A Verification Viewpoint to Network Congestion Games

Suman Sadhukhan

DOCTORAT BRETAGNE LOIRE / MATHSTIC

UNIVERSITÉ DE RENNES 1

THÈSE DE DOCTORAT DE

L'UNIVERSITÉ DE RENNES 1

ÉCOLE DOCTORALE N° 601
Mathématiques et Sciences et Technologies de l'Information et de la Communication
Spécialité : *Informatique*

Par
Suman Sadhukhan

A Verification Viewpoint on Network Congestion Games

Thèse présentée et soutenue à Rennes, France, le 09/12/2021
Unité de recherche : Inria Rennes

Rapporteurs avant soutenance :

Véronique BRUYÈRE	Professeure, Université de Mons, Belgique
Laurent DOYEN	Chargé de recherche CNRS, LMF, ENS Paris-Saclay, France

Composition du Jury :

Examinateurs :	Véronique BRUYÈRE	Professeure, Université de Mons, Belgique
	Laurent DOYEN	Chargé de recherche CNRS, LMF, ENS Paris-Saclay, France
	Hugo GIMBERT	Chargé de recherche CNRS, LaBRI, Université de Bordeaux, France
	Sophie PINCHINAT	Professeure, IRISA, Université de Rennes 1, France
	Arnaud SANGNIER	Maître de conférences, IRIF, Université de Paris, France
Directrice de thèse :	Nathalie BERTRAND	Directrice de recherche Inria, IRISA, Université de Rennes 1, France
Co-directeur de thèse :	Nicolas MARKEY	Directeur de recherche CNRS, IRISA, Université de Rennes 1, France
Encadrant de thèse :	Ocan SANKUR	Chargé de recherche CNRS, IRISA, Université de Rennes 1, France

Résumé de la thèse

Motivation

Lorsque nous nous rendons d'un endroit à un autre, nous voulons généralement éviter autant que possible les embouteillages, et nous avons souvent tendance à changer notre itinéraire prédestiné si nous voyons que les mises à jour sur notre itinéraire choisi montrent un trafic important. Mais il peut arriver que d'autres personnes, qui étaient déterminées à choisir le même itinéraire, voient la même mise à jour, et changent leur itinéraire pour celui que nous sommes en train de changer. Il en résulte une situation dans laquelle une route précédemment peu fréquentée devient plus fréquentée, et une route plus fréquentée peut devenir moins fréquentée qu'auparavant. Et, nous sommes de retour à la case départ.

Mais il peut y avoir des moyens de résoudre ce problème : (1) les navetteurs peuvent décider entre eux qui changera et qui ne changera pas, ou (2) la mise à jour peut venir directement aux navetteurs avec une solution de changement de manière à ce que s'ils suivent tous la suggestion, ils deviennent en fait meilleurs. On peut se demander comment les navetteurs communiquent et conduisent de manière coopérative afin de tirer profit de ce type de situation. Dans un avenir prévisible avec les véhicules automatisés [30, 61], on imagine que les communications de véhicule à véhicule (V2V) ou de véhicule à infrastructure (V2I) seront réalisables [46]. Dans cette thèse, notre préoccupation n'est pas de savoir comment ces communications sont réalisables, nous nous intéressons plutôt à des problèmes tels que, si une telle communication est effectivement établie, quel type de solution une telle prise de décision coopérative garantirait la réduction de l'effet de congestion, et comment synthétiser une telle planification des itinéraires pour les navetteurs, qui assure cette garantie.

Tout comme le trafic routier, un autre domaine où le contrôle de la congestion est pertinent dans notre vie quotidienne est l'*internet*. Comme nous passons de plus en plus de temps sur l'internet (surtout depuis que la pandémie nous a tous condamnés), nous rencontrons chaque jour un problème ou un autre. Un jour, nous devons éteindre notre appareil photo pendant une réunion en ligne entre deux personnes, tandis qu'un autre jour de chance, nous participons à un séminaire en ligne avec trente personnes. Télécharger un fichier aujourd'hui peut prendre deux fois plus de temps qu'hier pour la même taille. Nous sommes tous passés par là.

Bien que les raisons de ce phénomène puissent être multiples, comme l'indisponibilité ou la rupture de liaisons, les paquets de données peuvent traverser différents nœuds de réseau interne (routeurs), mais l'une des principales raisons des fluctuations de performances sur l'internet est la congestion du réseau. Le contrôle de la congestion est donc crucial dans l'Internet d'aujourd'hui. La forme prédominante de contrôle de la congestion est incarnée par le protocole de contrôle de la transmission (TCP) [32], qui fournit un moyen fiable de transmission des données de bout en bout, et évite également tout *congestion collapse*. Intuitivement, ce protocole fonctionne dans l'intérêt d'un réseau dans son ensemble (ou du moins d'une partie suffisamment grande), et par

conséquent, du point de vue d'un point d'extrémité, il n'est pas toujours dans son intérêt de se conformer au protocole. Cela donne lieu à une analyse de la théorie des jeux [1, 33, 34, 45] des protocoles de type TCP dans la littérature : Akella et al. [1], en utilisant l'analyse de la théorie des jeux de TCP, compare l'efficacité des comportements stables entre les variations récentes de TCP (par ex. SACK) avec les variations traditionnelles, et concluent que les variations récentes sont plus sensibles au comportement égoïste ; Lopez et al. [45] étudient le protocole basé sur les fontaines (FBP), et comparent les performances du mécanisme stable entre FBP et TCP en présence de congestion ; Kesselman et al. [34] mesurent la dégradation des performances à cause de la congestion dans un modèle de trafic de type TCP, et proposent une politique de détection précoce pour réduire la dégradation, et ce ne sont là que quelques exemples parmi une vaste littérature sur l'analyse de la théorie des jeux des protocoles de type TCP. Il s'agit là de quelques exemples tirés d'une vaste littérature sur l'analyse de la théorie des jeux des protocoles de type TCP. L'un des points communs entre toutes ces études est qu'elles portent sur les performances des comportements stables d'Internet dans des contextes différents mais spécifiques lorsque les utilisateurs finaux interagissent de manière égoïste et provoquent des congestions. De plus, dans ces contextes spécifiques, ces études se concentrent principalement sur la mesure du pire scénario d'un modèle sans vérifier directement les propriétés d'une instance de ces modèles.

En théorie, ce type de scénarios pratiques est souvent abstrait dans des jeux de graphes, et des problèmes similaires sont étudiés sur ces graphes, qui commentent, le plus souvent, les pires/meilleurs scénarios dans cette classe particulière de jeux. Nous en discutons davantage dans la section de ce chapitre. Dans cette thèse, nous considérons ce domaine d'un point de vue *méthodes formelles*, où au lieu de traiter le pire/meilleur scénario pour un modèle de jeu de congestion, nous sommes plus intéressés par le type de performance de stabilité qui peut être garanti dans une instance donnée de ce modèle. C'est pourquoi nous commençons par présenter les méthodes formelles et la manière dont elles abordent les questions de théorie des jeux, avant d'examiner plus en détail la manière dont nous abordons l'analyse des jeux de congestion dans une perspective de vérification.

Contexte

Méthodes formelles et théorie des jeux De nos jours, nous sommes fortement dépendants des systèmes informatisés : du smartphone à la navette spatiale, du distributeur automatique de billets au véhicule automatisé, du contrôle du trafic aux centrales nucléaires - la liste s'allonge chaque jour, tout comme notre dépendance à leur égard. Nous voulons que ces systèmes soient à l'abri des défaillances. Lorsqu'un système tombe en panne, nous sommes confrontés à des crises allant de pertes financières à des pertes de vies humaines. Par exemple, un défaut fatal dans le logiciel de contrôle [23] du missile Ariane-5 a provoqué un crash à peine 36 secondes après son lancement le 4 juin 1996. Un défaut de logiciel dans la partie contrôle de l'appareil de radiothérapie Therac-25 [44] a causé la mort de six patients cancéreux entre 1985 et 1987.

En informatique, *méthodes formelles* est un domaine dans lequel nous raisonnons formellement sur un tel système informatisé : nous vérifions les propriétés d'un système donné (*model checking*), et construisons des modèles qui sont corrects par construction (synthèse) à partir d'une spécification donnée. Dans les problèmes de model checking, une entrée consiste généralement en un système et une spécification logique (par exemple une formule en logique temporelle [54]), et le problème

demande si le système satisfait la spécification. Selon le type de formules considérées comme spécification, le problème de model checking demande si une ou toutes les exécutions du système en question satisfont la spécification.

En réalité, un concepteur/contrôleur peut ne contrôler que des parties du système entier, et il peut vouloir vérifier une spécification sur les exécutions du système qui implique ces parties. Par exemple, considérons un ascenseur dans un bâtiment comme le système, et la spécification est que lorsque quelqu'un appelle d'un certain étage, l'ascenseur doit s'arrêter à cet étage pour prendre cette personne. Cette spécification concerne une partie particulière du système, qui n'a presque rien à voir avec des exécutions telles que : ne pas rester coincé entre les étages, ouvrir la porte lorsqu'il s'arrête à un étage ou ne pas ouvrir la porte lorsqu'il passe d'un étage à l'autre, etc. Par conséquent, pour vérifier si l'ascenseur satisfait à la spécification, nous ne devons ni vérifier toutes les exécutions, c'est suffisant, ni vérifier une seule exécution, ce qui ne peut garantir que l'arrêt d'un seul appel.

Ces types de problèmes dans les méthodes formelles sont résolus en utilisant l'approche de la théorie des jeux : dans notre exemple, la personne qui appelle l'ascenseur est le *environnement*, et l'ascenseur lui-même est le système. Dans ces jeux à deux joueurs entre le système et l'environnement, l'environnement vise à montrer que le système est défectueux, tandis que le système tente d'établir qu'il ne l'est pas. En d'autres termes, le problème de vérification exige que le concepteur/contrôleur puisse faire en sorte que le système gagne contre tout comportement de l'environnement. Ces jeux sont appelés *jeux à somme nulle*, et une *solution* dans un tel jeu est une stratégie gagnante.

D'autre part, il est possible que les concepteurs des différents composants d'un système soient eux-mêmes en concurrence les uns avec les autres pour certaines ressources. Dans ce cas, au lieu de gagner ou de perdre le jeu, chaque joueur obtient un gain. Le gain peut être positif, ce que l'on appelle généralement une *récompense*, et il peut être négatif, ce que l'on appelle un *coût*. Contrairement à la victoire dans un jeu, l'objectif du joueur est ici d'optimiser son gain : maximiser s'il s'agit d'une récompense, minimiser s'il s'agit d'un coût. Ces jeux sont appelés *jeux à somme non nulle*.

De plus, dans un jeu à somme non nulle, l'objectif des concepteurs des différents composants d'un système peut être en contradiction directe avec l'intérêt général du système. Par conséquent, alors qu'une solution du point de vue du système est un *optimum social*, du point de vue d'un composant particulier, qui joue *égoïstement*, une solution serait un choix qui est le meilleur par rapport aux choix de tous les autres joueurs. Comme cela est vrai pour tous les composants, ils voudraient ensemble atteindre une certaine forme de stabilité dans leurs choix les uns par rapport aux autres ; et ce serait la solution de leur point de vue. Toutefois, cette notion de stabilité n'est pas unique.

Les équilibres de Nash (NE) [50], la notion de stabilité la plus courante, est le concept de solution qui est immunisé contre toute déviation d'un seul joueur à n'importe quel moment du jeu si tous les joueurs suivent leurs choix correspondants donnés par cette notion, depuis le début du jeu. Mais il existe également d'autres notions, qui sont étudiées en fonction de divers contextes. Par exemple, *Subgame perfect Equilibria* (SPE) (SPE) [51] est un concept de solution pour la stabilité, qui est plus pertinent dans un contexte où chaque joueur peut prendre des décisions *dynamiquement*, étape par étape, en voyant comment les autres joueurs réagissent. La stabilité est également assurée contre toute déviation d'un seul joueur à n'importe quelle étape du jeu, comme l'EN, sauf qu'elle l'est même si . *Equilibres forts*[5, 25] est une notion de stabilité qui est immunisée non seulement contre les déviations d'un seul joueur mais aussi contre les déviations coordonnées de plusieurs

joueurs, tandis que les équilibres de Stackelberg [26, 59] sont la solution stable lorsqu'il y a quelques joueurs dont l'objectif est égal à celui du système lui-même (non égoïstement), tandis que d'autres joueurs jouent dans leur propre intérêt.

Néanmoins, il n'est pas garanti que l'une ou l'autre notion d'ensemble stable de choix existe toujours dans tous les jeux à somme non nulle. Par conséquent, l'une des tâches dans ce contexte est toujours de trouver les notions qui conviennent le mieux à l'objectif, et si cette notion de stabilité existe dans cette classe particulière de jeux à somme non nulle.

Jeux de congestion et routage égoïste Dans cette thèse, nous nous intéressons à des jeux particuliers à somme non nulle, appelés jeux de congestion, joués sur un réseau. Les jeux de congestion modélisent le partage égoïste des ressources entre plusieurs joueurs [56]. Un cas particulier est celui des jeux de congestion jeux de congestion, dans lesquels les joueurs visent à l'acheminement du trafic à travers un réseau encombré. Dans jeux de congestion de réseau, chaque joueur choisit un ensemble de transitions, formant un chemin simple d'un état source à un état cible, et le Le coût d'une transition augmente avec sa charge, c'est-à-dire avec le nombre de joueurs qui l'utilisent. de joueurs qui l'utilisent. Dans la section suivante, nous passons en revue l'état de l'art sur les jeux de congestion de réseau.

Travaux connexes sur les jeux de congestion de réseau

Les jeux de congestion de réseau peuvent être classés en variantes atomiques et non atomiques. non atomiques. La sémantique non-atomique est appropriée pour de grandes populations de joueurs. de joueurs et est donc considérée comme un continuum. On considère alors des portions de la population qui appliquent des stratégies prédéfinies, et l'effet d'un joueur individuel l'effet d'un joueur individuel sur le coût des autres. Sur En revanche, dans les jeux atomiques, le nombre de joueurs est fixe, et chaque joueur peut avoir un impact significatif sur le coût supporté par les autres joueurs. Nous nous concentrons uniquement sur les jeux atomiques dans cette thèse.

Jeux de congestion du réseau. Réseau congestion, également appelés jeux atomiques de routage égoïste dans la littérature. dans la littérature, ont été étudiés pour la première fois par Rosenthal [56]. Ces jeux sont définis par un graphe dirigé, un nombre de paires de sommets de paires de sommets source-cible, et des fonctions de coût non décroissantes pour chaque arête du graphe. Pour chaque paire source-cible, un joueur doit choisir une route de la source au sommet cible. route de la source au sommet de la cible. Étant donné leur choix de chemins simples, le coût encouru par un joueur dépend du nombre d'autres joueurs qui choisissent des chemins partageant des arêtes avec lui. autres joueurs qui choisissent des chemins partageant des arêtes avec leur chemin, et des fonctions de coût de ces bords. Dans ce contexte, un équilibre de Nash de Nash associe chaque joueur à un chemin de telle sorte qu'aucun joueur n'a intérêt à s'en écarter. joueur n'est incité à dévier : il ne peut pas diminuer son coût en choisissant un chemin différent.

Rosenthal Rosenthal a prouvé que les jeux de congestion de réseau sont des jeux potentiels, de sorte que les équilibres de Nash existent toujours. existent toujours. Monderer et Shapley [48] ont étudié d'une manière plus générale jeux potentiels, et ont expliqué comment utiliser itérativement les meilleures stratégies de meilleure réponse pour converger vers un équilibre. Il est intéressant de

noter que, sous des hypothèses raisonnables sur les fonctions de coût, Bertsekas et Tsitsiklis ont établi qu'il existe une correspondance directe correspondance directe entre les équilibres du routage égoïste et du routage distribué du plus court chemin, qui distribué, qui est utilisé en pratique pour le routage des paquets dans les réseaux informatiques. réseaux informatiques [12]. Nous renvoyons le lecteur intéressé à citeroughgarden-chap2007 pour une introduction et de nombreux résultats de base sur les jeux de routage généraux.

Une question naturelle est de savoir si le routage égoïste est très différent d'une stratégie de routage décidée par un organisme centralisé. stratégie de routage décidée par une autorité centralisée. En d'autres termes, En d'autres termes, à quelle distance un optimum égoïste peut-il être de l'optimum social, dans lequel les les joueurs coopéreraient. La notion de prix de l'anarchie, d'abord proposée par Koutsoupias et Papadimitriou [39], est le rapport entre le pire coût d'un équilibre de Nash et le coût de l'optimum social.

Cela permet de mesurer à quel point les équilibres de Nash peuvent être mauvais. Dans le contexte des jeux de congestion de réseau, le prix de l'anarchie a été étudié pour la première fois par Suri *et al.* [60], en établissant une limite supérieure de $\frac{5}{2}$ lorsque toutes les fonctions de coût sont affines. Une limite supérieure affinée a été fournie par Awerbuch *et al.* [9]. Bornes sur la notion duale de prix de stabilité, qui est le rapport entre le coût d'un meilleur équilibre de Nash et le coût de l'optimum social ont également été étudiés pour les jeux de jeux de routage [3].

Aspects du timing. Plusieurs travaux ont étudié des raffinements de ce cadre. Dans [31], les auteurs étudient les jeux de congestion de réseau dans lesquels chaque bord est traversé avec une durée fixe indépendante de sa charge, tandis que le coût de chaque bord dépend de la charge. On dit donc que le modèle a des coûts dépendants du temps, puisque la charge dépend des moments où les joueurs traversent un bord donné. auxquels les joueurs traversent un bord donné. Les auteurs prouvent l'existence des équilibres de Nash par réduction au cadre de [56]. Une extension de ce cadre avec des automates temporisés et des horloges contraintes temporelles a été étudiée dans [6, 7]. peuvent être imposées dans le graphe.

La mise en place de durées fixes avec des coûts dépendant du temps est intéressant dans les applications où les joueurs partageant une ressource (un bord) voient leur qualité de service diminuer. ressource (un bord) voient leur qualité de service diminuer, alors que le temps d'utilisation de la ressource n'est pas affecté. la ressource n'est pas affecté [7]. Cela peut être le cas, par exemple dans certaines applications de télécommunication et de streaming multimédia. La temporisation apparaît également, par exemple, dans les applications [40, 53] où la charge affecte les temps de parcours et où l'objectif des joueurs est de minimiser le temps total de déplacement. D'autres travaux se concentrent sur les modèles de flux avec un aspect temporel [13, 37].

Jeux dynamiques de congestion de réseau. Dans les jeux classiques de congestion de réseau, les joueurs choisissent leurs stratégies (leurs chemins simples) en une seule fois. Cependant, il peut être intéressant de laisser les agents choisir leurs chemins *dynamiquement*, c'est-à-dire étape par étape, en observant les choix précédents des autres joueurs. Dans cet article, nous étudions les jeux de congestion de réseau avec des coûts dépendant du temps comme dans [31], mais avec des retards unitaires, et dans un cadre dynamique. Plus précisément, à chaque étape, chacun des joueurs sélectionne simultanément le bord qu'il veut prendre ; chaque joueur doit ensuite payer un coût qui dépend de la charge du bord qu'il a choisi. l'arête qu'il a choisie, et traverse cette

arête en une seule étape. Nous nommons ces jeux : jeux dynamiques de congestion de réseaux (dynamic NCG en abrégé) ; le comportement des joueurs dans ces jeux est formalisé au moyen de *stratégies*, indiquant aux joueurs ce qu'ils doivent jouer en fonction de la configuration actuelle. Remarquez que, puisque l'effet de congestion s'applique aux arêtes utilisées *simultanément* par plusieurs joueurs, prendre des cycles peut être intéressant dans un NCG dynamique, ce qui rend notre cadre plus complexe que la plupart des NCG. modèles [8, 31, 56, 57].

Un tel cadre dynamique a été étudié dans [8] pour les jeux d'allocation de ressources. d'allocation des ressources, qui étend [56] aux choix dynamiques.

Concepts de solution standard . Nous étudions les concepts de solution classiques sur les jeux dynamiques de congestion de réseau. réseau dynamique. Un profil de stratégie (*i.e.*, une fonction assignant une stratégie à chaque joueur) est un *équilibre de Nash* (NE) lorsque chaque stratégie unique est une réponse optimale aux stratégies des autres joueurs. d'autres termes, sous un tel profil stratégique, aucun joueur ne peut réduire ses baisser ses coûts en changeant unilatéralement ses stratégies. Remarquez que les NE ne doivent pas nécessairement exister en général, et lorsqu'ils existent, ils peuvent ne pas être uniques. être uniques. Dans le cadre des jeux dynamiques, les équilibres de Nash sont généralement mis en œuvre en utilisant des stratégies d'optimisation. généralement renforcés par des stratégies de punition, par lesquelles tout joueur joueur qui dévie sera puni par tous les autres joueurs une fois que la déviation a été détectée. Cependant, ces stratégies de punition peuvent aussi augmenter le coût encouru par les joueurs qui punissent, et ne constituent donc pas une menace crédible. ne constituent pas une menace crédible. *Subgame-Perfect Equilibria* (SPE) raffine NE et Nous abordons ce problème en exigeant que le profil stratégique soit un NE le long de tout jeu.

NE et SPE visent à minimiser le coût individuel de chaque joueur (sans se soucier des coûts des autres). sans se soucier des coûts des autres) ; dans un contexte de collaboration, les joueurs peuvent plutôt essayer de réduire le coût social, c'est-à-dire la somme des coûts encourus par tous les joueurs. encourus par tous les joueurs. Les profils stratégiques qui y parviennent sont appelés *social optima* (SO). De toute évidence, le coût social de NE et SPE ne peut être inférieur à celui de l'optimum social ; Le *prix de l'anarchie* mesure la gravité des comportements égoïstes par rapport aux comportements collaboratifs. comportements égoïstes par rapport aux comportements collaboratifs.

Parameterized network congestion games (jeux de congestion de réseau paramétrés) Tous les problèmes ci-dessus ont été abordés en partant du principe que tous les joueurs (et les contrôleurs, le cas échéant) connaissent le nombre de joueurs dans le réseau, ce qui est crucial pour concevoir un équilibre ou un optimum social. Comme une solution pour un certain nombre de joueurs apriori ne nous donne aucune information sur ce à quoi ressemblerait une solution pour un autre nombre de joueurs, nous devrions la résoudre à partir de zéro sur le même réseau chaque fois que le nombre de joueurs change.

L'étude de l'incertitude dans le trafic suscite un intérêt croissant. Une approche commune pour traiter ce problème consiste à généraliser le cadre déterministe au cadre stochastique avec un nombre inconnu de joueurs. Wang et al. [63] et Correa et al. [22] ont étudié le PoA pour les jeux de congestion non atomiques avec demandes stochastiques. Cominetti et al. [21] ont étudié le cadre atomique des jeux de congestion de Bernoulli dans lesquels chaque joueur participe au

jeu avec une probabilité indépendante, et ont trouvé que la limite supérieure de la PoA pour les modèles déterministes tient toujours. Cominetti et al. [20] a trouvé que de tels jeux de congestion de Bernoulli convergent vers un ensemble de jeux de Poisson au sens de Myerson[49], lorsque les probabilités de participation des joueurs tendent vers zéro. Wang et al. [62] étudie un autre modèle pour les jeux de congestion de réseaux atomiques avec des demandes stochastiques, dans lequel au lieu des probabilités de participation individuelles, ils considèrent les distributions du nombre de joueurs comme une connaissance commune, ce qui suit le cadre de l'incertitude de la population proposé par Myerson[49]. L'incertitude de la demande a également été considérée dans d'autres jeux tels que les jeux d'allocation de ressources par Ashlagi et al. [4].

En dehors des approches stochastiques, il existe d'autres méthodes pour calculer directement les équilibres dans un cadre paramétrique. Klimm et Wardo[36] ont trouvé que la complexité du calcul des équilibres dans les jeux de congestion atomique avec des fonctions paramétriques de coût-affine est PPAD-complète, en exprimant l'évolution des équilibres localement par une nouvelle matrice Laplacienne en bloc.

Nous essayons d'aborder le même problème avec le nombre de joueurs comme paramètre en exprimant le NE pour un grand nombre de joueurs en termes de NE pour un plus petit nombre de joueurs en utilisant une propriété semi-linéaire, ce qui évite au contrôleur, qui assigne un profil stable à tous les joueurs, de calculer le NE à partir de zéro. Malheureusement, le cas général reste ouvert dans cette thèse, et nous donnons des résultats préliminaires pour les graphes *series-parallèles*.

Contributions

Dans la première partie, nous adoptons un point de vue de complexité informatique pour étudier les jeux dynamiques de congestion des réseaux. jeux de congestion de réseaux dynamiques. Nous établissons d'abord la complexité du calcul de l'optimum Nous montrons que le calcul de l'optimum social est in PSPACE et NP-hard. Nous prouvons ensuite que les stratégies de meilleure réponse peuvent être calculées en temps polynomial. temps polynomial, et que les GCN dynamiques sont des jeux potentiels, montrant ainsi l'existence d'équilibres de Nash. montrant ainsi l'existence d'équilibres de Nash dans tout GNC dynamique ; ceci montre également que certains équilibres de Nash peuvent être calculés en temps pseudo-polynomial. Nous donnons ensuite un algorithme EXPSPACE (resp. 2EXPSPACE) algorithme pour décider de l'existence d'équilibres de Nash (resp. équilibres sous-jeux-parfaits) dont les coûts sociaux satisfont des limites données. Ce site nous permet de calculer le meilleur et le pire de ces équilibres, puis le prix de l'anarchie et le prix de la stabilité. Ces résultats sont publiés dans [11].

Notez que certaines des complexités élevées découlent du codage binaire du nombre de joueurs. codage binaire du nombre de joueurs, qui est le principal paramètre d'entrée. Par exemple, la complexité exponentielle de l'espace du NE contraint tombe à un temps pseudo-polynomial. temps pseudo-polynomial pour un nombre fixe de joueurs. Ce paramètre devient important puisque nous préconisons l'étude de problèmes de calcul, tels que le calcul des équations de Nash. problèmes de calcul, tels que le calcul des équilibres de Nash avec une donné. Nous croyons également que le calcul de valeurs précises pour le prix de l'anarchie et le prix de la stabilité est important. prix de l'anarchie et le prix de la stabilité est intéressant, plutôt que de plutôt que de fournir des limites sur l'ensemble de toutes les instances, comme dans le cas suivant *e.g.* [60].

Dans la deuxième partie, nous nous limitons aux arènes de jeu qui sont des graphes série-parallèles, et notre modèle au modèle communément étudié des jeux de congestion de réseau : sans stratégies dynamiques et sans effet de congestion non-synchrone dans le calcul du coût. Dans ce modèle, nous montrons que l'ensemble des profils NE est un ensemble semi-linéaire. De plus, l'ensemble semi-linéaire a une structure spéciale : tout vecteur période de l'ensemble semi-linéaire est un multiple d'un vecteur unique, que nous appelons le vecteur *shift*. De plus, nous émettons également une conjecture selon laquelle ce vecteur dit de décalage est le seul vecteur de période de l'ensemble semi-linéaire. Si cette conjecture s'avère vraie, nous montrons une manière très simple de représenter l'ensemble de tous les profils NE, et nous donnons un algorithme pour calculer un NE pour un grand nombre de joueurs en utilisant les vecteurs de décalage et le NE d'un nombre relativement plus petit de joueurs.

Outline

Ce document est divisé en deux parties. Dans la partie I, qui est basée sur [11], nous étudions les jeux dynamiques de congestion de réseau. Dans cette partie, au chapitre **??**, nous fixons les notations et adaptons les définitions de la littérature, qui seront utilisées dans les chapitres suivants. Dans le chapitre 2, nous considérons le problème de l'optimum social contraint, et résolvons le problème en PSPACE, tout en montrant que la borne inférieure est NP. Dans les chapitres 3 et 4, nous considérons deux concepts de solution : Les équilibres de Nash et les équilibres parfaits de sous-jeu, et nous résolvons le problème de décision sous contrainte pour chacun d'eux, ce qui nous permet de conclure la partie I. Cette partie est principalement basée sur [11].

Dans la partie II du document, nous considérons des jeux paramétrés de congestion de réseau sur des graphes série-parallèles, dans lesquels le nombre de joueurs est le paramètre. Ici, dans le chapitre 5, nous introduisons les définitions que nous utilisons spécifiquement pour cette partie, et dans le chapitre 6, notre objectif était d'obtenir un algorithme pour calculer les NE pour les NCG avec beaucoup de joueurs à partir des NE des NCG avec moins de joueurs. Nous énumérons quelques résultats dans ce sens, et nous formulons une conjecture qui, si elle est vraie, nous donne une représentation de l'ensemble de tous les NE avec un nombre quelconque de joueurs en termes d'un vecteur dit *shift* et de NE avec un certain nombre fini de joueurs.

Enfin, dans le chapitre 6.5, nous concluons en résumant brièvement tous les résultats, et en fournissant une liste de suivis immédiats et d'objectifs futurs.

Introduction

Motivation

When we commute from one place to another, usually we want to avoid as much traffic as possible, and we often tend to change our predestined route if we see updates on our chosen route show a heavy traffic. But it may happen that others too, who were determined to choose the same route, see the same update, and switch their route to the one that we are changing into. As a result a situation arises where a previously lighter traffic route becomes heavier, and a heavier traffic might become lighter than before. And, we are back to square one.

But there might be ways to tackle this problem: (1) the commuters may decide among themselves who will switch and who will not, or (2) the update might come to the commuters directly with a solution to switch in a way such that if they all follow the suggestion, they actually become better off. One might ask how the commuters communicate and drive *cooperatively* in order to benefit in situations like this. In a foreseeable future with automated vehicles [30, 61], both vehicle to vehicle (V2V) or vehicle to infrastructure (V2I) communications are imagined to be realizable [46]. In this thesis, our concern is not how these communications are realizable, rather we are interested in problems like if such a communication is indeed established what kind of solution such a cooperative decision making would guarantee in lowering congestion effect, and how to synthesize such a planning of routes for the commuters, which ensures the guarantee.

Like traffic on road, another area where congestion control is relevant in our daily life is the *internet*. As we keep spending more and more time on the internet (specially after the pandemic doomed us all), everyday we experience one problem or another. One day we need to turn off our camera during an online meeting between two people, while on another lucky day, we participate an online seminar with thirty people. Downloading a file today might take twice as much time as it had taken yesterday for the same size. We all have been there.

While there might be many facets of reasons for these happening, like - unavailability/broken links, packets that carry data may traverse different inner network nodes ('routers'), but one of the main reason behind the performance fluctuations on the internet happens due to *network congestion*. Thus, controlling congestion is crucial in today's Internet. The predominant form of congestion control is embodied in Transmission Control Protocol (TCP) [32], which provides a reliable way of end-to-end data transmission, and also avoiding any *congestion collapse*. Intuitively, this protocol works in the interest of a network as a whole (or at least of a sufficiently large part), and therefore from an end-point's perspective it might not always be in their best interest to comply with the protocol. This gives rise to game-theoretic analysis [1, 33, 34, 45] of TCP-like protocols in the literature: Akella et al. [1], using game-theoretic analysis of TCP, compare efficiency of stable behaviors between recent variations of TCP (e.g. SACK) with traditional variations, and conclude that recent variations are more susceptible to selfish behavior; Lopez et al. [45] study the Fountain Based Protocol (FBP), and compare the performance of stable mechanism between FBP

and TCP in the presence of congestion; Kesselman et al. [34] measure degradation in performance because of congestion in a TCP-like traffic model, and propose an early detection policy to reduce degradation, and these are the few examples from a vast literature on game theoretic analysis of TCP-like protocols. One of the commonalities between all these is that they are studying the performance of stable behaviors of the internet in different but specific contexts when end-users are interacting selfishly and causing congestion. Moreover, in those specific contexts these studies focus mainly on measuring worst-case scenario of a model without directly checking the properties of an instance of those models.

In theory, these kind of practical scenarios are often abstracted in graph games, and similar problems are studied on those graphs, which comment on, more often than not, worst/best-case scenarios in that particular class of games. We discuss more on this in Section of this chapter. In this thesis, we view this area from a *formal methods* perspective, where instead of addressing worst/best-case scenario for a congestion game model, we are more interested in what kind of stability performance can be guaranteed in a given instance of this model. Therefore, we start with introducing formal methods, how they address game theoretical questions, and then delve into more details how we approach analyzing congestion games from a verification perspective.

Background

Formal methods and Game theory In today's age, we are heavily dependent on computerized systems: from smartphone to space-shuttle, from ATM to automated vehicle, from traffic control to nuclear power plants - the list continues to grow each passing day, and so is our dependency on them. We want these systems to be immune to failure. When a system fails, we face crisis from financial loss to loss of lives. For example, a fatal defect in the control software [23] of the Ariane-5 missile caused a crash just within 36 seconds of its launch on June 4, 1996. A software flaw in the control part of the radiation therapy machine Therac-25 [44] caused the death of six cancer patients between 1985 and 1987.

In computer science, *formal methods* is a field where we formally reason about such computerized system: we verify properties for a given system (*model checking*), and build models which are correct by construction (synthesis) from a given specification. In model checking problems, an input usually consists of a system and a logical specification (for example a formula in temporal logic [54]), and the problem asks whether the system satisfies the specification. Depending on what kind of formulas are being considered as the specification, the model checking problem asks whether one or all executions of the system in question satisfies the specification.

In reality, a designer/controller may control only parts of the whole system, and they might want to verify a specification on the executions of the system which involves those parts. For instance, consider an elevator in a building as the system, and the specification is whenever someone calls from a certain floor, the elevator needs to stop at that floor eventually to pick up that person. Now, this specification is concerned with a particular part of the system, which has almost nothing to do with executions like : not getting stuck in between floors, opening the door when it stops at a floor or, not opening the door when it is running between floor etc. Therefore, to check whether the elevator satisfies the specification, we should neither check all executions, it is enough nor to check a single execution, which can only guarantee for a single call to stop.

These types of problems in formal methods are solved using game-theoretic [43] approach: in our example, the person who calls the elevator is the *environment*, and the elevator itself is the system. In such two-player games between the system and the environment, the environment aims to show that the system is faulty, while the system tries to establish that it is not. In other words, the verification problem demands that the designer/controller can make sure the system wins against any behaviour of the environment. These are called *zero-sum games*, and a *solution* in such a game is a winning strategy.

On the other hand, it is possible that the designers of different components of a system themselves compete with each other for certain resources. Here instead of either *winning* or *loosing* the game, each player obtains some payoff. Payoff can be positive, which is generally referred to as a *reward*, and it can be negative, which is referred to as a *cost*. Unlike winning a game, here a player's objective is to optimize their payoff: maximize if it is a reward, minimize if it is a cost. These games are called *non-zero sum* games.

Moreover, in a non-zero sum game, the objective of designers of different components of a system's might be in direct contrast with the system's overall best interest. Therefore, while a solution from the system's perspective is a *social optimum*, from a particular component's perspective, who is playing *selfishly*, a solution would be a choice that is best with respect to all other player's choices. Because this is true for all components, they together would like to achieve some kind of stability in their choices with each other; and that would be the solution from their perspective. However, this notion of stability is not unique.

Nash Equilibria (NE) [50], the most common notion of stability, is the solution concept which is immune to any single player deviation at any point of the game if all players follow their corresponding choices given by it, from the beginning of the game. But there are other notions as well, which are studied depending on various contexts. For example, *Subgame perfect Equilibria* (SPE) [51] is a solution concept for stability, which is more relevant in a context where each player can make decision *dynamically*, step by step, seeing how other players are responding. *Strong Equilibria*[5, 25] is a notion of stability which is immune not only to single player deviations but also to coordinated deviation of several players, while Stackelberg equilibria [26, 59] are the stable solution when there are some players whose objective are on par with the system itself (not selfishly), while some other players play in their self-interest.

Nonetheless, it is not guaranteed that one or another notion of stable set of choices always exist in all non-zero sum games. Hence, one of the task in this context is always find out which notions fits best for the purpose, and whether that notion of stability exists in that particular class of non-zero sum game.

Congestion games and Selfish Routing. In this thesis, we focus on particular non-zero sum games, called congestion games played on a network. Congestion games model selfish resource sharing among several players [56]. A special case is the one of network congestion games, in which players aim at routing traffic through a congested network. In network congestion games, each player chooses a set of transitions, forming a simple path from a source state to a target state, and the cost of a transition increases with its load, that is, with the number of players using it. In the next section, we review the state of the art on network congestion games.

Related works on Network Congestion Games

Network congestion games can be classified into atomic and non-atomic variants. Non-atomic semantics is appropriate for large populations of players, thus seen as a continuum. One then considers portions of the population that apply predefined strategies, and there is no such thing as the effect of an individual player on the cost of others. In contrast, in atomic games, the number of players is fixed, and each player may significantly impact the cost other players incur. We only focus on atomic games in this thesis.

Network congestion games. Network congestion games, also called atomic selfish routing games in the literature, were first considered by Rosenthal [56]. These games are defined by a directed graph, a number of pairs of source-target vertices, and non-decreasing cost functions for each edge in the graph. For each source-target pair, a player must choose a route from the source to the target vertex. Given their choice of simple paths, the cost incurred by a player depends on the number of other players that choose paths sharing edges with their path, and on the cost functions of these edges. In this setting, a Nash equilibrium maps each player to a path in such a way that no player has an incentive to deviate: they cannot decrease their cost by choosing a different path.

Rosenthal[56] proved that network congestion games are potential games, so that Nash equilibria always exist. Monderer and Shapley [48] studied in a more general way potential games, and explained how to iteratively use best-response strategies to converge to an equilibrium. Interestingly, under reasonable assumptions on the cost functions, Bertsekas and Tsitsiklis established that there is a direct correspondence between equilibria in selfish routing and distributed shortest-path routing, which is used in practice for packet routing in computer networks [12]. We refer the interested reader to [57] for an introduction and many basic results on general routing games.

