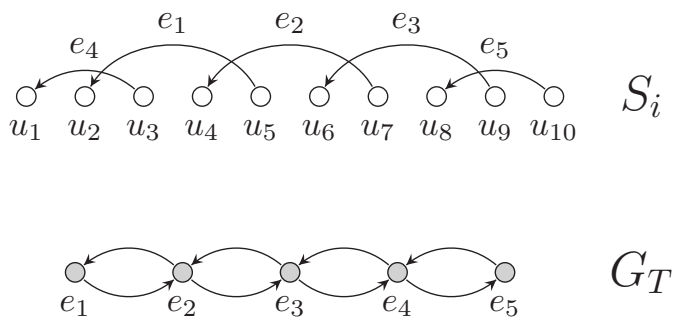



**Figure 6.6** – At the top, an example of an $S_i$ with $\ell = 10$ which implies a "digoned-path" in $G_T$ (at the bottom).

$(u_j, u_{j-3})$ for all odd $j \in [\![1, \ell]\!] \setminus \{1, 3\}$. In other words $S_i$ induces a "digoned-path" in $G_T$.

Now, consider $\mathcal{C}$ an optimal cycle packing of $T$. Let $A_1$ be the set of backward arcs of $\overleftarrow{A}(T)$ with head strictly before $u_1$ and tail strictly after $u_\ell$ in $\sigma(T)$. And let $\mathcal{C}_1$ be the cycles of $\mathcal{C}$ using at least one arc of $A_1$. It is easy to check that $\mathcal{C}' = (\mathcal{C} \setminus \mathcal{C}_1) \cup \{(h(e), u_1, t(e)) \colon e \in A_1\}$ is also an optimal cycle packing of $T$. Now every cycle of $\mathcal{C}'$ which uses a backward arc of $S_i$ only uses backward arcs of $S_i$ (otherwise it must one arc of $A_1$, which is not possible). Let $\mathcal{C}_i$ be the set of cycles of $\mathcal{C}$ using backward arcs of $S_i$. It is easy to see that $\{(u_i, u_{i+1}) \colon i \text{ even and } i \in [\![1, \ell - 2]\!]\}$ is a feedback arc set of $T[\{u_1, \ldots, u_\ell\}]$ and has size $\ell/2 - 1 = |S_i| - 1$. So we have $|\mathcal{C}_i| \leq |S_i| - 1$.

Repeating this argument for $i \in [\![1, k]\!]$ we obtain that $|\mathcal{C}| \leq |\overleftarrow{A}(T)| - k$. Thus, by Lemma 37.1, $\mathcal{C}$ has the same size than an optimal triangle packing of $T$. □

Using Corollary 4 and Lemma 37.2, we immediately have the following result.

**Corollary 5.** Maximum Arc-Disjoint Cycle Packing in Sparse Tournament *be solved in polynomial time.*

## 6.4 Concluding Remarks

### 6.4.1 Vertex-disjoint Triangle Packing

Concerning approximation algorithms for Disjoint Triangle Packing in Sparse Tournament, we have provided a $(1 + \frac{6}{c-1})$-approximation algorithm, where $c$ is a lower bound of the minspan of the instance. Recall that there is a $(4/3 + \varepsilon)$-approximation-approximation algorithm by Cygan [47] for 3-Set Packing. Thus, our approximation algorithm is better for sparse tournaments with minspan at least 19.

On the other hand, it is not hard to solve by dynamic programming DTS for instances where maxspan is bounded. Using these two opposite approaches, it could be interesting to derive an approximation algorithm for DTS with

approximation factor better than the $(4/3 + \varepsilon)$-approximation-approximation algorithm by Cygan [47] for sparse tournaments.

However, notice than in the NP-hardness proof of Disjoint Triangle Packing in Sparse Tournament the minspan is three. Thus, consider of the hardness of this problem for larger minspan could be a first interesting initial question. Concerning FPT-algorithms, the approach we used for sparse tournaments (reducing to the case where $m = O(k)$ and apply the $O(m)$ vertices kernel of Theorem 26) cannot work in the general case. Indeed, if we were able to "sparsify" the initial input such that $m' = O(k^{2-\varepsilon})$, applying the kernel in $O(m')$ would lead to a tournament of total bit size (by encoding the two endpoint of each arc) $O(m' \log m') = O(k^{2-\varepsilon})$, contradicting Theorem 28. Thus the situation for Disjoint Triangle Packing in Tournament could be similar than with vertex cover, where there exists a kernel in $O(k)$ vertices derived from Nemhauser and Trotter [124], but the resulting instance cannot have $O(k^{2-\varepsilon})$ edges, see [54] of Dell and Marx.

### 6.4.2 Arc-disjoint Triangle Packing

In this chapter, we proved that Arc-Disjoint Cycle Packing in Tournament and Arc-Disjoint Triangle Packing in Tournament both admit a polynomial time algorithm when the input is a sparse tournament. Unlike the vertex-disjoint version where the problem remain NP-hard, the sparse property seems to capture in a deeper way the difficulty of the arc-disjoint version. Note that such a gap of complexity between vertex and arc-disjoint version in such a subclass of tournaments is not obvious *a priori*.

# Chapter 7

# Complementary Cycles of any Length in Regular Bipartite Tournaments

This chapter corresponds to joint work with Stéphane Bessy presented to *Euro-Comb 2016* [25]. Given any $k$-regular bipartite tournament $D$, we show that, for every $p$ with $2 \leq p \leq |V(D)|/2 - 2$, $D$ has a cycle $C$ of length $2p$ such that $D \setminus C$ is hamiltonian unless $D$ is isomorphic to a special digraph, $F_{4k}$. This statement was conjectured by Manoussakis, Song and Zhang [157] in 1994.

## 7.1 Introduction and Preliminaries

### 7.1.1 Cycle Factors and Related work

Recall that a *cycle factor* of a digraph $D$ is a spanning subdigraph of $D$ whose components are vertex-disjoint (directed) cycles. For some positive integer $k$, a *$k$-cycle factor* of $D$ is a cycle factor of $D$ with $k$ vertex-disjoint cycles; it can also be considered as a partition of $D$ into $k$ hamiltonian subdigraphs. In particular, a 1-cycle factor is a Hamilton cycle of $D$. The cycles of a 2-cycle factor are often called *complementary cycles*. Finally, for a digraph $D$ and integers $n_1, \ldots, n_k$ such that $n_1 + \cdots + n_k = |V(D)|$, a *$(n_1, \ldots, n_k)$-cycle factor* is a $k$-cycle factor $(C_1, \ldots, C_k)$ of $D$ such that for each $i \in [\![1, k]\!]$ the cycle $C_i$ has length $n_i$. The cycle $C_1$ will be called the *first cycle* of the cycle factor.

A lot of work has been done concerning cycle factors in tournament. For instance, the classical result of Camion [38] states that a tournament is strong if and only if it admits an Hamilton cycle (*i.e.* a 1-cycle factor). Reid [93] proved that every 2-connected tournament with at least 6 vertices and not isomorphic to $T_7$, the Paley tournament on 7 vertices[1], has a 2-cycle factor. This was extended by Chen *et al.* [40] who proved that every $k$-connected tournament with at least $8k$ vertices contains a $k$-cycle factor.

[1]The Paley tournament $T_7$ has vertex set $\{v_1, v_2, v_3, v_4, v_5, v_6, v_7\}$ and is the union of the three directed cycles $(v_1, v_2, v_3, v_4, v_5, v_6, v_7)$, $(v_1, v_3, v_5, v_7, v_2, v_4, v_6)$ and $(v_1, v_5, v_2, v_6, v_3, v_7, v_4)$.

On the other hand finding cycles of many lengths in different digraphs is a natural problem in Graph Theory [19]. For example, Moon proved in [123] that every vertex of a strong tournament is in a cycle of every length. Concerning cycle factor with prescribed lengths in tournaments, Song [140], extended the results of Reid [93], and proved that every 2-connected tournament with at least 6 vertices and not isomorphic to $T_7$ has a 2-cycle factor containing cycles of lengths $p$ and $|V(T)| - p$ for all $p$ such that $3 \leq p \leq |V(T)| - 3$. Li and Shu [110] finally refined the previous result by proving that any strong tournament with at least 6 vertices, a minimum out-degree or a minimum in-degree at least 3, and not isomorphic to $T_7$, has 2-cycle factor containing cycles of lengths $p$ and $|V(T)| - p$ for all $p$ such that $3 \leq p \leq |V(T)| - 3$. Recently, Kühn *et al.* [103] have extended these results by showing that every $O(k^5)$-connected tournament admits a $k$-cycle factor with prescribed lengths.

#### 7.1.1.1 Our Contribution and Organization of the Chapter

In this paper, we focus on cycle factors in $k$-regular bipartite tournaments. A *$k$-regular bipartite tournament* is an orientation of a complete bipartite graph $K_{2k,2k}$ where every vertex has out-degree $k$ exactly. The existing results concerning this class of digraphs try to extend what is known about cycle factors in tournaments. For instance, Zhang and Song [158] proved that any $k$-regular bipartite tournament with $k \geq 2$ has a 2-cycle factor. Moreover, Manoussakis, Song and Zhang [157] conjectured in 1994 the following statement whose proof is the main result of this paper.

**Theorem 38.** *For $k \geq 2$ let $D$ be a $k$-regular bipartite tournament not isomorphic to $F_{4k}$. Then for every $p$ with $2 \leq p \leq k$, $D$ has a 2-cycle factor containing cycles of length $2p$ and $|V(D)| - 2p$.*

The digraph $F_{4k}$ corresponds to the $k$-regular bipartite tournament consisting of four independent sets $K, L$, $M$ and $N$ each of cardinality $k$ with all possible arcs from $K$ to $L$, from $L$ to $M$, from $M$ to $N$ and from $N$ to $K$. In fact, every cycle of $F_{4k}$ has length $0 \pmod 4$. So for instance $F_{4k}$ has no 2-cycle factor of length 6 and $4k - 6$. Zhang, Manoussakis and Song proved their conjecture when $p = 2$ in their original paper [157]. In 2014, Bai, Li and He proved the conjecture for $p = 3$ [12].

Our proof of Theorem 38 runs by induction on $p$ and so we will use as basis cases the results of Theorem 38 for $p = 2$ [157] and $p = 3$ [12]. To perform induction step, we will need also a weaker form of Theorem 38 given by the following lemma.

**Lemma 38.1.** *For $k \geq 2$ let $D$ be a $k$-regular bipartite tournament. If $D$ contains a cycle factor with a cycle of length $2p$ with $2 \leq p \leq k$, then $D$ contains a 2-cycle factor containing cycles of lengths $2p$ and $|V(D)| - 2p$.*

Theorem 38 and Lemma 38.1 will both need the following result due to Häggkvist and Manoussakis to be proven.

**Theorem 39** (Häggkvist and Manoussakis [79] and Manoussakis [118])**.** *A bipartite tournament containing a cycle factor has either a Hamilton cycle or a cycle factor consisting of cycles $C_1, \ldots, C_m$ such that for any $1 \leq i < j \leq m$, there is no arc from $C_j$ to $C_i$.*

We start by giving in subsubsection 7.1.1.2 some introducing tools and definitions we use for the proofs. Then, Lemma 38.1 and Theorem 38 are proven in section 7.2 and section 7.3, respectively. Finally, in section 7.4 we give some concluding remarks concerning cycle factors in bipartite tournaments.

#### 7.1.1.2 Specific Definitions and Notations

**Generic definitions** Given two subset of vertices $A$ and $B$, the number of arcs from $A$ to $B$ is denoted by $e(A, B)$. We simply write $e(A)$ instead of $e(A, A)$ to denote the number of arcs joining two vertices of $A$.

Given a cycle $C = (v_1, \ldots, v_n)$, and two distinct vertices $v_i$ and $v_j$ of $C$, the *subpath of $C$ from $v_i$ to $v_j$* is the path $v_i, v_{i+1}, \ldots, v_j$ if $i < j$, and $v_i, v_{i+1}, \ldots, v_n, v_1, \ldots, v_j$ otherwise. In other words, the subpath of $C$ from $v_i$ to $v_j$ is the path from $v_i$ to $v_j$ which only uses the arcs of $C$. Given a path $P = v_1, \ldots, v_n$ with $n \geq 3$, the *internal vertices* of $P$ are the vertices $v_2, \ldots, v_{n-1}$.

Given two vertices $u$ and $v$ of a digraph $D$, if there is no arc from $u$ to $v$, we say that there is an *anti-arc* from $u$ to $v$. Moreover, we say that $v$ is an *anti-out-neighbor* of $u$ and $u$ is an *anti-in-neighbor* of $v$. The *anti-out-neighborhood* (resp. *anti-in-neighborhood*) of $u$ in $D$, denoted $\overline{N_D^+}(u)$ (resp. $\overline{N_D^-}(u)$), corresponds to the set of vertices which are anti-out-neighbor (resp. *anti-in-neighbor*) of $u$. The *anti-out-degree* (resp. *anti-in-degree*) of a vertex $u$, denoted $\overline{d_D^+}(u)$ (resp. $\overline{d_D^-}(u)$), is the size of its anti-out-neighborhood (resp. *anti-in-neighborhood*). For two vertex-disjoint sets $A$, and $B$ if there are all the possible anti-arcs going from $A$ to $B$ (that is there are no arcs from $A$ to $B$), we say that $A$ *anti-dominates* $B$.

If there is no ambiguity, we may omit the reference to the considered digraph in the previous notations (for example, $\overline{N^-}(u)$ instead of $\overline{N_D^-}(u)$).

Given a digraph $D$ and a set of $t$ disjoint vertices $\{u_1, \ldots, u_t\}$ of $D$, we say that $u_1, \ldots, u_t$ is an *anti-path* if $u_i u_{i+1} \notin A(D)$ for any $1 \leq i \leq t-1$. In addition, if $u_t u_1 \notin A(D)$, then we say that $(u_1, \ldots, u_t)$ an *anti-cycle*. Given a set $X$ of vertices and a cycle $C$, we denote by $C(X)$ the set of the *successors* of $X$ along $C$. If $X$ is a singleton $\{x\}$, we simply write $C(x)$ instead of $C(\{x\})$.

**Bipartite tournaments and contracted digraphs** Let $D$ be a $k$-regular bipartite tournament with bipartition $(S, T)$. We have $|S| = |T| = 2k$, and for any vertex $u$ of $D$ we have $d^+(u) = d^-(u) = k$. Moreover, the (unoriented) graph on $S \cup T$ containing an edge for every arcs from $S$ to $T$ is a bipartite graph where every vertex has degree $k$. Hence, by Hall's Theorem [33], it admits a perfect matching. Let $M$ be a set of arcs of $D$ corresponding to such a perfect matching. For each vertex $u$ of $S$, the vertex $M(u)$ denotes the only vertex of $T$ such that the arc $(u, M(u))$ is an arc of $M$. We extend this notation to sets that is, given a subset $X$ of $S$, we define $M(X)$ by $M(X) = \bigcup_{x \in X} M(x)$.

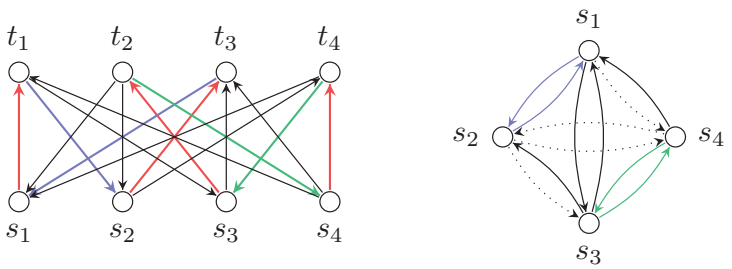



**Figure 7.1** – The right digraph is a contracted digraph according the red perfect matching of the left bipartite. The anti-arcs are depicted with dotted arcs. Note that a cycle factor (in blue and green) in the right digraph corresponds to a cycle factor in the left digraph, by using alternatively arcs of the perfect matching.

Now, given a perfect matching $M$ of $D$ made of arcs from $S$ to $T$, we define the *contracted digraph* according to $M$, denoted $D^M$ and obtained by contracting the arcs of $M$ and only keeping the arcs of $D$ from $T$ to $S$. Formally, the new digraph $D^M$ has vertex set $S$ and arc set $\{(u,v)\colon\ u \in S, v \in S \text{ and } (M(u),v) \in A(D)\}$. As the vertex set of $D^M$ is $S$, we also consider vertices of $D^M$ as vertices of $D$. Notice that $D^M$ has $2k$ vertices and that for every vertex $u$ of $D^M$ we have $N^+_{D^M}(u) = N^+_D(M(u))$ and so, $u$ has out-degree $k$ exactly. Similarly, $u$ has in-neighborhood $\{v \in S : M(v) \in N^-_D(u)\}$ and so has in-degree $k$ exactly. Notice also that $D^M$ does not contain any parallel arc but may contains digons. We refer reader to Figure 7.1 for an example of a contracted digraph.

Let $D'$ be a subdigraph of $D$ and denote by $M'$ the arcs of $M$ with both extremities in $D'$. If $M'$ is also a perfect matching of $D'$, then we abusively denote by $D'^M$ the contracted digraph $D'^{M'}$.

If now $M$ is a perfect matching of $D$ made of arcs from $T$ to $S$, then we can symmetrically define $D^M$ by exchanging $S$ and $T$ in the previous definitions.

Structurally, it is easy to notice that $(u_1, \ldots, u_t)$ is a cycle in $D^M$ if and only if $(u_1, M(u_1), \ldots, u_t, M(u_t))$ is a cycle of $D$. Thus, to prove Theorem 38, if $D$ is not isomorphic to $F_{4k}$, then for every $p$ with $2 \leq p \leq k-2$, it suffices to find a $(p, 2k-p)$-cycle factor in $D^M$. Finally, observe that the graph $D^M$ contains the same information than $D$ but, most of the time, it will be easier to identify particular structures in the former.

## 7.2 From a Cycle Factor to a 2-Cycle Factor

The aim of this section is to prove Lemma 39.1 which states that, given a cycle factor with a cycle of length $2p$, we can "merge" the other cycles in order to obtain a $(2p, |V(D)| - 2p)$-cycle factor. Moreover, in the case where $p$ is even, we could ask that the new cycle of length $2p$ is not isomorphic to $F_{2p}$ if the former was not. This condition will be useful in the induction step to prove Theorem 38.