A natural question is whether selfish routing is very different from a routing strategy decided by a centralized authority. In other words, how far can a selfish optimum be from the social optimum, in which players would cooperate. The notion of price of anarchy, first proposed by Koutsoupias and Papadimitriou [39], is the ratio of the worst cost of a Nash equilibrium and the cost of the social optimum. This measures how bad Nash equilibria can be. In the context of network congestion games, the price of anarchy was first studied by Suri *et al.* [60], establishing an upper bound of $\frac{5}{2}$ when all cost functions are affine. A refined upper bound was provided by Awerbuch *et al.* [9]. Bounds on the dual notion of price of stability, which is the ratio of the cost of a best Nash equilibrium and the cost of the social optimum were also studied for routing games [3].

Timing aspects. Several works investigated refinements of this setting. In [31], the authors study network congestion games in which each edge is traversed with a fixed duration independent of its load, while the cost of each edge depends on the load. The model is thus said to have *time-dependent* costs since the load depends on the times at which players traverse a given edge. The authors prove the existence of Nash equilibria by reduction to the setting of [56]. An extension of this setting with timed constraints was studied in [6, 7].

The setting of fixed durations with time-dependent costs is interesting in applications where the players sharing a resource (an edge) see their quality of service decrease, while the time to use the resource is unaffected [7]. This might be the case, for instance, in some telecommunication and multimedia streaming applications. Timing also appears, for instance, in [40, 53] where the load

affects travel times and players' objective is to minimize the total travel time. Other works focus on flow models with a timing aspect [13, 37].

Dynamic network congestion games. In classical network congestion games, players choose their strategies (*i.e.,* their *simple* paths) in one shot. However, it may be interesting to let agents choose their paths *dynamically*, that is, step by step, by observing other players' previous choices. In this thesis, we study network congestion games with time-dependent costs as in [31], but with unit delays, and in a dynamic setting. More precisely, at each step, each of the players simultaneously selects the edge they want to take; each player is then charged a cost that depends on the load of the edge they selected, and traverses that edge in one step. We name these games *dynamic network congestion games* (dynamic NCG in short); the behaviour of the players in such games is formalized by means of *strategies*, telling the players what to play depending of the current configuration. Notice that, because the congestion effect applies to edges used *simultaneously* by several players, taking cycles can be interesting in dynamic NCG, which makes our setting more complex than most NCG models [8, 31, 56, 57].

Such a dynamic setting was studied in [8] for resource allocation games, which extends [56] with dynamic choices.

Standard solution concepts. We study classical solution concepts on dynamic network congestion games. A strategy profile (*i.e.*, a function assigning a strategy to each player) is a *Nash Equilibrium* (NE) when each single strategy is an optimal response to the strategies of the other players; in other terms, under such a strategy profile, no player may lower their costs by unilaterally changing their strategies. Notice that NE need not exist in general, and when they exist, they may not be unique. In the setting of dynamic games, Nash Equilibria are usually enforced using *punishing strategies*, by which any deviating player will be punished by all other players once the deviation has been detected. However, such punishing strategies may also increase the cost incurred by the punishing players, and hence do not form a credible threat. *Subgame-Perfect Equilibria* (SPE) refine NE and address this issue by requiring that the strategy profile is an NE along any play.

NE and SPE aim at minimizing the individual cost of each player (without caring of the others' costs); in a collaborative setting, the players may instead try to lower the social cost, i.e., the sum of the costs incurred to all the players. Strategy profiles achieving this are called *social optima* (SO). Obviously, the social cost of NE and SPE cannot be less than that of the social optimum; the *price of anarchy* measures how bad selfish behaviours may be compared to collaborative ones.

Parameterized network congestion games All the above problems were addressed on an underlying assumption that all the players (and controllers if any) know the number of players in the network, which is crucial to design an equilibrium or social optimum. As a solution for a certain number of players apriori does not give us any information about what a solution for another number of players would look like, we would need to solve it from scratch on the same network every time the number of players changes.

There has been a growing interest in investigating uncertainty in traffic. One common approach to deal with this problem is to generalize the deterministic setting to the stochastic one with an unknown number of players. Wang et al. [63] and Correa et al. [22] studied the PoA for non-atomic

congestion games with stochastic demands. Cominetti et al. [21] studied the atomic setting of Bernoulli congestion games in which each player participates in the game with an independent probability, and found that the upper bound of the PoA for deterministic models still hold. Cominetti et al. [20] found such Bernoulli congestion games converge to a set of Poisson games in the sense of Myerson[49], when players' participating probabilities tend to zero. Wang et al. [62] studies another model for atomic network congestion games with stochastic demands, in which instead of individual participation probabilities, they consider distributions of numbers of players as common knowledge, which follows the framework of population uncertainty proposed by Myerson[49]. Demand uncertainty has also been considered in other games such as resource allocation games by Ashlagi et al. [4].

Apart from stochastic approaches, there have been other methods to directly compute equilibria in parametric setting. Klimm and Wardo[36] found the complexity of computing Equilibria in atomic congestion games with parametric cost-affine functions is PPAD-complete, by expressing the evolution of equilibria locally by a novel block Laplacian matrix.

We try to approach the same problem with number of players as the parameter by expressing NE for large number of players in terms of NE for smaller number of player using semi-linear property, and this spares the controller, who assigns a stable profile to all players, to compute NE from scratch. Unfortunately, the general case remains open in this thesis, and we give preliminary results for *series-parallel* graphs.

Contributions

In the first part, we take a computational-complexity viewpoint to study dynamic network congestion games. We first establish the complexity of computing the social optimum, which we show is in PSPACE and NP-hard. We then prove that best-response strategies can be computed in polynomial time, and that dynamic NCG are potential games, thereby showing the existence of Nash equilibria in any dynamic NCG; this also shows that some Nash equilibrium can be computed in pseudo-polynomial time. We then give an EXPSPACE (resp. 2EXPSPACE) algorithm to decide the existence of Nash Equilibria (resp. Subgame-Perfect Equilibria) whose social costs satisfy given bounds. This allows us to compute best and worst such equilibria, and then the price of anarchy and the price of stability. These results are appeared in [11].

Note that some of the high complexities follow from the binary encoding of the number of players, which is the main input parameter. For instance, the exponential-space complexity of constrained NE drops to pseudo-polynomial time for a fixed number of players. This parameter becomes important since we advocate the study of computational problems, such as computing Nash equilibria with a given cost bound. We also believe that computing precise values for the price of anarchy and the price of stability is interesting, rather than providing bounds on the set of all instances as in *e.g.* [60].

In the second part, we restrict to game arena that are series-parallel graphs, and our model to the commonly studied model of network congestion games: without dynamic strategies and non-synchronous congestion effect in the cost computation. In that model, we show that the set of NE profiles is a semi-linear set. Moreover, the semi-linear set has a special structure: any period vector of the semi-linear set is a multiple of a single vector, which we call the *shift- vector*. Moreover,

we also make a conjecture that this so-called shift vector is the sole period vector of the semi-linear set. If this conjecture turns out to be true, then we show a very simple way to represent the set of all NE profiles, and give an algorithm to compute an NE for a large number of players using the shift vectors and the NE of relatively smaller number of players.

Outline

This document is divided into two parts. In Part I, which is based on [11], we study Dynamic network congestion games. In that part, in Chapter 1, we fix the notations and adapt definitions from literature, which will be used in the subsequent chapters. In Chapter 2, we consider the constrained Social optimal problem, and solve the problem in PSPACE, while the lower bound is shown to be NP. In Chapters 3 and 4, we consider two solution concepts: Nash Equilibria and Subgame perfect Equilibria, and solve the constrained decision problem for each, and with this we conclude Part I. This part is mostly based on [11].

In Part II of the document, we consider parameterized network congestion games on series-parallel graphs, in which the number of players is the parameter. Here, in Chapter 5, we introduce the definitions that we use specifically for this part, and in Chapter 6, our aim was to obtain an algorithm to compute NE for NCG with many players from NE of NCG with fewer players. We list out some results on that direction, and we formulate a conjecture which, if true, gives us a representation of the set of all NE with any number of players in terms of a so-called *shift* vector and NE with some finite number of players.

Finally, in Chapter 6.5, we conclude by summarizing all the results briefly, and by providing a list of immediate follow-ups and future objectives.

Contents

List of Figures

Part I

Network congestion games with fixed number of players

1 Preliminaries

In the first part of this thesis, we consider dynamic Network Congestion Games (dynamic NCG) with a fixed number of players. And, in the second part of this thesis, we consider Network congestion games (NCG) with number of players as a parameter. Here, in this chapter, we introduce the model for the first part only, and for the second part we have a respective chapter for preliminaries, definitions etc.

Let $\mathcal{F}$ be a family of non-decreasing functions from $\mathbb{N}$ to $\mathbb{N}$, which are piecewise-affine, with finitely many pieces. We assume that each $\mathsf{f} \in \mathcal{F}$ is represented by the endpoints of the intervals, and the coefficients, all encoded in binary.

1.1 Game Arena: Network with Cost Functions

An arena is a weighted graph $\mathcal{A} = \langle V, E, \mathsf{src}, \mathsf{tgt} \rangle$, where V is the set of vertices, $E \subseteq V \times \mathcal{F} \times V$ is a set of transitions labeled with non-decreasing piecewise affine cost functions, and $\mathsf{src}, \mathsf{tgt} \subseteq V$ are the sets of source and target vertices. In our discussion, we consider that each vertex except possibly the vertices of tgt has at least one outgoing edge, and every target vertex $t \in \mathsf{tgt}$ is reachable from every vertex $v \in V$.

1.2 Dynamic Congestion Games Played on a Network

Definition 1.1 (Network Congestion Game)**.** *A network congestion game (NCG) is a triplet $\mathcal{G} = \langle \mathcal{A}, n, \mathsf{g} \rangle$ where $\mathcal{A}$ is an arena as above, $n \in \mathbb{N}$ is the number of players, and $\mathsf{g} : \mathsf{src} \times \mathsf{tgt} \to \{0, \ldots, n\}$ maps each source-target vertex pair (s, t) to a number $k \leq n$, which means k players have s and t as their source and target vertices respectively.*

Naturally, it is required that, $\sum_{(s,t) \in \mathsf{src} \times \mathsf{tgt}} \mathsf{g}(s, t) = n.$

Assigning source-target to players. We assume an arbitrary order among the elements of src and tgt. This gives us a lexicographic ordering among the elements of $\mathsf{src} \times \mathsf{tgt}$. We use this order to identify which player have which vertices among the source and target. To be fair, our model is not specific to any player's identity, hence we do not keep this order as a part of the syntax.

Let us consider the elements of $\mathsf{src} \times \mathsf{tgt}$ with the order are $(s_1, t_1) < (s_1, t_2) < \ldots < (s_{|\mathsf{src}|}, t_{|\mathsf{tgt}|})$. We consider (s_l, t_m), for $l \leq |\mathsf{src}|, m \leq |\mathsf{tgt}|$, to be the source-target vertex pair for Player i if

$$\sum_{(s,t) < (s_l, t_m)} \mathsf{g}(s, t) < i \leq \sum_{(s,t) \leq (s_l, t_m)} \mathsf{g}(s, t)$$

Abusing notation, we denote the source and target vertex of Player i as $\mathsf{src}(i)$ and $\mathsf{tgt}(i)$.

It is noteworthy that, whenever we will take an instance of an NCG $\langle \mathcal{A}, n, \mathsf{g} \rangle$ as input, since for all $(s,t) \in \mathsf{src} \times \mathsf{tgt}$, $\mathsf{g}(s,t) \leq n$, the input size is logarithmic in n. This is clearly the case if all players have same source and target, which we refer as *symmetric* network congestion games. However, in case of non-symmetric games, where players do not have same source-target vertices, one way to encode the input in such a case is to take a list of source-target vertex pairs for each player. In such encoding, the input size os polynomial in n. But we can encode n in binary only because in this model of congestion games, apart from the source-target pair of vertices, identity of a player doesn't matter. That is why, instead of associating each pair of source-target vertices to each player, we chose to encode via a map which associates each pair of source-target vertex to the number of players who use those as such.

Semantics. In a network congestion game, all players start from their source $\mathsf{src}(i)$ and *simultaneously* select the edges they want to traverse in order to reach their target $\mathsf{tgt}(i)$. In this journey from their source to target, each player pays some cost for traversing, and this cost depends on the cost functions f encoded in the transitions of the form (v, f, v'), and in turn, *may* depend on the number of players that also take the same edge, in case f is not a constant function.

Each player's objective in an NCG is to traverse from its source to target with minimum individual accumulated cost.

In classical network congestion games[39, 57], each player selects a path from their source to target at once, at the beginning of a play, and continue with that path till they reach their target vertex. In that setting, we can also see that a player's congestion cost is measured by taking into account how many other players have taken the same edge throughout their traversal, and not necessarily at the same step. On the contrary, in Part I of this thesis, we consider a setting which differs from the classical model in two aspects:

- first, the game is played in rounds, during which all players take exactly one transition; the number of players using an edge e is measured dynamically, at each round.

  We can also view this difference in another perspective: while in our model of NCG, players pay "immediately" for the congestion, in classical model players pay "at the end" by seeing how many other players have also traversed the same edge.

- second, during the play, each player may adapt their choices depending on what the other players have been doing in the previous rounds.

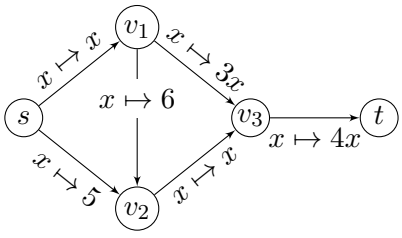


Figure 1.1: An arena $\mathcal{A}$ for a dynamic NCG

Example 1.2. *Consider the arena* $\mathcal{A} = \langle V, E, \mathsf{src} = \{s\}, \mathsf{tgt} = \{t\}\rangle$ *depicted at Fig. 1.1. Note that some of the edges like* $(s, x \mapsto x, v_1), (v_2, x \mapsto x, v_3)$ *are associated with linear cost functions, in which* $\mathsf{f} : x \mapsto x$ *means any player who takes this edge pays as much cost as the number of players who take the same edge in their traversal. Other edges like* $(v_1, x \mapsto 6, v_2)$ *are associated with constant cost functions, where each player who takes the edge* $(v_1, x \mapsto 6, v_2)$ *pays cost* 6*, no matter how many other players also take the same in their traversal.*

On this arena, we consider dynamic NCG $\langle \mathcal{A}, 2, \mathsf{g}\rangle$ *with two players, where* $\mathsf{g}(s, t) = 2$*. Assume that Player* 1 *follows the path* $\pi_1 : s \to v_1 \to v_3 \to t$ *and Player* 2 *goes via* $\pi_2 : s \to v_1 \to v_2 \to v_3 \to t$*.*

In classical NCGs, the cost for Player 1 *would be* $2 + 3 + 8 = 13$*, while for Player* 2*, it would be* $2 + 6 + 1 + 8 = 17$*. Notice that, here, even though the edge* $v_3 \to t$ *is not used simultaneously, both players bear congestion effect in their payable cost due to this edge. On the contrary, in our model, this gives rise to the following path (along with the costs):*

$$\begin{pmatrix} P1 \mapsto s \\ P2 \mapsto s \end{pmatrix} \xrightarrow[P2\mapsto 2]{P1\mapsto 2} \begin{pmatrix} P1 \mapsto v_1 \\ P2 \mapsto v_1 \end{pmatrix} \xrightarrow[P2\mapsto 6]{P1\mapsto 3} \begin{pmatrix} P1 \mapsto v_3 \\ P2 \mapsto v_2 \end{pmatrix}$$
$$\xrightarrow[P2\mapsto 1]{P1\mapsto 4} \begin{pmatrix} P1 \mapsto t \\ P2 \mapsto v_3 \end{pmatrix} \xrightarrow[P2\mapsto 4]{P1\mapsto 0} \begin{pmatrix} P1 \mapsto t \\ P2 \mapsto t \end{pmatrix}$$

Here, inside the brackets, we depict the current positions of the players, for example $\begin{pmatrix} P1 \mapsto s \\ P2 \mapsto s \end{pmatrix}$ *depicts that Player* 1 *and* 2 *both are at vertex* s*. On the other hand, labels over the transitions depict how much cost each player pays for taking that particular transition, for example at the first transition, both players observe congestion effect on their cost and each of them pays cost* 2*, according to the associated cost function* $x \mapsto x$*.*

Notice how edge $v_3 \to t$ *of* $\mathcal{A}$ *is used by both players, but not simultaneously, so that the cost of using that edge is* 4 *for each of them.*

An example of a strategy that dynamically adapts to the other players' behaviors, σ*, consists in first taking the transition* $s \to v_1$*, and then either taking* $v_1 \to v_3$ *if the other player took the same initial transition, or taking* $v_1 \to v_2$ *otherwise.* □

From hereon, we call this setting *dynamic* NCG in order to distinguish it from other network congestion game models [39, 57], which we often refer to as simply NCG.

1.3 Concurrent Game Semantics For Dynamic NCG

In congestion games, each player selects their edges simultaneously in order to traverse from their source to target, hence we can think of the dynamics of the game as a concurrent reachability game: where from each *location*, each player chooses an *action* to decide next location, for each joint action each player pays some individual cost, and their objective is to visit a designated location with minimum accumulated individual cost. We describe below, formally, how we associate a concurrent game structure with a dynamic NCG, and how we use that subsequently.

Configurations. For any $n \in \mathbb{N}$, we write $[\![n]\!] = \{i \in \mathbb{N} : 1 \leq i \leq n\}$ as the set of n Players. A *configuration* of a dynamic network congestion game $\langle \mathcal{A}, n, \mathsf{g} \rangle$ is a mapping $c : [\![n]\!] \to V$, indicating the position of each player in the arena. We define $c_{\mathsf{src}} : i \in [\![n]\!] \mapsto \mathsf{src}(i)$ and $c_{\mathsf{tgt}} : i \in [\![n]\!] \mapsto \mathsf{tgt}(i)$ as the initial and target configurations.

With $\mathcal{G} = \langle \mathcal{A}, n, \mathsf{g} \rangle$, we associate a multi-weighted graph $\mathcal{M} = \langle C, T \rangle$, where $C = V^{[\![n]\!]}$ is the set of all configurations, and $T \subseteq C \times \mathbb{N}^{[\![n]\!]} \times C$ is a set of edges, where the n-tuple of weights on an edge indicates individual accumulated cost for each player. Formally, there is an edge (c, w, c') in T if, and only if, there exists a collection $\mathbf{e} = (e_i)_{i \in [\![n]\!]}$ of edges of E such that for all $i \in [\![n]\!]$, writing $e_i = (v_i, \mathsf{f}_i, v_i')$ and $\mathsf{load}_{e_i}(\mathbf{e}) = \#\{j \in [\![n]\!] \mid e_j = e_i\}$ which denotes the number of players taking edge e_i simultaneously with Player i, we have $c(i) = v_i$, $c'(i) = v_i'$, and $w(i) = \mathsf{f}_i(\mathsf{load}_{e_i}(\mathbf{e}))$. Moreover, if $c(i) = \mathsf{tgt}(i)$, then as the game has stopped for player i, we know $w(i) = 0$, and $c'(i)$ remains $\mathsf{tgt}(i)$. We denote this edge with $c \overset{\mathbf{e}}{\Rightarrow} c'$. We may use the notation $\mathsf{cost}_i(c \overset{\mathbf{e}}{\Rightarrow} c')$ for $w(i)$ depending on the context, to directly specify the associated configurations and edge. We call $\mathcal{M}$ the *configuration graph* of the dynamic NCG $\mathcal{G}$.

Remark 1.3. *Here, we can note the difference in the way we compute cost between our model of network congestion games and the classical congestion games played on network [39, 57]. If we try to associate a configuration graph in the classical model to how we defined here, we can see that the edges cannot be associated with costs (weights in $\mathcal{M} = \langle C, T \rangle$ for our model) of Players likewise. This is due to the fact that, in the classical model, a player's cost of taking an edge does not solely depend on the current transition, rather takes into account how many players might have taken the same edge in other transitions too, in the full-course of the game.*

Two edges (c, w, c') and (d, x, d'), in that order, are said to be *consecutive* whenever $d = c'$. Given a configuration c, a *path* from c in a dynamic network congestion game is either the single configuration c (we call this a trivial path) or a non-empty, finite ($\rho = (t_j)_{1 \leq j < |\rho|}$) or infinite ($\rho = (t_j)_{j \geq 1}$) sequence of consecutive edges in $\mathcal{M}$, where t_1 is a transition from c; the size of a path ρ is zero for trivial paths, and $|\rho| \in \mathbb{N} \cup \{+\infty\}$ otherwise. Extending the notation $c \overset{\mathbf{e}}{\Rightarrow} c'$ of an edge, we often refer a path $\rho = (t_j)_{1 \leq j < |\rho|}$ by a sequence like $c_1 \overset{\mathbf{e}_1}{\Rightarrow} c_2 \overset{\mathbf{e}_2}{\Rightarrow} \dots c_{|\rho|-1} \overset{\mathbf{e}_{|\rho|-1}}{\Longrightarrow} c_{|\rho|}$ where $t_j = (c_j, w_j, c_{j+1})$ and $w_j = \mathsf{cost}_j(c_j \overset{\mathbf{e}_j}{\Rightarrow} c_{j+1})$.

We write $\mathsf{Paths}(\langle \mathcal{A}, n, \mathsf{g} \rangle, c)$ and $\mathsf{Paths}^{\omega}(\langle \mathcal{A}, n, \mathsf{g} \rangle, c)$ for the set of finite and infinite paths from c in $\langle \mathcal{A}, n, \mathsf{g} \rangle$, respectively. A finite path $\rho = (t_j)_{1 \leq j \leq |\rho|}$ and another path (finite or infinite) $\rho' = (t_j')_{j \geq 1}$ can be concatenated and can be written as $\rho \cdot \rho'$ if $t_{|\rho|}$ and t_1' are consecutive. Moreover, a finite path $\rho = (t_j)_{1 \leq j \leq |\rho|} \in \mathsf{Paths}(\langle A, n, \mathsf{g} \rangle, c)$ is called *complete* if for $t_{|\rho|} = (c, w, c')$, $c'(i) = \mathsf{tgt}(i)$ for all i. When the configuration c is clear from the context, we simply write $\mathsf{Paths}\mathcal{G}$ for game $\mathcal{G} = \langle \mathcal{A}, n, \mathsf{g} \rangle$.

Given a path ρ, an index $1 \leq j \leq |\rho|$ and a player $i \in [\![n]\!]$, we write $\rho(j)$ for the j-th configuration of ρ, and $\rho(j)(i)$ for the state of Player i in that configuration. For $j \geq 2$, we define $\rho_{\leq j}$ as the prefix of ρ that ends in the j-th configuration; we let $\rho_{\leq 1} = \rho(1)$. Similarly, for $1 \leq j \leq |\rho| - 1$, we let $\rho_{\geq j}$ denote the suffix that starts at the j-th configuration. Finally, if $|\rho|$ is finite, we let $\rho_{\geq |\rho|} = \rho(|\rho|)$.

Cost. With each path $\rho = (c_j, w_j, c'_j)_j$, and each player $i \in [\![n]\!]$, we associate a *cost*, written $\mathsf{cost}_i(\rho)$, which is zero for trivial paths, $+\infty$ for infinite paths along which $c_j(i) \neq \mathsf{tgt}(i)$ for all j, and $\sum_{j=1}^{|\rho|-1} w_j(i)$ otherwise. Moreover, when a player reaches their target vertex, they stay there with cost 0. We define the *social cost* of ρ as

$$\mathsf{soccost}(\rho) = \sum_{i \in [\![n]\!]} \mathsf{cost}_i(\rho)$$

Concurrent Game We now extend the configuration graph to a *concurrent game structure*[18], which is a multi-agent extension of transition system[2].

A *move* for Player $i \in [\![n]\!]$ from configuration c is an edge $e = (v, \mathsf{f}, v') \in E$ such that $v = c(i)$. A *move vector* from c is a sequence $\mathbf{e} = (e_i)_{i \in [\![n]\!]}$ such that for all $i \in [\![n]\!]$, e_i is a move for Player i from c.

Definition 1.4. *The concurrent game structure (CGS) associated with a dynamic NCG $\mathcal{G} = \langle \mathcal{A}, n, g \rangle$ is a tuple $\mathcal{S} = \langle C, T, M, U \rangle$, where*

- *$\langle C, T \rangle$ is the configuration graph associated to $\mathcal{G}$.*
- *$M : C \times [\![n]\!] \to 2^E$ lists the set of available moves for each player from each configuration.*
- *$U : C \times E^{[\![n]\!]} \to T$ is the transition function such that for every configuration c and every move vector $\mathbf{e} = (e_i)_{i \in [\![n]\!]}$ with $e_i \in M(c, i)$ for all $i \in [\![n]\!]$, $U(c, \mathbf{e}) = c \overset{\mathbf{e}}{\Rightarrow} c'$.*

Strategy and outcome. A *strategy* for Player i in $\mathcal{S}$ from configuration c is a function $\sigma_i : \mathsf{Paths}(\langle \mathcal{A}, n, \mathsf{g} \rangle, c) \to E$ that associates, with any *history* h, which is finite path from c in $\mathcal{S}$, a move for Player i from the last configuration of h. A *strategy profile* is a family $\sigma = (\sigma_i)_{i \in [\![n]\!]}$ of strategies, one for each player. We write $\mathfrak{S}_i$ for the set of Player i strategies, and $\mathfrak{S}^n$ for the set of strategy profiles.

Let c be a configuration, h be a history from c and a strategy profile $\sigma = (\sigma_i)_{i \in [\![n]\!]}$ from c. The *residual strategy profile* of σ after h is the strategy profile $\sigma^h = (\sigma_i^h)_{i \in [\![n]\!]}$ from the last configuration of h defined by $\sigma_i^h(h') = \sigma_i(h \cdot h')$.

The *outcome* of a strategy profile σ from c is the infinite path $\rho = (c_i, w_i, c_{i+1})_{i \geq 1}$, hereafter denoted with $\mathsf{outcome}(\sigma)$, obtained by running the strategy profile; it is formally defined as the only infinite path such that $(c_1, w_1, c_2) = U(c, \sigma(c))$, and such that for any $j \geq 2$, $(c_j, w_j, c_{j+1}) = U(c_j, \sigma(h'))$, where $h' = (c_1, w_1, c_2) \cdots (c_{j-1}, w_{j-1}, c_j)$.

Pick a strategy profile $\sigma = (\sigma_i)_{i \in [\![n]\!]}$, and let $\rho = (t_j)_{j \geq 1}$ be its outcome, writing $t_j = (c_j, (w_j^i)_{i \in [\![n]\!]}, c'_j)$ for all $j \geq 1$. Let $k \in [\![n]\!]$. If $c'_l(k) = \mathsf{tgt}(k)$ for some $l \in \mathbb{N}$, then σ_k is said to be winning for Player k. In that case, we define $\mathsf{cost}_k(\sigma)$ as $\mathsf{cost}_k(\mathsf{outcome}(\sigma))$. If $c'_l(i) = \mathsf{tgt}(i)$ for all $i \in [\![n]\!]$, we define the *social cost* of σ as $\mathsf{soccost}(\sigma) = \mathsf{soccost}(\rho)$.

A strategy σ_i for Player i is called *blind* whenever the following holds: for any two finite paths ρ and ρ' having same length k, and such that for any position $0 \leq j < k$ we have $\rho(j)(i) = \rho'(j)(i)$, then $\sigma_i(\rho) = \sigma_i(\rho')$.

Intuitively, this means that a blind strategy σ_i follows a path in $\mathcal{A}$, independently of what the other players do. A blind strategy can thus be represented as a path of the underlined arena $\mathcal{A}$

and we write $|\sigma_i|$ for the length of that path (until its first visit to tgt, if any). A blind strategy may not be *complete*, in the sense that it does not necessarily assign a move to *all* the paths of $\mathsf{Paths}(\langle \mathcal{A}, n, \mathsf{g} \rangle, c)$. We write $\mathfrak{B}$ for the set of blind strategies.

1.4 Optimizing strategies

In this section, we recall some standard solution concepts for optimizing strategies which concern with optimization of costs, individual or aggregate, depending on the context.

1.4.1 Social Optima

In a collaborative situation, all players want to collectively minimize the total cost for all of them reaching their respective target vertex from their sources.

Definition 1.5. *A strategy profile $\sigma = (\sigma_i)_{i \in [\![n]\!]}$ is called a* social optimum (SO) *if*

$$\mathsf{soccost}(\sigma) = \inf_{\tau \in \mathfrak{S}^n} \mathsf{soccost}(\tau)$$

Observation 1.6. *We make the following two observations:*

- *In a dynamic NCG, there might be several SO profiles (See Example 1.7 below), nonetheless, all of them have same social cost, by definition, even though individual costs of players might vary.*
- *If for two strategy profiles σ and σ', $\mathsf{outcome}(\sigma) = \mathsf{outcome}(\sigma')$, then σ is an SO profile if, and only if σ' is an SO profile.*

Example 1.7. *Consider the dynamic NCG $\mathcal{G} = \langle \mathcal{A}_2, 3, g \rangle$ played on arena $\mathcal{A}_2$, depicted in Fig 1.2a, where $\mathsf{g}(s,t) = 3$. Consider the strategy profile $\sigma = (\sigma_i)_{i \in [\![n]\!]}$ depicted in Fig 1.2b in which $\sigma_1 = \sigma_2$. Here σ_1 is a blind strategy represented by the path $\rho_1 = s_1 \xrightarrow{e_1} v_1 \xrightarrow{e_2} v_2 \xrightarrow{e_3} v_3 \xrightarrow{e_4} t$ (depicted as the red path), while σ_3 is also a blind strategy represented by the path $\rho_3 = s \xrightarrow{e_5} v_4 \xrightarrow{e_6} v_5 \xrightarrow{e_7} v_6 \xrightarrow{e_8} t$, depicted as the green path in the figure. Following the profile σ, we have $\mathsf{cost}_1(\sigma) = \mathsf{cost}_2(\sigma) = 14$, while $\mathsf{cost}_3(\sigma) = 8$.*

We claim that this σ is an SO profile. To verify this, from observation 1.6, we know that it is enough to compare with social costs of all the blind strategy profiles as outcome of any strategy profile can be viewed as blind strategy profiles. Now, apart from the paths ρ_1 and ρ_3, mentioned just above, the other paths that can represent a strategy are $\rho_2 = s \xrightarrow{e_1} v_1 \xrightarrow{e_9} v_5 \xrightarrow{e_7} v_6 \xrightarrow{e_8} t$ and $\rho_4 = s \xrightarrow{e_1} v_1 \xrightarrow{e_2} v_2 \xrightarrow{e_{10}} v_6 \xrightarrow{e_8} t$. By directly representing a blind strategy profile by a 3-tuple of paths, we have $\sigma = (\rho_1, \rho_1, \rho_3)$ and $\mathsf{soccost}(\sigma) = 14 \times 2 + 8 = 36$. The social costs for all other blind strategy profiles are:

$$\begin{aligned}
\mathsf{soccost}((\rho_1,\rho_2,\rho_3)) &= 37\\
\mathsf{soccost}((\rho_1,\rho_4,\rho_3)) &= 36\\
\mathsf{soccost}((\rho_2,\rho_4,\rho_3)) &= 45\\
\left.\begin{aligned}\mathsf{soccost}((\rho_1,\rho_1,\rho_1))\\ \mathsf{soccost}((\rho_4,\rho_4,\rho_4))\end{aligned}\right\} &= 54\\
\mathsf{soccost}((\rho_2,\rho_2,\rho_2)) &= 63\\
\mathsf{soccost}((\rho_3,\rho_3,\rho_3)) &= 72\\
\mathsf{soccost}(\rho_1,\rho_1,\rho_2) &= 43
\end{aligned}
\qquad
\begin{aligned}
\left.\begin{aligned}\mathsf{soccost}((\rho_1,\rho_1,\rho_4))\\ \mathsf{soccost}((\rho_2,\rho_2,\rho_1))\\ \mathsf{soccost}((\rho_4,\rho_4,\rho_1))\end{aligned}\right\} &= 46\\
\mathsf{soccost}((\rho_2,\rho_2,\rho_3)) &= 52\\
\mathsf{soccost}((\rho_2,\rho_2,\rho_4)) &= 58\\
\mathsf{soccost}((\rho_3,\rho_3,\rho_1)) &= 42\\
\mathsf{soccost}((\rho_3,\rho_3,\rho_2)) &= 55\\
\mathsf{soccost}((\rho_3,\rho_3,\rho_4)) &= 50\\
\mathsf{soccost}((\rho_4,\rho_4,\rho_2)) &= 51\\
\mathsf{soccost}((\rho_4,\rho_4,\rho_3)) &= 47
\end{aligned}$$

The rest of the blind strategy profiles are just permutations of the above, for which soccost *does not change. Thus, σ is an SO profile.* □

1.4.2 Equilibria

In a selfish situation, each player aims at optimizing their individual accumulated cost. Despite having only control over their own strategies (be it blind or dynamic), each player's cost gets affected by what other players chose. Hence, intuitively, optimizing individual cost can be thought of as responding to other people's strategies by an optimal choice of strategy. Following this, we study *best-response* strategy, which is commonly used in the literature in the context of individual cost-optimization for a quantitative non-zero sum multiplayer game.

Given a strategy profile $\sigma = (\sigma_i)_{i\in[\![n]\!]}$, a player $k \in [\![n]\!]$, and a strategy $\sigma'_k \in \mathfrak{S}$, we denote by $\langle \sigma_{-k}, \sigma'_k\rangle$ the strategy profile $(\tau_i)_{i\in[\![n]\!]}$ such that $\tau_k = \sigma'_k$ and $\tau_i = \sigma_i$ for all $i \in [\![n]\!] \setminus \{k\}$. The strategy σ_k is a *best-response* to $(\sigma_i)_{i\in[\![n]\!]\setminus\{k\}}$ if $\mathsf{cost}_k(\sigma) = \inf_{\tau_k\in\mathfrak{S}} \mathsf{cost}_k(\langle\sigma_{-k},\tau_k\rangle)$.

Nash Equilibria

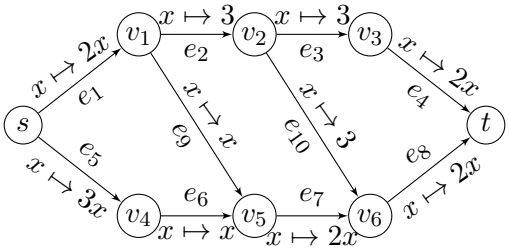
Definition 1.8. *A strategy profile $\sigma = (\sigma_i)_{i\in[\![n]\!]}$ is a* Nash Equilibium *(NE) if for every $k \in [\![n]\!]$, σ_k is a best-response to $(\sigma_i)_{i\in[\![n]\!]\setminus\{k\}}$.*

In such cases, there are no profitable unilateral deviations, i.e, no player can improve their cost by switching from their current strategy alone.

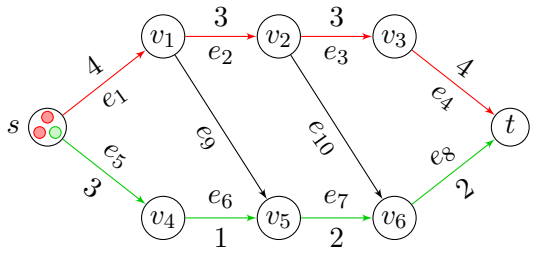
NE can be defined for sub-classes of strategy profiles. In particular, a *blind* NE is a strategy profile which is immune to any unilateral blind strategy deviation: formally, a strategy profile $\sigma = (\sigma_i)_{i\in[\![n]\!]}$ is a *blind NE* if for any player $k \in [\![n]\!]$, $\mathsf{cost}_k(\sigma) = \inf_{\tau_k\in\mathfrak{B}} \mathsf{cost}_k(\langle\sigma_{-k},\tau_k\rangle)$.

A priori, a blind NE need not be an NE for general strategies. That is, from a blind strategy profile, a player could have a profitable unilateral deviating strategy but not a profitable unilateral *blind* deviating strategy. But in fact, we will see in Chapter 3 that all blind NE are general-strategy NE for dynamic NCG.

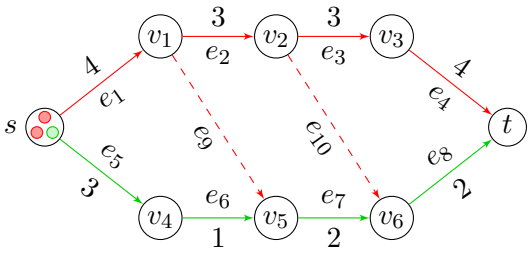
Example 1.9. *Consider the arena $\mathcal{A}_2$ displayed at Fig. 1.2a and the dynamic NCG $\mathcal{G} = \langle\mathcal{A}_2, 3, g\rangle$, where $g(s,t) = 3$ depicted at Fig 1.2b, illustrated with the NE profile $\sigma = (\sigma_i)_{i\in[\![3]\!]}$ defined below.*



(a) An arena $\mathcal{A}_2$ for a dynamic NCG



(b) Depiction of an SO profile in $\mathcal{G} = \langle \mathcal{A}_2, 3, \mathsf{g} \rangle$ where $\mathsf{g}(s,t) = 3$



(c) Depiction of an NE profile in $\mathcal{G} = \langle \mathcal{A}_2, 3, \mathsf{g} \rangle$ where $\mathsf{g}(s,t) = 3$

Figure 1.2: Example of an NE and an SO on arena $\mathcal{A}_2$

The strategies played by Player 1 *and* 2 *are symmetric. Intuitively, each of these two players play following the path* $s \xrightarrow{e_1} v_1 \xrightarrow{e_2} v_2 \xrightarrow{e_3} v_3 \xrightarrow{e_4} t$*, unless one of them realizes at some point (either at* v_1 *or* v_2*) that the other player has deviated (via* e_5 *or* e_9 *respectively). In that case, the current player immediately follows the edge (*e_9 *or* e_{10} *respectively) in order to "punish" the deviating player. Player* 3 *blindly follows the path* $s \xrightarrow{e_5} v_4 \xrightarrow{e_6} v_5 \xrightarrow{e_7} v_6 \xrightarrow{e_8} t$*.*

In Fig 1.2b, these strategies are depicted by the red arrows (dashed arrows depict the conditional deviation) for Players 1 *and* 2*, and green arrows for Player* 3*.*

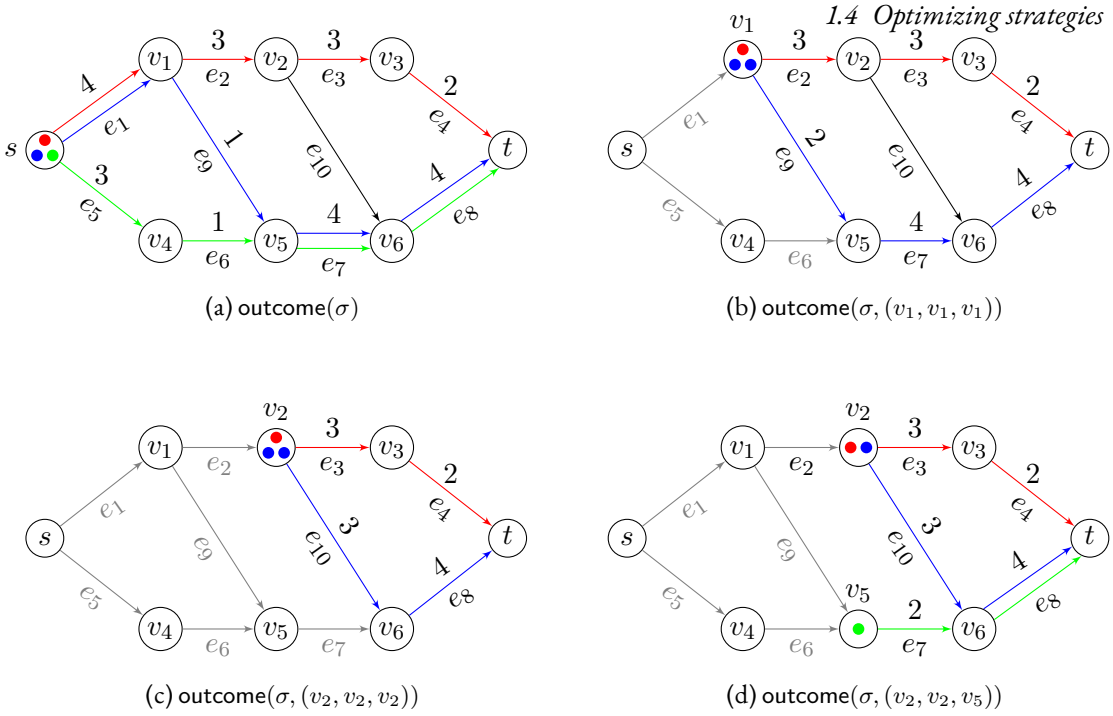
Note that,

$$\mathsf{outcome}(\sigma) = (s,s,s) \xrightarrow{(4,4,3)} (v_1,v_1,v_5) \xrightarrow{(3,3,1)} (v_2,v_2,v_5) \xrightarrow{(3,3,2)} (v_3,v_3,v_6) \xrightarrow{(4,4,2)} (t,t,t)$$

which implies $\mathsf{cost}_1(\sigma) = \mathsf{cost}_2(\sigma) = 14$ *and* $\mathsf{cost}_3(\sigma) = 8$*. Clearly, Player* 3 *doesn't have any unilateral deviation from which they can improve their cost. On the other hand, by design, if any of Player* 1 *or* 2 *deviates from their current strategy at* s *or* v_1*, the other player would punish by deviating at the next vertex. Hence, their unilateral deviation only increases their own cost, making it* 20 *(for deviating at* s*) or* 15 *(for deviating at* v_1*). Deviating at* v_2 *also keeps the deviating player's cost at* 14 *without improving it. Hence, none of the players has a unilateral profitable deviation from* σ*.*

Therefore, σ *is an NE profile.*

□



(a) outcome(σ)

(b) outcome($\sigma, (v_1, v_1, v_1)$)

(c) outcome($\sigma, (v_2, v_2, v_2)$)

(d) outcome($\sigma, (v_2, v_2, v_5)$)

Figure 1.3: Depiction of an SPE σ, explained in Example 1.12: Fig 1.3a represents the outcome, while Fig 1.3b, 1.3c and 1.3d represent outcome of σ when applied to the corresponding configurations

Subgame perfect Equilibria

In an NE, once a player deviated from their original strategy, other players can punish them even if that increases their own cost. Indeed, the condition for being NE only requires that the deviation should not be profitable for a deviating player. Subgame-perfect equilibrium (SPE) refines the concept of NE and rules out such non-credible threats.

Definition 1.10. *Fix an initial configuration, say c, then a Nash equilibrium profile from c is a* subgame perfect equilibria *(SPE) if for any finite history h from c and ending at c', the residual strategy profile after h is an NE from c'.*

By definition, any strategy of an SPE needs to be complete, i.e, the strategy needs to associate a move to all the paths of $\mathsf{Paths}(\langle \mathcal{A}, n, \mathsf{g} \rangle)$. Hence,

Remark 1.11. *A blind NE, which only associates a path from a player's source vertex to target, generally is not an SPE, simply because the profile is not complete.*

Example 1.12. *Let us consider again the game $\mathcal{G} = \langle \mathcal{A}_2, 3, g \rangle$ with $g(s,t) = 3$ as usual. Let us consider the strategy profile $\sigma = (\sigma_i)_{i \in [\![3]\!]}$ depicted in Fig 1.3.*

Here, σ_1 and σ_2 map the empty path ϵ to edge e_1, while σ_3 maps ϵ to e_5. To describe rest of the strategy profile, instead of explicitly writing each edge to which it maps a history h of $\mathsf{Paths}(\langle \mathcal{A}, 3, g \rangle, c_{\mathsf{src}})$, we associate a path $\pi = (e_j)_{j \geq 1}$ of $\mathcal{A}$ that is being followed by σ_i. We denote this new way of describing σ as $\hat{\sigma}$. Intuitively, we write $\hat{\sigma}_i(h) = \pi = (e_j)_{j \geq 1}$ to denote the following, in short: $\sigma_i(h) = e_1$, and if for $j \geq 1$, h_j denotes the finite path after applying σ for j steps, then $\sigma_i(h_j) = e_{j+1}$.

In the following, we define $\hat{\sigma}$; moreover, for convenience we refer to Fig.1.3 rather than using formal notation.

First, $\hat{\sigma}_1((s,s,s))$, $\hat{\sigma}_2((s,s,s))$ and $\hat{\sigma}_3((s,s,s))$ are shown as the red path (the only path involving e_3), blue path (involving e_9) and green path (involving e_5) respectively in Figure 1.3a; in fact, Fig. 1.3a captures $\mathsf{outcome}(\sigma)$.

But this doesn't make the definition of σ complete. For that, we need to define the residual strategies σ_i^h when h is not a prefix of $\mathsf{outcome}(\sigma)$. We also define those σ^h in terms of $\hat{\sigma}^h$.

Now, any path h ending at configuration $c_1 = (v_1, v_1, v_1)$ is not prefix of $\mathsf{outcome}(\sigma)$, and we define $\hat{\sigma}_1^h(c_1)$ to be the red path shown in Fig. 1.3b, and both $\hat{\sigma}_2^h(c_1)$ and $\hat{\sigma}_3^h(c_1)$ to be the blue path shown on $\mathcal{A}_2$, shown in the same figure.

Similarly, for any path h ending at configuration $c_2 = (v_2, v_2, v_2)$, we define $\hat{\sigma}^h(c_2) = (\hat{\sigma}_i^h(c_2))_{i\in[\![3]\!]}$ to follow the red, blue and blue path respectively as shown in Fig. 1.3c.

Finally, for any path ending at $c_3 = (v_2, v_2, v_5)$, we define $\hat{\sigma}^h(c_3) = (\hat{\sigma}_i^h(c_3))_{i\in[\![3]\!]}$ to follow the red, blue and green path respectively as shown in Fig. 1.3d □

1.5 Conclusion

In this Chapter, we have introduced a model of network congestion games, namely dynamic NCG, which differs from commonly studied model of NCG in two aspects: (a) simultaneous cost-computation, and (b) dynamic strategies. We believe worth considering. We have also discussed briefly the solution concepts, mainly definitions and example, both for optimizing individual and aggregate cost: namely Social optima, Nash equilibria and Subgame perfect equilibria.

Now, in the subsequent chapters of this part, we will study more on these solution concepts one by one, while in Part II, we will be focusing on the model where the number of player is a parameter.

2 Social Optima

In a congestion game, players play with the objective to traverse from their source to target vertex with minimal individual accumulated cost. But, their selfishly chosen strategies may result in a sub-optimal social cost. Therefore, a system which wants to optimize the overall cost, might be interested in finding out a solution by which it can assign strategies to individual players in its own interest. In this chapter, we address this problem for dynamic NCG by studying the optimal solution concept for a system as a whole, namely *Social optima* (SO).

Here, we recall from Section 1.3 of Chapter 1, that for a strategy profile $\sigma = (\sigma_i)_{i \in [\![n]\!]}$, we define the *social cost* of σ as $\mathsf{soccost}(\sigma) = \sum_{i \in [\![n]\!]} \mathsf{cost}_i(\rho)$, with $\rho = \mathsf{outcome}(\sigma)$.

In order to compute the optimal social cost for a dynamic NCG, we consider a decision version of this problem (Problem 2.1 below). In this chapter, we show different approaches to solve this problem. Starting with a brief explanation of a very naive approach at the very beginning of Section 2.1, we go on to explain an abstract weighted graph approach in Section 2.1.1 for symmetric NCG. Then, we explain why this approach fails for non-symmetric NCG on an example, and subsequently we explain an adaptation called src-abstract weighted graph approach in Section 2.1.2 to finally solve for the non-symmetric case. We conclude the section by showing that the decision problem can be solved in PSPACE. In section 2.2, we show a reduction from Partition problem to symmetric NCG to conclude that the problem is NP-hard.

Problem 2.1 (CONSTRAINED-SO)**.** *Given a dynamic NCG $\mathcal{G}$ and a bound $K \in \mathbb{N}$, does there exists a strategy profile with social cost less than or equal to K?*

Remark 2.2. *Concerning* CONSTRAINED-SO*, some immediate observations are:*

- *We take a non-negative integer K as the bound in* CONSTRAINED-SO *because the cost functions, $f \in \mathcal{F}$, maps an edge to a non-negative integer by our assumption on the model. Hence, the individual cost, $\mathsf{cost}_i(\rho)$ for any path is a non-negative integer too, so is the social cost $\mathsf{soccost}(\sigma)$ of strategy profile σ.*

- *Once we are able to solve Problem 2.1, we can compute the optimal social cost of a dynamic NCG $\mathcal{G} = \langle \mathcal{A}, n, g \rangle$ by the binary search Algorithm 1. Note that, the initial value for U (see line 2 of Algorithm 1) is an upper bound of social cost for any strategy profile, hence the optimal social cost must be less than or equal to U, which justifies the above binary search as a way to compute the optimal social cost of $\mathcal{G}$.*

In section 2.1, we first give an algorithm to solve CONSTRAINED-SO in PSPACE, then we show that even for symmetric dynamic NCGs, CONSTRAINED-SO is NP-hard.

Algorithm 1 Binary search to compute optimal social cost

procedure Opt-SOCCOST($\mathcal{G}$)
  $U = \sum_{(v,\mathsf{f},v')\in E} \mathsf{f}(n)$
  $L = 0$;
  while $L \leq U$ **do**
    if $L = U - 1$ **then return** U
    else
      $K := \lfloor \frac{L+U}{2} \rfloor$
      Solve CONSTRAINED-SO problem for $\mathcal{G}$ and K
      if the answer is "Yes" **then**
        $U := K$
      else
        $L := K$

2.1 Solving the CONSTRAINED-SO problem

To begin with, recall what we defined as the configuration graph $\mathcal{M} = \langle C, T\rangle$ associated to $\mathcal{G}$ and thereafter its extension to a concurrent game structure (CGS) $\mathcal{S}$ from Section 1.3. Note that, it suffices to find a path ρ in $\mathcal{S}$ with social cost less or equal to a given value K. Indeed, for a path $\rho = (c_j, w_j, c_{j+1})_{j\geq 1}$, we can define a blind strategy profile $\sigma = (\sigma_i)_{i\in[\![n]\!]}$ such that $\mathsf{outcome}(\sigma) = \rho$ as follows: by definition, each edge (c_j, w_j, c_{j+1}) of ρ can be viewed as $U(c_j, \mathbf{e}_j)$ for some edge vector $\mathbf{e}_j = (v_{j,i}, \mathsf{f}_{j,i}, v'_{j,i})_i \in E^{[\![n]\!]}$, and we define the blind strategy σ_i (which can be written as path) to be $(v_{j,i}, f_{j,i}, v'_{j,i})_{j\geq 1}$.

As the size of $\mathcal{M}$ (which is also the size of $\mathcal{S}$) is $\mathcal{O}(|V|^n)$, where the number of players n is encoded in binary, this algorithm where we non-deterministically guess a path $\mathcal{S}$, uses exponential space in the number of bits needed to encode n. However, we may not need to use the exact configuration to assert the existence of such a path. In the following, for pedagogical purposes, we first define an *abstract weighted graph* associated with $\mathcal{G}$, and solve the problem for symmetric (where there is a unique source-target pair for all players) NCGs and then we extend the same to what we call a *src-abstract weighted graph* associated with $\mathcal{G}$ and solve the same problem for non-symmetric NCG.

2.1.1 Abstract Weighted graphs: Solving CONSTRAINED-SO for Symmetric NCG

With a configuration c, we associate an abstract configuration $\bar{c} \in [\![n]\!]^V$, defined as $\bar{c}(v) = \#\{i \in [\![n]\!] : c(i) = v\}$. The *abstract weighted graph* associated with an NCG (not necessarily symmetric) $\mathcal{G} = \langle \mathcal{A}, n, \mathsf{g}\rangle$ is a weighted graph $\mathcal{P} = \langle A, B\rangle$, where A contains all abstract configurations and there is an edge $(a, w, a') \in B \subseteq A \times \mathbb{N} \times A$ if, and only if, there is a mapping $b : E \to [\![n]\!]$ such that $\sum_{e\in E} b(e) = n$, which intuitively means that $b(e)$ players are taking the edge e in the current transition from abstract configuration a to a'. Moreover, they satisfy the following:

$$a(v) = \sum_{e=(v,\mathsf{f},v')} b(e) \qquad w = \sum_{e=(v,\mathsf{f},v')} b(e) \times \mathsf{f}(b(e)) \qquad a'(v) = \sum_{e=(v',\mathsf{f},v)} b(e).$$

Similarly to the CGS representation, an *abstract path* of a network congestion game is either a single abstract configuration or a non-empty, finite or infinite sequence of consecutive edges in the abstract weighted graph $\mathcal{P}$. The cost of an abstract path is the sum of weights of its edges (if any). Then we have,

Lemma 2.3. *For NCG $\mathcal{G}$, For any $w \in \mathbb{N} \cup \{+\infty\}$, there is a path in $\mathcal{M}$ with social cost w if, and only if, there is an abstract path in $\mathcal{P}$ with the same cost w.*

Proof. From any path $\rho = (c_i, w_i, c_{i+1})_{i\geq 0}$ in $\mathcal{M}$, a path $\tau = (a_i, w_i, a_{i+1})_{i\geq 0}$ can be constructed in $\mathcal{P}$, where the a_i's are Parikh images [41] of the c_i's.

For the other direction, let us consider a path $\tau = (a_i, w_i, a_{i+1})_{i\geq 0}$ of $\mathcal{P}$. From this, we can construct a path in $\mathcal{M}$ by starting with a configuration c_0, of which a_0 is the Parikh image and then continuing with c'_i for each $i > 0$ such that a'_i's are Parikh images of c_i's respectively. By definition of $\mathcal{P}$ itself, we get the existence of such a c_0, and subsequently c_i for $i > 0$. Moreover, w_i's remain unaltered as it is in the path in $\mathcal{M}$. Note that, there might be multiple paths in $\mathcal{M}$ that can be associated to a path in $\mathcal{P}$ in such a way, but existence of one suffices for our purpose. □

With this, we consider a symmetric NCG $\mathcal{G} = \langle A, n, \mathsf{g}\rangle$, where there is a s in src and t in tgt such that, $\mathsf{g}(s,t) = n$. Recall K is the value in the CONSTRAINED-SO problem so that we look for a strategy profile in $\mathcal{G}$ with social cost less or equal to K.

Algorithm. We consider the abstract weighted graph $\mathcal{P} = \langle A, B\rangle$ associated with $\mathcal{G}$, and start a path finding algorithm from the vertex $\bar{c}_s$, where $\bar{c}_s(s) = n$, with a counter L initialized at 0. At each step, from the current vertex a of A, we guess an edge $(a, w, a') \in B$, check whether $L+w$ is still less or equal to K. If it is so, the guess is correct, hence we change the current configuration to a' and update the counter value to $L+w$. We keep another counter len, also initialized at 0, which we increment by 1 each time we make a successful guess. The algorithm terminates whenever it reaches a' such that $a'(t) = n$, or when len exceeds $[\![n]\!]^V$. As the counter is singly exponential in size of the input (# of bits needed to encode n), this non-deterministic algorithm runs in polynomial space. Hence, by Savitch's theorem [58], the problem is in PSPACE.

When the algorithm terminates after reaching $\bar{c}_t$, we know that there is a path in $\mathcal{P}$ with cost less or equal to K. By lemma 2.3, that means there is a path ρ in $\mathcal{M}$ with social cost less or equal to K, which again asserts the existence of a strategy profile σ with $\mathsf{soccost}(\sigma) \leq K$.

Note that, in the algorithm, we look for a path of length up to $[\![n]\!]^V$, which is the maximum length of any path without a cycle. Even though, any path in $\mathcal{P}$ corresponds to a path in $\mathcal{M}$ with same social cost as stated in Lemma 2.3, it is also true that any cycle in $\mathcal{P}$ may not necessarily correspond to a cycle in $\mathcal{M}$, because two different configurations of $\mathcal{M}$, which are permutations of each other, correspond to same abstract configuration of $\mathcal{P}$ (because their Parikh images are identical). However, in a symmetric network congestion game, as all players have same source-target, a transition from one permutation to another permutation of positions between players only contributes in increasing the social cost just like a cycle does. Hence, from an abstract path τ of $\mathcal{P}$ containing a cycle, we can construct a path τ' of $\mathcal{M}$ with social cost less than that of ρ.

The construction goes as follows. Let $a \in A$ is the configuration of $\mathcal{P}$, on which τ completes its cycle. Let us also suppose that ρ is a path in $\mathcal{M}$ which corresponds to τ, by Lemma 2.3 we know such a path exists. Now the cycle on a of τ corresponds to some path from c_1 to c_2 of ρ, where for c_1, c_2 of C there is a permutation $\phi : [\![n]\!] \to [\![n]\!]$ such that $c_1(i) = c_2(\pi(i))$ for all i. This necessarily implies $\bar{c_1} = \bar{c_2} = a$. Let us assume in the path from c_{src} to c_1 to c_2 in C, Player i follows the path $s \xrightarrow{\pi_1} u_i \xrightarrow{\pi_2} v_i$, then by construction Player $\phi(i)$ follows the path $s \xrightarrow{\pi'_1} v_i \xrightarrow{\pi'_2} u_i$. We construct the path ρ' in C by letting Player i follow the path $s \xrightarrow{\pi'_1} v_i$, and Player $\pi(i)$ follow the path $s \xrightarrow{\pi_1} u_i$ for each i, and this gives us a direct path from c_{src} to c_2 without occurring c_1. For the rest, we let them follow their paths from c_2 to c_{tgt} of ρ. As we remove some positive weighted parts of the path τ, the resulted path must have less weight than τ. Again from Lemma 2.3, this means there is a path ρ' with social cost less than that of ρ. Following this construction for every cycle of τ until there is no cycle remains, we have,

Lemma 2.4. *In symmetric NCG $\mathcal{G}$, if an abstract path τ of $\mathcal{P}$ containing a cycle corresponds to a path ρ in $\mathcal{M}$, then there is a path τ' in $\mathcal{M}$ without any cycle and τ' has lower cost than that of τ, which also corresponds to a path ρ' in $\mathcal{M}$.*

This justifies why we only guess a path in $\mathcal{P}$ without a cycle. But, Lemma 2.4 is not necessarily true for non-symmetric NCG. Consider the following example which shows why it is not true for a non-symmetric NCG.

Example 2.5. *Consider the non-symmetric NCG $\mathcal{G} = \langle \mathcal{A}, 2, \mathsf{g} \rangle$, depicted in Figure 2.1, where $\mathsf{g}(s_1, t_2) = 1$ and $\mathsf{g}(s_2, t_1) = 1$. Here cost-functions f's are omitted from the arena because they are not relevant for current discussion. In the abstract weighted graph $\mathcal{P}$ associated to $\mathcal{G}$, any path that corresponds to a complete path of $\mathcal{M}$ contains a cycle $(\bar{c}, w, \bar{c})$, where $\bar{c} = (0, 1, 1, 0)$ and w depends on the f's. Hence, the above algorithm discards any path containing a cycle, it would not answer "yes", even though there might be a profile with social cost less than the given K. In conclusion, the algorithm is not sound for non-symmetric NCG.* □

Thus, we extend the abstract weighted graph to what we call *src-abstract weighted graph* in the sequel and solve the general CONSTRAINED-SO problem based on this.

2.1.2 SRC-ABSTRACT WEIGHTED GRAPHS: SOLVING CONSTRAINED-SO FOR NCG

An *src-abstract weighted graph* associated to an NCG is an extension of the abstract weighted graph, where additional to the Parikh image of the configuration, we store information about the source vertices of the players.

With a configuration $c \in C$ of the multi-weighted configuration graph $\mathcal{M}$ associated to $\mathcal{G} = \langle A, n, \mathsf{g} \rangle$, we associate an src-abstract configuration $\mathscr{C}$, which is a $(|\mathsf{src}| + 1) \times |V|$ matrix over $[\![n]\!]$, and is defined as follows:

$$\mathscr{C}(0, v) = \#\{i \in [\![n]\!] : c(i) = v\} \text{ for each } v \in V, \text{ and}$$
$$\mathscr{C}(s, v) = \#\{i \in [\![n]\!] : c(i) = v, \mathsf{src}(i) = s\} \text{ for each } s \in \mathsf{src}, v \in V$$

Here recall that we defined $\mathsf{src}(i)$ to be the source vertex of Player i, which we get from g and the arbitrary ordering that we considered in Section 1.2. Intuitively, $\mathscr{C}(0, v)$ is the Parikh image of

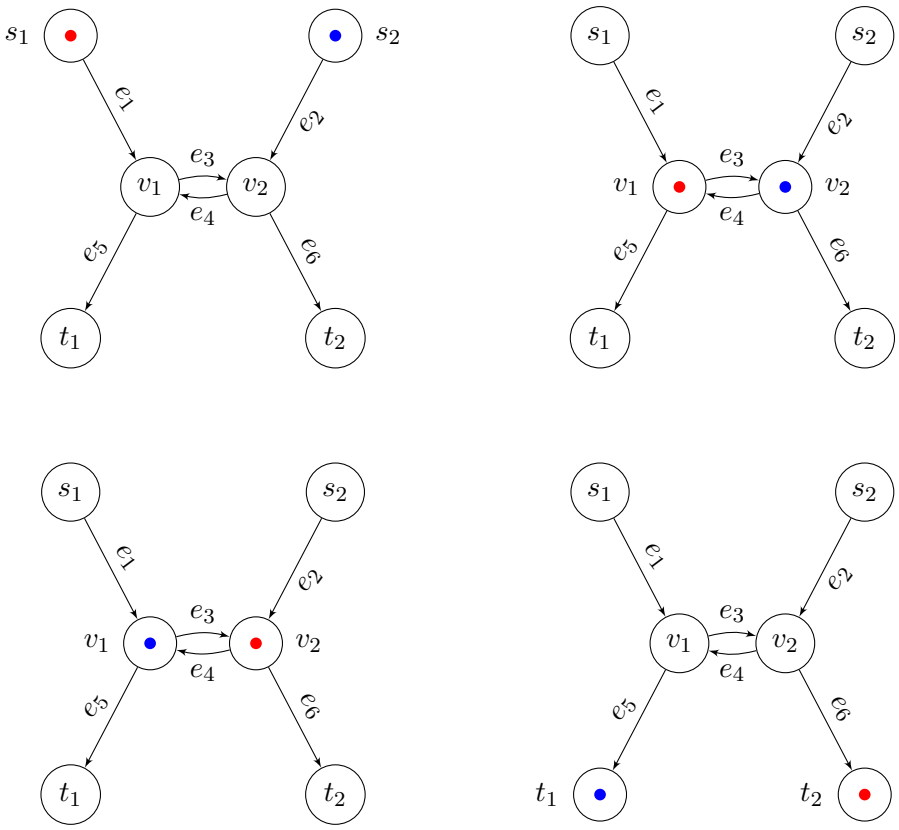


Figure 2.1: An example where a cycle appears in the abstract weighted graph for an SO

the configuration c, just like $\bar{c}$ of abstract weighted graph $\mathcal{P}$, and $\mathscr{C}(s,v)$ is the number of players who are at v in the configuration c, and whose source vertex is v'.

The src-abstract weighted graph associated to NCG $\mathcal{G}$ is $\mathscr{P} = \langle \mathscr{A}, \mathscr{B} \rangle$, where $\mathscr{A}$ is the set of all src-abstract configurations and there is an edge $(\mathscr{C}, w, \mathscr{C}') \in \mathscr{B} \subseteq \mathscr{A} \times \mathbb{N} \times \mathscr{A}$ if, and only if, there is a mapping $b : \mathsf{src} \times E \to [\![n]\!]$ such that $\sum_{s \in \mathsf{src}, e \in E} b(s,e) = n$, which intuitively means that $b(s,e)$ players are assigned to take edge e and their source vertex is s. Moreover, for every $v \in V$

$$\mathscr{C}(s,v) = \sum_{e=(v,\mathsf{f},v')} b(s,e) \quad w = \sum_{s \in V, e \in E} b(s,e) \times \mathsf{f}(b(s,e)) \quad \mathscr{C}'(s,v) = \sum_{e=(v',\mathsf{f},v)} b(s,e)$$

Similar to the abstract weighted graph, an src-abstract weighted path of an NCG is either a single src-abstract configuration or a non-empty, finite or infinite sequence of consecutive edges in $\mathscr{P}$. The cost of an src-abstract weighted path is the sum of weights w's of its edges (if any). Therefore, here we have a counterpart of Lemma 2.3 as follows:

Lemma 2.6. *For any $w \in \mathbb{N} \cup \{+\infty\}$, there is a path in $\mathcal{M}$ with social cost w if, and only if, there is an src-abstract weighted path in $\mathscr{P}$ with cost w.*

We now describe below an algorithm for solving the CONSTRAINED-SO problem for $\mathcal{G} = \langle A, n, \mathsf{g} \rangle$ and $K \in \mathbb{N}$.

Algorithm. We consider the src-abstract weighted graph $\mathscr{P} = \langle \mathscr{A}, \mathscr{B} \rangle$ and start a path finding algorithm from the vertex $\mathscr{C}_{\mathsf{src}} \in \mathscr{A}$, where $\mathscr{C}_{\mathsf{src}}$ is the src-abstract configuration corresponding to c_{src} of C. Hence, by definition, for $s \in \mathsf{src}$, $\mathscr{C}_{\mathsf{src}}(0,s) = \mathscr{C}_{\mathsf{src}}(s,s) = \sum_{t \in \mathsf{tgt}} \mathsf{g}(s,t)$ and for $v \in V \setminus \mathsf{src}$, $\mathscr{C}_{\mathsf{src}}(0,v) = 0$. Additionally, at each step of the algorithm, we maintain two counters L, len, both of which initialized at 0, and a matrix of size $|\mathsf{src}| \times |\mathsf{tgt}|$ M which is initialized at all zero entries. Here, L keeps track of the social cost up to the current step and len keeps track of the length of ongoing the src-abstract weighted path. Starting from $\mathscr{C}_{\mathsf{src}}$, at each step, we make a guess $(\mathscr{C}, w, \mathscr{C}') \in \mathscr{B}$. Then we check the following three things to declare the guess as "correct":

- Check whether $L + w \leq K$
- Check whether $len \leq [\![n]\!]^{|V|^2 + |V|}$
- If for some $s \in \mathsf{src}, t \in \mathsf{tgt}$, $\mathscr{C}'(s,t) > \mathscr{C}(s,t)$, check whether $M(s,t) \leq \mathsf{g}(s,t)$.

The first two checks are to maintain that we are guessing a path without a cycle in $\mathscr{P}$, and with social cost less than the threshold K. The third condition is to check when a new player reaches a target vertex t, whether the number of players with source s and target t is being maintained.

If all the above checks answers positively, we update the current vertex from $\mathscr{C}$ to $\mathscr{C}'$, and increment L and len by w and 1 respectively. Finally, for every s of src and t of tgt, we update $M(s,t)$ by adding $\mathscr{C}'(s,t) - \mathscr{C}(s,t)$. Then, we proceed to make the next guess unless $\forall t \in \mathsf{tgt}, \mathscr{C}'(0,t) = \sum_{s \in \mathsf{src}} \mathsf{g}(s,t)$. If one of the checks fails, it means the guess is not correct.

Again, here we have relied upon the following fact:

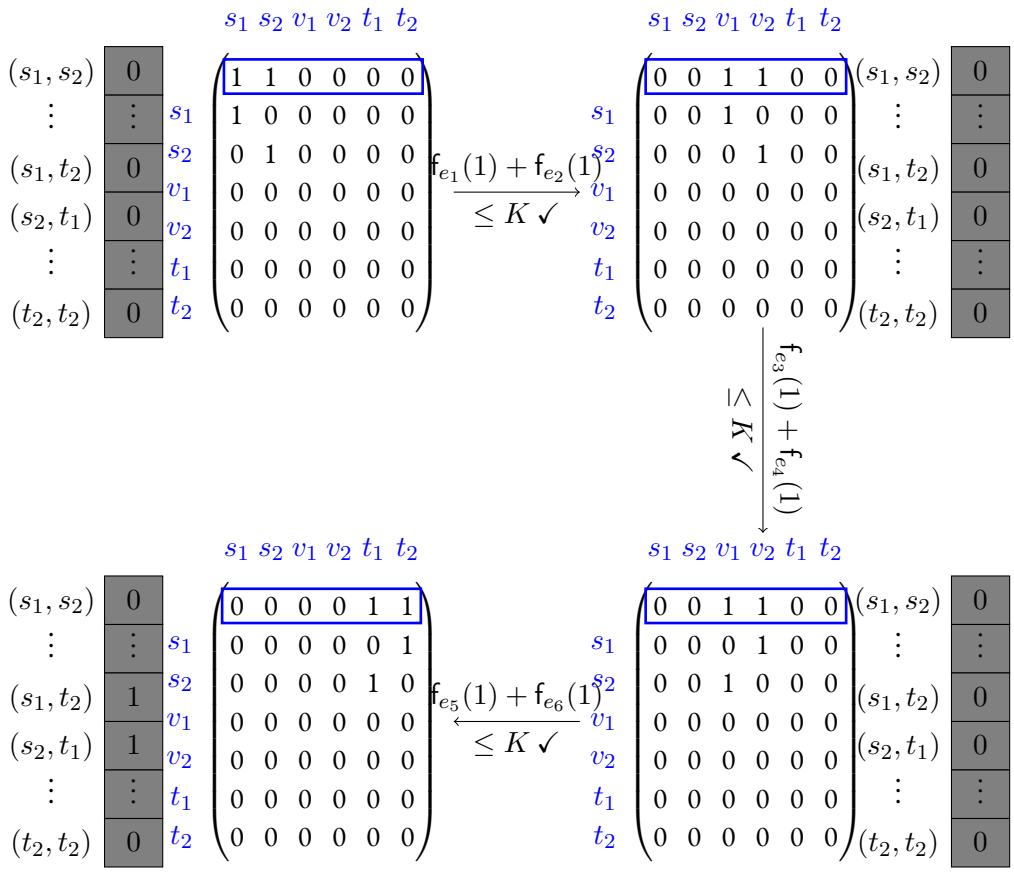


Figure 2.2: Depiction of a run of the CONSTRAINED-SO-algorithm on Example 2.5

Lemma 2.7. *If there is a weighted path in $\mathcal{M}$ with cost less or equal to K, then there is an* src*-abstract weighted path without a cycle in $\mathscr{P}$ with social cost less or equal to K.*

Proof. Suppose $\rho = (c_i, w_i, c_{i+1})_{0 \leq i \leq |\rho|-1}$ is a path in $\mathcal{M}$ with social cost less or equal to K. First of all, we can assume that ρ contains no cycle, because if it does, we can remove the cycle and get a path with lower social cost. Now our job is to show such a path in $\mathcal{M}$ has a corresponding path in $\mathscr{P}$, which also does not contain any cycle. This is not readily obvious because two configurations c and c' of $\mathcal{M}$ may have the same src-abstract configuration in $\mathscr{P}$. Here, we show that if two such configurations appear in ρ, we can construct a shorter path with less social cost. This construction intuitively truncates the part between these two configurations, but this is not exactly the same as removing a cycle, hence we need the following justification.

Let us assume there are two configurations c_j and c_l with $j < l \leq |\rho| - 1$ such that there exists a permutation map $\phi : [\![n]\!] \to [\![n]\!]$ satisfying $c_l(\phi(k)) = c_j(k)$ for all $k \in [\![n]\!]$. Then we can assume that there must be an index $k \in [\![n]\!]$ such that Player k and Player $\phi(k)$ do not have same source vertex. Because, if it is not the case , i.e, if for all $k \in [\![n]\!]$ $\mathsf{src}(\phi(k)) = \mathsf{src}(k)$, then we could construct another path $\hat{\rho} = (\hat{c}_i, w_i, \hat{c}_{i+1})_{0 \leq i \leq |\rho'|-1}$ in $\mathcal{M}$ as follows: Suppose Player k's

path in ρ is $s \xrightarrow{\pi_1} u \xrightarrow{\pi_2} v$ and Player $\phi(k)$'s path is $s \xrightarrow{\pi'_1} v \xrightarrow{\pi'_2} u$, then in $\hat{\rho}$, we assign Player k's path to $s \xrightarrow{\pi'_1} v$ and Player $\phi(k)$' path to $s \xrightarrow{\pi_1} u$, and so on for every such pair. Hence, we could take $\hat{\rho}$ instead of ρ.

Therefore, we assume for any two configurations c_j and c_l, with $j < l$, either they are not permutations of each other, or if they are, then there is at least one pair of that players that differ in their source vertex. From Lemma 2.6, we know there is a corresponding path $\tau = (\mathscr{C}_i, w_i, \mathscr{C}_{i+1})_{1 \leq i \leq |\tau|}$ in $\mathscr{P}$ with same social cost. Now, by construction, in τ, any two $\mathscr{C}_j$ and $\mathscr{C}_l$ must differ at some $(s, v) \in \mathsf{src} \times V$. Hence, τ doesn't contain a cycle. □

As the size of $\mathscr{P}$ is at most $[\![n]\!]^{(|V|+1)\times|V|}$, which is exponential in the input size, we need a counter of size that is polynomial in the input. Therefore, the non-deterministic path finding algorithm yields the following complexity result:

Theorem 2.8. *CONSTRAINED-SO can be solved in PSPACE.*

2.2 Complexity Lower Bound for CONSTRAINED-SO

In this section, we show that even for symmetric dynamic NCGs, the CONSTRAINED-SO problem is NP-hard by reducing a well-known NP-complete problem, namely the Partition problem. The reduction has been adapted from [47, Theorem 4.1] showing NP-hardness of the same problem in classical (non-dynamic) NCG.

Theorem 2.9. *The CONSTRAINED-SO problem is NP-hard in dynamic NCGs.*

Proof. Recall[38] that the partition problem asks whether a given set of positive integers can be partitioned into two subsets such that the sum of the numbers in the two subsets are equal.

Consider an instance $L = (r_i)_{i \in [\![m]\!]}$ of this Partition problem, and let $S = \sum_{r_i \in L} r_i/2$, $M = 14S+12m+1$, and $n = 2S+2m$. For any $r \in \mathbb{N}$, we define the threshold cost function T_r as $\mathsf{T}_r(i) = 1$ if $i \leq r$, and $\mathsf{T}_r(i) = M$ otherwise. We construct a dynamic NCG arena $\mathcal{A}_L$ with vertices $V = \{\mathsf{src}, \mathsf{tgt}, d_1, d_2\} \cup \{s_i, a_{i,1}, a_{i,2}\}_{i \in [\![m]\!]}$. The transitions are defined as follows:

- for each $i \in [\![m]\!]$, there is a transition from src to s_i with cost function T_{r_i+2};
- from each s_i, there is a transition to $a_{i,1}$ and another one to $a_{i,2}$ with the same cost function that assigns cost 2 for one player, and 4 for more;
- for $i \in [\![m]\!]$ and $j \in \{1, 2\}$, there is a transition from $a_{i,j}$ to d_j with constant cost 1, and a transition to tgt with cost function T_1
- from each d_j, there is a transition to tgt with cost function T_S.

The construction is illustrated in Figure 2.3.

We prove that there exists a strategy profile in $\mathcal{G}_L = \langle \mathcal{A}_L, n, \mathsf{g} \rangle$, where $\mathsf{g}(\mathsf{src}, \mathsf{tgt}) = n$, with social cost less than M if, and only if, L is a positive instance of the partition problem. Assume that the instance L has a solution $\langle L_1, L_2 \rangle$. We describe the behaviour of the strategy profile in $\mathcal{G}_L$ achieving social cost less than M.