**Lemma 39.1.** *For $k \geq 2$ let $D$ be a $k$-regular bipartite tournament. If $D$ contains a cycle factor with a cycle $C$ of length $2p$ with $2 \leq p \leq k$, then $D$ contains a*

*$(2p, |V(D)| - 2p)$-cycle factor $(C', C'')$. Moreover, if $p$ is at least 3 and even and $D[C]$ is not isomorphic to $F_{2p}$, then $D[C']$ is not isomorphic to $F_{2p}$ neither.*

*Proof.* As the cases where $p = 2$ and $p = 3$ of Theorem 38 are already proven in [157] and [12], we assume that $p \geq 4$.

Consider a cycle factor $\mathcal{C}'$ of $D$ containing a cycle $C$ of length $2p$, such that $C$ is not isomorphic to $F_{2p}$ if $p$ is even, and such that $\mathcal{C}'$ has a minimum total number of cycles. We denote by $\mathcal{C}$ the set of cycles of $\mathcal{C}'$ different from $C$. Thus, we want to show that $|\mathcal{C}| = 1$. By Theorem 39, if $|\mathcal{C}| \neq 1$ then we can assume that $\mathcal{C} = \{C_1, \ldots, C_\ell\}$ with $\ell \geq 2$ and that $C_i$ dominates $C_j$ whenever $i < j$. Let $(S, T)$ denotes the bipartition of $D$ and for every $i$, we denote by $c_i$ the number of vertices of $V(C_i) \cap S$, that is $C_i$ is of length $2c_i$.

**Claim 39.1.** *We have $e(C, C_1) = c_1(2k - c_1)$ and $e(C_\ell, C) = c_\ell(2k - c_\ell)$.*

*Proof.* A vertex $x_1$ in $C_1$ is an in-neighbor of every vertex in $C_2, \ldots, C_\ell$. Hence, we have $\sum_{x \in C_1} d_D^-(x) = e(C_1) + e(C, C_1)$. Thus we get $2kc_1 = c_1^2 + e(C, C_1)$ and the first result holds. The other equality is obtained similarly. □

Now, using Claim 39.1 we have the following.

$$\Big( \frac{1}{c_\ell} \sum_{x \in T \cap C_\ell} d_C^+(x) + \frac{1}{c_1} \sum_{x \in T \cap C} d_{C_1}^+(x) \Big) + \Big( \frac{1}{c_\ell} \sum_{x \in S \cap C_\ell} d_C^+(x) + \frac{1}{c_1} \sum_{x \in S \cap C} d_{C_1}^+(x) \Big)$$

$$= \frac{1}{c_\ell} e(C_\ell, C) + \frac{1}{c_1} e(C, C_1) = 4k - (c_1 + c_\ell)$$

Hence, either we have

$$\Big( \frac{1}{c_\ell} \sum_{x \in T \cap C_\ell} d_C^+(x) + \frac{1}{c_1} \sum_{x \in T \cap C} d_{C_1}^+(x) \Big) \geq 2k - \frac{(c_1 + c_\ell)}{2}$$

or we have

$$\Big( \frac{1}{c_\ell} \sum_{x \in S \cap C_\ell} d_C^+(x) + \frac{1}{c_1} \sum_{x \in S \cap C} d_{C_1}^+(x) \Big) \geq 2k - \frac{(c_1 + c_\ell)}{2}$$

Without loss of generality, we can assume that the former holds (otherwise we exchange in the following the role of $S$ and $T$).

Denote by $M$ the set of arcs of the digraph induced by the cycle factor $C, C_1, \ldots, C_\ell$ and going from $S$ to $T$ in $D$. It is clear that $M$ forms a perfect matching of $D$ and that $C^M \cup \mathcal{C}^M$ is a cycle factor of $D^M$, where $\mathcal{C}^M = \{C_1^M, \ldots, C_\ell^M\}$. Moreover, notice that the length of $C^M$ is $p$ and for $i$ with $1 \leq i \leq \ell$ the length of $C_i^M$ is $c_i$. By the previous assumption, in $D^M$ we have the following

$$\frac{e(C_\ell^M, C^M)}{c_\ell} + \frac{e(C^M, C_1^M)}{c_1} \geq 2k - \frac{(c_1 + c_\ell)}{2} \tag{7.1}$$

Now, let us find suitable vertices in $C^M$ to design the desired 2-cycle factor. To do so, let $B_1$ (resp. $B_\ell$) be the set of pairs $\{x, y\}$ of distinct vertices of

$C^M$ which are "well connected" to $C_1^M$ (resp. from $C_\ell^M$), that is such that $d^+_{C_1^M}(x) + d^+_{C_1^M}(y) > c_1$ (resp. $d^-_{C_\ell^M}(x) + d^-_{C_\ell^M}(y) > c_\ell$). We denote by $b_1$ (resp. $b_\ell$) the cardinal of $B_1$ (resp. $B_\ell$).

**Claim 39.2.** *We have* $b_1 + b_\ell \geq \frac{p(p-1)}{2}$.

*Proof.* For every pair $\{x, y\}$ of distinct vertices of $V(C^M)$, we have $d^+_{C_1^M}(x) + d^+_{C_1^M}(y) \leq 2c_1$ and, if $\{x, y\}$ is not a pair of $B_1$, we have more precisely $d^+_{C_1^M}(x) + d^+_{C_1^M}(y) \leq c_1$. Thus, in total,

$$\begin{aligned}\sum_{\substack{\{x,y\} \text{ pair} \\ \text{of } V(C^M)}} (d^+_{C_1^M}(x) + d^+_{C_1^M}(y)) &= \sum_{\{x,y\} \in B_1} (d^+_{C_1^M}(x) + d^+_{C_1^M}(y)) \\ &\quad + \sum_{\{x,y\} \notin B_1} (d^+_{C_1^M}(x) + d^+_{C_1^M}(y)) \\ &\leq 2b_1c_1 + (\frac{p(p-1)}{2} - b_1)c_1\end{aligned} \tag{7.2}$$

and

$$\sum_{\substack{\{x,y\} \text{ pair} \\ \text{of } V(C^M)}} \left(d^+_{C_1^M}(x) + d^+_{C_1^M}(y)\right) = (p-1)e(C^M, C_1^M)$$

Thus, we get

$$(p-1)\frac{e(C^M, C_1^M)}{c_1} \leq b_1 + \frac{p(p-1)}{2}$$

Similarly, we obtain

$$(p-1)\frac{e(C_\ell^M, C^M)}{c_\ell} \leq b_\ell + \frac{p(p-1)}{2}$$

Hence, using the inequality (7.1), we have

$$(p-1)(2k - \frac{c_1 + c_\ell}{2}) \leq b_1 + b_\ell + p(p-1)$$

Finally, since $C^M \cup C_1^M \cup C_\ell^M$ is a subgraph of $D^M$, we have $p + c_1 + c_\ell \leq 2k$ and so $2k - (c_1 + c_\ell)/2 \geq k + p/2$. With the previous inequality, we obtain $b_1 + b_\ell \geq (p-1)(k - p/2)$. Finally, using that $k \geq p$, we get the result, that is $b_1 + b_\ell \geq p(p-1)/2$. □

Now, for every pair $\{x, x'\}$ of distinct vertices of $C^M$, we color $\{x, x'\}$ in blue if it is a pair of $B_1$, and we color $\{x, x'\}$ in red if $\{y, y'\} \in B_\ell$ where $y$ (resp. $y'$) is the out-neighbor of $x$ (resp. $x'$) along $C^M$.

**Claim 39.3.** *There exists a pair of vertices colored in blue and red.*

*Proof.* If $b_1 + b_\ell > p(p-1)/2$, then we have colored more than $p(p-1)/2$ pairs of distinct vertices of $C^M$. Thus, at least one pair have been colored both in red and blue, yielding the result.

Now, let suppose that $b_1 + b_\ell \leq p(p-1)/2$. By Claim 39.2, it means that we have $b_1 + b_\ell = p(p-1)/2$ and that all the inequalities leading to the proof of Claim 39.2 are equalities. In particular, we have $p + c_1 + c_\ell = 2k$ and $p = k$. Notice that, as $c_1$ and $c_\ell$ are at least 2, we have $k \geq 4$. Moreover, as (7.2) is also an equality, we have $d^+_{C_1^M}(x) + d^+_{C_1^M}(y) = 2c_1$ for every pair $\{x, y\}$ of $B_1$ and $d^+_{C_1^M}(x) + d^+_{C_1^M}(y) = c_1$ for every pair $\{x, y\}$ of vertices of $C_1^M$ which is not in $B_1$. In particular, if $\{x, y\} \in B_1$ we have exactly $d^+_{C_1^M}(x) = c_1$ and $d^+_{C_1^M}(y) = c_1$. Similarly, if $\{x, y\} \in B_\ell$ we have exactly $d^-_{C_\ell^M}(x) = c_\ell$ and $d^-_{C_\ell^M}(y) = c_\ell$. In particular, we can prove that $b_1 \neq 0$. Indeed, otherwise, we have $b_\ell = p(p-1)/2$, that is, every pair of elements of $C^M$ is a pair of $B_\ell$. Then, by the previous remark, every vertex $x$ of $C^M$ satisfies $d^-_{C_\ell^M}(x) = c_\ell$, and $C_\ell^M$ dominates $C^M$. So, the out-neighborhood of any vertex $x$ of $C_\ell^M$ would contain the successor of $x$ along $C_\ell^M$ and all the cycle $C^M$, which is of length $p = k$, a contradiction to $d^+_{D^M}(x) = k$. Similarly, we have $b_\ell \neq 0$.

Now, let $A_1$ (*resp.* $A_\ell$) be the collection of vertices which belong to at least one pair of $B_1$ (*resp.* $B_\ell$). Thus, $A_1$ is not empty and for every vertex $v \in A_1$, we have $d^+_{C_1^M}(v) = c_1$. As every pair $\{x, y\}$ of distinct vertices of $C^M$ satisfies $d^+_{C_1^M}(x) + d^+_{C_1^M}(y) \in \{c_1, 2c_1\}$, it is easy to see that every vertex $w \notin A_1$ satisfies $d^+_{C_1^M}(w) = 0$, and that there is at most one vertex $a_1$ which does not belong to $A_1$. With the same arguments we see that there is at most vertex $a_\ell$ which does not belong to $A_\ell$. So, as $p = k \geq 4$ there exists a pair of vertices of $C^M$ containing neither $a_1$ nor the predecessor of $a_\ell$ along $C^M$. Thus, this pair is colored blue and red. □

In the following, let $\{y_1, z_1\}$ be a pair of vertices of $V(C^M)$ colored both in red and blue and we will denote by $y_\ell$ (resp. $z_\ell$) the successor of $y_1$ (resp. $z_1$) along $C^M$. Therefore we have $\{y_1, z_1\} \in B_1$ and $\{y_\ell, z_\ell\} \in B_\ell$. Notice that $\{y_1, z_1\}$ and $\{y_\ell, z_\ell\}$ are two distinct pairs of vertices but that $y_\ell = z_1$ or $z_\ell = y_1$ is possible.

**Claim 39.4.** *For every $i \in [\![0, c_1 - 2]\!]$, there exists $y$ and $z \in C_1^M$ with $(y_1, y)$ and $(z_1, z) \in A(D^M)$ and such that the subpath of $C_1^M$ from $y$ to $z$ contains $i$ internal vertices exactly. Similarly for every $i' \in [\![0, c_\ell - 2]\!]$, there exists $y'$ and $z' \in C_1^M$ with $(y', y_\ell), (z', z_\ell) \in A(D^M)$ and such that the subpath of $C_\ell^M$ from $z'$ to $y'$ contains $i'$ internal vertices exactly.*

*Proof.* Suppose that for every vertex $y$ of $N^+_{C_1^M}(y_1)$, the vertex $z$ which is $i+1$ vertices away from $y$ along $C_1^M$ does not belong to $N^+_{C_1^M}(z_1)$. Thus we have $|N^+_{C_1^M}(z_1)| \leq c_1 - |N^+_{C_1^M}(y_1)|$, which contradicts $d^+_{C_1^M}(y_1) + d^+_{C_1^M}(z_1) > c_1$ as $\{y_1, z_1\}$ is a pair of $B_1$. The proof is similar for the pair $\{y_\ell, z_\ell\}$. □

Now, we can construct our 2-cycle factor from the collection of cycles. To do so, we will build a cycle $\gamma$ containing $p$ vertices, such that $D^M[V(D^M) \setminus V(\gamma)]$ contains a spanning cycle denoted by $\gamma'$. Let $s$ (*resp.* $s'$) be the number of vertices

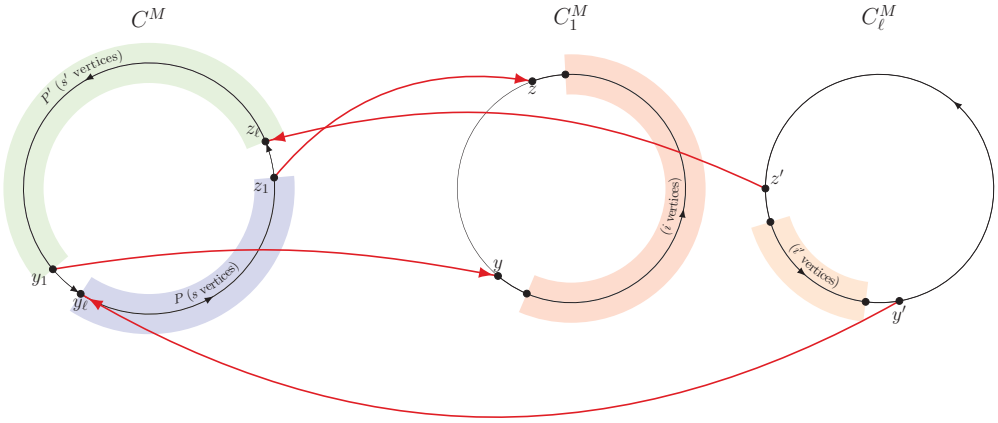



**Figure 7.2** – An illustrative case of the proof of Claim 39.4.

in the path $P$ (resp. $P'$) along $C^M$ from $y_\ell$ to $z_1$ (*resp.* from $z_\ell$ to $y_1$). We have $s + s' = p$, thus either we have $s \leq p/2$ or $s' \leq p/2$. We will suppose that the former holds, since an analogous reasoning can be applied for the other case. In the following, we will denote by $i_0$ the smallest index $j$ such that $s + \sum_{i=0}^{j} c_i > p$. Such index exists, since $s < p$ and $s + \sum_{i=0}^{\ell} c_i > \sum_{i=0}^{\ell} c_i = 2k - p \geq p$. The cycle $\gamma$ (resp. $\gamma'$) will be obtained as the union of the path $P$ (resp. $P'$) and a path $Q$ (resp. $Q'$) well chosen in $C_1^M, \ldots, C_\ell^M$. To design $Q$ and $Q'$ we consider several cases.

First assume that we have $1 < i_0 < \ell$. According to Claim 39.4 applied with $i = 0$ and $i' = 0$ there exists a pair of vertices $\{y, b_1\}$ of $C_1^M$ such that $y$ is the successor of $z$ along $C_1^M$ and with $(y_1, z), (z_1, y) \in A(D^M)$. Similarly, there is $\{y', z'\}$ in $C_\ell^M$ such $y'$ is the successor of $z'$ along $C_\ell^M$ and $(z', z_\ell), (y', y_\ell) \in A(D^M)$. As $C_i^M$ dominates $C_j^M$ for any $i < j$, we consider in $D^M$ the path $Q$ starting in $y$, containing every vertices of $C_1^M$ except $z$, every vertices of $C_j^M$, for any $j \in [\![2, i_0 - 1]\!]$ and $p - s - \sum_{i=1}^{i_0-1} c_i$ consecutive vertices of $C_{i_0}^M$ and finally ending with the vertex $y'$. Similarly we construct the path $Q'$ containing $z$, the remaining vertices of $C_{i_0}^M$, every vertices of $C_j^M$, for any $i_0 < j < \ell$ and every vertices of $C_\ell^M$ except $y'$, that is $Q'$ ends in $z'$. As $(z_1, y)$, $(y', y_\ell)$, $(y_1, z)$ and $(z', z_\ell)$ are arcs of $D^M$ $\gamma = P \cup Q$ and $\gamma' = P' \cup Q'$ are cycles and they form a 2-cycle factor of $D^M$. To conclude this case, it remains to notice that the number of vertices in $\gamma$ is $s + (c_1 - 1) + (\sum_{i=2}^{i_0-1} c_i) + (p - s - \sum_{i=1}^{i_0-1} c_i) + 1 = p$.

In the case where $i_0 = 1$, Claim 39.4 applied with $i = p - s - 2$ and $i' = 0$ asserts that there exists $\{y, z\}$ in $C_1^M$ such that there are $p - s - 2$ vertices from $y$ to $z$ along $C_1^M$ with $(y_1, z), (z_1, y) \in A(D^M)$. As we assume that $p \geq 3$ and we have $s \leq p/2$, we know that $p - s \geq 2$. There also are $\{y', z'\}$ in $C_\ell^M$ such that $y'$ is the successor of $z'$ along $C_\ell^M$ and $(z', z_\ell), (y', y_\ell) \in A(D^M)$. Thus we construct $Q$ starting from $y$, containing $p - s - 1$ vertices of $C_1^M$ and ending in $y'$. The path $Q'$ starts in $z$, contains the remaining vertices of $C_1^M$, every vertices of $C_j^M$, for any $1 < j < \ell$ and every vertices of $C_\ell^M$ except $y'$, that is $Q'$ ends in $z'$. As

previously, we easily check that $\gamma = P \cup Q$ and $\gamma' = P' \cup Q'$ form a 2-cycle factor of $D^M$ of lengths $p$ and $2k - p$.

The case $i_0 = \ell$ is symmetric to the previous one.

To check the last part of the statement, we have to guarantee that $D[C']$ is not isomorphic to $F_{2p}$, where $C'$ denote the cycle of $D$ corresponding to $\gamma$ (*i.e.* such that $C'^M = \gamma$). To do so, notice that, in all cases, we added $y$ and $y'$ to $P$ in order to close $\gamma$. In $D^M$, as $y \in C_1^M$ and $y' \in C_\ell^M$, we have $(y, y')$ which is an arc of $D^M$ and $(y', y)$ an anti-arc of $D^M$. Then $C'$ contains four vertices $y$, $C'(y), y'$ and $C'(y')$ such that $(y, C(y))$, $(C'(y), y')$, $(y', C'(y'))$ and $(y, y')$ are arcs of $D$. However $F_{2p}$ does not contain such a subdigraph. So $D[C']$ is not isomorphic to $F_{2p}$. □

## 7.3 Proof of Theorem 38

We prove a slightly stronger version of Theorem 38 where we ask for the first cycle of the cycle factor to be different from $F_{2p}$ if $p$ is even (notice that $F_{2p}$ is not defined for odd $p$). Namely, we prove the following result.

**Theorem 40.** *For $k \geq 3$ let $D$ be a $k$-regular bipartite tournament not isomorphic to $F_{4k}$. Then for every $p$ with $3 \leq p \leq k$, $D$ has a 2-cycle factor $(C_1, C_2)$ where $C_1$ has length $2p$ and if $p$ is even, $C_1$ is not isomorphic to $F_{2p}$.*

We prove this statement by induction on $p$. By the result of Bai *et al.* [12] the statement is true for $p = 3$ and the basis case for the induction holds. So for $3 \leq p < k$ we consider $D = (V, A)$ a $k$-regular bipartite tournament which admits a $(2p, 4k - 2p)$-cycle factor $(C_1, C_2)$ where $C_1$ is not isomorphic to $F_{2p}$ if $p$ is even. In particular, notice that $D$ is not isomorphic to $F_{4k}$. We want to show that $D$ admits a $(2(p+1), 4k - 2(p+1))$-cycle factor whose first cycle is not isomorphic to $F_{2(p+1)}$ if $p + 1$ is even. In the following, such a cycle factor is said to be *good*.

We denote by $(S, T)$ the bipartition of $D$ and by $(C_1, C_2)$ the $(2p, 4k - 2p)$-cycle factor of $D$, with $D[C_1]$ not isomorphic to $F_{2p}$. We also denote by $M_u$ the arcs of $C_1 \cup C_2$ going (up) from $S$ to $T$ and by $M_d$ the arcs of $C_1 \cup C_2$ going (down) from $T$ to $S$. It is clear that $M_u$ ans $M_d$ are perfect matchings of $D$ and that their union is $C_1 \cup C_2$. For $M$ being either $M_u$ or $M_d$, the digraph $D^M$ admits the 2-cycle factor $(C_1^M, C_2^M)$ with $|C_1^M| = p$ and $|C_2^M| = 2k - p$. Notice that, for even $p$, having $C_1$ not isomorphic to $F_{2p}$ is equivalent to having $D^M[C_1^M]$ being not isomorphic to the balanced complete bipartite digraph on $p$ vertices.

To form a good cycle factor from $(C_1^M, C_2^M)$, we will have a case-by-case study according to the structure of the non-arc in the digraph $D^M$. Prior to this study, we introduce some tools needed.

### 7.3.1 Switch Along an Anti-cycle

In this subsection, we first consider a matching $M$ of the $k$-regular bipartite tournament $D$ made from arcs from $S$ to $T$ and we define an operation allowing some local change in $M$.

**Lemma 40.1.** *If $D^M$ contains an anti-cycle $(u_1, \dots, u_t)$ with $t \geq 2$, then the set $M'$ of arcs defined in $D$ by $M' = (M \setminus \bigcup_{i=1}^{t}(u_i, M(u_i))) \cup (\bigcup_{i=1}^{t-1}(u_{i+1}, M(u_i)) \cup (u_1, M(u_t)))$ is a perfect matching of $D$. Moreover, for every $v \notin \{u_1, \dots, u_t\}$, we have $N^+_{D^{M'}}(v) = N^+_{D^M}(v)$ and, for every $i \in [\![2, t]\!]$, we have $N^+_{D^{M'}}(u_i) = N^+_{D^M}(u_{i-1})$ and $N^+_{D^{M'}}(u_1) = N^+_{D^M}(u_t)$.*

*Proof.* If $(u_1, \dots, u_t)$ is an anti-cycle of $D^M$, then it follows by definition that $(u_{i+1}, M(u_i))$ is an arc of $D$ for every $i \in [\![1, t-1]\!]$, as well as $(u_1, M(u_t))$. Thus, $M'$ is a perfect matching of $D$. For every $i \in [\![1, t-1]\!]$, we have $M'(u_{i+1}) = M(u_i)$ in $D$ (and $M'(u_1) = M(u_t)$), then $N^+_{D^{M'}}(u_{i+1}) = N^+_{D^M}(u_i)$ (and $N^+_{D^{M'}}(u_1) = N^+_{D^M}(u_t)$). The out-neighborhood of the other vertices are unchanged. □

The "shifting" operation between matchings $M$ and $M'$ described in the previous lemma is called a *switch along the anti-cycle* $(u_1, \dots, u_t)$. The first easy observation we can make on the new contracted digraph is the following.

**Corollary 6.** *If $D^M$ has a cycle factor and contains an anti-cycle $(u_1, \dots, u_t)$ with $t \geq 2$, then the digraph obtained after the switch along the anti-cycle $(u_1, \dots, u_t)$ has a cycle factor.*

*Proof.* Let $C$ be the anti-cycle $(u_1, \dots, u_t)$ and let $M'$ be the perfect matching obtained after the switch along $C$. Moreover, let $X$ be the subdigraph induced by the arcs of the cycle factor in $D^M$, and let $X'$ be the subdigraph induced by the switch of $X$ along $C$. By Lemma 40.1 we make a cyclic permutation on the out-neighborhoods of the vertices of $C$. So for every vertex $x$ in $S$ we have $d^+_{X'}(x) = d^+_X(x) = 1$ and $d^-_{X'}(x) = d^-_X(x) = 1$. Therefore, $X'$ is a cycle factor of $D^{M'}$. □

Claim 40.1 below is our main application of a switch along an anti-cycle in a contracted digraph. Before stating it, we need the following simple result.

**Lemma 40.2.** *If $D^M$ contains a cycle factor $\{B_1, \dots, B_\ell, B_{\ell+1}, \dots B_{\ell'}\}$ such that $|B_1| + \dots + |B_\ell| = p + 1$ and $D^M[B_1 \cup \dots \cup B_\ell]$ is strongly connected and not isomorphic to a balanced complete bipartite digraph, then $D$ admits a good cycle factor.*

*Proof.* For $i \in [\![1, \ell']\!]$, we denote by $\tilde{B}_i$ the cycle of $D$ such that $\tilde{B}_i^M = B_i$. Since $D^M[B_1 \cup \dots \cup B_\ell]$ is strongly connected, the digraph $D[\tilde{B}_1 \cup \dots \cup \tilde{B}_\ell]$ is also strongly connected and admits a cycle factor. So, by Theorem 39, it has a Hamilton cycle $C$ which is of length $2p+2$. Moreover, as $D^M[B_1 \cup \dots \cup B_\ell]$ is not isomorphic to a balanced complete bipartite digraph, then $D[C]$ is not isomorphic to $F_{2p+2}$ if $p$ is odd. Using Lemma 38.1 on the cycle factor $\{C, \tilde{B}_{\ell+1}, \dots, \tilde{B}_{\ell'}\}$, it proves that $D$ contains a good cycle factor. □

Now, back to the proof of Theorem 40, we see a first case where it is possible to extend the cycle $C_1$ to obtain a good cycle factor. Recall that the digraph $D^M$ contains the 2-cycle factor $(C_1^M, C_2^M)$, where here again $M$ stands for the matching $M_u$ or the matching $M_d$ of $D$.

**Claim 40.1.** *If $D^M$ contains an anti-cycle $H$ such that $H \backslash V(C_1^M)$ is an anti-path from $x$ to $y$ with $x = C_2^M(y)$ then $D$ admits a good cycle factor.*

*Proof.* To shorten notations, we denote $C_1^M$ by $C$ and $C_2^M$ by $C'$. Let $H = (a_1, \ldots, a_t)$ be an anti-cycle of $D^M$ such that $a_1, \ldots, a_s$ is an anti-path of $D^M[C']$, $a_1$ is the successor of $a_s$ along $C'$ and $a_{s+1}, \ldots, a_t$ is an anti-path of $D^M[C]$. Moreover, for every $i \in [\![1, t]\!]$, we denote by $b_i$ be the out-neighbor of $a_i$ along $C$ or $C'$. An illustrative case is depicted in Figure 7.3.

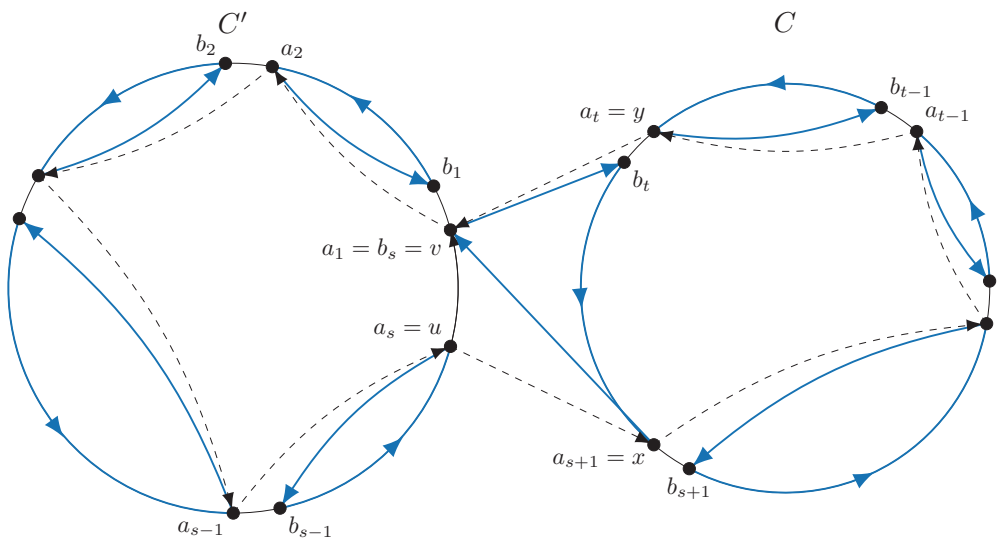



**Figure 7.3** – An illustrative case of the proof of Claim 40.1. The dotted arcs form the anti-cycle $H$ and the blue arcs form the cycle factor $B$ of $D^{M'}$.

We perform a switch along $H$ to obtain the digraph $D^{M'}$. By Corollary 6, the digraph $D^{M'}$ contains a cycle factor. More precisely, we pay attention to $B$ the cycle factor derived from $C \cup C'$. By Lemma 40.1, the out-neighbor in $B$ of every vertex not in $\{a_1, \ldots a_t\}$ is its out-neighbor in $C \cup C'$ and the out-neighbor in $B$ of every vertex $a_i$ in $\{a_1, \ldots a_t\}$ is $b_{i-1}$ (where indices are given modulo $t$). As there is only one arc of $H$ from $C$ to $C'$ and one arc of $H$ from $C'$ to $C$, the only arc of $B$ from $V(C)$ to $V(C')$ of $B$ is $(a_{s+1}, b_s)$ and the only arc of $B$ from $V(C')$ to $V(C)$ is $(a_1, b_t)$. So $B$ contains a subset $B_1$ of cycles covering $V(C) \cup \{a_1\}$ and a subset $B_2$ of cycles covering $V(C') \setminus \{a_1\}$. Thus the cycles of $B_1$ (resp. $B_2$) form a cycle factor of $D^{M'}[C' \setminus \{a_1\}]$ (resp. $D^{M'}[C \cup \{a_1\}]$) and we denote by $\tilde{B}_1$ (resp. $\tilde{B}_2$) the corresponding cycle factors of $D$. Moreover, for $i \in [\![s+1, t]\!]$, the arcs $(a_i, M(a_i))$ belongs to $D$ and as $M(a_i) = M'(a_{i+1})$ they link the cycles of $\tilde{B}_2$ in a strongly connected way. Thus $D[\tilde{B}_2]$ is strongly connected and so by Theorem 39, it has an Hamilton cycle $B_3$ on $2(p+1)$ vertices. In addition, if $p+1$ is even $D[B_3]$ is not isomorphic to $F_{2(p+1)}$ as it contains $C_1$ as a subdigraph, $C_1$ being a cycle on $2p$ vertices with $p$ odd. Therefore, $\tilde{B}_1 \cup B_3$ forms a cycle factor of $D$ with a cycle, $D[B_3]$, of length $2(p+1)$ not isomorphic to $F_{2(p+1)}$ and we can conclude with Lemma 38.1. □

The next claim is an easy case where we can insert a vertex of $C_2^M$ into $C_1^M$.

**Claim 40.2.** *If $C_2^M$ contains three consecutive vertices $a$, $b$ and $c$ and $C_1^M$ contains two consecutive vertices $x$ and $y$ such that $(a,c)$, $(x,b)$ and $(b,y)$ are arcs of $D^M$ then $D$ admits a good cycle factor.*

*Proof.* It is clear that using the arcs $(a,c)$, $(x,b)$ and $(b,y)$ we can form a 2-cycle factor of $D^M$, with one cycle of length $2k-(p+1)$ covering $C_2^M \setminus \{b\}$ and the other of length $p+1$ covering $C_1^M \cup \{b\}$. Let us denote by $\tilde{C}$ this latter one. If $p+1$ is even notice that the cycle of $D$ corresponding to $\tilde{C}$ cannot be isomorphic to $F_{2(p+1)}$. Indeed, otherwise $\tilde{C}$ would be isomorphic to a complete bipartite digraph but $\tilde{C}$ contains $C$ has a subdigraph which is a cycle on $p$ vertices with $p$ odd, a contradiction. □

Now we can prove the following claim, that we will intensively use in the remaining of the proof of Theorem 40.

**Claim 40.3.** *Assume that $\overline{D^M[C_2^M]}$ is not strongly connected and denote by $S_1,\dots,S_l$ its strongly connected components. If there exists an arc $(a,b)$ of $C_2^M$ such that there is an anti-path in $C_2^M$ from $b$ to $a$ and $a \in S_i$, $b \in S_j$ for $i \neq j$, then $D$ admits a good cycle factor.*

*Proof.* In $D^M$, we denote $C_2^M$ by $C'$ and $C_1^M$ by $C$. All the proof stands in $D^M$. First assume that there exists a vertex $c \in C$ such that $(a,c)$ and $(c,b)$ are anti-arcs, then the anti-cycle formed by the anti-path from $b$ to $a$ in $C'$ completed with the anti-arcs $(a,c)$ and $(c,b)$ satisfies the hypothesis of Claim 40.1 and we can conclude.

Hence, we assume that $\overline{N_C^+}(a) \cap \overline{N_C^-}(b) = \emptyset$. In particular, we have $\overline{d_C^+}(a) + \overline{d_C^-}(b) \leq p$. Let us denote by $A$ the set of all the vertices of $C'$ for whom there is a anti-path from $a$ to them, and by $B$ the set $C' \setminus A$. The set $A$ dominates the set $B$, we have $|A|+|B| = 2k-p$ and also $S_i \subseteq A$ and $S_j \subseteq B$ (otherwise $a$ and $b$ would have been in the same connected component of $\overline{D^M[C']}$). Therefore, we have

$$\begin{aligned} 2k-2 &= \overline{d^+}(a) + \overline{d^-}(b) \\ &\leq (\overline{d_C^+}(a) + \overline{d_C^-}(b)) + \overline{d_{C'}^+}(a) + \overline{d_{C'}^-}(b) \\ &\leq p + |A| - 1 + |B| - 1 = 2k-2 \end{aligned}$$

and thus we have equalities everywhere. In particular we obtain that $(A,B)$ is a partition of $V(C')$ with $A \setminus \{a\} \subseteq \overline{N^+}(a)$ and $B \setminus \{b\} \subseteq \overline{N^-}(b)$. As a consequence for every $x_A \in A$ and $x_B \in B$ there exists an anti-path from $x_B$ to $x_A$. Another consequence is that $\overline{d_C^+}(a) + \overline{d_C^-}(b) = p$ and that $C$ admits a partition into $\overline{N_C^+}(a)$ and $\overline{N_C^-}(b)$.

So let $b'$ be the successor of $b$ along $C'$, that is $b' = C'(b)$, and assume first that $b' \in B$. Hence, for every $x \in \overline{N_C^+}(a)$ the arc $(x,b)$ exists in $D^M$ (as we assume that $\overline{N_C^+}(a) \cap \overline{N_C^-}(b) = \emptyset$) and so $(b, C(x))$ is an anti-arc, otherwise we can insert $b$ into $C$ and shortcut the path $a, b, b'$ using Claim 40.2. Thus, we have

$C(\overline{N_C^+}(a)) \subseteq \overline{N_C^+}(b)$ and it leads to $\overline{d_C^+}(a) \leq \overline{d_C^+}(b)$. Hence, we have

$$\begin{aligned} 2k-2 = \overline{d^+}(a) + \overline{d^-}(b') &= \overline{d_C^+}(a) + |A| - 1 + \overline{d_C^-}(b') + \overline{d_B^-}(b') \\ &\leq \overline{d_C^+}(b) + \overline{d_C^-}(b') + |A| - 1 + |B| - 2 \end{aligned}$$

and then $\overline{d_C^+}(b) + \overline{d_C^-}(b') \geq p+1$. In particular, there exists $c \in C$ such that $(b,c)$ and $(c,b')$ are anti-arcs and we can apply Claim 40.1 to the anti-cycle $(b,b',c)$ to obtain a good cycle factor of $D$.