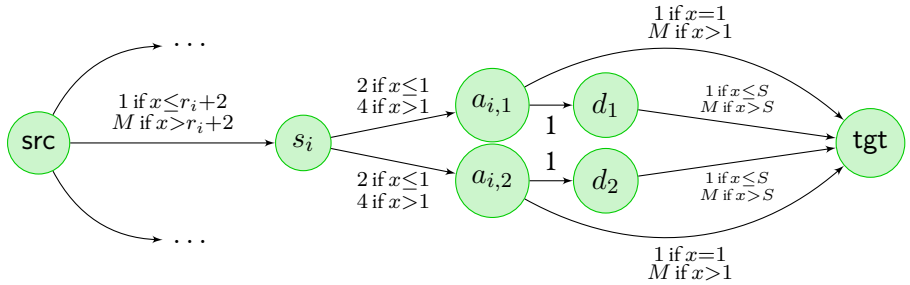


Figure 2.3: The reduction of Theorem 2.9. The figure shows the edges involving states $s_i, a_{i,1}, a_{i,2}$. The full graph is obtained by reproducing the shown construction for all $i \in [\![m]\!]$.

- Initially, all players are at src. During the first step, for each $i \in [\![m]\!]$, $(r_i + 2)$ players move to s_i. This incurs a total cost of $\sum_{i\in[\![m]\!]} r_i + 2 = \sum_{i\in[\![m]\!]} r_i + 2 \times m = 2S + 2m$;
- Consider $i \in [\![m]\!]$, and let $j \in \{1, 2\}$ be such that $r_i \in L_j$. From state s_i we let $r_i + 1$ players move to $a_{i,j}$, and one player to $a_{i,3-j}$. The $r_i + 1$ players each pay a cost of 4, and other player a cost of 2. Overall, we get a total cost of $\sum_{i\in[\![m]\!]}((r_i + 1) \times 4 + 1 \times 2) = 8S + 4m + 2m = 8S + 6m$;
- From each $a_{i,j}$, one player takes the transition directly to tgt. All other players move to d_j. Overall, $2m$ players directly move to tgt, while all other players, that are $2S$ of them, move to $\{d_1, d_2\}$. The total cost of this step is therefore $2S + 2m \times 2 = 2S + 4m$;
- From each d_j, players necessarily move directly to tgt. For each $i \in [\![m]\!]$ and $j \in \{1, 2\}$, exactly r_i players arrive to d_j from $a_{i,j}$. Thus there are exactly S players in each d_j, so that the total cost of this step is $2S$.

Summing over all steps, the social cost of this strategy profile is $14S + 12m < M$.

Now for the opposite direction, consider any strategy profile σ in $\langle \mathcal{A}, n\rangle$, outcome of which is π in the associated configuration graph, with social cost less than M. We are going to construct a partition of L. First we divide the path taken by the players from src to tgt in three phases: **Phase 1:** from src to some s_i; **Phase 2:** from some s_i to some $a_{i,j}$; **Phase 3:** from $a_{i,j}$ to tgt, either directly or via d_j. We now analyze the cost incurred by players under profile σ (or, in path π) in all three phases.

In phase 1, each player pays either cost 1 or cost M. By assumption on π, all players must have a cost of 1, and this is only possible if $r_i + 2$ players move to s_i, for each $i \in [\![m]\!]$. The total cost of this phase is thus $2S + 2m$.

In phase 3, a player pays a cost of either 2, M or $M + 1$. The latter two cases are not possible by assumption. So phase 3's contribution to the social cost has to be $2 \times n = 4S + 4m$. Then, the social cost of phase 2 is strictly less than $(14S + 12m + 1) - (2S + 2m) - (4S + 4m) = 8S + 6m + 1$. By phase 1, there are $r_i + 2$ players at each s_i, so the minimum contribution to the social cost for these $r_i + 2$ players is $2 \times 1 + 4 \times (r_i + 2) = 4r_i + 6$. Summing over all $i \in [\![m]\!]$, this yields $\sum_{r_i\in L}(4r_i + 6) = 8S + 6m$. Thus, the social cost of phase 2 is $8S + 6m$. But this

cost is achieved only when from each s_i exactly one transition is taken by one player and the other transition is taken by $r_i + 1$ players.

Therefore, each $a_{i,j}$ contains either one player or $(r_i + 1)$ players under σ. Given the cost functions, at most $2S$ players can move to tgt via d_1 or d_2 (otherwise players would get a cost of M). So $2m$ remaining players must take the transition from some $a_{i,j}$ to tgt. But at each $a_{i,j}$, there is a unique player that takes this transition due to the cost function. If $a_{i,j}$ contains $r_i + 1$ players, r_i players take the path via d_j, each paying a cost of 2. As there are in total $2S$ players taking the route via some d_j, and each d_j can contain at most S players because of the cost functions, there must be exactly S players which arrive at each d_j under σ. That is, for each i, j, there are exactly r_i players coming from some $a_{i,j}$ to d_j, and their total is S. This defines the required partition of L.

Finally, let us also point out that the reduction is polynomial. Indeed the size of the arena $\mathcal{A}_L$ is polynomial in the number of elements in the instance L and the number of players in $\mathcal{G}_L$ is exactly the sum of the elements considered in L. □

2.3 Conclusion

In this chapter, we consider the constrained SO problem, the decision version of which asks whether there is a strategy profile with social cost less than or equal to the given threshold.

To solve this problem, we exploit the game structure, and using an abstraction of configurations, we give a non-deterministic algorithm which verifies correctly if it is provided a strategy profile which indeed has the social cost less than or equal to the threshold. Thus, the computational complexity of the constrained SO problem lies in PSPACE. For lower bound, we show even for the symmetric case, the problem is NP-hard. Therefore, a future work would be to reduce the complexity gap for the constrained SO problem.

3 Nash Equilibria

In this chapter, we will talk about one of the solution concepts that is most common in the context of non-zero-sum multi-player games, namely Nash Equilibria. Here, in dynamic NCG, as mentioned earlier, we can imagine each player plays *selfishly*, i.e, their objective is to go from their source to target vertices with minimal accumulated individual cost. As a selfish player, each player tries to choose their strategy in a way that changing this strategy would not benefit to themselves if everyone else plays the same - this brings a stability in the players' choices, and that is formalized in a Nash equilibrium.

Recall Definition 1.8 from Chapter 1: a strategy profile is a Nash Equilibrium (NE) if each player's strategy in that profile is a best response to the rest of the players' strategies. In other words, no player can lower their cost by unilaterally deviating from it.

We first investigate whether NE exist in dynamic NCG in Section 3.1, and answer the question positively. We then address the problem whether in a given dynamic NCG, given a bound K there is a Nash equilibrium with social cost not more than K. We obtain complexity results on this problem in Section 3.2.

3.1 Existence of Nash Equilibria

In order to show existence of dynamic NCG, we first establish that NE exist when the strategy profiles, including deviations, are restricted to blind strategies, i.e, blind NE exist in dynamic NCG (recall definitions from Section 1.4.2). With that, we will then show that those blind NE are in fact NE in dynamic NCG.

To prove that blind Nash equilibria always exist, we establish that dynamic NCG with blind strategies are *potential games*[48, 56] which are known to have Nash Equilibria.

3.1.1 Existence of blind Nash Equilibria in dynamic NCG

A *potential function*[48] is a function which associates a value to a strategy profile, which is also referred to as a *potential value* of a strategy profile with respect to the corresponding function. Potential function must satisfy the following property: when a player unilaterally deviates from the strategy profile, the change in that player's accumulated cost between both strategy profiles is equal to the change in the potential value of the two profiles. A game is called a *potential games* if there exists a potential function associated to the game.

Consider a dynamic NCG $\mathcal{G} = \langle \mathcal{A}, n, \mathsf{g} \rangle$, a blind strategy profile $\pi = (\pi_i)_{i \in [\![n]\!]}$, and let N_π denote the maximum length of the paths prescribed by π. We define the following function, which is an adaptation of that used in [56]:

$$\psi(\pi) = \sum_{j=1}^{N_\pi} \sum_{e \in E} \sum_{i=1}^{\mathsf{load}_e(\pi, j)} \mathsf{f}_e(i)$$

where $\mathsf{load}_e(\pi, j)$ denotes the number of players who take the edge e at j^{th} step of π, and f_e is the cost function on edge e.

Lemma 3.1. *ψ is a potential function for $\mathcal{G}$ restricted to only blind strategies.*

Proof. Here we consider a blind strategy profile $\pi = (\pi_i)_{i \in [\![n]\!]}$ and another profile $\pi' = (\pi'_i, \pi_{-i})$, where Player i deviates from π_i to π'_i. As earlier, we also denote N_π and $N_{\pi'}$ to be the length of longest paths prescribed by π and π' respectively.

Now when Player i unilaterally deviates from its blind strategy π_i to π'_i, the only edges on which a change in the number of players happens are the edges which are part of exactly one of π_i and π'_i. Moreover, even on these edges the load differs by exactly 1: it gets increased on the edges which are in π'_i but not in π_i, and gets decreased on the edges which are in π_i but not in π'_i. Therefore, we have the following:

$$\begin{aligned}
\psi(\pi) - \psi(\pi') &= \sum_{j=1}^{N_\pi} \sum_{e \in E} \sum_{i=1}^{\mathsf{load}_e(\pi, j)} \mathsf{f}_e(i) - \sum_{j=1}^{N'_\pi} \sum_{e \in E} \sum_{i=1}^{\mathsf{load}_e(\pi', j)} \mathsf{f}_e(i) \\
&= \sum_{j=1}^{|\pi_i|} \sum_{e \in \pi_i} \mathsf{f}_e(\mathsf{load}_e(\pi_i, j)) - \sum_{j=1}^{|\pi'_i|} \sum_{e \in \pi'_i} \mathsf{f}_e(\mathsf{load}_e(\pi'_i, j)) \\
&= \mathsf{cost}_i(\pi) - \mathsf{cost}_i(\pi')
\end{aligned}$$

Hence, ψ indeed is a potential function. □

Using the above-defined potential function, one can derive an algorithm to find a Nash equilibrium, by a classical *best-response* iteration. Starting with an arbitrary blind strategy profile, at each step we replace some player's strategy with their best-response, and we continue as long as some player's cost can be decreased. When this procedure terminates, the profile at hand is a blind Nash equilibrium. In this argument, what remains is to show that in dynamic NCG with blind strategy profiles, best responses exist.

Lemma 3.2. *Given a dynamic NCG $\mathcal{G} = \langle \mathcal{A}, n, \mathsf{g} \rangle$, a blind strategy profile π, and $i \in [\![n]\!]$, Player i has a blind strategy π'_i that is their best response to π. This strategy has a size at most $N_\pi + |V|$ and can be computed in time $O(|V|^2.N_\pi^2)$.*

Proof. We let $\mathcal{A} = \langle V, E, \mathsf{src}, \mathsf{tgt} \rangle$. We define a weighted graph $\mathcal{G}$ obtained by concatenating $N_\pi + 1$ copies of $\mathcal{A}$, in which the moves of all players but Player i are hard-coded. Formally, $\mathcal{S} = \langle V \times [\![N_\pi + 1]\!], E' \rangle$ where:

- For each edge (v, f, v') in E, there is an edge $((v, N_\pi + 1), \mathsf{f}(1), (v', N_\pi + 1))$ in E': this encodes the fact that, after $N_\pi + 1$ steps, all other players have reached the target state, and Player i plays alone;

- For each edge $e = (v, \mathsf{f}, v')$ in E and each index $1 \leq k \leq N_\pi$, there is an edge $((v, k), w, (v', k + 1))$ in E' with $w = \mathsf{f}(load_e(\pi_{-i}, k) + 1)$, where $\mathsf{load}_e(\pi_{-i}, k)$ is the number of players (except Player i) taking edge e at step k when following π_{-i}. This way, w corresponds to the cost incurred to Player i if they were to take edge e at step k.

By construction, any blind strategy π'_i for Player i in $\langle \mathcal{A}, n, \mathsf{g} \rangle$ corresponds to a path ρ' from $(\mathsf{src}(i), 1)$ to $(\mathsf{tgt}(i), N_\pi + 1)$ in $\mathcal{S}$ such that the weight of ρ' is equal to $\mathsf{cost}_i(\pi_{-i}, \pi'_i)$: Player i can follow the exact same path as they would in $\mathcal{S}$ by ignoring the second component in the state space of $\mathcal{S}$.

Conversely, any path in $\mathcal{S}$ from $(\mathsf{src}(i), 1)$ to $(\mathsf{tgt}(i), N_\pi + 1)$ corresponds to a path from $\mathsf{src}(i)$ to $\mathsf{tgt}(i)$ in $\mathcal{A}$, which is a blind strategy π'_i. The cost of the path is equal to $\mathsf{cost}_i(\sigma_{-i}, \sigma'_i)$ by construction.

This shows that the best-response strategy can be computed by a shortest weighted path computation in $\mathcal{S}$. This path has size at most $N_\pi + |V|$: after N_π steps, the path enters configurations of the form $(c, N_\pi + 1)$, so all other players have already reached the target; since the shortest path is acyclic, the bound follows.

This computation can be done with Dijkstra's algorithm, which runs in $O((|V| \cdot N_\pi)^2)$. □

Finally, we have the following:

Theorem 3.3. *In dynamic NCG, blind Nash equilibria always exist, and we can compute one in pseudo-polynomial time.*

Proof. We apply the previous lemma as follows. Consider an initial strategy profile π_0 assigning to each player any acyclic path (thus, of length at most $|V|$) from their corresponding source to target. We have $\psi(\pi_0) \leq |V| \cdot \sum_{e \in E} \sum_{i=1}^{n} \mathsf{f}_e(i) \leq |V| \cdot n \cdot \max_{e \in E} \mathsf{f}_e(n)$. This quantity is pseudo-polynomial in the size of $\langle \mathcal{A}, n, \mathsf{g} \rangle$.

Let $\pi_1, \pi_2, \ldots$ denote the strategy profiles generated by this iterative procedure. Let $m_i = N_{\pi_i}$. By Lemma 3.2, we have $m_i \leq m_1 + (i-1)|V|$. We then have

$$\begin{aligned}
\sum_{i=1}^{k} |V|^2 m_i^2 &\leq |V|^2 \sum_{i=1}^{k} (m_1 + i|V|)^2 \\
&\leq |V|^2 \sum_{i=1}^{k} (n|V| + i|V|)^2 \\
&\leq |V|^4 \cdot k \cdot (n+k)^2
\end{aligned}$$

where the second step follows from $m_1 \leq n|V|$. Applying Lemma 3.2 again, the running time of the iterative procedure until the k-th step is $O(|V|^4 \cdot k \cdot (n+k)^2)$. In the worst case, we stop after $k = \psi(\pi_0)$ steps, so that the algorithm runs in pseudo-polynomial time. □

Remark 3.4. *As an alternative proof to existence of blind NE, we could have bounded the length of outcomes of blind NE as follows: all players have a strategy realizing cost at most $|V| \cdot \kappa$, where $\kappa = \max_{e \in E} \mathsf{f}_e(n)$, since the shortest path from $\mathsf{src}(i)$ to $\mathsf{tgt}(i)$ has length at most $|V|$, and the cost for a player at each step along edge e is at most κ. It follows that no path along which the cost for some player is larger than $|V| \cdot \kappa$ can be the outcome of a blind NE. As a consequence, if a dynamic NCG has a blind NE, then it has one of length at most $|V| \cdot \kappa \cdot |V|^n$ (by removing zero-cycles). Using this bound, we can transform dynamic NCG into classical congestion games, in which blind NE always exist [31, 56].*

We now show that blind NE are in fact NE. This is proved using the observation that given a blind strategy profile, the most profitable deviation for any player can be assumed to be a blind strategy.

Lemma 3.5. *In dynamic NCG, blind NE are NE.*

Proof. Consider a blind NE $\pi = (\pi_i)_{i \in [\![n]\!]}$ for dynamic game $\mathcal{G} = \langle \mathcal{A}, n, \mathsf{g} \rangle$. This means no player has a unilateral deviating blind strategy by which they can improve their cost.

Let us suppose π is not an NE of $\mathcal{G}$, which implies there is some player i who has a unilateral deviating strategy σ_i such that $\mathsf{cost}_i(\sigma_i, \pi_{-i}) < \mathsf{cost}_i(\pi)$. Now, let us consider $\mathsf{outcome}_i(\sigma_i, \pi_{-i})$ which is a path from $\mathsf{src}(i)$ to $\mathsf{tgt}(i)$ in $\mathcal{A}$. As all but Player i are playing with blind strategy profile, if we replace Player i's strategy σ_i by the blind strategy $\mathsf{outcome}_i(\sigma_i, \pi_{-i})$, none of the players' outcome and cost are going to change. Hence, $\mathsf{cost}_i(\mathsf{outcome}_i(\sigma_i, \pi_{-i}), \pi_{-i}) < \mathsf{cost}_i(\pi_i, \pi_{-i})$, which is a contradiction to the fact that π is a blind NE. Therefore, π, a blind NE, is indeed an NE for $\mathcal{G}$. □

Lemma 3.6. *There exists a dynamic NCG with an NE σ such that for all blind NE π', we have $\mathsf{cost}(\pi) < \mathsf{cost}(\pi')$.*

Proof. The proof is based on the dynamic NCG depicted on Fig. 3.1, for which we prove there is an NE with total cost 36, while any *blind* NE has higher social cost.

We consider the arena $\mathcal{A}$ shown in Fig. 3.1 with $n = 3$ players, with $\mathsf{g}(s, t) = 3$.

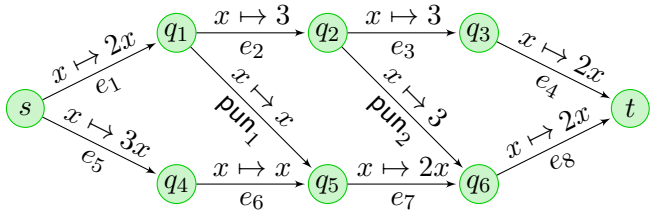


Figure 3.1: An arena on which blind Nash equilibria are sub-optimal.

The strategy profile $\sigma = (\sigma_i)_{i \in [\![3]\!]}$ is defined as follows.

For $i \in \{1, 2\}$, strategy σ_i chooses $e_1 e_2 e_3 e_4$ with the following exception: if Player $3 - i$ picks e_5 at s, then σ_i takes pun_1 at q_1; if Player $3 - i$ picks pun_1 at q_1, then σ_i takes pun_2 at q_2. Strategy σ_3 follows $e_5 e_6 e_7 e_8$.

We have $\mathsf{cost}_i(\pi) = 14$ for $i \in \{1, 2\}$, and $\mathsf{cost}_3(\pi) = 8$, so $\mathsf{cost}(\pi) = 36$.

We first show that π is a Nash equilibrium.

First, Player 3 has no incentive to deviate since taking e_1 at q_0 would alone cost 6, and the rest of the path, either through e_2 or $\mathsf{pun}_1 e_7$, has a cost more than 3 so that the total cost is more than $\mathsf{cost}_3(\pi)$. For Players 1 and 2, there are three states where they can deviate. If Player $i \in \{1, 2\}$ chooses e_5 at q_0, then the cost is $6 + 2 + 6 + 6 = 20$ as Player $3 - i$ chooses pun_1. Similarly, if they choose pun_1 at q_1 then Player $3 - i$ chooses pun_2 at q_2 which yields a cost of $4 + 1 + 4 + 6 = 15 > \mathsf{cost}_i(\pi)$. Last, if Player i chooses pun_2 at q_2, their cost is $4 + 3 + 3 + 4 = 14 = \mathsf{cost}_i(\pi)$.

We now show that the profile π has a lower cost than any blind Nash equilibrium.

There are only four possible blind strategies in our game following one of the four paths $\rho_1 = e_1e_2e_3e_4$, $\rho_2 = e_5e_6e_7e_8$, $\rho_3 = e_1\mathsf{pun}_1e_7e_8$ and $\rho_4 = e_1e_2\mathsf{pun}_2e_8$. We will represent these profiles as tuples of paths, *e.g.* (ρ_1, ρ_1, ρ_2). We are going to consider all possible tuples and show that each tuple is either not a Nash equilibrium or has cost more than $\mathsf{cost}(\pi) = 36$.

Observe that when we permute the players in a Nash equilibrium, it remains a Nash equilibrium with the same social cost. So we only need to consider the case where all players use the same strategy (4 possibilities), the case where they all choose distinct strategies ($\binom{4}{3}$), and the case where two of them choose the same strategy and the other one a different one ($\binom{4}{1} \times \binom{3}{1}$). So there are 20 profiles to check, and the rest of the profiles are permutations of these.

The following is an exhaustive list of the costs of these profiles.

$$\mathsf{cost}(\rho_1, \rho_1, \rho_1) = 18 \times 3 = 54$$
$$\mathsf{cost}(\rho_2, \rho_2, \rho_2) = 24 \times 3 = 72$$
$$\mathsf{cost}(\rho_3, \rho_3, \rho_3) = 21 \times 3 = 63$$
$$\mathsf{cost}(\rho_4, \rho_4, \rho_4) = 18 \times 3 = 54$$
$$\mathsf{cost}(\rho_1, \rho_2, \rho_3) = 12 + 13 + 12 = 37$$
$$\mathsf{cost}(\rho_1, \rho_3, \rho_4) = 12 + 14 + 15 = 41$$
$$\mathsf{cost}(\rho_1, \rho_2, \rho_4) = 12 + 12 + 13 = 37$$
$$\mathsf{cost}(\rho_2, \rho_3, \rho_4) = 14 + 15 + 16 = 45$$
$$\mathsf{cost}(\rho_1, \rho_1, \rho_2) = 14 \times 2 + 8 = \mathbf{36}$$
$$\mathsf{cost}(\rho_1, \rho_1, \rho_3) = 16 \times 2 + 11 = 43$$
$$\mathsf{cost}(\rho_1, \rho_1, \rho_4) = 14 \times 2 + 14 = 42$$
$$\mathsf{cost}(\rho_2, \rho_2, \rho_1) = 16 \times 2 + 10 = 42$$
$$\mathsf{cost}(\rho_2, \rho_2, \rho_3) = 20 \times 2 + 15 = 55$$
$$\mathsf{cost}(\rho_2, \rho_2, \rho_4) = 18 \times 2 + 14 = 50$$
$$\mathsf{cost}(\rho_3, \rho_3, \rho_1) = 16 \times 2 + 14 = 46$$
$$\mathsf{cost}(\rho_3, \rho_3, \rho_2) = 18 \times 2 + 16 = 52$$
$$\mathsf{cost}(\rho_3, \rho_3, \rho_4) = 18 \times 2 + 15 = 51$$
$$\mathsf{cost}(\rho_4, \rho_4, \rho_1) = 16 \times 2 + 14 = 46$$
$$\mathsf{cost}(\rho_4, \rho_4, \rho_2) = 16 \times 2 + 12 = 44$$
$$\mathsf{cost}(\rho_4, \rho_4, \rho_3) = 18 \times 2 + 15 = 51$$

All profiles have cost at least 36, and the only one that matches 36 is (ρ_1, ρ_1, ρ_2). However, this is not a Nash equilibrium. In fact, the cost of Player 1 here is 14, but they could profit from deviating to ρ_3 since $\mathsf{cost}_1(\rho_3, \rho_1, \rho_2) = 13$. □

3.2 Constrained NE Problem

Computing some (blind) NE may not be satisfactory for two reasons: one might want to compute the best (or the worst) NE in terms of the social cost; and as Lemma 3.6 claims, blind NE are suboptimal, *i.e.*, a lower social cost can be achieved by NE with general strategies. This justifies the study of more complex strategy profiles. In this section, we address these two concerns by considering a constrained NE problem, similar to `CONSTRAINED-SO` considered in the last chapter,

which decides existence of an NE with bounded social cost. As it has been shown in Chapter 2, a binary search using the solution to this decision problem gives us the optimal cost - here, in particular, optimal social cost for an NE in a given dynamic NCG. Hence, we consider the following problem:

Problem 3.7 (CONSTRAINED-NE)**.** *Given a dynamic NCG $\mathcal{G}$ and a bound $K \in \mathbb{N}$, does there exist an NE with social cost less than or equal to K?*

We first characterize the outcome of any NE, and then we use the characterization both to decide CONSTRAINED-NE and to define a strategy profile with the same outcome.

Characterization of outcomes of Nash Equilibria. Let us consider a dynamic NCG $\langle \mathcal{A}, n, \mathsf{g} \rangle$, and the corresponding game structure $\mathcal{S} = \langle C, T, M, U \rangle$. Given two configurations c, c' with $c \Rightarrow c'$, we let $\mathsf{cost}_i(c, c')$ denote the cost of Player i on this transition from $c(i)$ to $c'(i)$. We define $\mathsf{dev}_i(c, c')$ as the set of all configurations reachable when all players but Player i choose moves prescribed by the given transition $c \Rightarrow c'$:

$$\mathsf{dev}_i(c, c') = \{c'' \in C \mid c \Rightarrow c'' \text{ and } \forall j \in [\![n]\!] \setminus \{i\}.\, c''(j) = c'(j)\}.$$

Moreover, we define the *value* of configuration c for Player i is

$$\mathsf{val}_{i,c} = \sup_{\sigma_{-i} \in \mathfrak{S}^{-i}} \inf_{\sigma_i \in \mathfrak{S}_i} \mathsf{cost}_i((\sigma_{-i}, \sigma_i), c).$$

where recall $\mathfrak{S}_i$ is the set of all Player i strategies, and we denote $\mathfrak{S}^{-i}$ to be the set of strategy profiles for the coalition of all but Player i. Here, $\mathsf{val}_{i,c}$ corresponds to the value of a two player zero-sum game where Player i plays against the opposing coalition, starting at c. Intuitively, it is the worst cost that Player i cannot avoid to pay when the opposing coalition plays against them from configuration c. By [35], those values can be computed in polynomial time in the size of the game.

Lemma 3.8. *For a dynamic NCG $\mathcal{G} = \langle \mathcal{A}, n, \mathsf{g} \rangle$, the value of configuration c for Player i, $\mathsf{val}_{i,c}$, can be computed in EXPTIME,*

Proof. Here the two-player game between Player i and the opposing coalition is a game with state space $|V| \times [\![n-1]\!]^{|V| \times (|V|+1)}$ because each state keeps track of the position of Player i and the abstract position of the coalition similar to what we defined as an src-abstract weighted graph in Section 2.1.2. It follows that each $\mathsf{val}_{i,c}$ can be computed in exponential time in the size of the input $\langle \mathcal{A}, n, \mathsf{g} \rangle$. □

Moreover, memoryless optimal strategies exist (in $\mathcal{S}$), that is, the opposing coalition has a memoryless strategy σ_{-i} to ensure a cost of at least $\mathsf{val}_{i,c}$ from c.

The characterization of Nash equilibria outcomes is given in the following lemma.

Lemma 3.9. *For dynamic NCG $\mathcal{G} = \langle A, n, \mathsf{g} \rangle$, a path ρ in the corresponding concurrent game structure $\mathcal{S}$ is the outcome of a Nash equilibrium if, and only if,*

$$\forall i \in [\![n]\!].\, \forall 1 \leq l < |\rho|.\, \forall c \in \mathsf{dev}_i(\rho(l), \rho(l+1)).\quad \mathsf{cost}_i(\rho_{\geq l}) \leq \mathsf{val}_{i,c} + \mathsf{cost}_i(\rho(l), c).$$

The intuition is that if the suffix $\mathsf{cost}_i(\rho_{\geq l})$ of ρ has cost more than $\mathsf{val}_{i,c} + \mathsf{cost}_i(\rho(l), c)$, then Player i has a profitable deviation regardless of the strategy of the opposing coalition, since $\mathsf{val}_{i,c}$ is the maximum cost that the coalition can inflict to Player i at configuration c where the deviation is observed. The lemma shows that the absence of such a suffix means that a NE with given outcome exists, which the proof constructs.

Proof. Consider a Nash equilibrium $\sigma = (\sigma_i)_{i \in [\![n]\!]}$ with outcome ρ. Consider any player i, and any strategy σ'_i for this player. Let ρ' denote the outcome of $\sigma[i \to \sigma'_i]$. Let l denote the index of the last configuration where ρ and ρ' are identical. Since σ is a Nash equilibrium, we have $\mathsf{cost}_i(\rho) \leq \mathsf{cost}_i(\rho')$, that is,

$$\mathsf{cost}_i(\rho_{\geq l}) \leq \mathsf{cost}_i(\rho(l), \rho'(l+1)) + \mathsf{cost}_i(\sigma[i \to \sigma'_i], \rho'_{\leq l+1})$$

where $\mathsf{cost}_i(\sigma[i \to \sigma'_i], \rho'_{\leq l+1})$ is the cost for Player i of the outcome of the residual strategy $(\sigma[i \to \sigma'_i])^{\rho'_{\leq l+1}}$. Since the choice of σ'_i is arbitrary here, we have,

$$\mathsf{cost}_i(\rho_{\geq l}) \leq \mathsf{cost}_i(\rho(l), \rho'(l+1)) + \inf_{\sigma'_i \in \mathfrak{S}} \mathsf{cost}_i(\sigma[i \to \sigma'_i], \rho'_{\leq l+1}).$$

Moreover, we have $\inf_{\sigma'_i \in \mathfrak{S}} \mathsf{cost}_i(\pi[i \to \sigma'_i], \rho'_{\leq l+1}) = \inf_{\sigma'_i \in \mathfrak{S}} \mathsf{cost}_i(\pi[i \to \sigma'_i], \rho'(l+1))$ since memoryless strategies suffice to minimize the cost [35]. We then have

$$\inf_{\sigma'_i \in \mathfrak{S}} \mathsf{cost}_i(\pi[i \to \sigma'_i], \rho'(l+1)) \leq \sup_{\sigma_{-i} \in \mathfrak{S}^{-i}} \inf_{\sigma_i \in \mathfrak{S}} \mathsf{cost}_i((\sigma_{-i}, \sigma_i), \rho'(l+1)).$$

We obtain the required inequality

$$\begin{aligned} \mathsf{cost}_i(\rho_{\geq l}) &\leq \mathsf{cost}_i(\rho(l), \rho'(l+1)) + \sup_{\sigma_{-i} \in \mathfrak{S}^{-i}} \inf_{\sigma_i \in \mathfrak{S}} \mathsf{cost}_i((\sigma_{-i}, \sigma_i), \rho'(l+1)) \\ &\leq \mathsf{cost}_i(\rho(l), c) + \mathsf{val}_{i,c}. \end{aligned}$$

Conversely, consider a path ρ that satisfies the condition. We are going to construct a Nash equilibrium having outcome ρ. The idea is that players will follow ρ, and if some player i deviates, then the coalition $-i$ will apply a joint strategy to maximize the cost of Player i, thus achieving at least $\mathsf{val}_{i,c}$, where c is the first configuration where deviation is detected.

Let us define the punishment function $\mathcal{P}_\rho \colon \mathsf{Paths}(\langle \mathcal{A}, n, \mathsf{g} \rangle, c_{\mathsf{src}}) \to [\![n]\!] \cup \{\bot\}$ which keeps track of the deviating players and the step where such a player has deviated, here c_{src} is the initial configuration where Player i is at $\mathsf{src}(i)$. For path $h' = h(c, w, c')$, we write

$$\mathcal{P}_\rho(h') = \begin{cases} \bot & \text{if } h' \leq_{\mathsf{pref}} \rho, \\ i & \text{if } h \leq_{\mathsf{pref}} \rho, h(c, w, c') \not\leq_{\mathsf{pref}} \rho, \text{ and } i \in [\![n]\!] \text{ min. s.t. } c'(i) \neq \rho(|h|+1)(i), \\ \mathcal{P}_\rho(h) & \text{otherwise.} \end{cases}$$

Intuitively, $\bot$ means that no players have deviated from ρ in the current path. If $\mathcal{P}_\pi(h) = j$, then Player j was among the first players to deviate from ρ in the path h; so for some l, $h(l)(j) = \rho(l)(j)$ but $h(l+1)(j) \neq \rho(l+1)(j)$. Notice that if several players deviate at the same step, there are

no conditions to be checked, and the strategy can be chosen arbitrarily. For each configuration c and coalition $-i$, let $\sigma_{-i,c}$ be the strategy of coalition $-i$ maximizing the cost of Player i from configuration c; thus achieving at least $\mathsf{val}_{i,c}$. Player j's strategy in this coalition, for $j \neq i$, is denoted $\sigma_{-i,c,j}$. For path $h' = h(c, w, c')$, define

$$\tau_i(h') = \begin{cases} (c'(i), m(i), c''(i)) & \text{if } \mathcal{P}_\rho(h') = \bot, \rho(|h'|+1) = (c', w', c''), \\ & \text{and } m \in E^n \text{ is such that } T(c', m) = (w', c''), \\ \text{arbitrary} & \text{if } \mathcal{P}_\rho(h') = i, \\ \sigma_{-j,c,i}(h') & \text{if } \mathcal{P}_\rho(h') = j \text{ for some } j \neq i. \end{cases}$$

The first case ensures that the outcome of the profile $(\tau_i)_{i\in[\![n]\!]}$ is ρ. The third case means that Player i follows the coalition strategy $\sigma_{-j,c}$ after Player j has deviated to configuration c. The second case corresponds to the case where Player i has deviated: the precise definition of this part of the strategy is irrelevant.

Let us show that this profile is indeed a Nash equilibrium. Consider any player $j \in [\![n]\!]$ and any strategy τ'_j. Let ρ' denote the outcome of (τ_{-j}, τ'_j), and l the index of the last configuration where ρ and ρ' are identical. We have

$$\begin{aligned} \mathsf{cost}_j((\tau_{-j}, \tau'_j)) &= \mathsf{cost}_j(\rho_{\leq l}) + \mathsf{cost}_j(\rho(l), \rho'(l+1)) + \mathsf{cost}_j((\tau_{-j}, \tau_j), \rho'_{\leq l+1}) \\ &\geq \mathsf{cost}_j(\rho_{\leq l}) + \mathsf{cost}_j(\rho(l), \rho'(l+1)) + \mathsf{val}_{j,\rho'(l+1)(j)} \\ &\geq \mathsf{cost}_j((\tau_i)_{i\in[\![n]\!]}), \end{aligned}$$

where the second line follows from the fact that the coalition switches to a strategies ensuring a cost of at least $\mathsf{val}_{j,\rho'(l)(j)}$ at step l; and the third line is obtained by assumption. This shows that $(\tau_i)_{i\in[\![n]\!]}$ is indeed a Nash equilibrium and concludes the proof. □

Even though, in this chapter, our main concern is CONSTRAINED-NE, we can easily generalize the problem with a bound, not just for the social cost, but also for any one or more player's individual accumulated cost. For an integer vector $\vec{\gamma} = (\gamma_i)_{i\in[\![n]\!]}$, and for a path ρ in the multi-weighted configuration graph $\mathcal{M} = \langle C, T\rangle$ associated to a dynamic NCG $\mathcal{G}$, *$\vec{\gamma}$-social cost* is defined by:

$$\mathsf{soccost}_{\vec{\gamma}}(\rho) \coloneqq \sum_{i\in[\![n]\!]} \gamma_i \mathsf{cost}_i(\rho)$$

Definition 3.10 ($\vec{\gamma}$-minimal NE)**.** *For an integer vector $\vec{\gamma} = (\gamma_i)_{i\in[\![n]\!]}$, a strategy profile is called a* $\vec{\gamma}$-minimal NE *if it is an NE with optimal $\vec{\gamma}$-social cost.*

In the following, we describe how we compute $\vec{\gamma}$-minimal NE, which for $\gamma_i = 1$ for all $i \in [\![n]\!]$ gives us the optimal NE.

Algorithm. We define a graph that describes the set of outcomes of Nash equilibria by augmenting the n-weighted configuration graph $\mathcal{M} = \langle C, T\rangle$. For any integer vector $\vec{\gamma} = (\gamma_i)_{i\in[\![n]\!]}$, we define the weighted graph $\mathcal{M}_{\langle\mathcal{A},n,\mathbf{g}\rangle,\vec{\gamma}} = \langle C', T'\rangle$ with $C' = C \times ([\![Y]\!] \cup \{0, \infty\})^n$ where $Y = |V| \cdot \max_{e\in E} \mathsf{f}_e(n)$, and $T' \subseteq C' \times \mathbb{N} \times C'$; remember that all players have a strategy

realizing cost at most Y in $\langle \mathcal{A}, n, \mathsf{g} \rangle$. The initial state is $(c_{\mathsf{src}}, \infty^n)$. The set of transitions T' is defined as follows: $((c, b), z, (c', b')) \in T'$ if, and only if, there exists $(c, w, c') \in T$, $z = \vec{\gamma} \cdot w$ (where $\cdot$ is dot product), and for all $i \in [\![n]\!]$,

$$b'_i = \min(b_i - w_i, \min_{c'' \in \mathsf{dev}_i(c(i), c'(i))} \mathsf{cost}_i(c, c'') + \mathsf{val}_{i,c''} - w_i). \tag{3.1}$$

Notice that by definition of C', b'_i must be non-negative for all $i \in [\![n]\!]$, so there are no transitions $((c, b), z, (c', b'))$ if the above expression is negative for some i. Notice also that the size of $\mathcal{M}_{\langle \mathcal{A}, n, \mathsf{g} \rangle, \vec{\gamma}}$ is doubly-exponential in that of the input $\langle \mathcal{A}, n \rangle$, since this is already the case for C, while Y is singly-exponential.

Intuitively, for any path ρ that visits some state (c, b) in this graph, in order for ρ to be compatible with a Nash equilibrium, each player i must have cost no more than b_i in the rest of the path. In fact, the second term of the minimum in (3.1) is the least cost Player i could guarantee by not following (c, w, c') but going to some other configuration $c'' \in \mathsf{dev}_i(c, c')$, so the bound b_i is used to guarantee that these deviations are not profitable. The definition of b'_i in (3.1) is the minimum of $b_i - w_i$ and the aforementioned quantity since we check both the previous bound b_i, updated with the current cost w_i (which gives the left term), and the non-profitability of a deviation at the previous state (which is the right term). If this minimum becomes negative, this precisely means that at an earlier point in the current path, there was a strategy for Player i which was more profitable than the current path regardless of the strategies of other players; so the current path cannot be the outcome of a Nash equilibrium. This is why the definition of $\mathcal{M}_{\langle \mathcal{A}, n, \mathsf{g} \rangle, \vec{\gamma}}$ restricts the state space to nonnegative values for the b_i.

We prove that computing the cost of a Nash equilibrium minimizing the $\vec{\gamma}$-weighted social cost reduces to computing a shortest path in $\mathcal{M}_{\langle \mathcal{A}, n, \mathsf{g} \rangle, \vec{\gamma}}$. In particular, letting $\gamma_i = 1$ for all $i \in [\![n]\!]$, a $\vec{\gamma}$-minimal Nash equilibrium is a best Nash equilibrium (minimizing the social cost), while taking $\gamma_i = -1$ for all $i \in [\![n]\!]$, we get a worst Nash equilibrium (maximizing the social cost).

Theorem 3.11. *For dynamic NCG $\langle \mathcal{A}, n, g \rangle$ and vector $\vec{\gamma}$, the cost of the shortest path from $(c_{\mathsf{src}}, \infty^n)$ to some (c_{tgt}, b) in $\mathcal{M}_{\langle \mathcal{A}, n, \mathsf{g} \rangle, \vec{\gamma}}$ is the cost of a $\vec{\gamma}$-minimal Nash equilibrium.*

Proof. We show that for each path of $\langle \mathcal{A}, n, \mathsf{g} \rangle$ from c_{src} to c_{tgt}, there is a path in $\mathcal{M}_{\langle \mathcal{A}, n, \mathsf{g} \rangle, \vec{\gamma}}$ from $(c_{\mathsf{src}}, \infty^n)$ to some (c_{tgt}, b) with the same cost, and vice versa.

Consider a Nash equilibrium $\pi = (\sigma_j)_{j \in [\![n]\!]}$ with outcome $\rho = (c_j, w_j, c_{j+1})_{1 \leq j < l}$. We build a sequence $b_1, b_2, \ldots$ such that $\rho' = ((c_j, b_j), \vec{\gamma} \cdot w_j, (c_{j+1}, b_{j+1}))_{1 \leq j < l}$ is a path of $\mathcal{M}_{\langle \mathcal{A}, n, \mathsf{g} \rangle, \vec{\gamma}}$. We set $b_1(j) = \infty$ for all $j \in [\![n]\!]$. For $j \geq 1$, define

$$b_{j+1}(i) = \min\left(b_j(i) - w_j(i), \min_{c'' \in \mathsf{dev}_j(c_j(i), c_{j+1}(i))} \mathsf{cost}_i(c_j, c'') + \mathsf{val}_{i,c''} - w_j(i)\right).$$

We are going to show that for all $1 \leq j \leq l$, $\mathsf{cost}_i(\rho_{\geq j}) \leq b_j$, which shows that $b_j \geq 0$, and thus ρ' is a path of $\mathcal{M}_{\langle \mathcal{A}, n, \mathsf{g} \rangle, \vec{\gamma}}$.

We show this by induction on j. This is clear for $j = 1$. Assume this holds up to $j \geq 1$. We have, by induction that $\mathsf{cost}_i(\rho_{\geq j}) \leq b_j(i)$ for all $i \in [\![n]\!]$. Moreover, since π is a Nash equilibrium, by Lemma 3.9,

$$\forall i \in [\![n]\!], \mathsf{cost}_i(\rho_{\geq j}) \leq \min_{c'' \in \mathsf{dev}_i(\rho(j),\rho(j+1))} \mathsf{val}_{i,c''} + \mathsf{cost}_i(\rho(j), c'').$$

Therefore,

$$\begin{aligned}\mathsf{cost}_i(\rho_{\geq j+1}) &= \mathsf{cost}_i(\rho_{\geq j}) - w_j(i) \\ &\leq \min(b_j(i) - w_j(i), \min_{c'' \in \mathsf{dev}_i(\rho(j),\rho(j+1))} \mathsf{val}_{i,c''} + \mathsf{cost}_i(\rho(j), c'') - w_j(i))\end{aligned}$$

as required, and both paths have the same $\vec{\gamma}$-weighted cost.

Consider now a path $((c_i, b_i), z_i, (c_{i+1}, b_{i+1}))_{1 \leq i < l}$ in $\mathcal{M}_{\langle \mathcal{A},n,\mathrm{g}\rangle,\vec{\gamma}}$. By the definition of $\mathcal{M}_{\langle \mathcal{A},n,\mathrm{g}\rangle,\vec{\gamma}}$, there exists $w_1, w_2, \ldots$ such that $\rho = (c_j, w_j, c_{j+1})_{1 \leq j < l}$ is a path of $\langle \mathcal{A}, n, \mathrm{g}\rangle$, and $z_j = \vec{\gamma} \cdot w_j$. So it only remains to show that that ρ is the outcome of a Nash equilibrium. We will show that ρ satisfies the criterion of Lemma 3.9. We show by backwards induction on $1 \leq j \leq l$ that for all $i \in [\![n]\!]$,

1. $\mathsf{cost}_i(\rho_{\geq j}) \leq b_j(i)$,
2. $\mathsf{cost}_i(\rho_{\geq j}) \leq \min_{c'' \in \mathsf{dev}_i(\rho(j),c'')} \big(\mathsf{cost}_i(\rho(j), c'') + \mathsf{val}_{i,c''}\big)$.

For $j = l$, we have $\mathsf{cost}_i(\rho_{\geq l}) = 0$ so this is trivial. Assume the property holds down to $j + 1$ for some $1 \leq j < l$. By induction hypothesis, we have

$$\mathsf{cost}_i(\rho_{\geq j+1}) \leq b_{j+1}(i) = \min\left(b_j(i) - w_j(i), \min_{c'' \in \mathsf{dev}_i(\rho(j),c'')} \mathsf{cost}_i(\rho(j), c'') + \mathsf{val}_{i,c''} - w_j(i)\right).$$

Therefore,

$$\mathsf{cost}_i(\rho_{\geq j}) = \mathsf{cost}_i(\rho_{\geq j+1}) + w_j(i) \leq \min\left(b_j(i), \min_{c'' \in \mathsf{dev}_i(\rho(j),c'')} \mathsf{cost}_i(\rho(j), c'') + \mathsf{val}_{i,c''}\right),$$

as required. By Lemma 3.9, ρ is the outcome of a Nash equilibrium. □

Thanks to Theorem 3.11, we can compute the costs of the best and worst NE of $\langle \mathcal{A}, n, \mathrm{g}\rangle$ in exponential space. We can also decide the existence of an NE with constraints on the costs (both social and individual), by non-deterministically guessing an outcome and checking in $\mathcal{M}_{\langle \mathcal{A},n,\mathrm{g}\rangle,\vec{\gamma}}$ that it is indeed an NE. We obtain the following conclusion:

Corollary 3.12. *In dynamic NCG, the* CONSTRAINED-NE *problem is in* EXPSPACE.

Proof. As noted earlier, the number of vertices in $\mathcal{M}_{\langle \mathcal{A},n,\mathrm{g}\rangle,\vec{\gamma}}$ is doubly exponential since $|C| = |V|^n$ is doubly exponential. Storing a configuration and computing its successors can be performed in exponential space. One can thus guess a path of size at most the size of the graph and check whether its cost is less than the given bound. This can be done using counters that can be stored (in binary) in exponential space. Hence, the overall algorithm lies in EXPSPACE. □

Note that one can effectively compute a NE profile satisfying the constraints in doubly-exponential time by finding the shortest path of $\mathcal{M}_{\langle \mathcal{A},n,\mathrm{g}\rangle,\vec{\gamma}}$, and applying the construction of (the proof of) Lemma 3.9.

3.3 Conclusion

To recap, in this chapter, we first obtained the existence of Nash equilibria in dynamic network congestion games: first only for blind strategy profiles, and then establishing the fact that blind NE are NE for dynamic NCG.

Subsequently, we observed that, there can be dynamic games with NE which are better than any blind NE in that dynamic games: better in terms of social cost. Therefore, we investigated how to tackle the following problem : Given a dynamic game and a constraint, does there exist an NE with social cost upper-bounded by the constraint? or are there NE with social cost at least as much as the given bound? We obtained upper-bound complexity results for the above problem, and hardness results of these problem are yet to be answered.

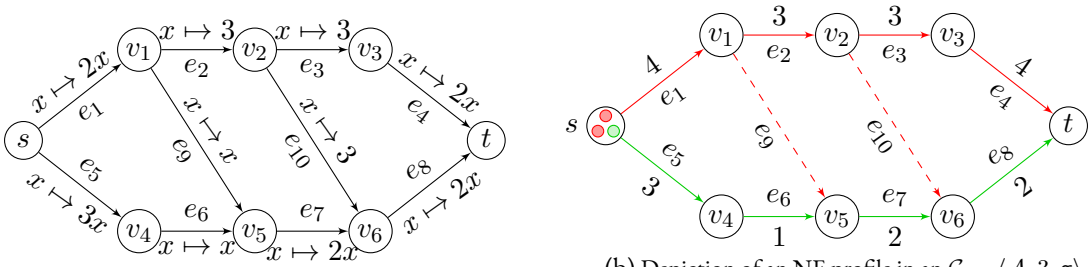
4 Subgame Perfect Equilibria

In the last chapter, we considered Nash Equilibria (NE) as a solution concept of congestion games. Intuitively, such an equilibrium gives a recipe to the players such that if each player follows the strategy, then they cannot do better by switching from it at any step. In other words, an NE is a stable strategy profile, it is immune to any unilateral deviation at every step of its outcome. The last point is crucial, i.e, it is stable with respect to any unilateral deviation only when a player tries to deviate from a configuration which appears in a prefix of the outcome of the strategy profile.

In this chapter, we will be discussing another solution concept that we often consider in the context of non-zero sum multi-player games: namely, *Subgame perfect Equilibria* (SPE). Intuitively this is also a strategy profile in which no player would benefit by changing their strategy unilaterally at any step of the play like NE. On top of that, if the players decide to play according to an SPE, not necessarily from the configurations that are in a prefix of the outcome of the profile, but also from any arbitrary configuration. When the players decide to play such a strategy profile, from that step on ward, no player would benefit by switching their strategy from the one prescribed by the profile.

This might make SPE to be an interesting solution concept to study on its own, but we explain further in the following with an example why SPE is a necessary optimal solution concept in a dynamic setting.

Non-credible threats. Let us first recall Example 1.9 where the arena and an NE are shown again at Figure 4.1 here.



(a) An arena $\mathcal{A}$ for a dynamic NCG

(b) Depiction of an NE profile in an $\mathcal{G} = \langle \mathcal{A}, 3, \mathsf{g} \rangle$ where $\mathsf{g}(s, t) = 3$

Figure 4.1: An example of NE with non-credible threats

In this example, in the NE profile depicted in 4.1b, Players 1 and 2 (who are depicted by red tokens) play a strategy in which they *threat* each other: to be specific, they "punish" the other player by switching to the dotted edge immediately in the next step (from v_1 or v_2) if they see that the other player had chosen to take the bottom path (from s or v_1 respectively). By forcing them

to take the above path together, these threats make both the players pay a higher cost than they would have paid if they did not threaten to "punish" each other in the profile. These threats, which are costlier for the player who threatens, are called *non-credible threats*, and may appear in an NE profile which is supposed to be somewhat cost-effective for all players.

These threats appear in NE profiles, because intuitively, an NE profile keeps a player's interest only in those histories that are prefixes of the outcome of the profile itself, not in any other history. For example, if we consider an history in the above game, in which Player 1 is at v_1 already, and Players 2 and 3 are at vertex v_4, then starting from that history, applying the NE profile depicted in Figure 4.1b is not a best-response for all players. The profile prescribes Player 1 to take edge e_9 in the next step, but the cheaper response would be to continue on the path $v_1 \xrightarrow{e_2} v_2 \xrightarrow{e_3} v_3 \xrightarrow{e_4} t$. This example illustrates that when the players can take dynamic decisions, an SPE is a better fitted solution concept for stability than an NE. This is why we consider studying SPE in this chapter. Now, let us recall the definition from Chapter 1, which says, an NE profile σ is an SPE if for any history h ending at any configuration c, the residual strategy $\sigma|_h$ is also NE from c.

In the next section, we briefly discuss the existence of SPE in dynamic NCG, which is an open question. Next, in Section 4.2, we consider a decision problem like the ones that we considered in case of SO and NE in earlier chapters, and give a 2EXPSPACE algorithm for solving it.

4.1 On Existence of SPE in Dynamic NCG

Like Nash Equilibria, the first natural question that arises for SPE is whether SPE always exist in dynamic NCG. Unfortunately, this is still an open question for our model. For our running example, the game of Figure 4.1a, we have already shown an SPE in Figure 1.3, in Chapter 1.

In the case of turn-based reachability games [17], existence of an SPE is shown by reducing decision making at possibly infinitely many nodes of unraveling tree to decision making only at finite nodes. Moreover, an SPE is defined by assigning plays from those finite nodes by applying Kuhn's Theorem[42] on an extensive game structure. We can not apply a similar argument because our model is concurrent - hence we do not have an extensive game structure. On the other hand, the closest model of ours, dynamic resource allocation games studied by Avni et al [8] shows an example where SPE do not exist. Their model somewhat generalizes dynamic network congestion games, and their example in which an SPE does not exist cannot be captured in a network game.

Nonetheless, we could always ask given a game and a bound whether there is an SPE with social cost bounded by the given threshold. If in future, it turns out there is a game where SPE does not exist, then for the above decision problem, for any threshold, the answer would always be No.

4.2 Constrained SPE problem

In this section, we address the associated constrained SPE problem, similar to what we did in the case of Nash equilibria and Social optima:

Problem 4.1 (CONSTRAINED-SPE)**.** *Given a dynamic NCG $\mathcal{G}$ and a bound $K \in \mathbb{N}$, decide if there exist an SPE profile with social cost less or equal to K.*

To solve the above problem, we first characterize the outcome of an SPE, and then obtain an algorithm for CONSTRAINED-SPE based on the characterization. But before that, let us first see the main factors that differentiate this characterization with that of NE.

DIFFICULTY OF THE CHARACTERIZATION OF SPE COMPARED TO NE. In order to characterize a play as the outcome of an NE in Lemma 3.9, we needed to make sure that when a player deviates, there exists a strategy profile of the other players by which the deviating player is made to pay more cost than they would have if they had not deviated.

Note that, after a deviation, the strategy profile made of a coalition "punishing" strategies and a deviating strategy has no other requirement to fulfill than to increase the cost of the deviating player. Hence we could make use of the $val_{i,c}$ and the memoryless coalition strategy profile which realizes $val_{i,c}$ as Player i (the deviating player)'s cost. Recall that $val_{i,c}$ is the cost that the opposing coalition can force Player i to pay if they deviate from configuration c. In contrast, when we characterize a play as the outcome of an SPE, additionally we need to make sure that the profile made of the corresponding "punishing" coalition strategies and any deviating strategy is an SPE. This additional condition makes the characterization of SPE outcomes technically more challenging than that of its counterpart for NE.

The characterization we propose is inspired from the one in [15] for SPE of a turn-based quantitative reachability game, where it is also guaranteed that SPE always exist [17]. Before going into details of the characterization, we first introduce some notions and establish their connection with SPE:

WEAK SPE [16]. Intuitively, an SPE is immune to any single player deviation, from any step of any history, which appears to be a strong requirement for a strategy profile (and hence its outcome). In the following, we define a weak version of SPE, which is easier to characterize. Then we establish an equivalence between this *weak SPE* and SPE, and hence reduce the characterization of SPE outcomes to that of weak SPE outcomes.

A strategy σ_i' is said to be *first-shot deviating* from another strategy σ_i if they coincide on all histories except the empty history. With this, we first define a weak version of Nash equilibria, and then refine later it to weak SPE, as follows:

Definition 4.2. *A strategy profile $\sigma = (\sigma_i)_{i \in [\![n]\!]}$ is called a* weak NE *if for all $i \in [\![n]\!]$, for every first-shot deviating strategy σ_i', it holds that $\mathsf{cost}_i(\langle \sigma_{-i}, \sigma_i' \rangle) \geq \mathsf{cost}_i(\sigma)$.*

Remark 4.3. *In [16],* weak *NE is a profile which is immune to* one-shot deviating strategies *and* very-weak *NE is a profile which is immune to* first-shot *deviating strategies; here we do not use one-shot deviating strategies, hence in our terminology there are only weak NE (and weak SPE), which are very-weak NE (and very-weak SPE respectively) in [16].*

In general a weak NE is not an NE.

Lemma 4.4. *There exists an NCG that admits a weak NE which is not an NE.*

Proof. The simple idea is a strategy profile might be immune to single player deviation at its first step, but that does not imply it will be immune to single player deviation at any step.

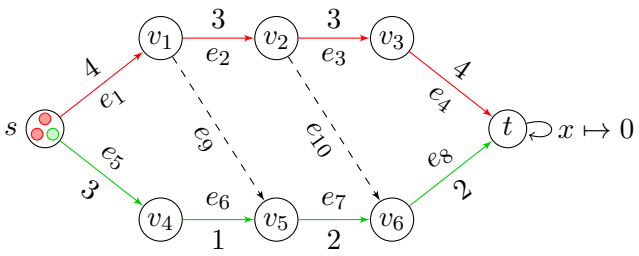


Figure 4.2: An weak NE but not an NE

Consider the game from Figure 4.1a and a weak NE of the game depicted as in Figure 4.2. Here two players decide at the beginning, and take the path colored in red, and another player takes the green path. The first two players' cost is 14, and the other player's cost is 8. If one of the first two players changes their strategy at the very first step, they will end up paying cost 16. One the other hand, if the player who is currently taking the green path changes their strategy at the very step and decides to take the red path, they will pay cost 18. Therefore, here no player has a first-shot deviating strategy by which they can lower their current cost, so that the profile is a weak NE.

On the contrary, one of the players, who are currently taking the red path, can switch their path from $s \xrightarrow{e_1} v_1 \xrightarrow{e_2} v_2 \xrightarrow{e_3} v_3 \xrightarrow{e_4} t$ to $s \xrightarrow{e_1} v_1 \xrightarrow{e_9} v_5 \xrightarrow{e_7} v_6 \xrightarrow{e_8} t$, and pay 13 which is lower than their current cost. Therefore, the strategy profile depicted here is not an NE. □

Definition 4.5. *A strategy profile is a* weak subgame-perfect equilibrium *(wSPE) if it is a weak NE from every finite history of the game.*

In Lemma 4.4, we saw that a weak NE is not necessarily an NE. But, we will see that this is not true for SPE, i.e, a weak SPE is actually an SPE. Intuitively, this is because if indeed there is a single player deviation from some step (other than the first) of a weak SPE outcome, then from that step's point of view the deviation is a first-shot deviation. As weak SPE requires a profile to be a weak NE from every step, a weak SPE would not have this single player deviation from any step, hence it will actually be an SPE. Formally,

Theorem 4.6. *In a dynamic NCG, a strategy profile is an SPE if, and only if, it is a weak SPE.*

Proof. Recall from Chapter 1, given a strategy profile σ and a history h, we write $\mathsf{outcome}(\sigma, h)$ for the outcome of the residual strategy σ^h from the last configuration of h. Similarly, we let $\mathsf{cost}_k(\sigma, h) = \mathsf{cost}_k(\mathsf{outcome}(\sigma, h))$.

It is easy to see that NEs are weak NEs, and subsequently SPEs are weak SPEs. This is because if a strategy profile is immune to single player deviation from any step of its outcome, it surely is immune to single player deviation from the first step.

For the opposite direction, let us consider a weak SPE $\sigma = (\sigma)_{i \in [\![n]\!]}$ of a dynamic NCG $\mathcal{G}$. Suppose that σ is not an SPE; then there exists a history h of $\mathcal{G}$ such that the residual strategy σ^h is not an NE from the last configuration of h.

This implies there is i, and a strategy σ'_i of Player i with $\mathsf{cost}_i(\sigma, h) > \mathsf{cost}_i(\langle \sigma_{-i}, \sigma'_i \rangle, h)$. In particular, $\mathsf{cost}_i(\langle \sigma_{-i}, \sigma'_i \rangle, h)$ is finite, and along the outcome of this strategy profile, Player i reaches $\mathsf{tgt}(i)$.

We pick σ_i' as a profitable deviating strategy (after history h) with minimum number of deviations from σ_i. Because Player i reaches $\mathsf{tgt}(i)$ along $\mathsf{outcome}(\langle \sigma_{-i}, \sigma_i' \rangle, h)$, there exist profitable strategies with finitely many deviations.

We write h' for the longest finite history along the path $\mathsf{outcome}(\langle \sigma_{-i}, \sigma_i' \rangle, h)$ where σ_i' deviates from σ_i, i.e, $\sigma_i(hh') \neq \sigma_i'(hh') = c$ and $\mathsf{outcome}(\sigma, hh'c) = \mathsf{outcome}(\langle \sigma_{-i}, \sigma_i' \rangle, hh'c)$.

Now if $\mathsf{cost}_i(\sigma, hh') \leq \mathsf{cost}_i(\langle \sigma_{-i}, \sigma_i' \rangle, hh')$, then deviating at history hh' is unnecessary for Player i for having profitable deviating strategy from h, and that contradicts the minimality of the number of deviations of σ_i' from σ_i. Hence we get $\mathsf{cost}_i(\sigma, hh') > \mathsf{cost}_i(\langle \sigma_{-i}, \sigma_i' \rangle, hh')$. But that implies we have a first-shot profitable deviating strategy for Player i, σ_i', from history hh', which contradicts the assumption that σ is a weak SPE. □

4.2.1 Characterization of the Outcomes of All SPE

In this section, we first informally explain how we characterize the outcomes of an SPE, then we introduce formally the tools that we need for the characterization. Finally, in Theorem 4.10, we state the characterization using those tools.

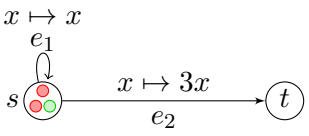
Therefore, before diving into technical jargon, let us first describe intuitively how we proceed:

- Broadly speaking, we associate a value through a function λ (and thus referring to them as λ-values) to each player and each edge of the configuration graph, such that the λ-values effectively capture the criteria for starting an SPE outcome from that edge. To be more specific, similar to $val_{i,c}$ of the NE outcome characterization, though technically more involved, these λ-values give an upper bound to the cost of each player in the outcome of any SPE that starts from that edge.

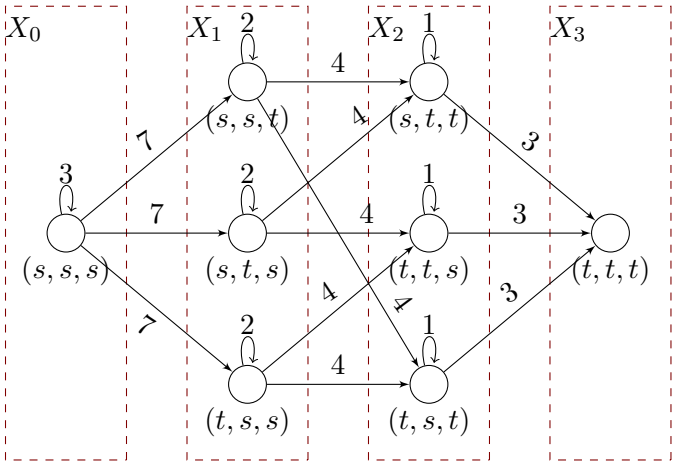
- Obviously, the next question is: how do we define these λ-values associated to a player and an edge of the configuration graph? The one line answer is we start with some initial λ-values, iteratively update them until they reach a fix-point. But, the main technical challenge lies in how we update these λ-values iteratively. We do not update λ-values of all edges together in the configuration graph, rather we only update in some *part* of the configuration graph. When the λ-values reach their fix-point there, then only we proceed to some *next* part of the graph where we use the values of a part which have been updated earlier. In the following, we explain what these *parts* are in the configuration graph, then we discuss how we update them inside an individual part, and thereafter we finally delve into more technical details.

  One can imagine that the game is played in *phases* inside the configuration graph, where each phase corresponds to the number of players that have already reached their target vertices. A play *transitions* from one phase to the next when a player reaches their target vertex. Intuitively, a play is split into these phases because once a player reaches their target, the game stops there for that particular player, they do not contribute in congestion for any other players any further, hence they also do not *affect* λ-values for the subsequent edges.

  There can be multiple components in a single phase, each of which corresponds to a different set of players (but same number of players) who are at their target vertex. Note that, there is no edge between two such components of a single phase because from a configuration where, say only Player 1 is at their target, there cannot be an edge to a configuration where



(a) An NCG $\mathcal{G} = \langle \mathcal{A}, 3, \mathsf{g} \rangle$, where $\mathsf{g}(s,t) = 3$



(b) Configuration graph in *stages*

Figure 4.3: Illustration of an NCG with its configuration graph in phases

only Player 2 is at their target, and so on. There will be at most $n+1$ such phases in the configuration graph, where the first phase represents no player is at their target, and the last phase represents all players are at their target vertices. These phases are in fact, referred to as the parts of the configuration graph on which one update on λ-values happens at a time.

We start with some initial λ-values for all edges of every phases in the configuration graph, then we start updating those values from the final phase, and moving backwards until the first phase. Moreover, in each phase of the graph, we update the λ-values iteratively till they reach a fix-point inside that phase. These updates in a phase depend on the λ-values of the later phases. Note that, because there is no edge between different components of a single phase of the configuration graph, the λ-values of an edge in one component do not affect λ-values of another edge in another component of that same phase.

Tools that are needed: Definitions and properties Let us consider a dynamic NCG $\langle \mathcal{A}, n, \mathsf{g} \rangle$ which we fix for the following discussion. We denote the associated multi-weighted configuration graph by $\mathcal{M} = \langle C, T \rangle$, where recall from Section 1.3, $C \subseteq V^{[\![n]\!]}$ is the set of configurations and $T \subseteq C \times \mathbb{N}^{[\![n]\!]} \times C$ is the set of transitions in the configuration graph.

We partition the set C of configurations into $(X_j)_{0 \leq j \leq n}$ such that a configuration c is in X_j if, and only if, $j = \#\{i \in [\![n]\!] \mid c(i) = \mathsf{tgt}(i)\}$. Since $\mathsf{tgt}(i)$ is a sink state in $\mathcal{A}$, if there is a transition from some configuration in X_j to some configuration in X_k, then $k \geq j$. We define $X_{\geq j} = \bigcup_{i \geq j} X_i$, and $Z_j = \{(c, w, c') \in T \mid c \in X_j\}$ and $Z_{\geq j} = \{(c, w, c') \in T \mid c \in X_{\geq j}\}$.

In Figure 4.3, we depict such a partition of the configuration graph C of the NCG $\mathcal{G}$ considered.

For all $0 \leq j \leq n$, we inductively define a sequence of families of labeling functions $\langle \lambda^{j^*} \rangle_{0 \leq j \leq n}$, where each $\lambda^{j^*} = (\lambda_i^{j^*})_{i \in [\![n]\!]}$ is a family of labeling function, and for all $i \in [\![n]\!]$, $\lambda_i^{j^*} : Z_{\geq j} \to \mathbb{N} \cup \{-\infty, +\infty\}$ is a labeling function associated to Player i, which maps an edge from $Z_{\geq j}$ to its so-called λ-value.

Definition 4.7. *For any family $\lambda = \langle \lambda_i \rangle_{i \in [\![n]\!]}$ with $\lambda_i : Z_{\geq j} \to \mathbb{N} \cup \{-\infty, +\infty\}$ and any $c \in X_{\geq j}$, a path $\rho = (t_k)_{k \geq 1}$ from c visiting c_{tgt} is said to be* λ-consistent *if for any $i \in [\![n]\!]$ and for any $1 \leq k \leq |\rho|$, it holds that $\mathsf{cost}_i(\rho_{\geq k}) \leq \lambda_i(t_k)$. We write $\Gamma_\lambda(c)$ to denote the set of all λ-consistent paths from c.*

We now define the sequence $\langle \lambda^{j^*} \rangle_{0 \leq j \leq n}$. For $j = n$, we have $X_{\geq n} = \{c_{\mathsf{tgt}}\}$ and $Z_{\geq n} = \{(c_{\mathsf{tgt}}, 0^n, c_{\mathsf{tgt}})\}$; there is a single path, which obviously is the outcome of an SPE since no deviations are possible. For all $i \in [\![n]\!]$, we let $\lambda_i^{n^*}(c_{\mathsf{tgt}}, 0^n, c_{\mathsf{tgt}}) = 0$.

Now, fix $j < n$, assuming that $\lambda^{(j+1)^*}$ has been defined, we go on to define λ^{j^*}. In order to define λ^{j^*}, we introduced an intermediary sequence of families of labeling functions $\mu = \langle \mu^k \rangle_{k \geq 0, i \in [\![n]\!]}$, where each family $\mu^k = (\mu_i^k)_{i \in [\![n]\!]}$ comprises of $\mu_i^k : Z_{\geq j} \to \mathbb{N} \cup \{-\infty, +\infty\}$, of which $(\lambda_i^{j^*})_{i \in [\![n]\!]}$ will be the limit. We call these μ^k's as the λ^{j^*}*-building functions*. For each of these λ^{j^*}-building functions μ^k, from any configuration c of $Z_{\geq j}$, we have the set of μ^k-consistent paths denoted by $\Gamma_{\mu^k}(c)$

Definition 4.8. *For any $0 \leq j \leq n$, we define λ^{j^*}-building functions $\mu^k : Z_{\geq j} \to \mathbb{N} \cup \{-\infty, +\infty\}$ as follows: they coincide with $\lambda^{(j+1)^*}$ on $Z_{\geq j+1}$, i.e, for any $\mathbf{e} \in Z_{\geq j+1}$, we let*

$\mu_i^k(\mathbf{e}) = \lambda_i^{(j+1)^*}(\mathbf{e})$*; and for* $\mathbf{e} \in Z_j = Z_{\geq j} \setminus Z_{\geq j+1}$ *they are defined inductively (induction on* k*):*

- *for the base case:*

$$\mu_i^0(\mathbf{e}) = \begin{cases} 0 & \text{if } c(i) = \mathsf{tgt}(i), \\ +\infty & \text{otherwise} \end{cases}$$

- *for* $k > 0$*, assuming* μ_i^{k-1} *has been defined,*

$$\mu_i^k(\mathbf{e}) = \begin{cases} 0 & \text{if } c(i) = \mathsf{tgt}(i) \\ \min\limits_{c'' \in \mathsf{dev}_i(c,c')} \sup\limits_{\rho \in \Gamma_{\mu^{k-1}}(c'')} (\mathsf{cost}_i(c,c'') + \mathsf{cost}_i(\rho)) & \\ \qquad \text{if for all } (c,\tilde{w},\tilde{c}) \in Z_{\geq j}.\, \Gamma_{\mu^{k-1}}(\tilde{c}) \neq \emptyset & \\ -\infty & \text{otherwise} \end{cases} \tag{4.1}$$

where $\mathsf{dev}_i(c,c') = \{c'' \in C \mid c \Rightarrow c'' \text{ and } \forall j \in [\![n]\!] \setminus \{i\}.\, c''(j) = c'(j)\}$ *is the set of configurations which can be obtained from Player* i*'s unilateral deviation from the transition* c *to* c'*, and* c' *itself.*

Lemma 4.9. *For any* $\mathbf{e} \in Z_{\geq j}$*, any* $i \in [\![n]\!]$*, and any* $k > 0$*, we have* $\mu_i^k(\mathbf{e}) \leq \mu_i^{k-1}(\mathbf{e})$.

Proof. We prove the statement by induction on k, starting with $k = 1$. Write $\mathbf{e} = (c, w, c')$. If $c' = \mathsf{tgt}(i)$, then by definition $\mu_i^k(\mathbf{e}) = \mu_i^{k-1}(\mathbf{e}) = 0$; otherwise, $\mu_i^0(\mathbf{e}) = \infty$; in both cases, the result holds.

Now, let us assume that for some $k > 0$, the following holds: for all $i \in [\![n]\!]$ and all $\mathbf{e} \in Z_{\geq j}$, it holds $\mu_i^k(\mathbf{e}) \leq \mu_i^{k-1}(\mathbf{e})$. Then for any $c \in X_{\geq j}$, it implies $\Gamma_{\mu^k}(c,w,c') \subseteq \Gamma_{\mu^{k-1}}(c,w,c')$.

Fix an arbitrary $\mathbf{e} = (c,w,c') \in T$. For any $c'' \in \mathsf{dev}_i(c,c')$, c'' also belongs to $X_{\geq j}$, hence, by induction hypothesis, we have $\Gamma_{\mu^k}(c'') \subseteq \Gamma_{\mu^{k-1}}(c'')$.

As supremum over a larger set would be at least as much as it is for a smaller set, we have:

$$\sup_{\rho \in \Gamma_{\mu^k}(c'')} (\mathsf{cost}_i(c,c'') + \mathsf{cost}_i(\rho)) \leq \sup_{\rho \in \Gamma_{\mu^{k-1}}(c'')} (\mathsf{cost}_i(c,c'') + \mathsf{cost}_i(\rho)).$$

As the above is true for any $c'' \in \mathsf{dev}_i(c,c')$, it remains true for the minimum, hence:

$$\min_{c'' \in \mathsf{dev}_i(c,c')} \sup_{\rho \in \Gamma_{\mu^k}(c'')} (\mathsf{cost}_i(c,c'') + \mathsf{cost}_i(\rho)) \leq \min_{c'' \in \mathsf{dev}_i(c,c')} \sup_{\rho \in \Gamma_{\mu^{k-1}}(c'')} (\mathsf{cost}_i(c,c'') + \mathsf{cost}_i(\rho))$$

which implies $\mu^{k+1}(\mathbf{e}) \leq \mu^k(\mathbf{e})$. □

As the costs are non-negative integers, the μ^k values are bounded by 0 if it is not $-\infty$, so they cannot decrease arbitrarily. Therefore, μ^k indeed reaches a fix-point, which is by definition λ^{j^*}. Following this for each j from n to 0, we finally have λ^{0^*}, and we write Γ^* for $\Gamma_{\lambda^{0^*}}$.

CHARACTERIZATION As promised, we now characterize the outcomes of any SPE using the λ-values as follows:

Theorem 4.10. *A path ρ in the configuration graph of $\mathcal{G} = \langle \mathcal{A}, n, \mathbf{g} \rangle$ is the outcome of an SPE if, and only if, $\rho \in \Gamma^*(c_{\mathsf{src}})$.*

Proof. Let us first consider an SPE σ of $\mathcal{G}$, of which ρ is the outcome. We shall show that for any c and any history h ending in configuration c, $\mathsf{outcome}(\sigma, h) \in \Gamma^*(c)$. In fact, we prove by induction on j that $\mathsf{outcome}(\sigma, h) \in \Gamma_{\lambda^{j^*}}(c)$ if $c \in X_j$. Then, for $h = c_{\mathsf{src}}$, we get the expected result.

For $j = n$, for any history h ending in X_n, $\mathsf{outcome}(\sigma, h)$ is a path looping on c_{tgt}. We defined $\lambda^{n^*}(\mathbf{e}) = 0$ for $\mathbf{e} \in X_n$, so that the result holds.

Assume the statement is true for some $j + 1 \leq n$: for any history h ending in some configuration $c \in X_{j+1}$, $\mathsf{outcome}(\sigma, h) \in \Gamma^{(j+1)^*}(c)$.

To prove the result at level j, we rely on the fact that λ^{j^*} is the limit of sequence μ^k as defined earlier, and we start another induction to show that for any $k \geq 0$, for any history h whose last configuration c is in X_j, the path $\mathsf{outcome}(\sigma, h)$ belongs to $\Gamma_{\mu^k}(c)$.

We begin with the case $k = 0$. Fix a history h ending in some $c \in X_j$. Write $\mathsf{outcome}(\sigma, h) = (c_r, w_r, c_{r+1})_{r \geq 1}$. Let r' be the least integer such that $c_{r'}$ in $\mathsf{outcome}(\sigma, h)$ belongs to $X_{>j}$. We need to show that for any $i \in [\![n]\!]$ and any r, $\mathsf{cost}_i((\mathsf{outcome}(\sigma, h))_{\geq r}) \leq \mu_i^0(c_r, w_r, c_{r+1})$. If $r < r'$, then either $\mu^0(c_r, w_r, c_{r+1}) = +\infty$, or $\mu^0(c_r, w_r, c_{r+1}) = 0$ and $c_r(i) = \mathsf{tgt}(i)$; in both cases, the result also holds. If $r \geq r'$, we have $\mu_i^0(c_r, w_r, c_{r+1}) = \lambda_i^{(j+1)^*}(c_r, w_r, c_{r+1})$, and by our outer induction hypothesis, $\mathsf{cost}_i(\mathsf{outcome}(\sigma, h)_{\geq r}) \leq \lambda^{(j+1)^*}((c_r, w_r, c_{r+1}))$.

We now prove the induction step. We assume that $\mathsf{outcome}(\sigma, h) \in \Gamma_{\mu^{k-1}}(c)$ for any history h ending in $c \in X_j$. Write $\mathsf{outcome}(\sigma, h) = (c_r, w_r, c_{r+1})_{r \geq 1}$. We first observe that $\mu_i^k(c_r, w_r, c_{r+1}) \neq -\infty$ for any $r \geq 1$. This is because for any $(c_r, \tilde{w}, \tilde{c}) \in T$ and for any $c_{\mathsf{src}} \overset{\tilde{h}}{\Rightarrow} \tilde{c}$, by our current induction hypothesis $\Gamma_{\mu^{k-1}}(\tilde{c})$ contains $\mathsf{outcome}(\sigma, \tilde{h})$, hence it is not empty.