Now, we assume that we have $b' \in A$. And more generally we can assume that $C'$ has no two consecutive vertices lying in $B$. Indeed, otherwise considering a non empty path of $C'[B]$, $(v,w)$ its first arc and $u$ the predecessor of $v$ along $C'$ we can proceed as before with $a=u$, $b=v$ and $b'=w$. Symmetrically, $C'$ has no two consecutive vertices in $A$ and $C'$ alternates between $A$ and $B$. In particular, $p$ is even and we write $C' = (a_1, b_1, a_2, b_2, \ldots, a_{k-p/2}, b_{k-p/2})$ with $A = \{a_1, \ldots, a_{k-p/2}\}$ and $B = \{b_1, \ldots, b_{k-p/2}\}$ (indices in $C'$ will be given modulo $k-p/2$). By the above arguments we also have $A \setminus \{a_i\} \subseteq \overline{N^+}(a_i)$ and $B \setminus \{b_i\} \subseteq \overline{N^-}(b_i)$ implying that $A$ and $B$ are two independent sets of $D^M$.

Now, for every $i \in [\![1, k-p/2]\!]$, we denote by $B_i$ the set $\overline{N_C^-}(b_i)$ and by $A_i$ the set $\overline{N_C^+}(a_i)$. As $(a_i, b_i)$ is an arc of $C'$, as there exists an anti-path from $b_i$ to $a_i$ in $D[C']$ and as $a_i$ and $b_i$ do not belong to the same strongly connected component of $\overline{D^M[C']}$, we can argue as before with $a_i$ playing the role of $a$ and $b_i$ the one of $b$. In particular, we obtain that $(A_i, B_i)$ is a partition of $V(C)$ for every $i \in [\![1, k-p/2]\!]$ with $|A_i| = |B_i| = p/2$. Moreover, assume that there exists $x \in C$ and $i \in [\![1, k-p/2]\!]$ such that $(a_i, x)$ and $(x, b_{i+1})$ are anti-arcs. In this case, we modify $C'$ into the cycle $C''$ by replacing the subpath $a_{i-1}, b_{i-1}, a_i, b_i, a_{i+1}, b_{i+1}$ of $C'$ by $a_{i-1}, b_i, a_{i+1}, b_{i-1}, a_i, b_{i+1}$. Now, $(a_i, b_{i+1})$ is an arc of $C''$ and there exists an anti-path $P$ in $D^M[C'']$ from $b_{i+1}$ to $a_i$. So we can conclude by applying Claim 40.1 to the anti-cycle $(P, x)$, that is the anti-cycle defined by the vertices of $P$ and $x$. So it means that $A_i \cap B_{i+1} = \emptyset$ for every $i \in [\![1, k-p/2]\!]$. We deduce then than all the $A_i$ coincide as well as all the $B_i$ and in particular we have $A$ anti-dominates $A_i$ and $A_i$ dominates $B$ for every $i \in [\![1, k-p/2]\!]$.

To conclude, consider $s, t \in [\![1, k-p/2]\!]$ such that $(b_s, a_t)$ is an anti-arc (such an anti-arc exists since there exists an anti-path from $b$ to $A$). If there exits a anti-arc from $x_a \in A_t$ to $x_b \in B_t$, then we conclude with Claim 40.1 applied to the anti-cycle $(x_a, x_b, b_t, b_s, a_t)$. Otherwise it means that $A_t$ dominates $B_t$ and so as $A_t$ dominates $B$, we have $A_t$ anti-dominates $A$. But then we obtain $\{b_s\} \cup A_t \cup A \setminus \{a_t\} \subseteq \overline{N^-}(a_t)$ and $\overline{d^-}(a_t) \geq k$, a contradiction. □

### 7.3.2 Properties of the 2-Cycle Factor $(C_1, C_2)$

Now, we define four different properties of the 2-cycle factor $(C_1, C_2)$ of $D$. For this, we first need the next claim.

**Claim 40.4.** *Let $C$ be $C_1$ or $C_2$. Either ($\overline{D^{M_u}[C^{M_u}]}$ or $\overline{D^{M_d}[C^{M_d}]}$ has exactly one terminal strong component) or ($|C|$ is congruent to 0 modulo 4 and $D[C]$ is isomorphic to $F_{|C|}$). Similarly, either ($\overline{D^{M_u}[C^{M_u}]}$ or $\overline{D^{M_d}[C^{M_d}]}$ has exactly*

*one initial strong component) or ($|C|$ is congruent to 0 modulo 4 and $D[C]$ is isomorphic to $F_{|C|}$).*

*Proof.* We prove the statement for terminal strong components. The proof for initial ones is similar.

We denote by $D'$ the digraph $D[C]$ with bipartition $(S', T')$ where $S' = S \cap V(C)$ and $T' = T \cap V(C)$. So, assume that $\overline{D'^{M_u}}$ has at least two terminal strong components and denote by $A$ and $B$ two such components. In $D'^{M_u}$, it means that $A$ dominates $D'^{M_u} \setminus A$ and that $B$ dominates $D'^{M_u} \setminus B$. In $D'$, by considering that $A$ and $B$ are subsets of $S'$, we then have $C(A)$, the successors of the vertices of $A$ along $C$, dominates $S' \setminus A$ and $C(B)$ dominates $S' \setminus B$. Notice that $A$ and $B$ are two disjoint sets of $S'$ and that $C(A)$ and $C(B)$ are two disjoint sets of $T'$ with $|C(A)| = |A|$ and $|C(B)| = |B|$.

Now, assume that there exists a vertex $u$ of $C(A)$ such that $u' = C(u)$ does not belong to $B$. Then let $v$ be a vertex of $C(B)$. As $u' \notin B$, the arc $(v, u')$ belongs to $D'$ and then there exists an anti-arc from $u$ to $v$ in $D'^{M_d}$. Moreover, for every vertex $w$ of $T'$, if $C(w)$ does not belongs to $A$ then there is an anti-arc in $D'^{M_d}$ from $w$ to $u$ and if it belongs to $A$, then there is an anti-arc in $D'^{M_d}$ from $w$ to $v$. Thus, $\overline{D'^{M_d}}$ has exactly one initial strong component, containing $v$.

Otherwise, in $D'$ every vertex $u$ of $C(A)$ satisfies $C(u) \in B$ and similarly every vertex $v$ of $C(B)$ satisfies $C(v) \in A$. In particular, we have $|C(A)| \leq |B| = |C(B)| \leq |A| = |C(A)|$ and so $|A| = |B| = |C(A)| = |C(B)|$. Moreover, as $C$ contains all the vertices of $D'$, $(A, B)$ is a partition of $S'$ and $(C(A), C(B))$ is a partition of $T'$ with $C(A)$ dominates $B$ and $C(B)$ dominates $A$. In particular, $|C|$ is congruent to 0 modulo 4. As all the arcs of $C$ from $T'$ to $S'$ go from $C(A)$ to $B$ or from $C(B)$ to $A$ and that $C(A)$ dominates $B$ and $C(B)$ dominates $A$ in $D'$, the sets $C(A)$ and $C(B)$ respectively induce a complete digraph in $\overline{D'^{M_d}}$. As they form a partition of the vertex set of $\overline{D'^{M_d}}$, it has exactly one initial strong component except if there is no arc between $C(A)$ and $C(B)$. In this later case, it means that all the arcs from $A$ to $C(A)$ and from $B$ to $C(B)$ are contained in $D'$. But, then $D'[C]$ is isomorphic to $F_{|C|}$. □

In the case where $\overline{D^{M_u}[C_2^{M_u}]}$ has exactly one initial strong component, we say that the 2-cycle factor $(C_1, C_2)$ has property $\mathcal{Q}_{\text{up}}$. Similarly, in the case where $\overline{D^{M_d}[C_2^{M_d}]}$ has exactly one initial strong component, we say that $(C_1, C_2)$ has property $\mathcal{Q}_{\text{down}}$. By Claim 40.4, we know that either $(C_1, C_2)$ has property $\mathcal{Q}_{\text{up}}$ or $\mathcal{Q}_{\text{down}}$ or that $|C_2|$ is congruent to 0 modulo 4 and that $D[C_2]$ is isomorphic to $F_{|C_2|}$.

Now, let us define another pair of properties for the 2-cycle factor $(C_1, C_2)$. As $D$ is a bipartite tournament every vertex of $C_2$ has an in-neighbor or an out-neighbor in $C_1$. Moreover, as $D$ is $k$-regular and $|C_1| = 2p < 2k$ there exists at least one arc from $C_1$ to $C_2$ and one arc from $C_2$ to $C_1$. So, it is easy to check that there exists an arc $(u, v)$ of $C_2$ such that $N^-_{C_1}(u) \neq \emptyset$ and that $N^+_{C_1}(v) \neq \emptyset$. If we have $u \in S$ and $v \in T$ we say that $(C_1, C_2)$ has property $\mathcal{P}_{\text{down}}$. In this case, it means that the arc $(u, v)$ is an arc of $M_u$, and in $D^{M_u}$ the vertex $u$ has an in-neighbor and an out-neighbor in $C_1^{M_u}$. Otherwise, that is when we have $v \in S$ and $u \in T$, we say that $(C_1, C_2)$ has property $\mathcal{P}_{\text{up}}$. In this case, let $u'$ be

the predecessor of $u$ along $C_2$, that is $u = C_2(u')$. So, in $D^{M_u}$, $(u', v)$ is an arc of $C_2^{M_u}$, the vertex $u'$ has an anti-out-neighbor in $C_1^{M_u}$ and $v$ has an anti-in-neighbor in $C_1^{M_u}$.

**Claim 40.5.** *If $(C_1, C_2)$ does not satisfies $\mathcal{P}_{\text{down}}$, then in $D^{M_u}$ every vertex of $C_2^{M_u}$ anti-dominates $C_1^{M_u}$ or is anti-dominated by $C_1^{M_u}$.*

*Proof.* Let $x$ be a vertex of $C_2^{M_u}$. Since $D$ does not satisfies $\mathcal{P}_{\text{down}}$, then in $D$, either $C_2(x)$ is dominated by $C_1 \cap S$ or $x$ dominates $C_1 \cap T$. In the first case, it means that there is no arc from $x$ to $C_1^{M_u}$ in $D^{M_u}$, while in the latter, it means there is no arc from $C_1^{M_u}$ to $x$. □

Notice that if $(C_1, C_2)$ satisfies property $\mathcal{Q}_{\text{up}}$, then exchanging the role of $S$ and $T$ in $D$ (and then of $M_u$ and $M_d$) leads to $(C_1, C_2)$ satisfies property $\mathcal{Q}_{\text{down}}$, and conversely. We also have the similar property with $\mathcal{P}_{\text{up}}$ and $\mathcal{P}_{\text{down}}$.

Thus, without loss of generality, we assume that $(C_1, C_2)$ has the property $\mathcal{P}_{\text{up}}$. Then, we study, in this order, the three different cases: either $(C_1, C_2)$ has property $\mathcal{Q}_{\text{up}}$, or $D[C_2]$ is isomorphic to $F_{|C_2|}$, or $(C_1, C_2)$ satisfies property $\mathcal{Q}_{\text{down}}$.

### 7.3.3 Case A: $(C_1, C_2)$ has Property $\mathcal{Q}_{\text{up}}$

So, we know that $\overline{D^{M_u}[C_2^{M_u}]}$ has exactly one initial strong component. If it is not strong itself, there exists an arc of $C_2^{M_u}$ entering into its unique initial strong component and we can directly conclude with Claim 40.3. Then, we assume that $\overline{D^{M_u}[C_2^{M_u}]}$ is strongly connected. We consider several cases.

**Case 1 : $\overline{D^{M_u}[C_1^{M_u}]}$ is strongly connected too**. As $(C_1, C_2)$ has the property $\mathcal{P}_{\text{up}}$, there exist $u$ and $v$ in $D^{M_u}$ such that $v$ is the successor of $u$ along $C_2^{M_u}$, $u$ has an anti-out-neighbor $u'$ in $C_1^{M_u}$ and $v$ has an anti-in-neighbor $v'$ in $C_1^{M_u}$. As both $\overline{D^{M_u}[C_1^{M_u}]}$ and $\overline{D^{M_u}[C_2^{M_u}]}$ are strongly connected, there exists an anti-path from $v$ to $u$ in $D^{M_u}[C_2^{M_u}]$ and an anti-path from $u'$ to $v'$ in $D^{M_u}[C_1^{M_u}]$. So, we can form an anti-cycle in $D^{M_u}$ which satisfies the hypothesis of Claim 40.1 and conclude that $D$ contains a good cycle factor.

**Case 2 : $\overline{D^{M_u}[C_1^{M_u}]}$ is not strongly connected.** In what follows, to shorten the notation, we denote $C_1^{M_u}$ by $C$ and $C_2^{M_u}$ by $C'$. So, as $\overline{D^{M_u}[C]}$ is not strong, there exists a partition $(A, B)$ of $V(C)$ such that there is no anti-arcs from $A$ to $B$, thus we have $A$ dominating $B$ in $D^{M_u}$.

**Case 2.1 :** *There exist two vertices $a \in A$ and $b \in B$ such that there is an anti-path $P$ from $b$ to $a$ in $D^{M_u}[C]$.* So, in $D^{M_u}$ we have

$$\overline{d^+}(a) + \overline{d^-}(b) = 2k - 2$$

Since $\overline{d^+}(a) \leq |A| - 1 + \overline{d^+_{C'}}(a)$ and $\overline{d^-}(b) \leq |B| - 1 + \overline{d^-_{C'}}(b)$ and $|A| + |B| = p$, we get

$$\overline{d^+_{C'}}(a) + \overline{d^-_{C'}}(b) \geq 2k - p$$

If $\overline{d^+_{C'}}(a) + \overline{d^-_{C'}}(b) > 2k - p$ then there exist two vertices $a'$ and $b'$ in $C'$ such that $(a, a')$ and $(b, b')$ are anti-arcs, and $b'$ is the successor of $a'$ along $C'$. As $\overline{D^{M_u}[C']}$ is strongly connected, there exists an anti-path $Q$ from $a'$ to $b'$ in $D^{M_u}[C]$. So $P \cup Q$ forms an anti-cycle of $D^{M_u}$ satisfying the conditions of Claim 40.1 and we can conclude that $D$ contains a good cycle factor.

So, we assume that $\overline{d^+_{C'}}(a) + \overline{d^-_{C'}}(b) = 2k - p$. It implies that $\overline{d^+_A}(a) = |A| - 1$ and $\overline{d^-_B}(b) = |B| - 1$ and that $a$ anti-dominates $A$ and $B$ anti-dominates $b$. We deduce that for every $a' \in A$ and $b' \in B$ there exists an anti-path from $b'$ to $a'$ and applying the same arguments to $a'$ and $b'$ we can assume that $A$ and $B$ are both independent sets of $D^{M_u}$. As $C$ has then to alternate between $A$ and $B$, we have $|A| = |B| = p/2$, and in particular $p$ is even.

Now, let $B_1$ be the set of vertices in $V(C')$ which have an anti-out-neighbor in $B$. More formally, $B_1 = \{c \in V(C') \colon \exists b' \in B,\ (c, b') \text{ is an anti-arc}\}$. For any $b' \in B$ we have $\overline{d^-_{C'}}(b') = k - p/2$ implying that $|B_1| \geq k - p/2$. Similarly, we define $A_2 = \{c \in C' \colon \exists a' \in A, (a', c) \text{ is an anti-arc}\}$. We have the analogous result $|A_2| \geq k - p/2$. Recall that $C'(B_1)$ is the successors of the vertices in $B_1$ along $C'$ and denote by $A'_2$ the set of predecessors of $A_2$ along $C'$, that is $C'(A'_2) = A_2$. Let $b_1$ be a vertex in $B_1$ and $z$ the successor of $b_1$ along $C'$. Denote by $b'$ the vertex in $B$ such that $(b_1, b')$ is an anti-arc. If there is any $c \in A \cup B$ such that $(c, z)$ is an anti-arc, then we consider the anti-path $P$ from $b$ to $c$ in $D^{M_u}[C]$ and the anti-path $Q$ from $z$ to $b_1$ in $D^{M_u}[C']$ (which exists as $\overline{D^{M_u}[C']}$ is strongly connected). Then we conclude with Claim 40.1 applied on the anti-cycle $P \cup Q$. Now, to obtain a contradiction assume that $(A \cup B)$ dominates $C'(B_1)$ and, symmetrically, that $A'_2$ dominates $(A \cup B)$. In particular, we have $A_2 \cap C'(B_1) = \emptyset$ and as $|A_2| \geq k - p/2$ and $|C'(B_1)| = |B_1| \geq k - p/2$ we obtain that $(A_2, C'(B_1))$ is a partition of $V(C')$. But then, $(C_1, C_2)$ cannot satisfy property $\mathcal{P}_{\text{up}}$. Indeed, $C'$ should contains two vertices $u$ and $v$ such that $v$ is the successor of $u$ along $C'$, $u$ has an anti-out-neighbor in $C$ and $v$ has an anti-in-neighbor in $C$. Thus, we have $v \notin C'(B_1)$ and then $v \in A_2$. This implies that $u \in A'_2$ and that $u$ dominates $A \cup B$, contradicting the fact that $u$ has an anti-out-neighbor in $C$.

**Case 2.2 :** *For every $a \in A$ and $b \in B$, there are no anti-paths from $b$ to $a$.* Thus, the set $A$ dominates the set $B$ and $B$ dominates $A$. Suppose without loss of generality that $|B| \leq |A|$ and let $b$ be a vertex in $B$. If $\overline{d^+_{C'}}(b) + \overline{d^-_{C'}}(b) > 2k - p$, then we can find two vertices $u$ and $v$ in $C'$ such that $(u, b)$ and $(b, v)$ are anti-arcs, and $v$ is the successor of $u$ along $C'$. In that case we conclude with Claim 40.1 considering the anti-cycle $P \cup b$ where $P$ is an anti-path from $v$ to $u$ in $D^{M_u}[C']$ (which exists as $\overline{D^{M_u}[C']}$ is strongly connected). Therefore, we can assume that for every $b \in B, \overline{d^+_{C'}}(b) + \overline{d^-_{C'}}(b) \leq 2k - p$. Thus, we have

$$\begin{aligned}
\overline{d^+_B}(b) + \overline{d^-_B}(b) &= (\overline{d^+}(b) + \overline{d^-}(b)) - (\overline{d^+_{C'}}(b) + \overline{d^-_{C'}}(b)) \\
&\geq (2k - 2) - (2k - p) \\
&\geq p - 2
\end{aligned}$$

Finally, since $\overline{d^+_B}(b) + \overline{d^-_B}(b) \leq 2|B| - 2 \leq |A| + |B| - 2 = p - 2$, we have equalities

everywhere in the previous computation. In particular, we have $|A| = |B| = p/2$ and $p$ is even. Moreover, for every $b \in B$ we have $\overline{d_B^+}(b) = \overline{d_B^-}(b) = p/2 - 1$, implying that $B$ is an independent set. Symmetrically for $A$, as $|A| = |B|$ either we can conclude as previously with Claim 40.1 or $A$ is also an independent set. In this latter case, $C$ would induce a complete bipartite graph in $D^{M_u}$ and $D[C_1]$ would be isomorphic to $F_{2p}$, a contradiction to our induction hypothesis.