Therefore, the only thing left to show now is the following:

$$\text{for all } i \in [\![n]\!], \text{ For all } r \geq 1, \mathsf{cost}_i(\mathsf{outcome}(\sigma, h)_{\geq r}) \leq \mu_i^k(c_r, w_r, c_{r+1})$$

This is obvious if $c_r(i) = \mathsf{tgt}(i)$. As previously, we let r' be the least integer such that the configuration $c_{r'}$ along $\mathsf{outcome}(\sigma, h)$ belongs to $X_{>j}$.

For $r < r'$: assume $\mathsf{cost}_i(\mathsf{outcome}(\sigma, h)_{\geq r}) > \mu_i^k(c_r, w_r, c_{r+1})$; then by definition of μ_i^k,

$$\mathsf{cost}_i(\mathsf{outcome}(\sigma, h)_{\geq r}) > \min_{c' \in \mathsf{dev}_i(c_r, c_{r+1})} \sup_{\rho \in \Gamma_{\mu^{k-1}}(c')} \{\mathsf{cost}_i(c_r, c') + \mathsf{cost}_i(\rho)\},$$

which implies the existence of an edge $(c_r(i), \mathsf{f}, v')$ in E such that

$$\mathsf{cost}_i(\mathsf{outcome}(\sigma, h)_{\geq r}) > \mathsf{cost}_i(c_r, c_{r+1}[i \to v']) + \mathsf{cost}_i(\sigma, h \cdot (c_r, w', c_{r+1}[i \to v'])).$$

Because of our induction hypothesis for $k-1$, we have,

$$\mathsf{outcome}(\sigma, h \cdot (c_r, w', c_{r+1}[i \to v'])) \in \Gamma_{\mu^{k-1}}(c_r, w', c_{r+1}[i \to v'])$$

This shows a profitable first-shot deviation for Player i, contradicting our hypothesis that σ is an SPE.

Otherwise for $r \geq r'$, the condition follows directly from the outer induction hypothesis, because μ^k coincides with λ^{j^*} for edges in those regions.

We now prove the converse implication.

Picking $\rho \in \Gamma^*(c_{\mathsf{src}})$, we build a weak SPE $\sigma = (\sigma_i)_{i \in [\![n]\!]}$ (which is an SPE by Theorem 4.6), step-by-step; for notational convenience, instead of defining $\sigma_i(h)$ for any history h, we directly define what $\mathsf{outcome}(\sigma, h)$ would be.

As a first step, we let $\mathsf{outcome}(\sigma, c_{\mathsf{src}}) = \rho$. Now let us construct the rest of σ. Consider an arbitrary history $h' = h \cdot (c, w, c')$, and suppose $\mathsf{outcome}(\sigma, h)$ is already defined to be $(c_j, w_j, c_{j+1})_{j \geq 1}$. Note that, here $c_1 = c$ because c is the last configuration of h by consideration. Now, if $c_2 = c'$, then we do not need to define $\mathsf{outcome}(\sigma, h')$, as it is already defined from h itself, and we have $\mathsf{outcome}(\sigma, h') = (c_j, w_j, c_{j+1})_{j \geq 2}$. Otherwise $c' \neq c_2$, and we define $\mathsf{outcome}(\sigma, h')$ in that case below.

If c' is a configuration which is obtained by two or more players' deviation from $\mathsf{outcome}(\sigma, h)$, i.e, if there exists $k \neq l \in [\![n]\!]$ such that $c'(k) \neq c_2(l)$, then pick any $\tilde{\rho} \in \Gamma^*(c')$ (which we know is not empty) and define $\mathsf{outcome}(\sigma, h \cdot (c, w, c')) = \tilde{\rho}$.

Otherwise c_2 differs from c' only at a single player's position, say Player i's. In that case we define

$$\mathsf{outcome}(\sigma, h \cdot (c, w, c')) = \underset{\rho \in \Gamma^*(c')}{\arg\max} \{\mathsf{cost}_i(c_1, c') + \mathsf{cost}_i(\rho)\}. \tag{4.2}$$

Note that, here we can take $\arg\max$ because $\lambda_i^*(\mathbf{e})$ is finite (which we shall establish in Corollary 4.13) for any $\mathbf{e} \in \mathcal{E}$, so there are finitely many plays in $\Gamma^*(c)$. This ends our definition of the strategy profile σ, which we now prove is a weak SPE.

Consider an arbitrary history h and denote $\mathsf{outcome}(\sigma, h)$ by $(c_j, w_j, c_{j+1})_{1 \leq j}$. Pick a configuration c' such that $(c_1, w', c') \in T$, $c'(j) = c_2(j)$ for all $j \neq i$, but $c'(i) \neq c_2(i)$. That is, c' is a configuration obtained by Player i's deviation from σ_i at the end of h.

We show that such a single player deviation is not profitable if, after that deviation, all players continue to play following σ:

$$\begin{aligned}
\mathsf{cost}_i(c_1, c') + \mathsf{cost}_i(\sigma, h \cdot (c_1, w', c')) &= \sup_{\rho \in \Gamma^*(c')} \{\mathsf{cost}_i(\rho) + \mathsf{cost}_i(c_1, c')\} \qquad \text{(by (4.2))} \\
&\geq \min_{\tilde{c} \in \mathsf{dev}_i(c_1, c_2)} \sup_{\rho \in \Gamma^*(\tilde{c})} \{cost_i(\rho) + cost_i(c, \tilde{c})\} \\
&= \lambda_i^*((c_1, w_1, c_2)) \\
&\geq \mathsf{cost}_i(\sigma, h)
\end{aligned}$$

So no such configuration c' would be profitable for Player i. Hence from any history h, σ is a weak NE. Therefore, we can conclude that σ is an SPE. □

SOME MORE PROPERTIES Having the characterization of outcomes of SPE by the notion of λ-consistency, we now are going to establish the upper bounds of the λ-values for an NCG $\mathcal{G}$. Remember, by definition a λ-value could be $+\infty$, but the finite values that any λ-function maps into is, in fact, bounded. Here, we show the upper bound on those finite values of λ.

A λ-map is nothing but a fix-point of so-called λ-building functions, denoted here by μ^k's. In Lemma 4.11, we establish the finite upper bound of μ^k's after $|V|$ iterations (V is the set of vertices of the NCG under consideration). Corollary 4.12 tells us when a μ^k function reaches their fix-point in a phase $Z_{\geq j}$. Finally, combining these two together, we get Corollary 4.13 which gives us an upper bound for λ-values.

Lemma 4.11. *For any edge $\mathbf{e} = (c, w, c') \in Z_j$, the $(|V|+1)^{th}$ family $\mu^{|V|}$ of λ^{j^*}-building functions satisfies $\mu_i^{|V|}(\mathbf{e}) \leq |V| \times \kappa$, where $\kappa = max_{e \in E} \mathsf{f}_e(n)$.*

Proof. We prove a slightly stronger statement. Let m be the length of the shortest path from $c(i)$ to tgt in $\mathcal{A}$. We show (by induction) that $\mu_i^m(\mathbf{e}) \leq m \times \kappa$. As $m \leq |V|$ and by Lemma 4.9, this entails our current lemma.

If $m = 1$, then there is an edge e from $c[i]$ to tgt in $\mathcal{A}$. Let us consider an edge $\mathbf{e}' = (c, w', c'')$ such that $c''[j] = c'[j]$ for all $j \neq i$, and $c''[i] = \mathsf{tgt}(i)$. Now if $\Gamma_{\mu^{m-1}}(\tilde{c}) = \emptyset$ for any $(c, \tilde{w}, \tilde{c}) \in T$, then anyway $\mu_i^m(\mathbf{e}) = -\infty$, and the result holds. Otherwise, we have $\Gamma_{\mu^{m-1}}(c'') \neq \emptyset$, and for any $\rho \in \Gamma_{\mu^{m-1}}(c'')$, we have $\mathsf{cost}_i(\rho) = 0$. Therefore, $\mu_i^m(\mathbf{e}) = \mathsf{cost}_i(c, c'')$, which is bounded by κ.

Now assume that the induction hypothesis holds up to step $m-1$, and consider a configuration c such that the length of a shortest path from $c(i)$ to $\mathsf{tgt}(i)$ in $\mathcal{A}$ is m. Fix an edge $(c, w, c') \in T$. Write $(c(i), \mathsf{f}, v') \in E$ for the first edge of a shortest path from $c(i)$ to tgt. We consider a configuration c'' such that $c''(j) = c'(j)$ for all $j \neq i$, and $c''(i) = v'$. By construction, there is a path from $c''(i)$ to $\mathsf{tgt}(i)$ of length $\leq m-1$. By induction hypothesis, for any edge $(c'', \tilde{w}, \tilde{c}) \in T$, we have $\mu_i^{m-1}((c'', \tilde{w}, \tilde{c})) \leq (m-1) \times \kappa$. This implies that for any path $\rho = (t_j)_{j \geq 1} \in \Gamma_{\mu^{m-1}}(c'')$, we have $\mathsf{cost}_i(\rho) \leq \mu_i^{m-1}(t_1) \leq (m-1) \times \kappa$. Therefore, $\mu_i^m(c, w, c') \leq \mathsf{cost}_i(c, c'') + (m-1) \times \kappa \leq m \times \kappa$. □

Lemmas 4.9 and 4.11 provide a bound on the number of steps until any sequence of λ^{j^*}-building functions stabilize:

Corollary 4.12. *Any sequence $(\mu^k)_{k \geq 0}$ of λ^{j^*}-building functions stabilizes after at most $|V|(1 + n \cdot \kappa \cdot |E|^n)$ steps.*

From Lemma 4.11, we also get that the sequence $(\mu^k)_{k \geq 0}$ built for computing λ^{j^*} cannot stabilize unless for all $i \in [\![n]\!]$, for all $\mathbf{e} \in Z_{\geq j}$, we have $\mu_i^k(\mathbf{e}) \leq |V| \times \kappa$. By Lemma 4.9:

Corollary 4.13. *For any $0 \leq i, j \leq n$, and any $\mathbf{e} \in Z_{\geq j}$, we have $\lambda_i^{j^*}(\mathbf{e}) \leq |V| \times \kappa$.*

4.2.2 Using the Characterization for CONSTRAINED-SPE

With the characterization of the outcomes of all SPE (Theorem 4.10), what remains to solve in CONSTRAINED-SPE is to check whether there is a path in $\Gamma^*(c_{\mathsf{src}})$ with cost less than the given threshold. Now, when we non-deterministically guess a play and verify whether the play is the outcome of an SPE, at each step we check whether the play is fulfilling the condition for being λ^*-consistent up to the current step. Moreover, at each step, while we guess a new configuration from the current one, we also make sure that there is a λ-consistent play for a suitable λ from all the configurations that are reachable (but not currently guessed) in one step. In order to perform these *λ-consistency checks* at each step, we make use of an infinite graph structure, namely the *counter graph*.

A counter graph is defined according to a family of labeling functions $\lambda = \langle \lambda_i \rangle_{i \in [\![n]\!]}$ and a configuration c such that any path from c to c_{tgt} in the counter graph is actually λ-consistent. In this section, after formally defining what a counter graph is, we establish this relation between a path in the graph and a λ-consistent play in Lemma 4.15. In Lemma 4.16, we establish the equivalence between the fact that supremum of Player i's cost over μ^k-consistent plays is $+\infty$ with the existence of a cycle in the corresponding counter graph.

Moreover, in Lemma 4.17, we show an upper bound for the finite value that the intermediary μ^k-functions can take. Recall, we establish an upper bound of the finite value that a λ-function can take in Corollary 4.13, and also an upper bound for the finite values that a μ^k can map to, when $k \geq |V|$. Because the μ^k's are non-increasing (Lemma 4.9), the intermediary μ^k-values could be higher than that of when $k \geq |V|$. Therefore, establishing this upper bound gives us the necessary computations for the space we need for the CONSTRAINED-SPE algorithm, which we describe in the final part of this section.

Counter Graph. We formally define a counter graph as follows:

Definition 4.14. *With any family of labeling functions $\mu = \langle \mu_i \rangle_{i \in [\![n]\!]}$ and configuration c, we associate an infinite-state* counter graph $\mathbb{C}[\mu, c] = \langle C', T' \rangle$ *to capture μ-consistent paths from c:*

- *the set of vertices is $C' = C \times (\mathbb{N} \cup \{+\infty, -\infty\})^{[\![n]\!]}$*
- *T' contains all edges $((d, b), w, (d', b'))$ such that (d, w, d') is an edge of $\mathcal{M}$ and for all $i \in [\![n]\!]$, either $b'(i) = 0$ if $d(i) = \mathtt{tgt}(i)$, or $b'_i = \min\{b_i - w_i, \mu_i(d, w, d') - w_i\}$ otherwise (provided that $b'_i \geq 0$ for all i, in order for (d', b') to be an edge of $\mathbb{C}[\mu, c]$).*

Intuitively, in configuration (d, b) of the counter graph, b is used to enforce μ-consistency: each edge taken along a path imposes a constraint on the cost of the players for the rest of the path; this constraint is added to the constraints of the earlier edges, and propagated along the path.

With the initial configuration c, we associate b^c such that $b^c_i = 0$ if $c(i) = \mathtt{tgt}(i)$ and $b^c_i = +\infty$ otherwise: this configuration imposes no constraints since no edges have been taken yet. We call (c, b^c) to be the *initial configuration* of the counter graph $\mathbb{C}[\mu, c]$.

Notice that $\mathbb{C}[\mu, c]$ is infinite, but as we show below, only finitely many states are reachable from the initial configuration. We write $|C'|_{\mathsf{reach}}$ for the number of reachable states in C'.

We extend the region decomposition of $\mathcal{M} = \langle C, T \rangle$ to any counter graph $\mathbb{C}[\mu, c] = \langle C', T' \rangle$ in the natural way: $(c', b') \in X'_j$ if $c' \in X_j$, and an edge $((c', b'), w', (c'', b'')) \in Z'_{\geq j}$ if $(c', w', c'') \in Z_{\geq j}$.

We call a path π from (c, b^c) to (c_{tgt}, b) (for some b) a *valid* path in $\mathbb{C}[\mu, c]$. For any path $\pi \in \mathbb{C}[\mu, c]$, where $c \in X_j$, we write $\pi = \pi[j] \cdot \pi[j+1] \cdots \pi[n]$ where $\pi[l]$ denotes the (possibly empty) section of path in Z'_l. We also use the notation $\mathrm{max}b_l(\mu, c)$ to denote the maximum finite counter value that appears in the vertices reachable from (c, b^c) and belonging to X'_j. More precisely,

$$\mathrm{max}b_l(\mu, c) = \max\{m \in \mathbb{N} \mid \exists (c', b') \in X_l \text{ s.t. } (c, b^c) \to^* (c', b') \text{ and } \exists i \in [\![n]\!],\, b'(i) = m\}$$

We extend this notation to $\mathrm{max}b_{\geq l}(\mu, c)$, which denotes $\max\limits_{j \geq l} \mathrm{max}b_j(\mu, c)$. When all the counter values are in $\{0, \infty\}$, we let $\mathrm{max}b_l(\mu, c) = 0$.

Lemma 4.15. *There exists a path $\pi = ((c_j, b_j), w_j(c_{j+1}, b_{j+1}))_j$ from (c, b^c) to a vertex (c_{tgt}, b) (for some b) in the counter graph $\mathbb{C}[\lambda, c] = \langle C', T' \rangle$ of $\mathcal{G}$ if, and only if, there is a λ-consistent path $\rho = (c_j, w_j, c_{j+1})_j$ from c to c_{tgt} in $\mathcal{G}$.*

Proof. Assume that such a path π exists. Along π, for each player i, let $k(i)$ be the least index such that $c_{k(i)}(i) = \mathsf{tgt}(i)$. Then it is enough to show that for $1 \leq k < k(i)$, we have $\mathsf{cost}_i(\rho_{\geq k}) \leq \lambda_i(c_k, w_k, c_{k+1})$. And indeed we have

$$\begin{aligned} 0 \leq b_{k(i)}(i) &\leq b_{k(i)-1}(i) - \mathsf{cost}_i(c_{k(i)-1}, c_{k(i)}) \\ &\leq b_{k(i)-2}(i) - \mathsf{cost}_i(c_{k(i)-2}, c_{k(i)-1}) - \mathsf{cost}_i(c_{k(i)-1}, c_{k(i)}) \\ &\vdots \\ &\leq b_{k+1}(i) - \sum_{j=k+1}^{k(i)-1} \mathsf{cost}_i(c_j, c_{j+1}) \end{aligned}$$

so that $\mathsf{cost}_i(\rho_{\geq k}) \leq b_{k+1}(i) \leq \lambda_i(c_k, w_k, c_{k+1})$.

Conversely, if there is a λ-consistent path $\rho = (c_j, w_j, c_{j+1})_{1 \leq j < |\rho|}$ from c, we define $\pi = ((c_j, b_j), w_j, (c_{j+1}, b_{j+1}))_{1 \leq j < |\rho|}$ with:

$$b_1(i) = \begin{cases} 0 & \text{if } c_1(i) = \mathsf{tgt}(i) \\ +\infty & \text{otherwise} \end{cases}$$

And for $1 \leq j \leq |\rho|$,

$$b_j(i) = \min\{b_{j-1}(i) - \mathsf{cost}_i(c_{j-1}, c_j), \lambda_i(c_{j-1}, w_{j-1}, c_j) - \mathsf{cost}_i(c_{j-1}, c_j)\}$$

For π to be a valid path in $\mathbb{C}[\lambda, c]$, we have to prove that $b_j(i) \geq 0$ for all $1 < j \leq |\rho|$ and all $i \in [\![n]\!]$. For $j = 1$, it is evident.

For $j > 1$, the second element of the minimum defining $b_j(i)$ is non-negative, because $\mathsf{cost}_i(c_{j-1}, c_j) \leq \mathsf{cost}_i(\rho_{\geq j-1}) \leq \lambda_i(c_{j-1}, w_{j-1}, c_j)$. Suppose the first element $b_{j-1}(i) - \mathsf{cost}_i(c_{j-1}, c_j)$ is negative. Using definition of $b_{j-1}(i)$, we can say the above inequality can be true only if either $b_{j-2}(i) - \sum_{k=j-2}^{j-1} \mathsf{cost}_i(c_k, c_{k+1}) < 0$, or $\lambda_i(c_{j-2}, w_{j-2}, c_{j-1}) - \sum_{k=j-2}^{j-1} \mathsf{cost}_i(c_k, c_{k+1}) < 0$. But again the latter cannot be true because it directly contradicts λ-consistency of ρ. Hence, our supposition can only be true if the former constraint holds. We repeat

the process with $b_{j-2}(i)$, coming down to the condition $b_2(i) - \sum_{k=2}^{j-1} \mathsf{cost}_i(c_k, w_k, c_{k+1}) < 0$ to make our supposition $b_{j-1}(i) - \mathsf{cost}_i(c_{j-1}, c_j) < 0$ true. But $b_2(i) = \lambda_i(c_1, w_1, c_2) - \mathsf{cost}_i(c_1, c_2)$, so that the inequality above entails $\lambda_i(c_1, w_1, c_2) < \sum_{k=1}^{j-1} \mathsf{cost}_i(c_k, c_{k+1})$. This contradicts the λ-consistency condition at the beginning of ρ. It follows that $b_j(i) \geq 0$ for all $1 \leq j \leq |\rho|$ and all $i \in [\![n]\!]$. □

Lemma 4.16. $\sup_{\rho \in \Gamma_{\mu^k}(c)} \mathsf{cost}_i(\rho) = +\infty$ *if, and only if, there exists a valid path* π *in* $\mathbb{C}[\mu^k, c]$ *with the following conditions:*

- π *is of the form* $h \cdot \beta \cdot h'$*, where* β *is a cycle in* $\mathbb{C}[\mu^k, c]$*;*
- *Player* i*'s (constant) counter value* $b(i)$ *is positive throughout* β*.*

Proof. $\sup_{\rho \in \Gamma_{\mu^k}(c)} \mathsf{cost}_i(\rho) = +\infty$ implies there exists a sequence of paths $(\rho_m)_{m \geq 1}$ in $\Gamma_{\mu^k}(c)$ such that $\mathsf{cost}_i(\rho_m) \to \infty$ as m grows. By Lemma 4.15, for each of those ρ_m, there exists corresponding π_m from (c, b^c) to (c_{tgt}, b) in $\mathbb{C}[\mu^k, c]$. As a player's cost in a single edge is bounded, the length of π_m has to grow unboundedly. First of all, that is only possible if there is a cycle β in $\mathbb{C}[\mu^k, c]$ - we have the first condition now. Now Player i's counter value cannot be 0 in β because then the cycle doesn't contribute to make $\mathsf{cost}_i(\rho) \to +\infty$. Thus the second condition also holds.

Conversely, for $\pi = h \cdot \beta \cdot h'$ in $\mathbb{C}[\mu^k, c]$, where β is a cycle and Player i's cost > 0 in β, we construct a sequence of paths $\pi^m = h \cdot \beta^{m+1} \cdot h'$ for $m \geq 0$. By Lemma 4.15, corresponding to this sequence of paths, there exists a sequence of paths $(\rho_m)_{m \geq 0}$ in $\Gamma_{\mu^{k-1}}(c'')$, and for those paths, $\mathsf{cost}_i(\rho_m) \to \infty$ as m grows. Hence, $\mu_i^k(c, w, c') = +\infty$ if these condition holds. □

At this point, we have bounded the finite values that the λ^{j^*}'s can take. But in the transitioning from $\lambda^{(j+1)^*}$ to λ^{j^*}, μ_i^k can take larger values when $k < |V|$. In the sequel, we bound the values that any family $\mu^k = \langle \mu_i^k \rangle_{i \in [\![n]\!]}$ can return.

To this aim, we begin with working on the supremum of the cost for Player i of the paths in $\Gamma_{\mu^k}(c)$.

When we consider a $\mathrm{max}b_l(\mu^k, c)$ with $c \in X_j$ of $\mathcal{M}$, it is implicit that $l \geq j$. For $l > j$, we have

$$\mathrm{max}b_l(\mu^k, c) \leq \mathrm{max}b_l(\mu^0, c) \leq \max_{\substack{\mathbf{e} \in Z_l \\ i \in [\![n]\!]}} \lambda_i^{l^*}(\mathbf{e}) \leq |V| \times \kappa$$

because $\mu_i^k(\mathbf{e}) = \mu_i^0(\mathbf{e}) = \lambda_i^{l^*}(\mathbf{e})$ for those $\mathbf{e} \in Z_{\geq l}$ and for all $i \in [\![n]\!]$. So it remains to bound $\mathrm{max}b_j(\mu^k, c)$; but for that too, when $k \geq |V|$, we have $\mathrm{max}b_j(\mu^k, c) \leq |V| \times \kappa$. Therefore, we only need to provide a bound for $\mathrm{max}b_j(\mu^k, c)$ when $k < |V|$.

Lemma 4.17. *For an edge* $\mathbf{e} = (c, w, c') \in Z_j$ *and a* λ^{j^*}*-building function* $\mu^k = \langle \mu_i^k \rangle_{i \in [\![n]\!]}$*, if* $\mu_i^k(\mathbf{e})$ *is non-zero finite then*

$$\mu_i^k(\mathbf{e}) \leq (n|C| + 2|V|) \times \sum_{l=1}^{k} (n|C|)^{l-1} \cdot \kappa^l.$$

Moreover, the above bound also applies to $\max b_j(\mu^k, c'')$ *for any* $c'' \in X_j$*:*

$$\max b_j(\mu^k, c) \leq (n|C| + 2|V|) \times \sum_{l=1}^{k} (n|C|)^{l-1} \cdot \kappa^l.$$

Proof. We can claim that the finite maximum counter value appearing in X'_j of any counter graph $\mathbb{C}[\mu^k, c'']$ (where $c \in X_j$) is bounded by the finite maximum $\mu_i^k(\mathbf{e})$ value appearing in the same region (maximum over i and $\mathbf{e} \in Z_j$), i.e,

$$\max b_j(\mu^k, c'') \leq \max\{\mu_i^k(c, w, c) \in \mathbb{N} \mid ((c, b), w, (c', b')) \in Z'_j \text{ of } \mathbb{C}[\mu^k, c''],\ i \in [\![n]\!]\}.$$

This is justified because for the initial vertex (c, b^c) of X'_j, $b^c \in \{0, \infty\}$. Hence, a counter value becomes finite and non-zero, when some $\mu_i^k(\mathbf{e})$ for $\mathbf{e} \in Z_j$ becomes finite and non-zero. But by definition of counter graph, $((c, b), w, (c', b')) \in Z'_j$ implies $(c, w, c') \in Z_j$, hence we can obtain,

$$\max b_j(\mu^k, c'') \leq \max\{\mu_i^k(c, w, c') \in \mathbb{N} \mid (c, w, c') \in Z_j, i \in [\![n]\!]\}.$$

By induction on k, we prove that for any $c'' \in X_j$ and any $i \in [\![n]\!]$,

$$\max b_j(\mu^k, c'') \leq \max\{\mu_i^k(\mathbf{e}) \in \mathbb{N} \mid \mathbf{e} \in Z_j\} \leq (n|C| + 2|V|) \times \sum_{l=1}^{k} (n|C|)^{l-1} \cdot \kappa^l.$$

As $\mu_i^0(\mathbf{e}) \in \{0, \infty\}$ for all $\in Z_j$, the given bounds hold for $k = 0$.

We fix an arbitrary $\mathbf{e} = (c, w, c') \in Z_j$ and a player $i \in [\![n]\!]$ such that $\mu_i^k(\mathbf{e}) \in \mathbb{N} \setminus \{0\}$. That $\mu_i^k(\mathbf{e})$ is a non-zero finite value means that $\sup_{\rho \in \Gamma_{\mu^{k-1}}(\tilde{c})} \mathsf{cost}_i(\rho) \in \mathbb{N}$ for some $\tilde{c} \in \mathsf{dev}_i(c, c')$. If $\tilde{c}(i) = \mathsf{tgt}(i)$ then $\sup_{\rho \in \Gamma_{\mu^{k-1}}(\tilde{c})} \mathsf{cost}_i(\rho) = 0$, and that makes $\mu_i^k(\mathbf{e}) \leq \mathsf{cost}_i(c, \tilde{c}) \leq \kappa$, satisfying the given bound.

Otherwise, $\tilde{c}(i) \neq \mathsf{tgt}$, and depending whether $\tilde{c}$ belongs to X_j or $X_{>j}$, we analyze two cases:

- $\tilde{c} \in X_l$ for some $l > j$: Then

$$\begin{aligned} \mu_i^k(\mathbf{e}) &\leq \sup_{\rho \in \Gamma_{\mu^{k-1}}(\tilde{c})} \{\mathsf{cost}_i(c, \tilde{c}) + \mathsf{cost}_i(\rho)\} \\ &\leq \mathsf{cost}_i(c, \tilde{c}) + \max_{\substack{\mathbf{e}' \in Z_l \\ i \in [\![n]\!]}} \mu_i^{k-1}(\mathbf{e}') \leq (1 + |V|) \times \kappa. \end{aligned}$$

- $\tilde{c} \in X_j$: Now consider $\mathbb{C}[\mu^{k-1}, \tilde{c}]$, and its initial vertex $(\tilde{c}, b^{\tilde{c}})$. By Lemma 4.15, for any path $\rho \in \Gamma_{\mu^{k-1}}(\tilde{c})$, we have a corresponding valid path π_ρ in $\mathbb{C}[\mu^{k-1}, \tilde{c}]$. We also consider the region decomposition $\pi_\rho = \pi_\rho[j] \dots \pi_\rho[n]$. From the fact that $\sup_{\rho \in \Gamma_{\mu^{k-1}}(\tilde{c})} < +\infty$, we argue that in π_ρ, from $(\tilde{c}, b^{\tilde{c}})$ within each $|C|$ step either one counter value strictly decreases, or Player i reaches $\mathsf{tgt}(i)$. Otherwise, there would have been a cycle in $\mathbb{C}[\mu^{k-1}, \tilde{c}]$ resulting in the supremum being $+\infty$ (thanks to lemma 4.16). Recall by design, the counter

values of π_ρ in X'_j lie in $[\![\max b_j(\mu^{k-1}, \tilde{c})]\!] \cup \{0, +\infty\}$, and when a counter value decreases along an edge, it decreases at least by 1. Therefore, within $n \times |C| \times (\max b_j(\mu^{k-1}, \tilde{c}) + 1)$ steps from $(\tilde{c}, b^{\tilde{c}})$ at least one of the counter value becomes 0. When a player-l counter value becomes 0, π_ρ must reach the next region (making the corresponding $c(l) = \mathsf{tgt}(i)$), otherwise π_ρ is not a valid path. Moreover, from the next region, $\mathsf{cost}_i(\rho[j+1] \dots \rho[n])$ is bounded by $\mu_i^{k-1}(\mathbf{e}')$, where $\mathbf{e}'$ denotes the first edge of ρ which belongs to $Z_{>j}$. Therefore,

$$\begin{aligned}\mathsf{cost}_i(\rho) &\leq (n \times |C| \times (\max b_j(\mu^{k-1}, \tilde{c}) + 1)) \times \kappa + \max_{l>j} \max b_l(\mu^{k-1}, \tilde{c}) \\ &\leq (n \times |C| + |V|) \times \kappa + (n \times |C|) \times \max b_j(\mu^{k-1}, \tilde{c}) \times \kappa.\end{aligned}$$

If Player i reaches tgt within X'_j, $\mathsf{cost}_i(\rho)$ would be much smaller. In the above, we have used the bound from Corollary 4.13 as $\mu^{k-1}(\mathbf{e}) = \lambda^{l^*}(\mathbf{e})$ for $\mathbf{e} \in X_l$ for $l > j$. Now as $\tilde{c} \in X_j$, we can use the induction hypothesis for giving a bound to $\max b_j(\mu^{k-1}, \tilde{c})$. As the above shown bound works for any $\rho \in \Gamma_{\mu^{k-1}}(\tilde{c})$, it works for the supremum too, hence we have

$$\begin{aligned}&\mu_i^k(c, w, c') \\ &\leq \sup_{\rho \in \Gamma_{\mu^{k-1}}(\tilde{c})} \mathsf{cost}_i(c', \tilde{c}) + \mathsf{cost}_i(\rho) \\ &\leq |V| \times \kappa + (n \times |C| + |V|) \times \kappa + (n \times |C|) \times \max b_j(\mu^{k-1}, \tilde{c}) \times \kappa \\ &\leq (n|C| + 2|V|) \times \kappa + (n|C| \times \kappa) \times (n|C| + 2|V|) \times \sum_{l=1}^{k-1} (n|C|)^{l-1} \cdot \kappa^l \\ &= (n|C| + 2|V|) \times \sum_{l=1}^{k} (n|C|)^{l-1} \cdot \kappa^l\end{aligned}$$

□

This provides a bound for any $\max b_l(\mu, c)$ for any λ^{j^*}-building function μ, $c \in X_j$, and $l \geq j$. Hence, we can conclude that the counter graph $\mathbb{C}[\mu, c] = \langle C', T' \rangle$ can be made finite, by taking $C' = \{(d, b) \in C \times ([0; Y] \cup \{+\infty\})^{[\![n]\!]} \mid c \Rightarrow^* d\}$, with $Y = n^{|V|} \cdot |V|^{n.|V|} \cdot \kappa^{|V|}$, which is doubly-exponential in the encoding of n. Note that, this makes $|C'|_{\mathsf{reach}}$ at most double-exponential too - which will be one of the key arguments in the complexity analysis of the final algorithm.

Algorithm. With all these ingredients, we now describe a non-deterministic algorithm for the decision problem CONSTRAINED-SPE. To be specific, if indeed there is an SPE with social cost less than the threshold K, then our algorithm can verify that. In this verification algorithm, a certificate is a play (or alternatively a path in the configuration graph) which is potentially the outcome of an SPE with social cost less than K. At the end, we conclude the complexity of the above algorithm in Theorem 4.18.

To start with, we compute λ^*-values for each edge in the configuration graph. In order to do that, we start with λ^{n^*}, and inductively compute $\lambda^{j^*} = \langle \lambda_i^{j^*} \rangle_{i \in [\![n]\!]}$ from $\lambda^{(j+1)^*}$ until $j = 0$. When computing λ^{j^*} from $\lambda^{(j+1)^*}$, we use the intermediary family of labeling functions μ^k. That is, for each $j \in [\![n]\!]$, starting from $\mu^0 = \langle \mu_i^0 \rangle_{i \in [\![n]\!]}$, we compute μ^{k+1} from μ^k until it becomes $\mu^k = \mu^{k+1}$, i.e, reaches a fix-point. Then we declare $\lambda^{j^*} = \mu^k$.

Recall that, for any j, for any intermediary family of functions $\mu^k = \langle \mu_i^k \rangle_{i \in n}$, (where $\mu_i^k : Z_{\geq j} \to \mathbb{N} \cup \{+\infty, -\infty\}$) we only need to compute $\mu_i^k(\mathbf{e})$ for $\mathbf{e} \in Z_j$ because for $\mathbf{e}' \in Z_{\geq j+1}$, $\mu_i^k(\mathbf{e}') = \lambda_i^{(j+1)^*}(\mathbf{e}')$. In the following, we explain how we compute $\mu_i^k(c, w, c')$ for $(c, w, c') \in Z_j$, assuming $\{\mu_i^{k-1}(\mathbf{e}) \mid \mathbf{e} \in Z_{\geq j}\}$ are stored:

- First we verify that whether $\mu_i^k(c, w, c')$ is not $-\infty$. By (4.1), we know that μ^k would be $-\infty$ only if there exists an edge $(c, w', c'') \in Z_j$ from c in the configuration graph such that $\Gamma_{\mu^{k-1}}(c'')$ is empty. Therefore, to verify this, we check for each edge $(c, w', c'') \in Z_j$, whether $\Gamma^{\mu^{k-1}}(c'')$ is not empty. In order to do that,we guess a valid path (but do not store) in $\mathbb{C}[\mu^{k-1}, c'']$. If there is such a path in $\mathbb{C}[\mu^{k-1}, c'']$, there will be one of length bounded by $|C|$, which is doubly-exponential in the size of the input. Hence we keep a counter which we increase by 1 every time we correctly guess a new edge along the path, and the counter stops either when we reach a (c_{tgt}, b), or at the latest when it reaches $|C|$. As the length of such a path is bounded by $|C|$, which is doubly exponential in the input size, we can encode the necessary counter using only exponential space. Therefore, if indeed $\mu_i^k(c, w, c')$ is not $-\infty$, we can verify that using exponential space.

- Once we verified the above, and concluded that $\mu_i^k(c, w, c')$ is not $-\infty$, we know that it must be in $\mathbb{N} \cup \{+\infty\}$.

  Now, we first check whether $\mu_i^k(c, w, c')$ is $+\infty$. For that, we need to verify the conditions stated in Lemma 4.16, i.e, for each $c'' \in \mathsf{dev}_i(c, c')$, we need a valid path π of the form $h \cdot \beta \cdot h'$ in $\mathbb{C}[\mu^{k-1}, c'']$, where β is a cycle and Player i's counter value is > 0 throughout β.

  We guess such a path π in $\mathbb{C}[\mu^{k-1}, c'']$ for each edge $(c, w', c'') \in \mathsf{dev}_i(c, c')$. Of this path. we first guess the very first vertex of the cycle β, say (c_1, b_1) (with $b_1(i) > 0$). Then we guess the cycle itself, keeping only the currently guessed edge in memory at each step. Note that, if there is a cycle on (c_1, b_1), there has to be one within length $|C|$ because within a cycle only the configurations change, not the counter-values. Then we guess a path from $(c'', b^{c''})$ to (c_1, b_1), and another path from (c_1, b_1) to (c_{tgt}, b) (for some b).

  The length of the part up to (c_1, b_1) is bounded by $|C'|_{\mathsf{reach}}$, whilst we can always get a path of length at most $|C|$ (if it exists) from (c_1, b_1) to (c_{tgt}, b). This discrepancy between the two bounds is mainly because for the latter part of the path, we do not exactly fix the final vertex (b can be any tuple of n non-negative values), we just want the first component to be c_{tgt}, while for the former part it gets fixed to (c_1, b_1).

  If we can guess such a path π for each of $c'' \in \mathsf{dev}_i(c, c')$, we return $\mu_i^k(c, w, c') = +\infty$.

- Otherwise, we have at least 1 and at most $|V|$ configurations $c'' \in \mathsf{dev}_i(c, c')$ such that $\sup_{\rho \in \Gamma_{\mu^{k-1}}(c'')} \mathsf{cost}_i(\rho) \in \mathbb{N}$. We call this set of configurations as $\mathsf{dev}_i(c, c')|_{\mathsf{finsup}}$.

To compute the finite value of $\mu_i^k(c, w, c')$, we compute $M_j = \sup_{\rho \in \Gamma_{\mu^{k-1}}(c''_j)} \mathsf{cost}_i(\rho)$ for each of $c''_j \in \mathsf{dev}_i(c, c')|_{\mathsf{finsup}}$, and then take the minimum. In order to compute M_j, we first tentatively set $M_j = \min_{e \in E} \mathsf{f}_e(1)$, and proceed iteratively as follows: we guess a valid path π of length at most $|C'|_{\mathsf{reach}}$ of $\mathbb{C}[\mu^{k-1}, c''_j]$ such that $\mathsf{cost}_i(\rho) \geq M_j$, where ρ is the corresponding path of π in $\mathcal{G}$. If such a path exists, we increase M_j by 1, and repeat. At some point, we get M_j such that there doesn't exist a valid path π of length at most $|C'|_{\mathsf{reach}}$ with Player i's cost larger than or equal to $M_j + 1$, but there exists a valid path with Player i's cost larger than or equal to M_j. We store that M_j.

When all values M_j have been computed, we return:

$$M = \min_{j \in [\![|\mathsf{dev}_i(c,c')|_{\mathsf{finsup}}|]\!]} \{\mathsf{cost}_i(c, c''_j) + M_j\}.$$

We use at most doubly-exponential space in the procedure of guessing a path, and we reuse that space for guessing the next path in the above algorithm.