To conclude this section, notice that if $\overline{D^{M_u}[C_2^{M_u}]}$ has exactly one terminal strong component, then we can conclude similarly. Indeed, we have seen that if $\overline{D^{M_u}[C_2^{M_u}]}$ is strongly connected, then $D$ admits a good cycle factor, and if $\overline{D^{M_u}[C_2^{M_u}]}$ is not strong, then there exists an arc of $C_2^{M_u}$ leaving its unique terminal component and we can directly conclude with Claim 40.3.

### 7.3.4 Case B: $D[C_2]$ is Isomorphic to $F_{2k-p}$

As $D[C_1]$ is not isomorphic to $F_p$, by Claim 40.4 we can assume that $\overline{D^{M_u}[C^{M_u}]}$ has exactly one strong initial component. So, we contract $M_u$ and obtain the 2-cycle factor $(C_1^{M_u}, C_2^{M_u})$. As usual, we denote $C_1^{M_u}$ by $C$ and $C_2^{M_u}$ by $C'$. As $D[C_2]$ is isomorphic to $F_{2k-p}$, the digraph $D^{M_u}[C']$ is the complete bipartite digraph $K_{(k-p/2,k-p/2)}$. We denote by $(A, B)$ its bipartition.

**Claim 40.6.** *If there exists an arc $(a, b)$ of $C$ such that there is an anti-path in $C$ from $b$ to $a$, an anti-arc from $a$ to $A$ and an anti-arc from $A$ to $b$, then $D$ admits a good cycle factor.*

*Proof.* First denote by $a'$ the end of an anti-arc from $a$ to $A$ and by $b'$ the beginning of an anti-arc from $A$ to $b$. We can assume than there exists a vertex $c$ of $B$ such that $b', c, a'$ is a subpath of $C'$. Indeed, as $D^{M_u}[C']$ is isomorphic to a complete bipartite digraph, there exists a Hamilton cycle $C''$ of $D^{M_u}[C']$ starting with $b', c, a'$. We can then consider $C''$ instead of $C'$. So we assume than $a', c$ and $b'$ are consecutive along $C'$, and as $a'$ and $b'$ belong to $A$, $(a', b')$ is an anti-arc of $D^{M_u}$. So we perform a switch exchange along the anti-cycle formed by the anti-path from $b$ to $a$ in $C$, and the anti-arcs $(a, a')$, $(a', b')$ and $(b', b)$. By Lemma 40.1 and Corollary 6, it is easy to see that we obtain a new cycle factor $\mathcal{C}$ of $D^{M_u}$ containing the triangle $C_3 = (a', b, c)$, a cycle $C_s$ containing all the vertices of $C' \setminus \{a', c\}$ and other cycles included into $C$. As $D^{M_u}[C_s]$ is a bipartite complete digraph on $2k - p - 2$ vertices, we can replace it with two cycles $C'_s$ on $p - 2$ vertices and a cycle $C''_s$ on $2k - 2p$ vertices (with $2k - 2p \geq 2$). Now, as $C_3$ contains a vertex of $A$ and $C''_s$ a vertex of $B$, the union of these two cycles is strongly connected in $D^{M_u}$. So using Theorem 39 in $D^{M_u}$ there exists a cycle of length $p + 1$ spanning $C''_s \cup C_3$. Using this cycle and the cycles of $(\mathcal{C} \setminus \{C_3, C_s\}) \cup C'_s$ we form a cycle factor of $D^{M_u}$ with one being of length $p + 1$. In particular, as $p + 1$ is odd, this cycle factor of $D^{M_u}$ corresponds to a good cycle factor of $D$. □

As $A$ and $B$ have a symmetric role, Claim 40.6 still holds by replacing $A$ by $B$ in its statement. Then, let us consider the structure of $\overline{D^{M_u}[C]}$.

**Claim 40.7.** *If $\overline{D^{M_u}[C]}$ is strongly connected then $D$ admits a good cycle factor.*

*Proof.* Let $v$ be a vertex of $C$ and $u$ its predecessor along $C$. As $|C| = p < k$, the vertex $v$ is the beginning of an anti-arc which ends in $C'$, that is in $A$ or $B$. So, if we cannot conclude with Claim 40.6, it means than $v$ is dominated by $A$ or $B$. As this is true for every vertex $v$, by a direct counting argument, there exist a set $X_A$ of $p/2$ vertices of $C$ which are dominated by $A$ and a set $X_B$ of $p/2$ vertices which are dominated by $B$. Moreover, no vertex $x$ of $A$ dominates a vertex of $X_B$ (otherwise we would have $d_{D^{M_u}}(x) > k$) and no vertex of $B$ dominates a vertex of $X_B$. So, for every vertex $v$ of $C$ we have $d^-_C(v) = k - (k - p/2) = p/2$.

Using the same argument between a vertex $v$ and its successor along $C$ we obtain that every vertex $v$ of $C$ satisfies $d^+_C(v) = p/2$. It means that in $D$, the bipartite tournament $D[C_1]$ contains $2p$ vertices, is not isomorphic to $F_{2p}$ and satisfies $d^+(u) = d^-(u) = p/2$ for each of its vertex $u$.

Now, assume first that $p \geq 5$. By induction, provided that $p \geq 5$, the bipartite tournament $D[C_1]$ admits a 2-cycle factor $(C_{\text{ind}}, C'_{\text{ind}})$ with $C_{\text{ind}}$ being of length 6. In $D^{M_u}$, we have a 2-cycle factor $(F_{\text{ind}}, F'_{\text{ind}})$ with $F_{\text{ind}}$ being of length 3. Let $(x, y)$ be an arc of $F_{\text{ind}}$. As $d^+_{C'}(x) \geq k - p + 1 > 0$ and $d^-_{C'}(y) \geq k - p + 1 > 0$, there exist $x'$ and $y'$ in $C'$ such that $(x, x')$ and $(y', y)$ are arcs of $D^{M_u}$. Now, $D^{M_u}[C']$ being a complete bipartite digraph, it admits a 2-cycle factor $(C_s, C'_s)$ such that $C_s$ contains $x'$ and $y'$ and is of length $p - 2$. So, using Lemma 40.2, as $D^{M_u}[C_s \cup F_{\text{ind}}]$ is strongly connected, $D^{M_u}$ admits a good cycle factor.

To conclude, assume that $p = 4$. As every vertex $x$ of $C$ satisfies $d^+_C(x) = d^-_C(x) = 2$ and $\overline{D^{M_u}[C]}$ is strongly connected, it is easy to see that the anti-arcs of $C$ form a cycle of length 4. If $C$ contains a vertex $x$ such that $x$ has an in-neighbor $x'$ in $A$ and an out-neighbor $y'$ in $B$, then we form a good cycle factor of $D^{M_u}$ with : a cycle of length 3 in $C \setminus x$, a cycle of length $p + 1 = 5$ containing $x'$, $x$, $y'$ and another vertex $x''$ of $A$ and another vertex $y''$ of $B$, and a cycle of length $2k - 8$ covering $C' \setminus \{x', x'', y', y''\}$. Otherwise, it means that we can write $V(C) = \{a, a', b, b'\}$ such that the in- and out-neighborhood of $a$ and $a'$ in $C'$ are exactly $A$ and that the in- and out-neighborhood of $b$ and $b'$ in $C'$ are exactly $B$. Moreover, as every pair of vertices are linked by at least one arc in $D^{M_u}[C]$, we can assume that $(a, a')$ and $(b, b')$ are arcs of $D^{M_u}$. Then, as $k > p$, we have $2k - p \geq 6$ and we can select three vertices $a_1, a_2$ and $a_3$ in $A$ and three vertices $b_1, b_2$ and $b_3$ in $B$. Then, we form a good cycle factor with the two cycles $(a, a', a_1, b_2, a_3)$ and $(b, b', b_1, a_2, b_3)$ of length $p + 1 = 5$ and a cycle covering $C' \setminus \{a_1, a_2, a_3, b_1, b_2, b_3\}$. □

So, $\overline{D[C]}$ is not strongly connected but contains only one strong initial component $S_1$. Let $S_2$ be $V(C) \setminus S_1$. In particular, all the arcs from $S_2$ to $S_1$ exist in $D^{M_u}$. Moreover, there exists an arc $(x, y)$ of $C$ with $x \in S_2$ and $y \in S_1$. As $S_1$ is the only initial component of $\overline{D[C]}$ there exists an anti-path from $y$ to $x$ in $\overline{D[C]}$. As $|C| = p < k$, there exists a vertex $x'$ in $C'$ such that $(x, x')$ is an anti-arc of $D^{M_u}$. Without loss of generality, we can assume that $x' \in A$. By Claim 40.6, if $y$ has an anti-in-neighbor in $A$, then $D$ admits a good cycle factor. So we can assume that $A$ dominates $y$. Similarly, $y$ has an anti-in-neighbor in $C'$ which must be in $B$ and if we cannot conclude with Claim 40.6, it means that $x$ dominates $B$. As $x$ dominates also $S_1$ and $y$ is dominated by $S_2$, we have $|S_1| = |S_2| = p/2$

and the out-neighborhood of $x$ is exactly $B \cup S_1$ and the in-neighborhood of $y$ is exactly $A \cup S_2$. Now, assume that $z$, the successor of $y$ along $C$ is in $S_1$. Notice that $(z, y)$ is an anti-arc of $D^{M_u}$. As $S_2$ and $y$ dominate $z$, it has at least one anti-in-neighbor in $A$ and at least one anti-in-neighbor in $B$ (otherwise, we would have $d^-(z) \geq k - p/2 + p/2 + 1 = k + 1$). As $y$ has an anti-out-neighbor in $C'$, wherever it is, in $A$ or $B$, we can conclude with Claim 40.6. Otherwise, it means that the successor of $y$ along $C$ is in $S_2$. Symmetrically, the predecessor of $x$ along $C$ is in $S_1$. Repeating the argument, we conclude that $C$ alternates between $S_1$ and $S_2$ and that the conclusions we had for $x$ and $y$ respectively hold for all vertices of $S_1$ and $S_2$. So, we have $S_1$ is dominated by $A$ and $S_2$ and is an independent set of $D^{M_u}$, and $S_2$ dominates $B$ and $S_1$ and is also an independent set of $D^{M_u}$. We deduce that the in-neighborhood of $B$ is exactly $A \cup S_2$ and then the out-neighborhood of $S_1$ is $A \cup S_2$ also. In particular, $S_1$ dominates $S_2$ and there is no anti-path from $S_1$ to $S_2$, providing a contradiction, as $S_1$ is the only initial strong component of $\overline{D^{M_u}[C]}$.

### 7.3.5 Case C: $(C_1, C_2)$ Satisfies $\mathcal{Q}_{\text{down}}$

As we assume that we are not in Case A, the digraph $\overline{D^{M_u}[C_2^{M_u}]}$ has at least two initial strong components, and at least two terminal strong components (as noticed at the end of Case A).

Besides, if $(C_1, C_2)$ satisfies $\mathcal{P}_{\text{down}}$, then by exchanging the role of $S$ and $T$, we are in the symmetrical case of Case A. Then, we can assume that $(C_1, C_2)$ does not satisfy $\mathcal{P}_{\text{down}}$. Once again, we denote $C_1^{M_u}$ by $C$ and $C_2^{M_u}$ by $C'$. So, by Claim 40.5, for every vertex $x$ of $C'$ either there is no arc from $x$ to $C$ or there is no arc from $C$ to $x$.

Before concluding the proof of Theorem 40, we need the two following claims.

**Claim 40.8.** . *If $D^{M_u}[C']$ contains a vertex $x$ such that there is no arc between $x$ and $C$, then $D$ admits a good cycle factor.*

*Proof.* Otherwise, let $x$ be such a vertex and call $x, y, z, t$ the subpath of $C'$ of length 3 starting from $x$. We know that either there is no arc from $C$ to $t$ or no arc from $t$ to $C$. Assume that the latter holds. If there also exists an anti-arc from $C$ to $t$, then we can find three vertices $a$, $b$ and $c$ in $C$ such that $a, b, c$ is a subpath of $C$ of length 2 and that $\{a, x\}$ and $\{c, t\}$ are independent sets of $D^{M_u}$. So, we will exchange some small paths between $C$ and $C'$.

First, notice that $p > 3$. Indeed, if we denote by $A_C$ (resp.) $B_C$ the set of vertices of $C'$ which anti-dominate $C$ (resp. are anti-dominating by $C$), we have $V(C') = A_C \cup B_C$, $x \in A_C \cap B_C$ and then $|A_C| + |B_C| \geq |A_C \cup B_C| + 1 = |C'| + 1 = 2k - 2$. So, we have $|A_C| = |B_C| = k - 1$ and $A_C \cap B_C = \{x\}$. Moreover, we have $t \in A_c \setminus \{x\}$ and so $\overline{N^+}(c)$ contains $B_C \cup \{t\}$ of size $k$, a contradiction.

Now, to perform the path exchange, let us depict the situation in $D$: $C_1$ contains the path $a, C_1(a), b, C_1(b), c, C_1(c)$ and $C_2$ contains the path $x, C_2(x), y, C_2(y), z, C_2(z), t, C_2(t)$. Moreover, in $D$, $x$ dominates $V(C_1) \cap T$, $C_2(x)$ is dominated by $V(C_1) \cap S$, there is an arc from $c$ to $C_2(t)$ and an arc from $t$ to $C_1(c)$. So, we replace in $C_1$ the path $a, C_1(a), b, C_1(b), c, C_1(c)$ by $a, C_2(x),$

$y, C_2(y),\ z, C_2(z),\ t, C_1(c)$ to obtain the cycle $\tilde{C_1}$ and we replace in $C_2$ the path $x, C_2(x), y,\ C_2(y), z, C_2(z), t, C_2(t)$ by $x, C_1(a), b, C_1(b), c, C_2(t)$ to obtain the cycle $\tilde{C_2}$. The cycles $(\tilde{C_1}, \tilde{C_2})$ form a $(2(p+1), 4k-2(p+1))$-cycle factor of $D$. Moreover, $\tilde{C_1}$ is not isomorphic to $F_{2(p+1)}$. Indeed, as $p > 3$, the cycle $\tilde{C_1}$ contains the predecessor $u$ of $a$ along $C_1$ (which is not $C_1(c)$ then). Let call $d$ the vertex of $C_1$ with $C_1(d) = u$. As $C_2(x)$ is dominated by $V(C_1) \cap S$ in $D$, there is an arc from $d$ to $C_2(x)$. Thus, $\tilde{C_1}$ is not isomorphic to $F_{2(p+1)}$, as it contains the path $d, C_1(d), a, C_2(x)$ and the arc $(d, C_2(x))$ while $F_{2(p+1)}$ does not contain such a substructure.

So, we can assume that $C$ dominates $t$. Call by $S$ the strong component of $\overline{D^{M_u}[C']}$ containing $t$. There exists $S_{\text{term}}$ a terminal strong component of $\overline{D^{M_u}[C']}$ distinct from $S$. Let $S'$ be the union of the strong components of $\overline{D^{M_u}[C']}$ different from $S_{\text{term}}$ and $S$. As $S_{\text{term}}$ is a terminal strong component of $\overline{D^{M_u}[C']}$, for any vertex $u$ of $S_{\text{term}}$, we have $k-1-|C| \leq \overline{d^+_{C'}}(u) \leq |S_{\text{term}}|-1$ and then $|S_{\text{term}}| \geq k-p$ (noticed that a symmetrical reasonning holds for initial strong components also). Moreover, as $C$ and $S_{\text{term}}$ dominate $t$ we must have exactly $|S_{\text{term}}| = k-p$ and the in-neighborhood of $t$ is exactly $C \cup S_{\text{term}}$. In particular, $z$ belongs to $S_{\text{term}}$ and $S$ is the other terminal strong component of $\overline{D^{M_u}[C']}$. Moreover, as $S' \cup S$ has size $k$ and is dominated by $S_{\text{term}}$, then $S' \cup S$ is exactly the out-neighborhood of each vertex of $S_{\text{term}}$. In particular, there is no arc from $S_{\text{term}}$ to $C$. Now, if there is an anti-arc from $C$ to $z$, then we will conclude as previously using a small path exchange between $C$ and $C'$. Indeed, if such an anti-arc exists between a vertex $b'$ of $C$ and $z$, call $a'$ the predecessor of $b'$ along $C$. So, $\{a', x\}$ and $\{b', z\}$ are independent sets of $D^{M_u}$. Then, in $D$ we exchange the path $a', C(a'), b', C(b')$ of $C_1$ with the path $a', C'(x), y, C'(y), z, C'(z)$ to obtain the cycle $\tilde{C_1}$. Similarly, we exchange the path $x, C'(x), y, C'(y), z, C'(z)$ of $C_2$ with the path $x, C(a'), b', C'(z)$ to obtain the cycle $\tilde{C_2}$. Then, $(\tilde{C_1}, \tilde{C_2})$ forms a good cycle factor of $D$, as in particular, denoting by $c'$ the predecessor of $a'$ along $C$, $\tilde{C_1}$ contains in $D$ the path $c', C(c'), a', C'(x)$ and the arc $(c', C'(x))$ and so $\tilde{C_1}$ is not isomorphic to $F_{2(p+1)}$ which does not contain such a substructure.