We keep a binary counter throughout transitioning from μ^{k-1} to μ^k to flag whether the fixpoint has been reached. Once we reach $\lambda^* = \langle \lambda_i^* \rangle$, we finally check whether $\Gamma_{\lambda^*}(c_{\mathsf{src}})$ is empty by guessing a path in $\mathbb{C}[\lambda^*, c_{\mathsf{src}}]$ from $(c_{\mathsf{src}}, b^{c_{\mathsf{src}}})$ to (c_{tgt}, b).

In conclusion, we use doubly-exponential space: (1) to store $\{\mu_i^{k-1}(\mathbf{e}) \mid i \in [\![n]\!], \mathbf{e} \in T\}$ which is double-exponential in the encoding of the number of players, and (2) to encode a counter which keeps checking whether the length of our guessed paths does not exceed $|C'|_{\mathsf{reach}}$.

Theorem 4.18. *The existence of SPEs in a dynamic NCG can be decided in 2EXPSPACE.*

4.3 Conclusion

In this chapter, we have studied SPE in dynamic network congestion games. Even though, we do not have certain result whether SPE always exist in dynamic NCG or not, we have shown that how constrained SPE can be of interest. Henceforth, we studied constrained SPE problem: first we characterized the outcome of SPE, then based on the characterization, we developed an algorithm which decides in 2EXPSPACE whether given a dynamic game and constraint, there exists an SPE in the game with social cost less or equal to that constraint value.

The natural two problems that remain open are : (1) existence of SPE, (2) finding lower bound complexity results for constrained SPE problem.

Part II

Network Congestion games with number of players as parameter

5 Framework for parameterized games on series-parallel graphs

In this part of the thesis, we continue our study on network congestion games, but from a different perspective. Here, instead of fixing a number of players, we consider the number of players as a parameter in the model. Our objective here is to study how the optimal strategy profiles behave with this parameter.

But the scope of this part is restricted compared to Part I. That is why, in this chapter, we first specify what the particular restrictions are, and what new notions we introduce for adjusting with the new restricted setting for tackling network congestion game with the number of players as a parameter.

Restrictions on the model. First of all, we remove the two specificities that we earlier considered in Part I, namely: *synchronous cost computation* and *dynamic strategies*. This is solely for technical reason, and we believe studying this problem on other model of congestion games (including the one that we considered in Part I) could be a future work of interest. Here:

- costs are computed *non-synchronously*, as it is done in *classical* network congestion games [3, 39, 57]. That means two players bears congestion effect in their cost for an edge even if they do not take that edge in their route at the same step.
- Strategies are just paths from source to target, unlike the dynamic strategies that we considered in Part I.

Therefore, both syntactically and semantically, we go back to network congestion games from the literature. Moreover, we restrict the cost functions associated with the edges to linear functions only, contrary to affine functions that we considered in Part I. Finally, we restrict the network arena to only *series-parallel* graphs instead of *any* directed graphs that have been considered in Part I. With this, let us introduce our setting formally, where some of the definitions might just be inherited from Chapter 1, but recalled again anyway.

5.1 Definitions

Series-Parallel network Let $\mathcal{H}$ be a family of non-decreasing linear functions from $\mathbb{N}$ to $\mathbb{N}$. Formally, a function $h : \mathbb{N} \to \mathbb{N}$ of $\mathcal{H}$ is of the form $h(x) = w \cdot x$, with $w \geq 0$.

We first redefine our notion of a network in this part, which is a directed acyclic graph with a single source-target vertex pair, contrary to multiple source-target vertex pairs considered in Part I. Formally,

Definition 5.1 (Network)**.** *A network is a tuple $\mathcal{N} = \langle V, E, s, t\rangle$, where $\langle V, E\rangle$ forms a directed acyclic graph, s and t are the* source *and* target *vertices respectively, and for every $v \in V$, there is a path from s to v, and a path from v to t. Moreover, there is no incoming edge to s, and no outgoing edge from t.*

As mentioned before, here we restrict our study on *series-parallel* networks. In order to define series-parallel network, we first define two ways of composing any two networks: two networks $\mathcal{N}_1 = \langle V_1, E_1, s_1, t_1\rangle$ and $\mathcal{N}_2 = \langle V_2, E_2, s_2, t_2\rangle$, and they can be composed to form a single network by:

- *Series composition*: Intuitively, two networks are composed to a single one by merging the target vertex t_1 of the former and the source vertex s_2 of the latter to a new vertex. As the new vertex is no longer a target vertex, the self-loop from t_1 is also removed. We refer to this network as $N_1 \oplus N_2$ Formally, $N_1 \oplus N_2 = \langle V, E, s, t\rangle$, where

$$V = V_1 \cup V_2 \cup \{u\} \setminus \{t_1, s_2\}$$
$$E = E_1 \cup E_2 \cup \{(v, u) : (v, t_1) \in E_1\} \cup \{(u, v) : (s_2, v) \in E_2\}$$

  and $s = s_1$, and $t = t_2$.

- *Parallel composition*: Intuitively, two networks are composed *parallelly* by merging the two source vertices into a new source, and two target vertices into a new target vertex. We refer the new network as $N_1 \parallel N_2$. Formally, $N_1 \parallel N_2 = \langle V, E, s, t\rangle$, where

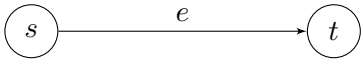
$$V = V_1 \cup V_2 \cup \{s, t\} \setminus \{s_1, t_2, s_2, t_2\}$$
$$E = E_1 \cup E_2 \cup \{(s, v) : (s_1, v) \in E_1\} \cup \{(s, v) : (s_2, v) \in E_2\}$$
$$\cup \{(v, t) : (v, t_1) \in E_1\} \cup \{(v, t) : (v, t_2) \in E_2\}$$

Definition 5.2 (Series-Parallel Network)**.** *A network $\mathcal{N} = \langle V, E, s, t\rangle$ is called a* series-parallel network *if either it is a single edge (s, t), or series or parallel composition of two series-parallel networks.*

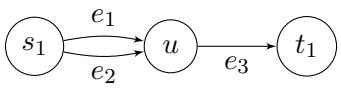
Example 5.3. *Consider Figure 5.1. Here 5.1a is a single edge between source and target, the* basic *unit for any series-parallel network. Any series-parallel network is built starting from single edge(s), and composing them either in series or in parallel. For example, the network $\mathcal{N}_1 \oplus \mathcal{N}_2$ of Figure 5.1d is built by composing $\mathcal{N}_1$ of 5.1b and $\mathcal{N}_2$ of 5.1c, while $\mathcal{N}_1 \parallel \mathcal{N}_2$ is built by composing them in parallel.* □

Parameterized Network Congestion Games. We define *parameterized network congestion game* (pNCG) on this series-parallel network, which is a network congestion game without fixed number of players. Formally,

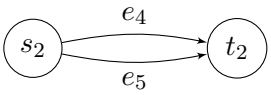
Definition 5.4 (Parameterized Network Congestion Game)**.** *A* parameterized network congestion game *(pNCG) is a tuple $\mathcal{G} = \langle V, T, s, t\rangle$, where V is the set of vertices, $T \subseteq V \times \mathcal{H} \times V$ is the set of transitions labeled by non-decreasing linear functions, and the* underlying graph *$\mathcal{N} = \langle V, E, s, t\rangle$, where $E = \{(v, v') :$ for each $(v, h, v') \in T\}$, is a series-parallel network.*



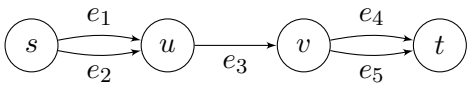
(a) A single edge: smallest series-parallel network



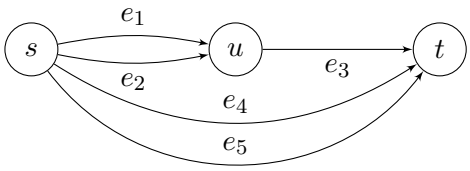
(b) $\mathcal{N}_1$



(c) $\mathcal{N}_2$



(d) $\mathcal{N}_1 \oplus \mathcal{N}_2$



(e) $\mathcal{N}_1 \parallel \mathcal{N}_2$

Figure 5.1: Series and Parallel composition of networks

Note that, syntactically a pNCG is no different from the underlying arena $\mathcal{N}$ on which the game is defined. This is in contrast with the network congestion game (Definition 1.1) defined in Chapter 1, where apart from the arena $\mathcal{A}$, the number of players n, and a source-target assigning map g were part of an instance.

Semantics. Intuitively, a pNCG captures infinitely many NCG , one for every integer $n \in \mathbb{N}$. Hence, with a pNCG $\mathcal{G}$, we associate infinitely many network congestion games, denoted by $\mathcal{G}_n$ for each $n \in \mathbb{N} \setminus \{0\}$. Here $\mathcal{G}_n$ denotes the congestion game $\langle \mathcal{G}, n, \mathsf{g} \rangle$, where $\mathsf{g}(s,t) = n$. Because, a pNCG only associates with *symmetric* NCG, we omit g from the notation, and simply write $\mathcal{G}_n = \langle \mathcal{G}, n \rangle$. This also means pNCG inherits the semantics, i.e, objective of a player, and how a game *proceeds* in rounds, from NCG itself, as it is explained in the semantics part on Page 4.

Strategy. Because a pNCG captures a class of NCG, σ is a Player i strategy in a pNCG $\mathcal{G}$ means σ is a Player i strategy in an NCG from the class that $\mathcal{G}$ represents. Moreover, as mentioned earlier, in this part, we restrict the strategy space only to what we called earlier *blind* strategies. Recall from Chapter 1, a blind strategy π is a path from s to t in the arena. A path π from s to t in $\mathcal{G}$ is denoted as a sequence of the form $s \xrightarrow{e_1} v_1 \dots \xrightarrow{e_{|\pi|}} t$. For such a path π, we write $e_j \in \pi$ for $1 \leq j \leq |\pi|$. We denote $\mathsf{Paths}\mathcal{G}$ to be the set of all paths in $\mathcal{G}$. As all the players in any associated NCG $\mathcal{G}_n$ have same source-target vertex pair, a strategy can be directly associated to the pNCG $\mathcal{G}$ itself.

A *strategy profile* σ of $\mathcal{G}_n$ for any $n \in \mathbb{N} \setminus \{0\}$ is a n-tuple $(\pi_i)_{i \in [\![n]\!]}$ where each π_i is a strategy of $\mathcal{G}$. We call σ to be a strategy profile of $\mathcal{G}$ itself, if it is a strategy profile of $\mathcal{G}_n$ for some $n \in \mathbb{N} \setminus \{0\}$.

In the next chapter of this part, our main objective is to investigate how optimal strategy profile for stability behaves as the parameter evolves. Hence, instead of denoting a strategy profile of $\mathcal{G}_n$ as a n-tuple of paths in $\mathcal{G}$, we use two notations that are more resilient to the change in n, and therefore easier to observe if and how strategy profiles relate to each other when n varies:

- *Path representation* of a strategy profile: Here a strategy profile is a $|\mathsf{Paths}\mathcal{G}|$-tuple $\vec{m} = (m_\pi)_{\pi \in \mathsf{Paths}\mathcal{G}}$, in which m_π denotes the number of players taking path π in the strategy profile. Necessarily, for a strategy profile $\vec{m}$ of $\mathcal{G}_n$, $\sum_{\pi \in \mathsf{Paths}\mathcal{G}} m_\pi = n$.

- *Edge representation* of a strategy profile: Here a strategy profile is a $|E|$-tuple $\vec{n} = (n_e)_{e \in E}$, in which n_e denotes the number of players taking e in their strategy in this profile. Here necessarily for every vertex $v \in V \setminus \{s, t\}$ the number players incoming to v in a strategy profile has to match up to the number of players outgoing from v. Formally, $\sum_{e \in In(v)} n_e = \sum_{e \in Out(v)} n_e$, where $In(v)$ and $Out(v)$ are the set of incoming and outgoing edges of v.

In both representations, we do not keep the information about which player is playing which strategy as all of them has same source-target vertex pair, but this makes multiple strategy profiles to have same path/edge representation. Two strategy profiles that have the same path representation are isomorphic to each other up to permutation of the players. As a player's identity does not matter for our current context, we can consider path representation as a *unique* representation of a strategy profile, and simply refer to a strategy profile unless otherwise mentioned. On the other hand, that is not the case with edge representation.

Here is an example where two strategy profiles (path representation) have same edge representation.

Example 5.5. *Consider a pNCG namely, say $\mathcal{G}$, on network 5.1d from Figure 5.1, and consider the associated NCG with three players $\mathcal{G}_3 = \langle \mathcal{G}, 3 \rangle$. In this network, the paths are $\pi_1 = s \xrightarrow{e_1} u \xrightarrow{e_3} v \xrightarrow{e_4} t$, $\pi_2 = s \xrightarrow{e_1} u \xrightarrow{e_3} v \xrightarrow{e_5} t$, $\pi_3 = s \xrightarrow{e_2} u \xrightarrow{e_3} v \xrightarrow{e_4} t$, and $\pi_4 = s \xrightarrow{e_2} u \xrightarrow{e_3} v \xrightarrow{e_5} t$.*

Now consider two path-strategy profiles $\vec{m}_1$ and $\vec{m}_2$ as follows:

$$\begin{aligned} m_1(\pi_1) &= 1 & m_2(\pi_1) &= 2 \\ m_1(\pi_2) &= 1 & m_2(\pi_2) &= 0 \\ m_1(\pi_3) &= 1 & m_2(\pi_3) &= 1 \\ m_1(\pi_4) &= 0 & m_2(\pi_4) &= 0 \end{aligned}$$

Both path profiles have the same edge-representation, which is $\vec{n} = (n_e)_{e \in E}$, where $n_{e_1} = 2$, $n_{e_2} = 1, n_{e_3} = 3, n_{e_4} = 1$ and $n_{e_5} = 2$. □

We write $\mathfrak{B}_{edge}$ for the set of edge-representations of all strategy profiles, while we write $\mathfrak{B}_{path}$ for the set of path-representations of all strategy profiles. Moreover, there is a map which associates a path-strategy profile uniquely to its corresponding edge-strategy profile as follows:

Definition 5.6. $\Gamma : \mathfrak{B}_{path} \to \mathfrak{B}_{edge}$ *maps each path-strategy profile* $\vec{m} = (m_\pi)_{\pi \in \mathsf{Paths}\mathcal{G}}$ *to a edge-strategy profile* $\vec{n} = (n_e)_{e \in E}$ *where:*

$$n_e = \sum_{\pi : e \in \pi} m_\pi$$

To have a simpler notation without any further confusion, we always use $\vec{m}$ to denote a path representation, while we use $\vec{n}$ to denote an edge-representation of a strategy profile.

Cost. As mentioned earlier, we get back to the *non-synchrounous* way of computing cost, which has been usual in the literature [3, 9, 39, 57, 60] with few exceptions[6, 7]. To recall, here a player bears congestion effect on their cost due to some other player if they both take that edge in their path. With each path-strategy profile $\vec{m} \in \mathfrak{B}_{path}$, and each path $\pi \in \mathsf{Paths}\mathcal{G}$, we associate a *cost*, written as $\mathsf{cost}_\pi(\vec{m}) = \sum_{e \in \pi} \mathsf{h}_e(\Gamma(\vec{m})_e)$, meaning a player who is taking path π in profile $\vec{m}$ pays $\mathsf{cost}_\pi(\vec{m})$.

Subsequently, the social cost of a strategy profile $\vec{m}$ is

$$\mathsf{soccost}(\vec{m}) = \sum_{\pi \in \mathsf{Paths}\mathcal{G}} m_\pi \cdot \mathsf{cost}_\pi(\vec{m})$$

Similarly, from an edge-strategy profile too, a player's cost can be defined, but we never use that cost in the sequel, hence it is not being defined formally.

5.2 Conclusion

In this chapter, we have formally defined Parameterized network congestion games on series-parallel network, in a restricted setting than Part I, putting restriction on (a) arena, and (b) types of strategies. Moreover, the way of computing cost has also been altered from what it was in the earlier part. Each of these restrictions rose from our technical limitations, and we will justify those in details when we approach the problem that we study in the next chapter.

6 Nash Equilibria for Parametric Network congestion games

In this chapter, we study Nash Equilibria (NE) of parameterized network congestion games (pNCG). Intuitively, a pNCG is a class of network congestion games, in which a game arena is fixed and the number of players varies from one game to another. Hence, any NE for an NCG of the class of games represented by a pNCG , is considered as an NE of that pNCG itself. Formally,

Definition 6.1 (path-NE)**.** *A path-strategy profile $\vec{m} = (m_\pi)_{\pi \in \mathcal{P}_\mathcal{G}}$ is an NE for pNCG $\mathcal{G}$ if it is an NE of NCG $\langle \mathcal{G}, k \rangle$, where $k = \sum_{\pi \in \mathsf{Paths}_\mathcal{G}} m_\pi$. We refer such $\vec{m}$ as a path NE of $\mathcal{G}$.*

Recall from Chapter 5, we defined a map Γ which associates each path-strategy profile to an edge-strategy profile by counting how many players takes each edge in the path-strategy profile. Using that map, we define the following:

Definition 6.2 (edge-NE)**.** *An edge-strategy profile $\vec{n} = (n_e)_{e \in E}$ is said to be an NE for pNCG iff there exists a path-strategy profile $\vec{m} = (m_\pi)_{e \in \mathsf{Paths}_\mathcal{G}}$ such that $\Gamma(\vec{m}) = \vec{n}$ and $\vec{m}$ is an NE of for $\mathcal{G}$. Subsequently, we refer to such $\vec{n}$ as an edge NE of $\mathcal{G}$.*

We know NE always exists for an NCG[56], we have the following result regarding the existence of an NE for a pNCG :

Corollary 6.3. *Nash equilibria always exists for a parameterized network congestion games.*

From hereon, we use $\mathfrak{N}_{path}$ and $\mathfrak{N}_{edge}$ to denote the set of path NE and edge NE profiles, respectively.

From earlier chapter of this thesis, we know how to compute an NE for an NCG, and therefore we can effectively compute NE for any game from the class of games represented by any given pNCG . But from those algorithms, irrespective of the number of players, we essentially need to follow the same method to compute an NE. A priori, we did not know whether an NE of one game is somehow related to an NE of another game from the same class of games represented by a given pNCG . And if they are, can we use that relation to compute an NE for a game from an NE for another game without computing it from scratch?

More specifically, we are concerned with the problem if and how we can compute an NE for a game with large number of players from NE of games with smaller number of players. In this chapter, we investigate this problem for series-parallel network. Formally, we address the following problem:

Definition 6.4 (Problem 1)**.** *Given a pNCG $\mathcal{G}$ on a series-parallel network, can we compute a finite representation of a mapping $\mathsf{NE} \colon \mathbb{N} \to \mathbb{N}^{|E|}$ such that $\mathsf{NE}(k)$ is an NE of $\langle \mathcal{G}, k \rangle$.*

Remark 6.5. *Even though currently, in this chapter, our objective is to study the problem only for series-parallel graphs, many results that we obtain on the way are also true for general graphs. Hence, we specify in the statements when we prove a result only for series-parallel graphs.*

In the next section, we first characterize path NE profiles. From that characterization, we move on to show that the set of edge NE profiles is a *semi-linear set*, so it has a finite *basis* and a finite set of *periods*. We then compute the set of periods, the set of bases, and finally discuss about computability and expressibility of NEs.

6.1 Characterization of NE Profiles: Path and Edge Representations

Here, we first show when a path-strategy profile $\vec{m}$ is a path NE . Intuitively, $\vec{m}$ is a path NE iff for a path π, which is taken by some player in the profile (i.e. $m_\pi > 0$), and for any other path π' (which may or may not be taken by a player in $\vec{m}$), the cost that a player pays in π is less or equal to the cost they would pay in π' if they unilaterally deviate. Moreover, cost in π is sum of the cost of each edge e of π, which in turn, is just $w_e \cdot n_e$, where $\Gamma(\vec{m}) = \vec{n} = (n_e)_{e\in E}$. Formally, we have the following (here $\pi \setminus \pi'$ denotes the set of edges which are in π, but not in π'):

Theorem 6.6. *In any network $\mathcal{G}$ with linear cost functions, a path-strategy profile $\vec{m} = (m_\pi)_{\pi\in \mathsf{Paths}\mathcal{G}}$ is an NE iff*

$$\forall \pi, \pi' \in \mathsf{Paths}\mathcal{G},\ m_\pi > 0 \implies \sum_{e\in\pi\setminus\pi'} w_e \cdot n_e \leq \sum_{e\in\pi'\setminus\pi} w_e \cdot (n_e + 1) \tag{6.1}$$

where $\Gamma(\vec{m}) = \vec{n} = (n_e)_{e\in E}$.

Proof. The proof is pretty straightforward. Consider a path-strategy profile $\vec{m} = (m_\pi)_{\pi\in \mathsf{Paths}\mathcal{G}}$ which satisfies (6.1). We also assume $\Gamma(\vec{m}) = \vec{n} = (n_e)_{e\in E}$. If $\vec{m}$ is not a path NE , then there exists $i \in [\![n]\!]$ such that Player i is a deviating player who benefits by unilaterally deviate from their current strategy. Suppose π be the path Player i is currently taking in $\vec{m}$, and π' be a cost-effective unilateral deviation for Player i. Hence, $m_\pi \geq 1$ (as Player i is taking π in $\vec{m}$) and $\sum_{e\in\pi\setminus\pi'} w_e \cdot n_e > \sum_{e\in\pi'\setminus\pi} w_e \cdot n_e$, otherwise π' is not more cost-effective than π. But, that contradicts (6.1), hence $\vec{m}$ is an NE.

For the opposite direction, suppose a strategy profile $\vec{m} = (m_\pi)_{\pi\in \mathsf{Paths}\mathcal{G}}$ is a path NE with $\Gamma(\vec{m}) = \vec{n} = (n_e)_{e\in E}$, and also suppose towards a contradiction that there are paths $\pi, \pi' \in \mathsf{Paths}\mathcal{G}$ such that $m_\pi > 0$ and $\sum_{e\in\pi\setminus\pi'} w_e \cdot n_e > \sum_{e\in\pi'\setminus\pi} w_e \cdot (n_e + 1)$. Now, if a player deviates from π to π', for the common part of the two paths, $\pi \cap \pi'$, the number of players remain same. Hence, a player who was taking π in $\vec{m}$ should deviate unilaterally from π to π', and reduce their own cost. That contradicts to our hypothesis that $\vec{m}$ is an NE profile. □

Note that we have not used series-parallel graph in the above, which implies Theorem 6.6 holds for any graph. This also holds true for the next section (6.2) too, but from Section 6.3 onward, we will see results that are proven only on series-parallel network.

6.2 Set of NE Profiles as a Semi-linear Set

A strategy profile in edge representation is a subset of $\mathbb{N}^{|E|}$. In this section, using the above characterization, we establish that the set $\mathfrak{N}_{edge} \subseteq \mathbb{N}^{|E|}$ of a pNCG $\mathcal{G}$ is a *semi-linear* set. Upon proving this, we will be able to give a structure to $\mathfrak{N}_{edge}$, which in turn, enables us to compute an NE for *sufficiently* large number of players from NE for comparatively small number of players.

First, we recall some definitions from the literature related to semi-linear sets [52]. Here, we are only interested in semi-linear subsets of $\mathbb{N}^{|E|}$, so we consider the definitions accordingly.

Definition 6.7. *A set $S \subseteq \mathbb{N}^{|E|}$ is called* linear *if there is a base vector $\vec{b} \in \mathbb{N}^n$ and a finite set of period vectors $P = \{p_1, p_2, \dots p_m\}$ such that*

$$\forall s \in S, \forall i \in [m],\ \exists \lambda_i \in \mathbb{N},\ s = \vec{b} + \sum_{i \in [\![m]\!]} \lambda_i \cdot p_i$$

Such a linear set S can also be written as $L(b; P)$

Definition 6.8. *A set $S \subseteq \mathbb{N}^{|E|}$ is said to be* semi-linear *if it is a finite union of linear sets.*

Therefore, a semi-linear set S can be expressed as

$$S = \bigcup_{i \in I} L(b_i; P_i)$$

where both $|I|, |P_i|$ are finite.

Now, we show that the set of NE profiles of a pNCG is a semi-linear set.

Theorem 6.9. *For a parameterized network congestion game, $\mathfrak{N}_{edge}$ is a semi-linear set.*

Proof. In order to prove this, we show that the set of NE profiles in edge-representation is definable in *Presburger arithmetic* (**PrA**). We know from [27] that a set definable in **PrA** is semi-linear, and vice-versa.

We first recall the definitions from literature regarding **PrA** : it is the *first-order* theory of the model $(\mathbb{N}, +, =)$. For every number $n \in \mathbb{N}$, the relations $<, \leq$, modulo comparison relation $\equiv_m$, for natural numbers $n \geq 1$, and the functions $x \mapsto nx$ of multiplication by natural numbers are definable in $(\mathbb{N}, +, =)$. Therefore, we show here how we can define $\mathfrak{N}_{edge}$ using formulas in **PrA** .

First, we write the formulas below, and then we justify that each component of each formula is in **PrA** , in which we will be using the above symbols for having a shorter version of **PrA** formulas instead of what we would have got if we use only $(\mathbb{N}, +, =)$.

$$\textbf{EdgeProfile}(\vec{n} = (n_e)_{e \in E}) := \forall e \in E,\ \sum_{e \in In(v)} n_e = \sum_{e \in Out(v)} n_e$$

Here, universal quantification over a finite set corresponds to a conjunction over finitely many formulas, each of which are **PrA** formulas, hence so is **EdgeProfile**$(\vec{n})$. Any $\vec{n}$ such that $\vec{n} \models$ **EdgeProfile**$(\vec{n})$ is an edge profile of the network.

Next, we enrich the grammar by introducing a unary operator $\mathbf{\Gamma}$ which maps $\mathbb{N}^{|\mathcal{P}|}$ to $\mathbb{N}^{|E|}$, and definable in $(\mathbb{N}, +, =)$:

$$\mathbf{\Gamma}(\vec{m} = (m_\pi)_{\pi \in \mathcal{P}}) \coloneqq (n_e)_{e \in E}, \quad \text{where } n_e = \sum_{e \in \pi} m_\pi$$

This unary operator maps a path profile to its corresponding edge profile in the network.

We further introduce two formulas which essentially capture the set of path NE and edge NE .

$$\begin{aligned}\textbf{PathNE}(\vec{m} = (m_\pi)_{\pi \in \mathcal{P}}) \coloneqq \forall \pi, \pi' \in \mathcal{P},\ m_\pi > 0 \\ \implies \sum_{e \in \pi \setminus \pi'} w_e \cdot \mathbf{\Gamma}(\vec{m})_e \leq \sum_{e \in \pi' \setminus \pi} w_e \cdot (\mathbf{\Gamma}(\vec{m})_e + 1)\end{aligned}$$

$$\begin{aligned}\textbf{EdgeNE}(\vec{n} = (n_e)_{e \in E}) \coloneqq \textbf{EdgeProfile}(\vec{n}) \\ \wedge (\exists \vec{m} = (m_\pi)_{\pi_\in \mathcal{P}},\ (\mathbf{\Gamma}(\vec{m}) = \vec{n}) \wedge \textbf{PathNE}(\vec{m}))\end{aligned}$$

Here, $\textbf{PathNE}(\vec{m})$ is only satisfied by path-representation of a strategy profile which is an NE of $\mathcal{G}$, and finally $\textbf{EdgeNE}(\vec{n})$ captures the edge representations of those strategy profiles which are NE. Therefore, the set $\mathfrak{N}_\mathcal{G}$ can be defined as:

$$\mathfrak{N}_\mathcal{G} \coloneqq \{(n_e)_{e \in E} \in \mathbb{N}^{|E|} : (n_e)_{e \in E} \models \textbf{EdgeNE}(\vec{n})\}$$

which shows it is a **PrA** -definable set, hence by [27], $\mathfrak{N}_\mathcal{G}$ is a semi-linear set too. □

6.3 Period vectors of edge NE profiles

As $\mathfrak{N}_\mathcal{G}$ is a semi-linear set, and infinite (at least one NE for each $\langle \mathcal{G}, k \rangle$) for any network $\mathcal{G}$, there exists at least one base vector $\vec{b}$ and one set of period vectors P_i which is non-empty, when we express $\mathfrak{N}_\mathcal{G} = \bigcup_{i \in I} L(b_i; P_i)$. Also note that, by definition a period vector $\vec{d} = (d_e)_{e \in E}$ of $\mathcal{G}$ is itself a strategy profile, i.e, $\vec{d} \models \textbf{EdgeProfile}(\vec{n})$ (where **EdgeProfile** is the formula in the proof of Theorem 6.9).

Our next objective is to characterize the period vectors for a given pNCG . Intuitively, we first establish that in a period vector, the cost across any path is constant. This is the key component that we establish *only* for series-parallel networks, hence restricting the full chapter to this class of graphs.

Theorem 6.10. *For pNCG $\mathcal{G}$ on a series-parallel network, for any period vector $\vec{d} = (d_e)_{e \in E}$, we have the following:*

$$\exists \kappa_{\vec{d}} \in \mathbb{N}\ \forall \pi \in \mathsf{Paths}_\mathcal{G},\ \sum_{e \in \pi} w_e \cdot d_e = \kappa_{\vec{d}} \tag{6.2}$$

Proof. Recall from Definition 5.2 of Chapter 5, a series-parallel network is defined as either a single edge between s and t, or series/parallel composition of two series-parallel networks. Here, in this

proof, we use structural induction on that compositions of series-parallel networks. The base case of this structural induction is a single edge.

For a network $\mathcal{G}$ comprised of a single edge e between s and t, the cost must be constant *across all paths* as there is only one path.

To formalize the induction hypothesis, suppose $\mathcal{G}_1 = \langle V_1, E_1, s_1, t_1\rangle$ and $\mathcal{G}_2 = \langle V_2, E_2, s_2, t_2\rangle$ are the two series-parallel graphs, and $\vec{d}_1, \vec{d}_2$ are the two period vectors associated with these two networks respectively. Moreover, we denote $\mathcal{P}_1$ and $\mathcal{P}_2$ as the set of all paths from the corresponding source to target in $\mathcal{G}_1$ and $\mathcal{G}_2$ respectively. As the induction hypothesis, we have the following: for any period vector $\vec{d}_1 = (d_e^{(1)})_{e\in E_1}$ of $\mathcal{G}_1$ and similarly for $\vec{d}_2 = (d_e^{(2)})_{e\in E_2}$ of $\mathcal{G}_2$, there are $\kappa_{\vec{d}_1}$ and $\kappa_{\vec{d}_2}$ such that:

$$\forall \pi \in \mathcal{P}_1, \sum_{e\in\pi} w_e \cdot d_e^{(1)} = \kappa_{\vec{d}_1}, \text{ and} \tag{6.3}$$

$$\forall \pi \in \mathcal{P}_2, \sum_{e\in\pi} w_e \cdot d_e^{(2)} = \kappa_{\vec{d}_2} \tag{6.4}$$

In the induction step, one by one, we consider $\mathcal{G}$ first as the series composition $\mathcal{G}_1 \oplus \mathcal{G}_2$ and then as the parallel composition $\mathcal{G}_1 \parallel \mathcal{G}_2$, we denote the set of paths to be $\mathsf{Paths}\mathcal{G}$, and establish that for any period vector $\vec{d}$ of $\mathcal{G}$, there is a $\kappa_{\vec{d}}$ satisfying above.

Firstly, we consider $\mathcal{G}$ to be $\mathcal{G}_1 \oplus \mathcal{G}_2$. From the series composition, we have t_1 of $\mathcal{G}_1$ and s_2 of $\mathcal{G}_2$ merged into a single vertex, say v, of $\mathcal{G}$. Therefore, any path π of $\mathsf{Paths}\mathcal{G}$ can be thought of as concatenating two paths π_1 of $\mathcal{P}_1$ and π_2 of $\mathcal{P}_2$, slightly abusing notations though.

Moreover, any period vector $\vec{d} = (d_e)_{e\in E}$ of $\mathcal{G}$ can be expressed as $(\vec{d_1} = (d_e)_{e\in E_1}, \vec{d_2} = (d_e)_{e\in E_2})$, simply because $E = E_1 \sqcup E_2$, where $\vec{d_1}$ and $\vec{d_2}$ are the period vectors of $\mathcal{G}_1$ and $\mathcal{G}_2$ respectively. From the induction hypothesis, we have $\kappa_{\vec{d_1}}, \kappa_{\vec{d_2}} \in \mathbb{N}$ for these $\vec{d_1}, \vec{d_2}$. Therefore, for any $\pi \in \mathcal{P}$, we have $\sum\limits_{e\in\pi} w_e \cdot d_e = \sum\limits_{e\in\pi_1} w_e \cdot d_e + \sum\limits_{e\in\pi_2} w_e \cdot d_e = \kappa_{\vec{d_1}} + \kappa_{\vec{d_2}}$. So, $\kappa = \kappa_{\vec{d_1}} + \kappa_{\vec{d_2}}$ works!

Now we consider $\mathcal{G} = \mathcal{G}_1 \parallel \mathcal{G}_2$, and a period vector $\vec{d}$ of $\mathcal{G}$. Slightly abusing the notations, we can say that the set of paths $\mathsf{Paths}\mathcal{G}$ can be decomposed into two disjoint sets: the set of paths in $\mathcal{G}_1$, denoted by $\mathcal{P}_1$ and the set of paths in $\mathcal{G}_2$, denoted by $\mathcal{P}_2$. For $\vec{d} = (d_e)_{e\in E}$, from the induction hypothesis, we have κ_1 and κ_2 such that $\forall \pi \in \mathcal{P}_1 \sum\limits_{e\in\pi} w_e \cdot d_e = \kappa_1$ and $\forall \pi \in \mathcal{P}_2 \sum\limits_{e\in\pi} w_e \cdot d_e = \kappa_2$.

Now we consider two paths $\pi_1 \in \mathcal{P}_1$ and $\pi_2 \in \mathcal{P}_2$. If we prove that $\sum\limits_{e\in\pi_1} w_e \cdot d_e = \sum\limits_{e\in\pi_2} w_e \cdot d_e$, we are done.

Without loss of generality, suppose towards a contradiction that $\sum\limits_{e\in\pi_1} w_e \cdot d_e > \sum\limits_{e\in\pi_2} w_e \cdot d_e$, and we denote $M = \sum\limits_{e\in\pi_1} w_e \cdot d_e - \sum\limits_{e\in\pi_2} w_e \cdot d_e > 0$. Note that, even though we fix a $\pi_1 \in \mathcal{P}_1$ and $\pi_2 \in \mathcal{P}_2$, and defined M, this would remain the same if we take some other $\pi_1' \in \mathcal{P}_1$ and $\pi_2' \in \mathcal{P}_2$ because of the induction hypothesis.

Now, we consider an NE profile $\vec{n} = (n_e)_{e\in E} \in \mathfrak{N}_{\mathcal{G}}$, and also an arbitrary path NE $\vec{m} = (m_\pi)_{\pi\in \mathsf{Paths}_{\mathcal{G}}}$ with $\Gamma(\vec{m}) = \vec{n}$. With this, we can take a path $\pi \in \mathcal{P}_1$ with $m_\pi > 0$. Now for this $\pi \in \mathcal{P}_1$, and the $\pi_2 \in \mathcal{P}_2$ that we fixed earlier, we have from (6.1):

$$\sum_{e\in\pi\setminus\pi_2} w_e \cdot n_e \leq \sum_{e\in\pi_2\setminus\pi} w_e \cdot (n_e + 1)$$

We denote $L = \sum_{e\in\pi_2\setminus\pi} w_e \cdot (n_e + 1) - \sum_{e\in\pi\setminus\pi_2} w_e \cdot n_e$. By definition of period vector, for any $\lambda \in \mathbb{N}$, we have

$$\sum_{e\in\pi\setminus\pi_2} w_e \cdot (n_e + \lambda d_e) \leq \sum_{e\in\pi_2\setminus\pi} w_e \cdot (n_e + \lambda d_e + 1)$$

We take $\lambda = \lceil \frac{L}{M} \rceil$ Then,

$$\sum_{\pi\setminus\pi_2} w_e \cdot n_e - \sum_{\pi_2\setminus\pi} w_e \cdot (n_e + 1) + \lambda\left(\sum_{e\in\pi} w_e \cdot d_e - \sum_{e\in\pi_2} w_e \cdot d_e\right) = -L + \left\lceil \frac{L}{M} \right\rceil \cdot M > 0$$
$$\implies \sum_{e\in\pi\setminus\pi_2} w_e \cdot (n_e + \lambda d_e) > \sum_{e\in\pi_2\setminus\pi} w_e \cdot (n_e + \lambda d_e + 1)$$

This contradicts to the fact that $\vec{n}$ is an edge NE and $m_\pi > 0$. Hence, M must be 0, which implies we can take $\kappa_{\vec{d}} = \kappa_1 = \kappa_2$. □

6.3.1 A System of Equations for Period Vectors

We now have two sets of linear equations that a period vector of a pNCG must satisfy, the first one being the set of *flow equations* for the arena, and the second one, with a parameter κ_x, is the one that we got from Theorem 6.10. We call the latter set of equations the *κ_x-parameterized cost equations*. Together, we have:

Corollary 6.11. *Any period vector $\vec{d} = (d_e)_{e\in E}$ of a pNCG on a series-parallel network game is a solution to the following system of linear equations:*

- $\forall v \in V \setminus \{s, t\} \sum_{e\in In(v)} x_e = \sum_{e\in Out(v)} x_e$
- $\exists \kappa_x \forall \pi \in \mathsf{Paths}_{\mathcal{G}} \sum_{e\in\pi} w_e.x_e = \kappa_x$

We refer to the above system of equations as $\mathcal{E}(\kappa_x)$ with $(x_e)_{e\in E}$, corresponding to each edge, and a parameter κ_x.

Note that, in $\mathcal{E}(\kappa_x)$ there are $|V| - 2$ many flow equations, $|\mathsf{Paths}_{\mathcal{G}}|$ many equations from Theorem 6.10, and $|E|$ many variables. If $|V| - 2 + |\mathsf{Paths}_{\mathcal{G}}| > |E|$, then the system of equations may have multiple solutions. Consider Example 6.12, where this is indeed the case.