So, we assume that $C$ dominates $z$. As $z$ is dominated by $S$ (recall that $S$ is a terminal strong component of $\overline{D^{M_u}[C']}$) and $S$ has size at least $k-p$, then we have that $S \cup C$ is exactly the in-neighborhood of $z$ and that $|S| = k-p$. Finally, $S'$ is non empty (of size $2k-2p$) and there is no arc from $S'$ to $\{z, t\}$. To conclude, let $(u, v)$ be an arc of $C'$ with $u \in S_{\text{term}} \cup S$ and $v \in S'$ (such an arc exits as $S' \neq \emptyset$). By the previous arguments there exists an anti-path from $v$ to $u$ in $\overline{D^{M_u}[C']}$ and by Claim 40.3, we conclude that $D$ admits a good cycle factor. □

The last claim will show that every arc of $C'$ is contained in a digon.

**Claim 40.9.** *If $D^{M_u}[C']$ contains an arc $(x, y)$ such $(y, x)$ is not an arc of $D^{M_u}$, then $D$ admits a good cycle factor.*

*Proof.* Assume that $(x, y)$ is an arc of $C'$ such that $(y, x)$ is an anti-arc of $D^{M_u}$. If $x$ and $y$ are not in the same strong component of $\overline{D^{M_u}[C']}$, then we conclude with Claim 40.3. Otherwise, we assume that $x$ and $y$ lie in a same strong

component $S$ of $\overline{D^{M_u}[C']}$. In $D^{M_u}$, there exist an initial strong component $S_{\text{init}}$ and a terminal strong component $S_{\text{term}}$ of $\overline{D^{M_u}[C']}$ which are different from $S$ (with possibly $S_{\text{init}} = S_{\text{term}}$). As previously noticed, we have $|S_{\text{init}}| \geq k - p$ and $|S_{\text{term}}| \geq k - p$. And as $S_{\text{init}}$ and $S_{\text{term}}$ are different from $S$ then, $S$ dominates $S_{\text{init}}$ and is dominated by $S_{\text{term}}$. In particular, as $x$ and $y$ lie in $S$ and $(x, y)$ is an arc of $C'$, $x$ has at least an anti-out-neighbor $x'$ in $C$ and $y$ has at least an anti-in-neighbor $y'$ in $C$ (otherwise, considering $S_{\text{init}}$ or $S_{\text{term}}$ we would have $d^+_{D^{M_u}}(x) \geq k + 1$ or $d^-_{D^{M_u}}(y) \geq k + 1$). If $(x', y)$ or $(x, y')$ is not an arc of $D^{M_u}$ then we conclude with Claim 40.1, using an anti-path from $y$ to $x$ in $S$. So we assume that $(x', y)$ and $(x, y')$ are arcs of $D^{M_u}$. Similarly, if we have $\overline{d^+_C(x)} + \overline{d^-_C(y)} > p$ then there exists a vertex $u$ in $C$ such that $(x, u)$ and $(u, y)$ are anti-arcs of $D^{M_u}$ and we conclude with Claim 40.1. Thus we assume that we have $\overline{d^+_C(x)} + \overline{d^-_C(y)} \leq p$ and then that $\overline{d^+_{C'}(x)} + \overline{d^-_{C'}(y)} \geq 2k - p$. If there is no vertex $z$ such that $(y, z, x)$ is an anti-cycle, then we can partition $V(C') \setminus \{x, y\}$ into two sets $X$ and $Y$ such that $x$ anti-dominates $X$ and $Y$ anti-dominates $y$. Call $\mathcal{S}_X$ the set of strongly connected components of $\overline{D^{M_u}[C']}$ which are included into $X$. Notice that $\mathcal{S}_X$ is not empty as it contains all the terminal strong component of $\overline{D^{M_u}[C']}$, but does not contain any initial strong component of $\overline{D^{M_u}[C']}$ . Now, consider any arc $(u, v)$ of $C'$ going from a component $S_1$ of $\mathcal{S}_X$ to a component $S_2$ not belonging to $\mathcal{S}_X$. As $S_2$ contains a vertex of $\{x, y\} \cup Y$, there exists an anti-path from $v$ to $x$. And as $S_1$ belongs to $\mathcal{S}_X$, there is an anti-arc from $x$ to $u$. So using Claim 40.3, we can conclude that $D$ admits a good cycle factor.

Thus, we assume that there exists a vertex $z$ such that $(y, x, z)$ is an anti-cycle of $D^{M_u}$. Let us prove first that there is no arc between $z$ and $C$. If it is not the case, assume without loss of generality, that there is an arc $(z, z')$ from $z$ to $C$. We will perform a switch exchange along the anti-cycle $(y, x, z)$ and show that the 2-cycle factor that we obtain will satisfy $\mathcal{P}_{\text{down}}$ and $\mathcal{Q}_{\text{down}}$. Denote by $M'_u$ the perfect matching of $D$ obtain from $M_u$ by switch exchange along $(y, x, z)$ (that is $M'_u = (M_u \setminus \{(x, M_u(x)), (y, M_u(y)), (z, M_u(z))\}) \cup \{(x, M_u(y)), (y, M_u(z)), (z, M_u(x))\}$). So, when performing the switch exchange along $(y, x, z)$, by Lemma 40.1, we obtain $N^+_{D^{M'_u}}(x) = N^+_{D^{M_u}}(y)$, $N^+_{D^{M'_u}}(y) = N^+_{D^{M_u}}(z)$ and $N^+_{D^{M'_u}}(z) = N^+_{D^{M_u}}(x)$. In $D^{M_u}$, call by $P_1$ the subpath of $C'$ going from the successor of $y$ (along $C'$) to the predecessor of $z$ and by $P_2$ the subpath of $C'$ going from the successor of $z$ to the predecessor of $x$. Then, after the switch exchange, the cycle $C'$ becomes in $D^{M'_u}$ the cycle $C'' = (x, P_1, z, y, P_2)$. We denote by $C'_2$ its corresponding cycle in $D$. Notice that the strong components of $\overline{D^{M'_u}[C'']}$ are the same than the ones of $\overline{D^{M_u}[C']}$. Indeed, as the anti-cycle $(y, x, z)$ becomes the anti-cycle $(x, y, z)$ in $D^{M'_u}$, the permutation of the anti-out-neighborhoods of $x$, $y$ and $z$ does not affect the strong components of $\overline{D^{M_u}[C']}$ and their relationships. So, $(C_1, C'_2)$ still satisfies $\mathcal{Q}_{\text{down}}$. Finally, remind that in $D^{M_u}$, the vertex $z$ has an out-neighbor $z'$ in $C$ and $y$ an in-neighbor $y'$ in $C$. As we have $N^+_{D^{M'_u}}(y) = N^+_{D^{M_u}}(z)$, the arcs $(y', y)$ and $(y, z')$ belong to $D^{M_u}$. Thus, $(C_1, C'_2)$ now satisfies $\mathcal{P}_{\text{down}}$ and we can conclude with the symmetrical case of Case A.

To finish, we can assume that there is no arc between $z$ and $C$ and we use Claim 40.8 to conclude. □

Finally we can assume that every arc of $C'$ is in a digon. We write $C' = (u_1, \ldots, u_\ell)$ with $\ell = 2k - p$. The indices of vertices of $C'$ will be given modulo $\ell$. Then we consider two cases:

**Case 1: $p$ is odd.** Then $\ell = 2k - p$ is also odd. As $\overline{D^{M_u}[C']}$ is not strongly connected, there exists $i \in [\![1, \ell]\!]$ such that $(u_i, u_{i-2})$ is an arc of $D^{M_u}$. Without loss of generality, we can assume that $(u_\ell, u_{\ell-2})$ is an arc of $D^{M_u}$. We consider then the set $X = \{u_1, u_5, u_7, \ldots, u_{\ell-p+1}\}$, that is all the vertices $u_i$ with odd $i$ between 1 and $\ell - p + 1$, except $u_3$. Notice that $X$ has size $(\ell - p + 1 - 1)/2 - 1 = k - p$. If there is no arc between $X$ and $u_3$, as $u_3$ has no arc to $C$ or no arc from $C$, we would have $\overline{d^+_{D^{M_u}}}(u_3) \geq k$ or $\overline{d^-_{D^{M_u}}}(u_3) \geq k$, a contradiction. So, there exists an arc between $u_3$ and some $u_i \in X$. If $i = 1$, then we consider the cycle factor $\mathcal{C}'$ on $2k - p - 1$ vertices containing $C$ and the cycles with vertex sets $\{u_1, u_2, u_3\}, \{u_4, u_5\}, \ldots, \{u_{\ell-p-2}, u_{\ell-p-1}\}$ and the cycle factor $\mathcal{C}$ on $p + 1$ vertices containing the cycles with vertex set $\{u_{\ell-p}, u_{\ell-p+1}\}, \{u_{\ell-p+2}, u_{\ell-p+3}\}, \ldots, \{u_{\ell-1}, u_\ell\}$. Notice that $D^{M_u}[u_{\ell-p}, \ldots, u_\ell]$ is strongly connected and is not a complete bipartite graph, as it contains the cycle $(u_{\ell-2}, u_{\ell-1}, u_\ell)$. So, by Lemma 40.2, the digraph $D$ admits a good cycle factor. Now, if $i = u_{\ell-p+1}$, then we consider the cycle factor $\mathcal{C}$ on $p + 1$ vertices containing the cycles with vertex sets $\{u_{\ell-p+2}, u_{\ell-p+3}\}, \ldots, \{u_{\ell-1}, u_\ell\}$, $\{u_1, u_2\}$, and the cycle factor $\mathcal{C}'$ on $2k - p - 1$ vertices containing $C$ and only the cycle with vertex set $\{u_3, u_4, \ldots, u_{\ell-p}, u_{\ell-p+1}\}$. We conclude as previously. Finally, if $i \in \{5, 7, \ldots, \ell - p - 1\}$, then we choose $\mathcal{C}$ to be the cycle factor on $p+1$ vertices containing the cycles with vertex set $\{u_{\ell-p}, u_{\ell-p+1}\}, \{u_{\ell-p+2}, u_{\ell-p+3}\}, \ldots, \{u_{\ell-1}, u_\ell\}$ and $\mathcal{C}'$ the one containing the cycles with vertex set $\{u_1, u_2\}, \{u_3, u_i\}, \{u_{i+1}, u_{i+2}\}, \ldots, \{u_{\ell-p-2}, u_{\ell-p-1}\}$. Once again, we conclude as previously.

**Case 2: $p$ is even.** If there exists an arc between two vertices at distance 2 along $C'$, then we will proceed almost as in the case where $p$ is odd. The difference here, is that $\mathcal{C}$ will contain cycles all of length 2 except one of length 3. Indeed, assume for instance that $(u_\ell, u_{\ell-2})$ is an arc of $D^{M_u}$. We consider once again the set $X = \{u_1, u_5, u_7, \ldots, u_{\ell-p+1}\}$ and show, as previously, that there exists an arc between a vertex $u_i$ of $X$ and $u_3$. If $i = 1$, then we consider the cycle factor $\mathcal{C}'$ on $2k - p - 1$ vertices containing $C$ and the cycles with vertex sets $\{u_1, u_2, u_3\}, \{u_4, u_5\}, \ldots, \{u_{\ell-p-2}, u_{\ell-p-1}\}$ and the cycle factor $\mathcal{C}$ on $p + 1$ vertices containing the cycles with vertex set $\{u_{\ell-p}, u_{\ell-p+1}\}, \{u_{\ell-p+2}, u_{\ell-p+3}\}, \ldots, \{u_{\ell-4}, u_{\ell-3}\}, \{u_{\ell-2}, u_{\ell-1}, u_\ell\}$. Once again, we conclude with Lemma 40.2 that $D$ admits a good cycle factor. The cases where $i = \ell - p + 1$ and where $i \in \{5, 7, \ldots, \ell - p - 1\}$ are similar to the corresponding cases where $p$ is odd.

Finally, assume that there is no arc between two vertices at distance 2 along $C'$. Then, we denote by $A$ the vertices $u_i$ with odd indices and by $B$ the vertices $u_i$ with even indices. The sets $A$ and $B$ form two strong connected components of $\overline{D^{M_u}[C']}$ and so their are both initial and terminal strong components of $\overline{D^{M_u}[C']}$. In particular, $D^{M_u}$ contains all the arcs from $A$ to $B$ and all the arcs from $B$ to $A$. The vertex $u_1$ must have a neighbor in $A$. Indeed, otherwise, as there is no arc from $u_1$ to $C$ or from $C$ to $u_1$, we will conclude that $\overline{d^+_{D^{M_u}}}(u_1) \geq k$ or $\overline{d^-_{D^{M_u}}}(u_1) \geq k$,

which is not possible. So, assume that $u_1$ is adjacent to a vertex $u_i$ of $A$. Then, instead $C'$ we consider the cycle $C'' = (u_1, u_2, u_i, u_4, \ldots, u_{i-1}, u_3, u_{i+1}, \ldots, u_\ell)$. As there exist all the possible arcs between $A$ and $B$, then all the arcs of $C''$ are contained in a digon and there exists an arc between two vertices at distance 2 along $C''$ (the arc $(u_1, u_i)$). Thus, we are in the previous case and we conclude that $D^{M_u}$ admits a good anti-cycle.

## 7.4 Concluding Remarks

We finish this paper with some conjectures about the problem of cycle factor in bipartite or multipartite tournaments. First of all, we have to mention the two related conjectures appearing in the original paper of Zhang *et al.* [157]. The first one adds a new hypothesis imposing an arc in the 2-cycle factor.

**Conjecture 8** (Zhang *et al.* [157])**.** *Let $D$ be a $k$-regular bipartite tournament, with $k$ an integer greater than 2. Let $uv$ be any specified arc of $D$. If $D$ is isomorphic neither to $F_{4k}$ nor to some other specified families of digraphs, then for every even $p$ with $4 \leq p \leq |V(D)| - 4$, $D$ has a cycle $C$ of length $p$ such that $D \setminus C$ is Hamiltonian and such that $C$ goes through the arc $uv$.*

The second conjecture, conversely, imposes that the cycles contain specific vertices.

**Conjecture 9** (Zhang *et al.* [157])**.** *Let $D$ be a $k$-regular bipartite tournament, with $k$ an integer greater than 2. Let $u$ and $v$ be two specified vertices of $D$. If $D$ is isomorphic neither to $F_{4k}$ nor to some other specified families of digraphs, then for every even $p$ with $4 \leq p \leq |V(D)| - 4$, $D$ has a cycle $C$ of length $p$ such that $D \setminus C$ is Hamiltonian and such that $C$ contains $u$ and $D \setminus C$ contains $v$.*

We can see that the regularity of the bipartite tournament in Theorem 38 is quite important since we can easily find an infinite family of bipartite tournament with $|d^+(u) - d^-(u)| \leq 1$ for every vertex $u$ and $|d^+(u) - d^+(v)| \leq 1$ for every pair of vertices $\{u, v\}$, which does not contain any cycle factor. For instance, for any $k \geq 1$ consider the bipartite tournament, inspired by $F_{4k}$, consisting of four independent sets $K$, $L$, $M$ and $N$ with $|K| = |N| = k$ and $|L| = |M| = k + 1$ with all possible arcs from $K$ to $L$, from $L$ to $M$, from $M$ to $N$ and from $N$ to $K$.

However, if the digraph is close to a regular bipartite tournament and contains a cycle factor, the following could hold.

**Conjecture 10.** *Let $D$ be a bipartite tournament which contains a 2-cycle factor and such that $|d^+(u) - d^-(u)| \leq 1$ for every vertex $u$ and $|d^+(u) - d^+(v)| \leq 1$ for every pair of vertices $\{u, v\}$. If $D$ is not isomorphic to $F_{4k}$ for some value $k$, then for every even $p$ with $4 \leq p \leq |V(D)| - 4$, the digraph $D$ has a $(p, |V(D)| - p)$-cycle factor.*

Let $D$ be a $c$-partite tournament, and denote by $I_1, \ldots, I_c$ its independent sets. We say that $D$ is *$k$-fully regular* if, for any distinct $i$ and $j$ with $i, j \in [\![1, c]\!]$,

$D[I_i \cup I_j]$ is a $k$-regular bipartite tournament. In particular, all the sets $I_i$ have size $2k$.

Yeo [153] proved that if $c \geq 5$ then, in every $c$-partite regular tournament $D$, every vertex is contained in a cycle of length $l$ for $l \in [\![3, |V(D)|]\!]$. He also conjectured the following.

**Conjecture 11** (Yeo [153])**.** *Every regular $c$-partite tournaments $D$, with $c \geq 4$, contains a $(p, |V(D)| - p)$-cycle factor for all $3 \leq p \leq |V(D)|$.*

According to our results, a weaker form of Yeo's Conjecture could be the following.

**Conjecture 12.** *Let $D$ be a $k$-fully regular $c$-partite tournament with $c \geq 5$. Then for every even $p$ with $4 \leq p \leq |V(D)| - 4$, $D$ has a $(p, |V(D)| - p)$-cycle factor.*

We can see that if $c$ is even and there is at least one pair $\{I_i, I_j\}$ such that $D[I_i, I_j]$ is not isomorphic to $F_{4k}$, then our result implies the Conjecture 12. However, the case where $c$ is odd still has to be proven.

# List of Figures

# Index

# Bibliography


[1] Faisal N. Abu-Khzam. Kernelization algorithms for $d$-hitting set problems. In Frank K. H. A. Dehne, Jörg-Rüdiger Sack, and Norbert Zeh, editors, *Algorithms and Data Structures, 10th International Workshop, WADS 2007, Halifax, Canada, August 15-17, 2007, Proceedings*, volume 4619 of *Lecture Notes in Computer Science*, pages 434–445. Springer, 2007.

[2] Faisal N. Abu-Khzam. A quadratic kernel for 3-set packing. In Jianer Chen and S. Barry Cooper, editors, *Theory and Applications of Models of Computation, 6th Annual Conference, TAMC 2009, Changsha, China, May 18-22, 2009. Proceedings*, volume 5532 of *Lecture Notes in Computer Science*, pages 81–87. Springer, 2009.

[3] Nir Ailon, Moses Charikar, and Alantha Newman. Aggregating inconsistent information: Ranking and clustering. *J. ACM*, 55(5):23:1–23:27, 2008.

[4] Islam Akaria and Raphael Yuster. Packing edge-disjoint triangles in regular and almost regular tournaments. *Discrete Mathematics*, 338(2):217–228, 2015.

[5] Noga Alon. Disjoint directed cycles. *J. Comb. Theory, Ser. B*, 68(2):167–178, 1996.

[6] Noga Alon. Ranking tournaments. *SIAM J. Discrete Math.*, 20(1):137–142, 2006.

[7] Noga Alon, Daniel Lokshtanov, and Saket Saurabh. Fast FAST. In Susanne Albers, Alberto Marchetti-Spaccamela, Yossi Matias, Sotiris E. Nikoletseas, and Wolfgang Thomas, editors, *Automata, Languages and Programming, 36th International Colloquium, ICALP 2009, Rhodes, Greece, July 5-12, 2009, Proceedings, Part I*, volume 5555 of *Lecture Notes in Computer Science*, pages 49–58. Springer, 2009.

[8] Noga Alon, Raphael Yuster, and Uri Zwick. Color-coding. *J. ACM*, 42(4):844–856, 1995.

[9] Brian Alspach. The wonderful Walecki construction. *Bull. Inst. Combin. Appl*, 52:7–20, 2008.

[10] Esther M. Arkin and Refael Hassin. On local search for weighted $k$-set packing. In Rainer E. Burkard and Gerhard J. Woeginger, editors,

*Algorithms - ESA '97, 5th Annual European Symposium, Graz, Austria, September 15-17, 1997, Proceedings*, volume 1284 of *Lecture Notes in Computer Science*, pages 13–22. Springer, 1997.

[11] Giorgio Ausiello. *Complexity and approximation: combinatorial optimization problems and their approximability properties.* Springer, 1999.

[12] Yandong Bai, Hao Li, and Weihua He. Complementary cycles in regular bipartite tournaments. *Discrete Mathematics*, 333:14–27, 2014.

[13] Yandong Bai and Yannis Manoussakis. On the number of vertex-disjoint cycles in digraphs. *arXiv e-prints*, page arXiv:1805.02999, May 2018.

[14] Jorgen Bang-Jensen and Gregory Gutin. *Digraphs - theory, algorithms and applications.* Springer, 2002.

[15] Jorgen Bang-Jensen and Carsten Thomassen. A polynomial algorithm for the 2-path problem for semicomplete digraphs. *SIAM J. Discrete Math.*, 5(3):366–376, 1992.

[16] Reuven Bar-Yehuda, Dan Geiger, Joseph Naor, and Ron M. Roth. Approximation algorithms for the vertex feedback set problem with applications to constraint satisfaction and bayesian inference. In Daniel Dominic Sleator, editor, *Proceedings of the Fifth Annual ACM-SIAM Symposium on Discrete Algorithms. 23-25 January 1994, Arlington, Virginia, USA.*, pages 344–354. ACM/SIAM, 1994.

[17] Piotr Berman and Marek Karpinski. On some tighter inapproximability results (extended abstract). In Jirí Wiedermann, Peter van Emde Boas, and Mogens Nielsen, editors, *Automata, Languages and Programming, 26th International Colloquium, ICALP'99, Prague, Czech Republic, July 11-15, 1999, Proceedings*, volume 1644 of *Lecture Notes in Computer Science*, pages 200–209. Springer, 1999.

[18] Jean-Claude Bermond and Carsten Thomassen. Cycles in digraphs — a survey. *Journal of Graph Theory*, 5(1):1–43, 1981.

[19] Jean-Claude Bermond and Carsten Thomassen. Cycles in digraphs — a survey. *Journal of Graph Theory*, 5(1):1–43, 1981.

[20] Stéphane Bessy, Marin Bougeret, R. Krithika, Abhishek Sahu, Saket Saurabh, Jocelyn Thiebaut, and Meirav Zehavi. Packing Arc-Disjoint Cycles in Tournaments. In Peter Rossmanith, Pinar Heggernes, and Joost-Pieter Katoen, editors, *44th International Symposium on Mathematical Foundations of Computer Science (MFCS 2019)*, volume 138 of *Leibniz International Proceedings in Informatics (LIPIcs)*, pages 27:1–27:14, Dagstuhl, Germany, 2019. Schloss Dagstuhl–Leibniz-Zentrum fuer Informatik.

[21] Stéphane Bessy, Marin Bougeret, and Jocelyn Thiebaut. Triangle packing in (sparse) tournaments: Approximation and kernelization. In Kirk Pruhs and

Christian Sohler, editors, *25th Annual European Symposium on Algorithms, ESA 2017, September 4-6, 2017, Vienna, Austria*, volume 87 of *LIPIcs*, pages 14:1–14:13. Schloss Dagstuhl - Leibniz-Zentrum fuer Informatik, 2017.

[22] Stéphane Bessy, Marin Bougeret, and Jocelyn Thiebaut. (arc-disjoint) cycle packing in tournament: classical and parameterized complexity. *CoRR*, abs/1802.06669, 2018.

[23] Stéphane Bessy, Fedor V. Fomin, Serge Gaspers, Christophe Paul, Anthony Perez, Saket Saurabh, and Stéphan Thomassé. Kernels for feedback arc set in tournaments. *J. Comput. Syst. Sci.*, 77(6):1071–1078, 2011.

[24] Stéphane Bessy, Nicolas Lichiardopol, and Jean-Sébastien Sereni. Two proofs of the Bermond-Thomassen conjecture for tournaments with bounded minimum in-degree. *Discrete Mathematics*, 310(3):557–560, 2010.

[25] Stéphane Bessy and Jocelyn Thiebaut. Complementary cycles in regular bipartite tournaments: a proof of manoussakis, song and zhang conjecture. *Electronic Notes in Discrete Mathematics*, 61:115–121, 2017.

[26] Péter Biró, David Manlove, and Romeo Rizzi. Maximum weight cycle packing in directed graphs, with application to kidney exchange programs. *Discrete Math., Alg. and Appl.*, 1(4):499–518, 2009.

[27] Andreas Björklund, Thore Husfeldt, and Sanjeev Khanna. Approximating longest directed paths and cycles. In Josep Díaz, Juhani Karhumäki, Arto Lepistö, and Donald Sannella, editors, *Automata, Languages and Programming: 31st International Colloquium, ICALP 2004, Turku, Finland, July 12-16, 2004. Proceedings*, volume 3142 of *Lecture Notes in Computer Science*, pages 222–233. Springer, 2004.

[28] Hans L. Bodlaender, Rodney G. Downey, Michael R. Fellows, and Danny Hermelin. On problems without polynomial kernels. *J. Comput. Syst. Sci.*, 75(8):423–434, 2009.

[29] Hans L. Bodlaender, Bart M. P. Jansen, and Stefan Kratsch. Cross-composition: A new technique for kernelization lower bounds. In Thomas Schwentick and Christoph Dürr, editors, *28th International Symposium on Theoretical Aspects of Computer Science, STACS 2011, March 10-12, 2011, Dortmund, Germany*, volume 9 of *LIPIcs*, pages 165–176. Schloss Dagstuhl - Leibniz-Zentrum fuer Informatik, 2011.

[30] John Adrian Bondy. Counting subgraphs a new approach to the Caccetta-Häggkvist conjecture. *Discrete Mathematics*, 165-166:71–80, 1997.

[31] John Adrian Bondy. Beautiful conjectures in graph theory. *Eur. J. Comb.*, 37:4–23, 2014.

[32] John Adrian Bondy and Uppaluri S. R. Murty. *Graph Theory with Applications*. Macmillan Education UK, 1976.

[33] John Adrian Bondy and Uppaluri S. R. Murty. *Graph Theory*, volume 244 of *Graduate Texts in Mathematics.* Springer-Verlag, Berlin, 2008.

[34] Darryn Bryant. Cycle decompositions of complete graphs. *London Mathematical Society Lecture Note Series*, 346:67, 2007.

[35] Louis Caccetta and Roland Häggkvist. *On minimal digraphs with given girth.* Department of Combinatorics and Optimization, University of Waterloo, 1978.

[36] Mao-cheng Cai, Xiaotie Deng, and Wenan Zang. An approximation algorithm for feedback vertex sets in tournaments. *SIAM J. Comput.*, 30(6):1993–2007, 2000.

[37] Mao-cheng Cai, Xiaotie Deng, and Wenan Zang. A min-max theorem on feedback vertex sets. *Math. Oper. Res.*, 27(2):361–371, 2002.

[38] Paul Camion. Chemins et circuits hamiltoniens des graphes complets. *Comptes Rendus Hebdomadaires des Séances de l'Académie des Sciences. Paris*, 249(21):2151–2152, 1959.

[39] Pierre Charbit, Stéphan Thomassé, and Anders Yeo. The minimum feedback arc set problem is NP-hard for tournaments. *Combinatorics, Probability & Computing*, 16(1):1–4, 2007.

[40] Guantao Chen, Ronald J. Gould, and Hao Li. Partitioning vertices of a tournament into independent cycles. *J. Comb. Theory, Ser. B*, 83(2):213–220, 2001.

[41] Guantao Chen, Jian Shen, and Raphael Yuster. Second neighborhood via first neighborhood in digraphs. *Annals of Combinatorics*, 7(1):15–20, Jun 2003.

[42] Jianer Chen, Yang Liu, Songjian Lu, Barry O'Sullivan, and Igor Razgon. A fixed-parameter algorithm for the directed feedback vertex set problem. *J. ACM*, 55(5):21:1–21:19, 2008.

[43] Yijia Chen, Jörg Flum, and Moritz Müller. Lower bounds for kernelizations and other preprocessing procedures. *Theory Comput. Syst.*, 48(4):803–839, 2011.

[44] Vasek Chvátal and Endre Szemerédi. Short cycles in directed graphs. *J. Comb. Theory, Ser. B*, 35(3):323–327, 1983.

[45] William W. Cohen, Robert E. Schapire, and Yoram Singer. Learning to order things. In Michael I. Jordan, Michael J. Kearns, and Sara A. Solla, editors, *Advances in Neural Information Processing Systems 10, [NIPS Conference, Denver, Colorado, USA, 1997]*, pages 451–457. The MIT Press, 1997.

[46] Vincent Conitzer. Computing slater rankings using similarities among candidates. In *Proceedings, The Twenty-First National Conference on Artificial Intelligence and the Eighteenth Innovative Applications of Artificial Intelligence Conference, July 16-20, 2006, Boston, Massachusetts, USA*, pages 613–619. AAAI Press, 2006.

[47] Marek Cygan. Improved approximation for 3-dimensional matching via bounded pathwidth local search. In *54th Annual IEEE Symposium on Foundations of Computer Science, FOCS 2013, 26-29 October, 2013, Berkeley, CA, USA*, pages 509–518, 2013.

[48] Marek Cygan, Fedor V. Fomin, Lukasz Kowalik, Daniel Lokshtanov, Dániel Marx, Marcin Pilipczuk, MichałPilipczuk, and Saket Saurabh. *Parameterized Algorithms*. Springer, 2015.

[49] Jean-Charles de Borda. Mémoire sur les élections au scrutin. *Histoire de l'Académie Royale des Sciences*, 1781.

[50] Maurits de Graaf, Alexander Schrijver, and Paul D. Seymour. Directed triangles in directed graphs. *Discrete Mathematics*, 110(1-3):279–282, 1992.

[51] Nathaniel Dean and Brenda J Latka. Squaring the tournament — An open problem. *Congressus Numerantium*, pages 73–80, 1995.

[52] Rina Dechter. Enhancement schemes for constraint processing: Backjumping, learning, and cutset decomposition. *Artif. Intell.*, 41(3):273–312, 1990.

[53] Rina Dechter and Judea Pearl. *The cycle-cutset method for improving search performance in AI applications*. University of California, Computer Science Department, 1986.

[54] Holger Dell and Dániel Marx. Kernelization of packing problems. In Yuval Rabani, editor, *Proceedings of the Twenty-Third Annual ACM-SIAM Symposium on Discrete Algorithms, SODA 2012, Kyoto, Japan, January 17-19, 2012*, pages 68–81. SIAM, 2012.

[55] Erik D. Demaine, Fedor V. Fomin, Mohammad Taghi Hajiaghayi, and Dimitrios M. Thilikos. Subexponential parameterized algorithms on bounded-genus graphs and $H$-minor-free graphs. *J. ACM*, 52(6):866–893, 2005.

[56] Erik D. Demaine and MohammadTaghi Hajiaghayi. The bidimensionality theory and its algorithmic applications. *Comput. J.*, 51(3):292–302, 2008.

[57] Michael Dom, Jiong Guo, Falk Hüffner, Rolf Niedermeier, and Anke Truß. Fixed-parameter tractability results for feedback set problems in tournaments. *J. Discrete Algorithms*, 8(1):76–86, 2010.

[58] Michael Dom, Daniel Lokshtanov, and Saket Saurabh. Incompressibility through colors and ids. In Susanne Albers, Alberto Marchetti-Spaccamela,

Yossi Matias, Sotiris E. Nikoletseas, and Wolfgang Thomas, editors, *Automata, Languages and Programming, 36th International Colloquium, ICALP 2009, Rhodes, Greece, July 5-12, 2009, Proceedings, Part I*, volume 5555 of *Lecture Notes in Computer Science*, pages 378–389. Springer, 2009.

[59] Rodney G. Downey and Michael R. Fellows. *Fundamentals of Parameterized Complexity.* Texts in Computer Science. Springer, 2013.

[60] Cynthia Dwork, Ravi Kumar, Moni Naor, and D. Sivakumar. Rank aggregation methods for the web. In *Proceedings of the Tenth International World Wide Web Conference, WWW 10, Hong Kong, China, May 1-5, 2001*, pages 613–622, 2001.

[61] Paul Erdös and Lajos Pósa. On independent circuits contained in a graph. *Canadian Journal of Mathematics*, 17:347–352, 1965.

[62] Guy Even, Joseph Naor, Baruch Schieber, and Madhu Sudan. Approximating minimum feedback sets and multicuts in directed graphs. *Algorithmica*, 20(2):151–174, 1998.

[63] Uriel Feige. Faster fast(feedback arc set in tournaments). *CoRR*, abs/0911.5094, 2009.

[64] Paola Festa, Panos M. Pardalos, and Mauricio G. C. Resende. *Feedback Set Problems*, pages 1005–1016. Springer, 2009.

[65] Dror Fidler and Raphael Yuster. Remarks on the second neighborhood problem. *Journal of Graph Theory*, 55(3):208–220, 2007.

[66] David C. Fisher. Squaring a tournament: A proof of Dean's conjecture. *Journal of Graph Theory*, 23(1):43–48, 1996.

[67] Fedor V. Fomin, Daniel Lokshtanov, Fahad Panolan, and Saket Saurabh. Efficient computation of representative families with applications in parameterized and exact algorithms. *J. ACM*, 63(4):29:1–29:60, 2016.

[68] Harold N. Gabow and Shuxin Nie. Finding a long directed cycle. In J. Ian Munro, editor, *Proceedings of the Fifteenth Annual ACM-SIAM Symposium on Discrete Algorithms, SODA 2004, New Orleans, Louisiana, USA, January 11-14, 2004*, pages 49–58. SIAM, 2004.

[69] Robert B. Gardner. Optimal packings and coverings of the complete directed graph with 3-circuits and with transitive triples. In *Proceedings of the Twenty-eighth Southeastern International Conference on Combinatorics, Graph Theory and Computing (Boca Raton, FL, 1997)*, volume 127, pages 161–170, 1997.

[70] Michael R. Garey and David S. Johnson. *Computers and Intractability: A Guide to the Theory of NP-Completeness.* W. H. Freeman, 1979.

[71] Ralucca Gera, Teresa W. Haynes, and Stephen T Hedetniemi. *Graph Theory: Favorite Conjectures and Open Problems–2.* Springer, 2018.

[72] Ralucca Gera, Stephen Hedetniemi, and Craig Larson. *Graph Theory: Favorite Conjectures and Open Problems–1*. Springer, 2016.

[73] Jonathan L. Gross and Jay Yellen. *Handbook of graph theory*. CRC press, 2004.

[74] Jiong Guo, Falk Hüffner, and Hannes Moser. Feedback arc set in bipartite tournaments is NP-complete. *Inf. Process. Lett.*, 102(2-3):62–65, 2007.

[75] Jiong Guo, Hannes Moser, and Rolf Niedermeier. *Iterative Compression for Exactly Solving NP-Hard Minimization Problems*, volume 5515 of *Lecture Notes in Computer Science*, pages 65–80. Springer, 2009.

[76] Rajesh Gupta, Rajiv Gupta, and Melvin A. Breuer. BALLAST: a methodology for partial scan design. In *Proceedings of the Nineteenth International Symposium on Fault-Tolerant Computing, FTCS 1989, Chicago, IL, USA, 21-23 June, 1989*, pages 118–125. IEEE Computer Society, 1989.

[77] Gregory Z. Gutin and Anders Yeo. Quasi-hamiltonicity: A series of necessary conditions for a digraph to be hamiltonian. *J. Comb. Theory, Ser. B*, 78(2):232–242, 2000.

[78] Roland Häggkvist. Hamilton cycles in oriented graphs. *Combinatorics, Probability & Computing*, 2:25–32, 1993.

[79] Roland Häggkvist and Yannis Manoussakis. Cycles and paths in bipartite tournaments with spanning configurations. *Combinatorica*, 9(1):33–38, Mar 1989.

[80] Peter Hamburger, Penny E. Haxell, and Alexandr V. Kostochka. On directed triangles in digraphs. *Electr. J. Comb.*, 14(1), 2007.

[81] Yahya Ould Hamidoune. A note on minimal directed graphs with given girth. *J. Comb. Theory, Ser. B*, 43(3):343–348, 1987.

[82] Frédéric Havet and Stéphan Thomassé. Median orders of tournaments: A tool for the second neighborhood problem and Sumner's conjecture. *Journal of Graph Theory*, 35(4):244–256, 2000.

[83] Frédéric Havet, Stéphan Thomassé, and Anders Yeo. Hoàng-Reed conjecture holds for tournaments. *Discrete Mathematics*, 308(15):3412–3415, 2008.

[84] Danny Hermelin and Xi Wu. Weak compositions and their applications to polynomial lower bounds for kernelization. In Yuval Rabani, editor, *Proceedings of the Twenty-Third Annual ACM-SIAM Symposium on Discrete Algorithms, SODA 2012, Kyoto, Japan, January 17-19, 2012*, pages 104–113. SIAM, 2012.

[85] Jan Hladký, Daniel Král', and Sergey Norin. Counting flags in triangle-free digraphs. *Combinatorica*, 37(1):49–76, 2017.

[86] Chính T. Hoàng and Bruce A. Reed. A note on short cycles in digraphs. *Discrete Mathematics*, 66(1):103 – 107, 1987.

[87] Sheng-Ying Hsiao. Fixed-parameter complexity of feedback vertex set in bipartite tournaments. In Takao Asano, Shin-Ichi Nakano, Yoshio Okamoto, and Osamu Watanabe, editors, *Algorithms and Computation - 22nd International Symposium, ISAAC 2011, Yokohama, Japan, December 5-8, 2011. Proceedings*, volume 7074 of *Lecture Notes in Computer Science*, pages 344–353. Springer, 2011.

[88] Russell Impagliazzo, Ramamohan Paturi, and Francis Zane. Which problems have strongly exponential complexity? *J. Comput. Syst. Sci.*, 63(4):512–530, 2001.

[89] Bill Jackson. Long paths and cycles in oriented graphs. *Journal of Graph Theory*, 5(2):145–157, 1981.

[90] Bart M. P. Jansen and Astrid Pieterse. Sparsification upper and lower bounds for graphs problems and not-all-equal SAT. 43:163–174, 2015.

[91] David S. Johnson. Approximation algorithms for combinatorial problems. *Journal of Computer and System Sciences*, 9(3):256 – 278, 1974.

[92] Donald B. Johnson. Finding all the elementary circuits of a directed graph. *SIAM J. Comput.*, 4(1):77–84, 1975.

[93] Reid K. Brooks. Two complementary circuits in two-connected tournaments. In B.R. Alspach and C.D. Godsil, editors, *Annals of Discrete Mathematics (27): Cycles in Graphs*, volume 115 of *North-Holland Mathematics Studies*, pages 321 – 334. North-Holland, 1985.

[94] Yoshihiro Kaneko and Stephen C. Locke. The minimum degree approach for Paul Seymour's distance 2 conjecture. *Congressus Numerantium*, pages 201–206, 2001.

[95] Viggo Kann. *On the approximability of NP-complete optimization problems.* PhD thesis, Royal Institute of Technology Stockholm, 1992.

[96] Richard M. Karp. *Reducibility Among Combinatorial Problems*, pages 85–103. The IBM Research Symposia Series. Plenum Press, New York, 1972.

[97] Marek Karpinski and Warren Schudy. Faster algorithms for feedback arc set tournament, Kemeny rank aggregation and betweenness tournament. In Otfried Cheong, Kyung-Yong Chwa, and Kunsoo Park, editors, *Algorithms and Computation - 21st International Symposium, ISAAC 2010, Jeju Island, Korea, December 15-17, 2010, Proceedings, Part I*, volume 6506 of *Lecture Notes in Computer Science*, pages 3–14. Springer, 2010.

[98] Claire Kenyon-Mathieu and Warren Schudy. How to rank with few errors. In *Proceedings of the 39th Annual ACM Symposium on Theory of Computing, San Diego, California, USA, June 11-13, 2007*, pages 95–103, 2007.

[99] Subhash Khot. On the power of unique 2-prover 1-round games. In John H. Reif, editor, *Proceedings on 34th Annual ACM Symposium on Theory of Computing, May 19-21, 2002, Montréal, Québec, Canada*, pages 767–775. ACM, 2002.

[100] Thomas. P. Kirkman. On a problem in combinations. *Cambridge and Dublin Mathematical Journal*, 2:191–204, 1847.

[101] R. Krithika, Abhishek Sahu, Saket Saurabh, and Meirav Zehavi. The parameterized complexity of packing arc-disjoint cycles in tournaments. *CoRR*, abs/1802.07090, 2018.

[102] Daniela Kühn and Deryk Osthus. A survey on hamilton cycles in directed graphs. *Eur. J. Comb.*, 33(5):750–766, 2012.

[103] Daniela Kühn, Deryk Osthus, and Timothy Townsend. Proof of a tournament partition conjecture and an application to 1-factors with prescribed cycle lengths. *Combinatorica*, 36(4):451–469, 2016.

[104] Mithilesh Kumar and Daniel Lokshtanov. Faster exact and parameterized algorithm for feedback vertex set in bipartite tournaments. In Akash Lal, S. Akshay, Saket Saurabh, and Sandeep Sen, editors, *36th IARCS Annual Conference on Foundations of Software Technology and Theoretical Computer Science, FSTTCS 2016, December 13-15, 2016, Chennai, India*, volume 65 of *LIPIcs*, pages 24:1–24:15. Schloss Dagstuhl - Leibniz-Zentrum fuer Informatik, 2016.

[105] Mithilesh Kumar and Daniel Lokshtanov. Faster exact and parameterized algorithm for feedback vertex set in tournaments. In Nicolas Ollinger and Heribert Vollmer, editors, *33rd Symposium on Theoretical Aspects of Computer Science, STACS 2016, February 17-20, 2016, Orléans, France*, volume 47 of *LIPIcs*, pages 49:1–49:13. Schloss Dagstuhl - Leibniz-Zentrum fuer Informatik, 2016.

[106] Arno Kunzmann and Hans-Joachim Wunderlich. An analytical approach to the partial scan problem. *J. Electronic Testing*, 1(2):163–174, 1990.

[107] Daniela Kühn and Deryk Osthus. Hamilton decompositions of regular expanders: A proof of Kelly's conjecture for large tournaments. *Advances in Mathematics*, 237:62 – 146, 2013.

[108] Tien-Nam Le, Daniel Lokshtanov, Saket Saurabh, Stéphan Thomassé, and Meirav Zehavi. Subquadratic kernels for implicit 3-hitting set and 3-set packing problems. In Artur Czumaj, editor, *Proceedings of the Twenty-Ninth Annual ACM-SIAM Symposium on Discrete Algorithms, SODA 2018, New Orleans, LA, USA, January 7-10, 2018*, pages 331–342. SIAM, 2018.

[109] Dong-Ho Lee and Sudhakar M. Reddy. On determining scan flip-flops in partial-scan designs. In *IEEE/ACM International Conference on Computer-Aided Design, ICCAD 1990, Santa Clara, CA, USA, November 11-15, 1990. Digest of Technical Papers*, pages 322–325. IEEE Computer Society, 1990.

[110] Hao Li and Jinlong Shu. The partition of a strong tournament. *Discrete Mathematics*, 290(2/3):211–220, 2005.

[111] Nicolas Lichiardopol. A new bound for a particular case of the Caccetta-Häggkvist conjecture. *Discrete Mathematics*, 310(23):3368–3372, 2010.

[112] Nicolas Lichiardopol, Attila Pór, and Jean-Sébastien Sereni. A step toward the Bermond–Thomassen conjecture about disjoint cycles in digraphs. *SIAM J. Discrete Math.*, 23(2):979–992, 2009.

[113] Errol L. Lloyd and Mary Lou Soffa. On locating minimum feedback vertex sets. *J. Comput. Syst. Sci.*, 37(3):292–311, 1988.

[114] Daniel Lokshtanov, Pranabendu Misra, Joydeep Mukherjee, Geevarghese Philip, Fahad Panolan, and Saket Saurabh. A 2-approximation algorithm for feedback vertex set in tournaments. *CoRR*, abs/1809.08437, 2018.

[115] Daniel Lokshtanov, M. S. Ramanujan, and Saket Saurabh. A linear time parameterized algorithm for directed feedback vertex set. *CoRR*, abs/1609.04347, 2016.

[116] Édouard Lucas. Récréations mathématiques, vol. ii, gauthier-villars, paris, 1892. *Reprinted several times by Albert Blanchard, Paris.*

[117] Alessandro Maddaloni. *Algorithmic properties of (structured) digraphs with special emphasis on connectivity, paths and cycles.* PhD thesis, Syddansk Universitet. Det Naturvidenskabelige Fakultet, 2013.

[118] Yannis Manoussakis. *Problèmes extrémaux dans les graphes orientés.* PhD thesis, Université d'Orsay, 1987.

[119] William McCuaig. Intercyclic digraphs. 147:203–245, 1991.

[120] Pranabendu Misra, Venkatesh Raman, M. S. Ramanujan, and Saket Saurabh. A polynomial kernel for feedback arc set on bipartite tournaments. *Theory Comput. Syst.*, 53(4):609–620, 2013.

[121] Matthias Mnich, Virginia Vassilevska Williams, and László A. Végh. A 7/3-approximation for feedback vertex sets in tournaments. In Piotr Sankowski and Christos D. Zaroliagis, editors, *24th Annual European Symposium on Algorithms, ESA 2016, August 22-24, 2016, Aarhus, Denmark*, volume 57 of *LIPIcs*, pages 67:1–67:14. Schloss Dagstuhl - Leibniz-Zentrum fuer Informatik, 2016.

[122] John W. Moon. Topics on tournaments. 1968. *Holt, Reinhart and Winston.*

[123] John W. Moon. On subtournaments of a tournament. *Canadian Mathematical Bulletin*, 9(3):297–301, August 1966.

[124] George L. Nemhauser and Leslie E. Trotter Jr. Properties of vertex packing and independence system polyhedra. *Math. Program.*, 6(1):48–61, 1974.

[125] Tsuyoshi Nishimura. Short cycles in digraphs. *Discrete Mathematics*, 72(1-3):295–298, 1988.

[126] Christos H. Papadimitriou. *Computational complexity.* Addison-Wesley, 1994.

[127] Christos H. Papadimitriou and Mihalis Yannakakis. Optimization, approximation, and complexity classes. *J. Comput. Syst. Sci.*, 43(3):425–440, 1991.

[128] Viresh Patel, Daniela Kühn, John Lapinskas, and Deryk Osthus. Proof of a conjecture of Thomassen on Hamilton cycles in highly connected tournaments. *Proceedings of the London Mathematical Society*, 109(3):733–762, 05 2014.

[129] Michał Pawel Pilipczuk. *Tournaments and optimality: new results in parameterized complexity.* PhD thesis, 2013.

[130] Venkatesh Raman and Saket Saurabh. Parameterized algorithms for feedback set problems and their duals in tournaments. *Theor. Comput. Sci.*, 351(3):446–458, 2006.

[131] Bruce A. Reed, Neil Robertson, Paul D. Seymour, and Robin Thomas. Packing directed circuits. *Combinatorica*, 16(4):535–554, 1996.

[132] Prashant Sasatte. Improved approximation algorithm for the feedback set problem in a bipartite tournament. *Oper. Res. Lett.*, 36(5):602–604, 2008.

[133] Prashant Sasatte. Improved FPT algorithm for feedback vertex set problem in bipartite tournament. *Inf. Process. Lett.*, 105(3):79–82, 2008.

[134] Jeanette P. Schmidt and Alan Siegel. The spatial complexity of oblivious $k$-probe hash functions. *SIAM J. Comput.*, 19(5):775–786, 1990.

[135] Paul D. Seymour. Packing directed circuits fractionally. *Combinatorica*, 15(2):281–288, 1995.

[136] Alan C. Shaw. The logical design of operating systems. 1974.

[137] Jian Shen. Directed triangles in digraphs. *J. Comb. Theory, Ser. B*, 74(2):405–407, 1998.

[138] Jian Shen. On the girth of digraphs. *Discrete Mathematics*, 211:167–181, 2000.

[139] Jian Shen. On the caccetta–häggkvist conjecture. *Graphs and Combinatorics*, 18(3):645–654, Oct 2002.

[140] Zeng Min Song. Complementary cycles of all lengths in tournaments. *J. Comb. Theory, Ser. B*, 57(1):18–25, 1993.

[141] Ewald Speckenmeyer. On feedback problems in digraphs. In Manfred Nagl, editor, *Graph-Theoretic Concepts in Computer Science*, pages 218–231, Berlin, Heidelberg, 1990. Springer Berlin Heidelberg.

[142] Carsten Thomassen. Edge-disjoint hamiltonian paths and cycles in tournaments. *Proceedings of the London Mathematical Society*, 3(1):151–168, 1982.

[143] Carsten Thomassen. Disjoint cycles in digraphs. *Combinatorica*, 3(3):393–396, 1983.

[144] Carsten Thomassen. The 2-linkage problem for acyclic digraphs. *Discrete Mathematics*, 55(1):73–87, 1985.

[145] Timothy W. Tillson. A hamiltonian decomposition of $K^*_{2m}$, $2m \geq 8$. *J. Comb. Theory, Ser. B*, 29(1):68–74, 1980.

[146] Craig A. Tovey. A simplified NP-complete satisfiability problem. *Discrete Applied Mathematics*, 8(1):85–89, 1984.

[147] Anke van Zuylen. Linear programming based approximation algorithms for feedback set problems in bipartite tournaments. *Theor. Comput. Sci.*, 412(23):2556–2561, 2011.

[148] Anke van Zuylen and David P. Williamson. Deterministic pivoting algorithms for constrained ranking and clustering problems. *Math. Oper. Res.*, 34(3):594–620, 2009.

[149] Jianxin Wang, Qilong Feng, and Jianer Chen. An $O^*(3.53^k)$-time parameterized algorithm for the 3-set packing problem. volume 412, pages 1745–1753, 2011.

[150] JZ Wang and C Cai. Hamiltonian circuits of oriented graphs. *Chinese J. Oper. Res*, 1:64–65, 1982.

[151] Manuel Welhan. The Hoàng–Reed Conjecture for $\delta^+ = 3$. *Discrete Mathematics*, 310(13-14):1932–1939, 2010.

[152] Mingyu Xiao and Jiong Guo. A quadratic vertex kernel for feedback arc set in bipartite tournaments. *Algorithmica*, 71(1):87–97, 2015.

[153] Anders Yeo. Diregular $c$-partite tournaments are vertex-pancyclic when $c \geq 5$. *Journal of Graph Theory*, 32(2):137–152, 1999.

[154] D.H. Younger. Graphs with interlinked directed circuits. In *Proceedings of the Mid-west Symposium on Circuit Theory*, volume 2, 1973.

[155] Raphael Yuster. Packing triangles in regular tournaments. *Journal of Graph Theory*, 74(1):58–66, 2013.

[156] CQ Zhang. Every regular tournament has two arc-disjoint hamiltonian cycles. *J. Qufu Normal College*, pages 70–81, 1980.

[157] Ke Min Zhang, Yannis Manoussakis, and Zeng Min Song. Complementary cycles containing a fixed arc in diregular bipartite tournaments. *Discrete Mathematics*, 133(1-3):325–328, 1994.

[158] Ke Min Zhang and Zeng Min Song. Complementary cycles containing of fixed vertices in bipartite tournaments. *Appl. J. Chin. Univ.*, 3:401–407, 1988.

**Abstract**

In this thesis, we are interested in some algorithmic and structural problems of (oriented) cycle packing in dense digraphs. These problems are mainly motivated by understanding the structure of such graphs, but also because many algorithmic problems are easy (*i.e.* resolvable in polynomial time) on acyclic digraphs while they are NP-difficult in the general case.

More specifically, we first study the packing of cycles and the packing of triangles in tournaments. These problems are the two dual problems (from a linear programming point of view) of feedback arc/vertex set that have received a lot of attention in literature. Among other things, we show that there is no polynomial algorithm to find a maximum collection of cycles (respectively triangles) vertex or arc-disjoint in tournaments, unless $\mathsf{P} = \mathsf{NP}$. We are also interested in algorithms of approximations and parameterized complexity of these different problems.

Then, we study these problems in the specific case where the tournament admits a feedback arc set which is a matching. Such tournaments are said to be sparse. Surprisingly, the problem remains difficult in the case of vertex-disjoint triangles, but the packing of triangles and the packing of arc-disjoint cycles become polynomial. Thus, we explore the approximation and parameterized complexity of the vertex-disjoint case in sparse tournaments.

Finally, we answer positively to a structural conjecture on $k$-regular bipartite tournaments by Manoussakis, Song and Zhang from 1994. Indeed, we show that all digraphs of this non-isomorphic class to a particular digraph have for every $p$ even with $4 \leq p \leq |V(D)| - 4$ a $C$ cycle of size $p$ such that $D \setminus V(C)$ is hamiltonian.

**Résumé**

Dans cette thèse, nous nous intéressons à quelques problèmes algorithmiques et structurels du *packing* de cycles (orientés) dans les graphes orientés denses. Ces problèmes sont notamment motivés par la compréhension de la structure de tels graphes, mais également car de nombreux problèmes algorithmiques sont faciles (résolubles en temps polynomial) sur des graphes orientés acycliques alors qu'il sont NP-difficiles sur les graphes orientés en général.

Plus spécifiquement, nous étudions dans un premier temps le *packing* de cycles et le *packing* de triangles dans les tournois. Ces problèmes sont les duaux (d'un point de vue programmation linéaire) des problèmes de *feedback arc/vertex set* qui ont reçu beaucoup d'attention dans la littérature. Nous montrons entre autres qu'il n'y a pas d'algorithme polynomial pour trouver une collection maximale de cycles (respectivement triangles) sommet ou arc-disjointe dans les tournois, sauf si $\mathsf{P} = \mathsf{NP}$. Nous nous intéressons également aux algorithmes d'approximations et de complexité paramétrée de ces différents problèmes.

Nous étudions ensuite plus en détail ces problèmes dans le cas spécifique où le tournoi admet un *feedback arc set* qui est un couplage, appelé *sparse*. Étonnamment, le problème reste difficile dans le cas des triangles sommet-disjoints, mais devient polynomial pour les triangles et cycles arc-disjoints. Ainsi, nous explorons l'approximation et la complexité paramétrée du cas sommet-disjoints dans les tournois *sparses*.

Enfin, nous répondons positivement à une conjecture structurelle sur les bipartis complets $k$-réguliers par Manoussakis, Song et Zhang datant de 1994. En effet, nous démontrons que tous les digraphes de cette classe non isomorphes à un digraphe particulier possèdent pour tout $p$ pair avec $4 \leq p \leq |V(D)| - 4$ un cycle $C$ de taille $p$ tel que $D \setminus V(C)$ est hamiltonien.