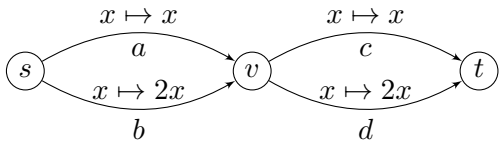


Figure 6.1: An arena $\mathcal{G}$ to show S is a set of covering paths

Example 6.12. *Consider the game $\mathcal{G}$ depicted in Figure 6.1. Here, the set of all paths* $\mathsf{Paths}\mathcal{G} = \{ac, bd, ad, bd\}$*, where in our notation, formally, ac is the path $s \xrightarrow{a} v \xrightarrow{c} t$, and so on. Moreover,* $|V| = 3,\ |E| = 4$*, which makes* $|V| - 2 + |\mathsf{Paths}\mathcal{G}| > |E|$. □

As we see in the above example, the κ_x-parameterized cost equations may not be optimal as a set, in the sense that there might be a smaller set of linear equations, which is a subset of κ_x-parameterized cost equations and the set of solutions of whose coincides with the same.

In the following, we compute such a minimal set, and then go on to prove that together with the flow equations, this set has a unique solution.

We define a set of *covering paths* for a pNCG $\mathcal{G}$. Intuitively, this is a subset of $\mathsf{Paths}\mathcal{G}$ such that for a given strategy profile $\vec{d}_{edge}$, if the cost across all the paths in this covering set of paths is constant, then so is across all paths from s to t. Formally,

Definition 6.13. *For pNCG $\mathcal{G}$, we call a set* $\mathsf{CovP}(\mathcal{G}) \subseteq \mathsf{Paths}\mathcal{G}$ *a set of* covering paths *of $\mathcal{G}$ if for any strategy profile $\vec{d} = (d_e)_{e \in E}$*

$$\forall \pi, \pi' \in \mathsf{CovP}(\mathcal{G}) \sum_{e \in \pi} w_e \cdot d_e = \sum_{e \in \pi'} w_e \cdot d_e \implies \forall \pi, \pi' \in \mathsf{Paths}\mathcal{G} \sum_{e \in \pi} w_e \cdot d_e = \sum_{e \in \pi'} w_e \cdot d_e \tag{6.5}$$

Note that, $\mathsf{Paths}\mathcal{G}$ itself is a set of covering paths for $\mathcal{G}$. But we are looking for a covering set of a specific size $|E| - |V| + 2$ for a given $\mathcal{G} = \langle V, E \rangle$, and that is what we compute next.

Algorithm to compute $\mathsf{CovP}(\mathcal{G})$ Intuitively, we choose a subset of all paths in such a way that cost of all the other paths in $\mathsf{Paths}\mathcal{G}$ is already determined by the paths that are already chosen in the set.

To construct such a set, for vertex $v \in V$, we first fix an ordering $\mathcal{O}$ among the outgoing edges $Out(v)$ from each vertex, and among its incoming edges $In(v)$ of each vertex. With respect to this ordering, for each v, we define two paths: a path from v to t, denoted by $\mathsf{fst\_fwd}(v)$, and a path from s to v, denoted by $\mathsf{fst\_bwd}(v)$.

Intuitively, we construct these two paths for $v \in V \setminus \{s, t\}$ in the following manner:

- $\mathsf{fst\_fwd}(v)$: We build the path step by step, adding an edge at each step from v, till we reach t. At the very first step, we start from vertex v, and consider the *first* edge, say $e_0 = (v, v_1)$ in $Out(v)$ with respect to ordering $\mathcal{O}$. If $v_1 = t$, then $\mathsf{fst\_fwd}(v) = v \xrightarrow{e_0} v_1$, otherwise

we extend $v \xrightarrow{e_0} v_1$ by considering rest of the from v_1 in following inductive way. At i^{th} step, say the currently built path is $\pi = v \xrightarrow{e_0} \dots \xrightarrow{e_{i-1}} v_i$. If $v_i = t$, then we declare π to be $\mathsf{fst\_fwd}(v)$, otherwise we add the first edge from $Out(v_i)$ with respect to $\mathcal{O}$ to π, and proceed to the next step.

- $\mathsf{fst\_bwd}(v)$*:* Similarly, we build this path step by step too, except here, at each step we build it in reverse order, starting from v, and terminating when we reach s.

Moreover, we define both $\mathsf{fst\_fwd}(t)$ and $\mathsf{fst\_bwd}(s)$ to be the empty path.

Example 6.14. *Consider again the running example here: the game $\mathcal{G}$ depicted in Figure 6.1. We take the ordering $\mathcal{O}$ on the edges as $a \prec b$ and $c \prec d$. This is applied to $In(u)$ and $Out(u)$ for all $u \in V$. Hence,* $\mathsf{fst\_fwd}(s) = \mathsf{fst\_bwd}(v) = a$*, and* $\mathsf{fst\_bwd}(t) = \mathsf{fst\_fwd}(v) = c$*.* □

With these two paths for each vertex v, we finally compute a set of covering paths as shown in Algorithm 2. Intuitively, for each vertex v with out-degree ≥ 2, and for each outgoing edge in $Out(v)$, say $v \xrightarrow{e} u$, we consider the path $\mathsf{fst\_bwd}(v) \cdot e \cdot \mathsf{fst\_fwd}(u)$, which is a path from s to t, and add it to the set of covering paths.

Algorithm 2 Computing a set of covering paths for arena $\mathcal{G}$

```
Initialize with S = ∅
for each vertex v with out-degree deg_v ≥ 2 do
    if v is the source vertex s then
        for 0 ≤ j ≤ deg_s − 1 do                          %Adding deg_s paths
            Add e_j · fst_fwd(u_j) to S                   %s --e_j--> u_j is the j^th edge in Out(s)
    else
        for 1 ≤ j ≤ deg_v − 1 do                          %Adding deg_v − 1-many paths
            Add fst_bwd(v) · e_j · fst_fwd(u_j) to S %v --e_j--> u_j is the j^th edge in Out(v)
return S
```

Lemma 6.15. *For a pNCG $\mathcal{G} = \langle V, E, s, t\rangle$, the set S computed in Algorithm 2 is a set of covering paths* $\mathsf{CovP}(\mathcal{G})$*, and it is of size $|E| - |V| + 2$.*

Proof. We first show that the set S indeed satisfies (6.5), i.e, we need to show that for any strategy profile if the cost across every path in S is the same in that profile, then the cost across all paths (including those that are not in S) is the same too.

We fix a strategy profile $\vec{d} = (d_e)_{e \in E}$, and we denote $\kappa_{\vec{d}} = \sum_{e \in \pi} w_e \cdot d_e = \sum_{e \in \pi'} w_e \cdot d_e$ for all $\pi, \pi' \in S$. Suppose towards a contradiction that (6.5) is not true, i.e, there is a path $\rho = s \xrightarrow{e_0} v_1 \xrightarrow{e_1} \dots t \in \mathsf{Paths}\mathcal{G}$ for which $\sum_{e \in \rho} w_e \cdot d_e \neq \kappa_{\vec{d}}$. We first consider $\sum_{e \in \rho} w_e \cdot d_e > \kappa_{\vec{d}}$, the other case is symmetric.

With this assumption, we show the following by induction: For all $0 \leq j \leq |\rho| - 1$, the path $\pi_j = e_j \cdot \mathsf{fst\_fwd}(v_{j+1})$, which is a path from v_j (of ρ) to t, satisfies

$$\sum_{j \leq i \leq |\rho|-1} w_{e_i} \cdot d_{e_i} > \sum_{e \in \pi_j} w_e \cdot d_e$$

Once we prove this, we can actually show that there are two paths π, π' in S, for which $\sum_{e \in \pi} w_e \cdot d_e \neq \sum_{e \in \pi'} w_e \cdot d_e$, which is a contradiction, hence proving that such a ρ, for which the cost is not same as that of $\kappa_{\vec{d}}$, cannot exist.

As $\mathsf{fst\_fwd}(v) = e \cdot \mathsf{fst\_fwd}(u)$, where $v \xrightarrow{e} u$ is the first edge in $Out(v)$, we have $\mathsf{fst\_bwd}(v).\mathsf{fst\_fwd}(v) \in S$ for any $v \in V$.

Now, base case of the induction is $j = 0$: we have $\pi_0 = e_0 \cdot \mathsf{fst\_fwd}(v_1)$. From our previous assumption, we have $\sum_{e \in \rho} w_e \cdot d_e > \kappa_{\vec{d}} = \sum_{e \in \pi_0} w_e \cdot d_e$.

As the induction hypothesis, let us suppose the statement is true for some $j < |\rho| - 1$. Then to show it is true for $j + 1$, we start with the induction hypothesis and proceed as follows:

$$\begin{aligned}
&\sum_{j \leq i \leq |\rho|-1} w_{e_i} \cdot d_{e_i} > \sum_{e \in \pi_j} w_e \cdot d_e \\
\implies &\sum_{j \leq i \leq |\rho|-1} w_{e_i} \cdot d_{e_i} - w_{e_j} \cdot d_{e_j} > \sum_{e \in \pi_j} w_e \cdot d_e - w_{e_j} \cdot d_{e_j} \\
\implies &\sum_{j+1 \leq i \leq |\rho|-1} w_{e_i} \cdot d_{e_i} > \sum_{e \in \mathsf{fst_fwd}(v_{j+1})} w_e \cdot d_e \\
&\qquad\qquad [\text{From the fact } \pi_j = e_j \cdot \mathsf{fst_fwd}(v_{j+1})]
\end{aligned}$$

Now adding an identical term to both sides, we get

$$\begin{aligned}
&\sum_{e \in \mathsf{fst_bwd}(v_{j+1})} w_e \cdot d_e + \sum_{j+1 \leq i \leq |\rho|-1} w_{e_i} \cdot d_{e_i} \\
&\qquad > \sum_{e \in \mathsf{fst_bwd}(v_{j+1})} w_e \cdot d_e + \sum_{e \in \mathsf{fst_fwd}(v_{j+1})} w_e \cdot d_e \\
&\qquad = \sum_{e \in \mathsf{fst_bwd}(v_{j+1}) \cdot \mathsf{fst_bwd}(v_{j+1})} w_e \cdot d_e = c_{\vec{x}} \qquad (*) \\
\implies &\sum_{e \in \mathsf{fst_bwd}(v_{j+1})} w_e \cdot d_e + \sum_{j+1 \leq i \leq |\rho|-1} w_{e_i} \cdot d_{e_i} > \sum_{e \in \mathsf{fst_bwd}(v_{j+1}).\pi_{j+1}} w_e \cdot d_e
\end{aligned}$$

which finally implies: $\sum_{j+1 \leq i \leq |\rho|-1} w_{e_i} \cdot d_{e_i} > \sum_{e \in \pi_{j+1}} w_e \cdot d_e$

This ends the proof of the induction.

Now consider $j = |\rho| - 1$. As $\pi_{|\rho|-1} = v_{|\rho|-1} \xrightarrow{e_{|\rho|-1}} t$, by the statement we just proved, we have $w_{e_{|\rho|-1}} \cdot d_{e_{|\rho|-1}} > w_{e_{|\rho|-1}} \cdot d_{e_{|\rho|-1}}$, which clearly is a contradiction.

Hence, S satisfies (6.5).

Finally we will show now that the size of S is $|E| - |V| + 2$. Note that in Algorithm 2, except for s, for each vertex v with degree ≥ 2, we add $deg(v) - 1$ many paths to S. And for s, we add $deg(s)$ many paths. Moreover, any path in the set S, barring the very first one that gets added in the procedure, has an unique identifying edge that is not shared by any other paths in the set. And the path, that gets added first in the set, is the only path of which every edge is the "first" outgoing edge (with respect to ordering $\mathcal{O}$) of the corresponding vertex. This means at each step the algorithm adds a new path in the set until it terminates. Therefore, the size of the set:

$$\begin{aligned}
|S| &= \sum_{v:deg(v)\geq 2} (deg(v) - 1) + 1 \\
&= \sum_{v\in V\setminus\{t\}} (deg(v) - 1) + 1 \qquad \text{[As } t \text{ is the only vertex with no outgoing edge]} \\
&= |E| - (|V| - 1) + 1 = |E| - |V| + 2
\end{aligned}$$

□

Example 6.16. *Let us consider the example in Figure 6.1. Here, the set of all paths* $\mathsf{Paths}\mathcal{G} = \{ac, bc, ad, bd\}$*, where the set computed by applying algorithm 2,* $S = \{ac, ad, bc\}$*. In fact, this is a minimal set satisfying (6.5), we consider the following sets* S_{-ac}, S_{-ad} *and* S_{-bc}*, where* $S_{-\pi} = S \setminus \{\pi\}$ *for* $\pi \in S$*. Now, we will show for each of these sets there exists a* $\vec{x} = (x_e)_{e\in E}$ *for which* $\forall \pi_1, \pi_2 \in S_{-\pi} \sum_{e\in\pi_1} w_e x_e = \sum_{e\in\pi_2} w_e x_e$ *holds but* $\forall \pi_1, \pi_2 \in \mathcal{P}_{\mathcal{A}} \sum_{e\in\pi_1} w_e x_e = \sum_{e\in\pi_2} w_e x_e$ *doesn't hold. Hence, none of them can be a set of covering paths, proving the minimality of* S*.*

For this particular example, we write a candidate for period vector as $\vec{x} = (x_a, x_b, x_c, x_d)$*. Now, for* S_{-ac}*,* $\vec{x} = (3, 3, 1, 2)$ *works as* $2.3 + 1 = 3 + 2.2$ *but for the path* ac*, the value is* 4*. Similarly, it is easy to check that for* S_{-bd} *and* S_{-bc}*,* $(2, 1, 3, 0)$ *and* $(3, 0, 1, 2)$ *works for supporting example which makes that no* $S_{-\pi}$ *is a set of covering paths for* $\pi \in S$*.* □

Remark 6.17. *In the following, in essence we are going to show that this minimality holds for all graphs. We say this* in essence *because this is not the result we seek, rather our larger goal is to obtain period vectors for the set of NE. As a mandatory byproduct, we can see that the covering set of paths computed in Algorithm 2 is, in fact, a minimal set of covering paths.*

"Small" system of linear equations. With this set of covering paths $\mathsf{CovP}(\mathcal{G})$ of size $|E| - |V| + 2$, we now have a reduced system of parameterized linear equations with parameter κ_x, which a period vector of arena $\mathcal{G}$ satisfies:

- $\forall v \in V \setminus \{s, t\} \sum_{e\in In(v)} x_e = \sum_{e\in Out(v)} x_e$
- $\exists \kappa_x \, \forall \pi \in \mathsf{CovP}(\mathcal{G}) \sum_{e\in\pi} w_e.x_e = \kappa_x$

We refer to this system of equations as $\mathcal{E}^{\mathsf{CovP}}(\kappa_x)$. Here, we have $|V| - 2$ flow equations as before, and $|E| - |V| + 2$ κ_x-parameterized cost equations. This constitutes a system of $|E|$ linear equations over $|E|$ unknowns and a parameter κ_x.

We can also express this system of parameterized linear equations as $M_{\mathcal{G}} \cdot X = b$, where $M_{\mathcal{G}}$ is an $|E| \times |E|$ matrix, which we refer to as the *period-generator* of $\mathcal{G}$, with entries only from $\{w_e : e \in E\} \cup \{0, -1, 1\}$, b is a column matrix with $|E|$ rows, in which the first $|V|$ entries are 0 and rest of the entries are κ_x, and each of the $|E|$ entries of X is the unknown variable x_e from the linear equations, corresponding to each edge of E.

Each column of $M_{\mathcal{G}}$ corresponds to an edge $e \in E$, the first $|V| - 2$ rows of $M_{\mathcal{G}}$ correspond to each vertex $v \in V \setminus \{s, t\}$, and each of the rest of $|E| - |V| + 2$ rows correspond to each of the paths in $\mathsf{CovP}(\mathcal{G})$. Therefore, each entry of $M_{\mathcal{G}}$ can be referred to be either as the $(v, e)^{th}$ entry for each $v \in V \setminus \{s, t\}$ and $e \in E$, or as the $(\pi, e)^{th}$ entry for each $\pi \in \mathsf{CovP}(\mathcal{G})$ and $e \in E$. To be specific, here we explain how exactly $M_{\mathcal{G}}$ looks like:

- Let us consider $m_{v,e}$ to be the $(v, e)^{th}$ entry, then

$$m_{v,e} = \begin{cases} 1 & \text{if } e \in In(v) \\ -1 & \text{if } e \in Out(v) \\ 0 & \text{otherwise} \end{cases} \tag{6.6}$$

  We denote the v^{th} row of $M_{\mathcal{G}}$ by $\vec{m}_v$.

- And for the $(\pi, e)^{th}$ entry, denoted by $m_{\pi,e}$, we have

$$m_{\pi,e} = \begin{cases} w_e & \text{if } e \in \pi \\ 0 & \text{otherwise} \end{cases} \tag{6.7}$$

  We denote the π^{th} row of $M_{\mathcal{G}}$ by $\vec{m}_\pi$ for $\pi \in \mathsf{CovP}(\mathcal{G})$.

Theorem 6.18. *For any network $\mathcal{G} = \langle V, E, s, t\rangle$, the $|E| \times |E|$ period-generator $M_{\mathcal{G}}$ is invertible.*

Proof. Suppose there exists an $\vec{f}$ satisfying $M_{\mathcal{G}} \cdot \vec{f} = \vec{0}$. Then there are exactly three possibilities regarding $\vec{f}$:

- either $\vec{f} \geq \vec{0}$, i.e, for all edges $e \in E$, $f_e \geq 0$, or
- $\vec{f} \leq \vec{0}$, i.e, for all edges $e \in E$, $f_e \leq 0$, or
- there exists $e_i, e_j \in \mathsf{CovP}(\mathcal{G})$ such that $f_{e_i} > 0$ and $f_{e_j} < 0$.

In all three cases, we will reach a contradiction if $\vec{f} \neq 0$, henceforth proving that $M_{\mathcal{G}} \cdot \vec{f} = 0$ implies $\vec{f} = 0$.

Case 1: $\vec{f} \geq 0$.

Suppose towards contradiction that $\vec{f} \neq \vec{0}$, and there is an edge e such that $f_e > 0$. Let us also assume $\pi \in \mathsf{CovP}(\mathcal{G})$ be a path of which e is an edge, and consider the row-matrix $\vec{m}_\pi$ as it was defined in (6.7).

Now, $M_{\mathcal{G}} \cdot \vec{f} = 0$ implies $\vec{m}_\pi \cdot \vec{f} = \sum_{e \in \pi} w_e \cdot f_e = 0$.

Because for all edges e', $w_{e'} > 0$, and $f_{e'} \geq 0$, $f_e > 0$ makes $\vec{\rho_\pi}.\vec{f} = \sum_{e \in \pi} w_e \cdot f_e$ greater than 0 too. This contradicts with the hypothesis $M_\mathcal{G} \cdot \vec{f} = 0$. Therefore, there cannot be such an edge e with $f_e > 0$, which means $\vec{f} \geq \vec{0}$ implies $\vec{f} = \vec{0}$.

Case 2: $\vec{f} \leq \vec{0}$.

This case is symmetric to **Case 1**, which means in this case also $\vec{f}$ has to be $\vec{0}$.

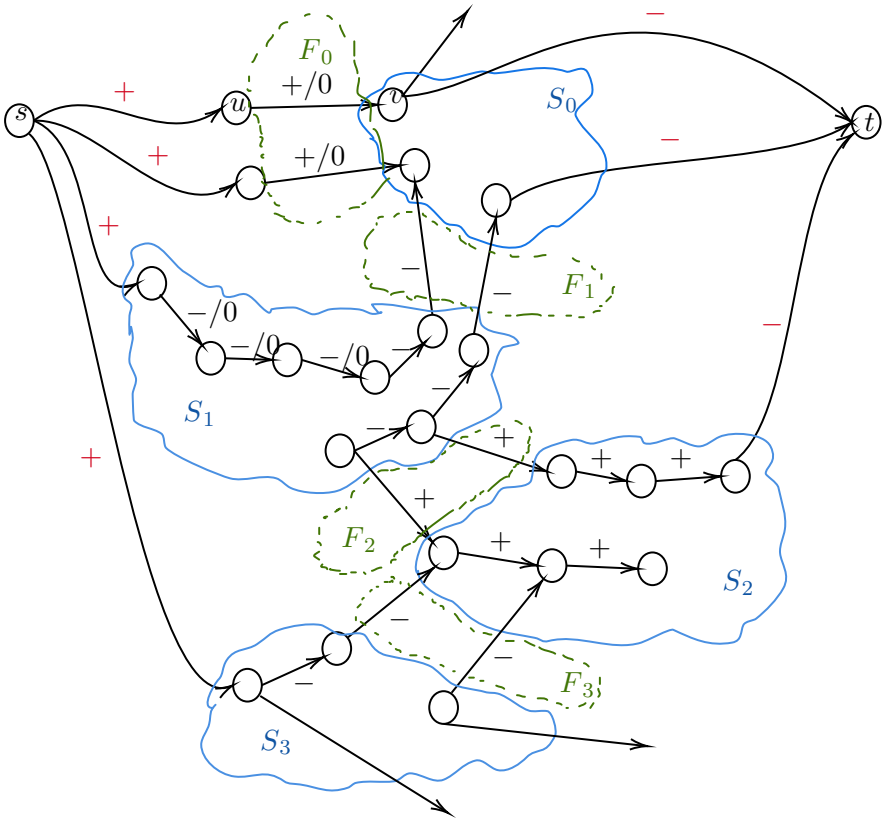


Figure 6.2: In an arbitrary graph, how S_m's and F_m's would have been if **Case 3** were true. Here, up to level 3 is shown, curved-arrows denote positive and negative paths with label $+$ and $-$ respectively, and straight-arrows denote edges, S_m's are the set of vertices in the enclosed area, and F_m's are the sum of absolute values in the area enclosed by dotted green lines

Case 3: $\exists e_i, e_j \in E : (f_{e_i} > 0) \wedge (f_{e_j} < 0)$.

Before going into details, we first briefly summarize the main steps of the whole argument by which we proceed with this case:

Step I: We argue that there is a path π with $e_i, e_l \in \pi$ such that $f_{e_i} > 0$ and $f_{e_l} < 0$.

Step II: We define the notions of *positive*, *negative* and *zero* paths, and argue why all paths are, in fact, zero paths from $M_{\mathcal{G}} \cdot \vec{f} = \vec{0}$.

Step III: We show the existence of either a positive or a negative path. To prove this, we build a sequence of sets of vertices along with an increasing sequence of integers which are bounded above, thus making it impossible to grow the sequence infinitely. But by design, the sequence only stops if there is either a positive or a negative path.

Now, the existence of either path contradicts with Step II, thus concluding that the hypothesis on the Case 3 itself is wrong.

STEP I: We can easily conclude from **Case 1** and **2** that, there cannot be a path $\pi \in \mathsf{Paths}\mathcal{G}$ such that there is an edge $e \in \pi$ with $f_e > 0$ (or $f_e < 0$ resp.), and for all other paths e' of π, $f_{e'} \geq 0$ (or $f_{e'} \leq 0$ resp).

Therefore, for any $\pi \in \mathsf{CovP}(\mathcal{G})$, either for all edges $e \in \pi$, $f_e = 0$, or there are edges $e, e' \in \pi$ such that $f_e > 0$ and $f_{e'} < 0$. If indeed for all paths π, for all edges $e \in \pi$, $f_e = 0$, we have nothing to show further, Case 3 stops here. Hence, we can assume there is a path π where the latter condition holds. We consider this path $\pi \in \mathsf{CovP}(\mathcal{G})$ in the subsequent part of the discussion.

STEP II: Here, we define some notions as follows: for any path $v \overset{\sigma}{\rightsquigarrow} v'$, we call σ *positive*, *negative*, and *zero* respectively if $\sum_{e\in\sigma} w_e \cdot f_e > 0$, $\sum_{e\in\sigma} w_e \cdot f_e < 0$, and $\sum_{e\in\sigma} w_e \cdot f_e = 0$ respectively. An edge is just a path of length 1, hence an edge e is called positive (resp. negative and zero) if $f_e > 0$ (resp. $f_e < 0$ and $f_e = 0$).

Because $M_{\mathcal{G}} \cdot \vec{f} = 0$ implies for all $\pi \in \mathsf{CovP}(\mathcal{G})$, $\vec{m}_\pi \cdot \vec{f} = \sum_{e\in\pi} w_e \cdot f_e = 0$, in terms of this notions, all paths $\sigma \in \mathsf{CovP}(\mathcal{G})$ are zero paths. Moreover, from the definition of covering paths of a network, in fact, all paths $\sigma \in \mathsf{Paths}\mathcal{G}$ are zero paths.

STEP III: Now let's get back to π considered in the first step. Without loss of generality, we assume that the first *non-zero* (i.e, positive or negative) edge of π is positive.

Recall from the plan, here, our objective is to show that there exists either a positive or negative path.

We construct an infinite sequence of non-empty sets of vertices $\langle S_l \rangle_{l\geq 0}$, along with another increasing sequence of non-negative integers $\langle F_l \rangle_{l\geq 0}$. Moreover, we define the latter sequence in such a way that each F_l is bounded above .

Hence, there is a contradiction because using only finitely many numbers it is not possible to form an infinite increasing sequence!

We start with the construction of the sequences $\langle S_l \rangle_{l\geq 0}$ and $\langle F_l \rangle_{l\geq 0}$. In Lemma 6.19, we show that S_l's are non-empty, which is crucial to justify F_l's are indeed well-defined. In

Lemma 6.20, we prove that the sequence $\langle F_l \rangle_{l \geq 0}$ are indeed increasing. We finally conclude giving the bound for F_l's, and with the argument how the contradiction comes in place.

Now, we start defining the sequence of sets of vertices as follows:

$$S_0 = \{v \in V : \exists \text{ a path } \sigma : s \overset{\sigma}{\rightsquigarrow} v \text{ such that } (\forall e \in \sigma \ f_e \geq 0) \text{ and } (\forall e \in Out(v) \ , f_e < 0)\}$$

Intuitively, S_0 is the set of vertices to which there is a path σ from s satisfying (a) all the edges in σ are positive or zero, and (b) all the edges outgoing from v are strictly negative.

First, we claim that S_0 is non-empty. We assumed the first non-zero edge of π to be positive. Let $u \xrightarrow{e} v$ be that edge, then if every outgoing edge of v is negative, then v is in S. Otherwise, there is an edge $v \xrightarrow{e'} v'$ in $Out(v)$, which is either positive or zero. This v' might be a candidate of S_0, if all the outgoing edges of v' are negative. If not, we take a non-negative outgoing edge of v', and continue. In this way, if we reach t before finding a v in S_0, then we have a positive path from s to t, a contradiction! Hence, S_0 must be non-empty.

And we define,

$$F_0 = \sum_{\substack{u \xrightarrow{e} v \in E \\ v \in S_0}} f_e$$

Here, we take the sum over all the f_e-values of all edges that are incoming to vertices of S_0. As by definition, all the incoming f_e's are positive or zero for S_0, we have $F_0 \geq 0$.

The sequence starts from S_0, then for each $m \geq 0$, we inductively define S_{2m+1} from S_{2m} and S_{2m+2} from S_{2m} as follows:

$$S_{2m+1} = \{v \in V : \exists v' \in S_{2m}, \exists \text{ a path } v \overset{\sigma}{\rightsquigarrow} v' \text{ such that for all } e \in \sigma, f_e \leq 0\}$$
$$S_{2m+2} = \{v \in V : \exists v' \in S_{2m+1}, \exists \text{ a path } v' \overset{\sigma}{\rightsquigarrow} v \text{ such that for all } e \in \sigma, f_e \geq 0\}$$

Moreover, we define:

$$F_{2m+1} = \sum_{\substack{u \xrightarrow{e} v \in E \\ u \in S_{2m+1}, v \in S_{2m}}} |f_e| \qquad \text{and} \qquad F_{2m+2} = \sum_{\substack{u \xrightarrow{e} v \in E \\ u \in S_{2m+1}, v \in S_{2m+2}}} f_e$$

Intuitively, here as F_{2m+1}, we take the sum of all absolute values of f_e over the set of incoming edges from S_{2m} to S_{2m+1}, while as F_{2m+2}, we take the sum of all absolute values of f_e over the set of all outgoing edges from S_{2m+1} to S_{2m+1}. Notice that in Figure 6.2, that the first set of f_e values are negative, while the latter set of f_e values are positive.

To show that the sequence $\langle F_l \rangle_{l \geq 0}$ is well-defined, we need to show that S_l is non-empty for all $l \geq 0$. Earlier, we established that S_0 is non-empty. For the rest, consider the following lemma:

Lemma 6.19. *For $l \geq 0$, S_l is non-empty.*

Proof. By induction on l, we, in fact, prove that: S_l is non-empty, and for every vertex v of S_l, there is a positive path from s to v, and a negative path from v to t.

First consider $l = 0$. We already observed that S_0 is non-empty. Moreover, by definition of S_0, for any vertex $v \in S_0$, there is a positive path σ^+_{sv} from s to v. Now, because for any path $\pi^c \in \mathsf{CovP}(\mathcal{G})$, $\vec{\rho_{\pi^c}} \cdot \vec{f} = \sum_{e \in \pi^c} w_e \cdot f_e = 0$, we have, in fact, for any path $\pi \in \mathsf{Paths}\mathcal{G}$:

$$\sum_{e \in \pi} w_e \cdot f_e = 0$$

Again, consider the positive path σ^+_{sv} from s to v, and let π be the path from s to t, via v such that $\pi = \sigma^+_{sv} \cdot \sigma'$ for some $v \overset{\sigma}{\rightsquigarrow} t$. Then, from the above identity, we have

$$\sum_{e \in \pi} w_e \cdot f_e = \sum_{e \in \sigma^+_{sv}} w_e \cdot f_e + \sum_{e \in \sigma'} w_e \cdot f_e = 0$$

Hence, σ', which is a path from v to t, is a negative path. Therefore, the base case of the induction is done.

For the inductive step, we assume that the statement is true for $l = 2m$, i.e, S_{2m} is non-empty, and for every vertex v of S_{2m}, there is a positive path from $s\sigma^+_{sv}v$, and a negative path from $v\sigma^-_{vt}t$.

Now, consider $l = 2m + 1$. From the inductive hypothesis, we have that S_{2m} is non-empty, however, we additionally claim that there is a vertex v in S_{2m} such that for all outgoing edges e of $Out(v)$, $f_e < 0$. Suppose this is not the case. Then, starting from any v of S_{2m}, we can construct a path σ till t such that for all edges $e \in \sigma$, $f_e \geq 0$. But that constitutes a positive path from s to t itself, which is a contradiction! Therefore, our claim is true.

Now, consider such a vertex v from S_{2m}, for which every outgoing edge e is negative, which implies $\sum_{e \in Out(v)} f_e < 0$.

Because there is a positive path from s to every vertex of S_{2m}, and a negative path from every vertex to t, s and t themselves cannot be in S_{2m}, i.e, the chosen v is in $V \setminus \{s, t\}$. Hence, we can consider the row-vector $\vec{\rho_v}$ of $M_\mathcal{G}$ corresponding to vertex v. Now, from $\vec{\rho_v} \cdot \vec{f} = 0$, we have

$$\sum_{e \in In(v)} f_e = \sum_{e \in Out(v)} f_e$$

which is negative. Hence, there has to be an edge $u \xrightarrow{e} v$ for which $f_e < 0$, which implies $u \in S_{2m+1}$. That is S_{2m+1} is non-empty.

Moreover, for every vertex u of S_{2m+1}, by definition, there is a vertex v in S_{2m} such that there is a negative path $u \overset{\sigma^-_{uv}}{\rightsquigarrow} v$. Therefore, adjoining σ^-_{uv} to the negative path $v \overset{\sigma^-_{vt}}{\rightsquigarrow} t$, we

get a negative path, say $\sigma^-_{ut} = \sigma^-_{uv} \cdot \sigma^-_{vt}$, form u to t. Now, to show there is a positive path from s to u, we proceed similarly as we did when we ought to show that for every vertex v of S_0 has a negative path to t, except here is a change of sign. Consider a path $\pi \in \mathsf{Paths}\mathcal{G}$ via u such that $\pi = \sigma \cdot \sigma^-_{ut}$ for some σ. Then we have

$$\sum_{e\in\pi} w_e \cdot f_e = \sum_{e\in\sigma} w_e \cdot f_e + \sum_{e\in\sigma^-_{ut}} w_e \cdot f_e = 0$$

Hence, σ is a positive path indeed. This concludes proving the statement for $l = 2m + 1$.

Finally, we need to show the same for $l = 2m + 2$, where we assume the statement is true for $l = 2m + 1$. Again we start with the argument that there is a vertex v in S_{2m+1} such that every incoming edge of v is positive. If not, then we can back track from v via non-negative edges till we reach s, and this implies existence of a negative path form s to t, hence a contradiction! Therefore, indeed there is a vertex v in S_{2m+1} such that every edges of $In(v)$ is positive.

Now, fix this vertex v from S_{2m+1}, and we have $\sum_{e\in In(v)} f_e > 0$. Again, we have $\sum_{e\in In(v)} f_e = \sum_{e\in Out(v)} f_e$, from which we get there must be an edge $v \xrightarrow{e} u \in Out(v)$ such that $f_e > 0$. Hence, S_{2m+2} is non-empty.

For a vertex u in S_{2m+2}, there is a vertex v in S_{2m+1}, therefore we get a positive path σ^+_{su} by simply adjoining the positive paths σ^+_{sv} and σ^+_{vu}. For the negative path, our argument is exactly similar as it was in case of a vertex of S_0. We consider a path π from s to t, that extends σ^+_{su} by, say $v \overset{\sigma}{\rightsquigarrow} t$. Therefore, $\sum_{e\in\pi} w_e \cdot f_e = \sum_{e\in\sigma^+_{su}} w_e \cdot f_e + \sum_{e\in\sigma} w_e \cdot f_e = 0$. Hence, σ, which is a path from u to t, is a negative path. □

As S_l is non-empty for all $l \geq 0$, we establish that F_l is well-defined for all $l \geq 0$. In the following, we show that the sequence $\langle F_l \rangle_{l\geq 0}$ is, in fact, an increasing sequence.

Lemma 6.20. *For any $l \geq 0$, $F_{l+1} > F_l$.*

Proof. We prove this by induction on l, the approach is quite similar to how we proved Lemma 6.19.

First note that, from $M_{\mathcal{G}} \cdot \vec{f} = \vec{0}$, we have for any vertex v, $\sum_{e\in In(v)} f_e = \sum_{e\in Out(v)} f_e$, which we can also write in terms of absolute value of f_e's as:

$$\sum_{\substack{e\in In(v)\\ f_e>0}} |f_e| + \sum_{\substack{e\in Out(v)\\ f_e<0}} |f_e| = \sum_{\substack{e\in Out(v)\\ f_e>0}} |f_e| + \sum_{\substack{e\in In(v)\\ f_e<0}} |f_e| \tag{6.8}$$

As this is true for any vertex v, it holds if we take a sum over any collection of vertices of V. In particular, for all $l \geq 0$,

$$\sum_{v \in S_l} \left\{ \sum_{\substack{e \in In(v) \\ f_e > 0}} |f_e| + \sum_{\substack{e \in Out(v) \\ f_e < 0}} |f_e| \right\} = \sum_{v \in S_l} \left\{ \sum_{\substack{e \in Out(v) \\ f_e > 0}} |f_e| + \sum_{\substack{e \in In(v) \\ f_e < 0}} |f_e| \right\}$$

$$i.e, \sum_{\substack{e \in In(v), f_e > 0 \\ v \in S_l}} |f_e| + \sum_{\substack{e \in Out(v), f_e < 0 \\ v \in S_l}} |f_e| = \sum_{\substack{e \in Out(v), f_e > 0 \\ v \in S_l}} |f_e| + \sum_{\substack{e \in In(v), f_e < 0 \\ v \in S_l}} |f_e| \tag{6.9}$$

Now, for $l = 0$, in the left hand side (L.H.S) of identity (6.9), the first term is F_0 and the second term is strictly positive (as for all vertices in S_0, all outgoing edges are negative). Moreover, in the right hand side (R.H.S), the first term is 0 (because there is no positive outgoing edge), and the second term is F_1. Therefore, we have

$$F_0 + M = 0 + F_1$$

where $M = \sum_{\substack{e \in Out(v), f_e < 0 \\ v \in S_0}} |f_e| > 0$. Hence, $F_1 > F_0$. Therefore, the base case of the induction is done.

As the inductive hypothesis, we assume that the statement is true for $l = 2m$, which means: $F_{2m+1} > F_{2m}$.

For $l = 2m + 1$, we consider (6.9) with substituting l by $2m + 1$, by which we get

$$\sum_{\substack{e \in In(v), f_e > 0 \\ v \in S_{2m+1}}} |f_e| + \sum_{\substack{e \in Out(v), f_e < 0 \\ v \in S_{2m+1}}} |f_e| = \sum_{\substack{e \in Out(v), f_e > 0 \\ v \in S_{2m+1}}} |f_e| + \sum_{\substack{e \in In(v), f_e < 0 \\ v \in S_{2m+1}}} |f_e|$$

Here, the first term in the L.H.S is positive from the fact that there is a vertex v in S_{2m+1} for which all the edges in $In(v)$ is positive. We argued this inside the proof of Lemma 6.19, the crux being otherwise we would have a negative path from s to t, which, in turn, would give us a contradiction!

The second term of the L.H.S is at least F_{2m+1} because the term here is the sum of all the $|f_e|$'s of all negative outgoing edges of all vertices in S_{2m+1}, while F_{2m+1} is just the sum over a particular subset of those edges.

The first term of the R.H.S is exactly F_{2m+2} because for $v \in S_{2m+1}$, for any edge $v \xrightarrow{e} u$ with $f_e > 0$, we put u in S_{2m+2}.

Finally, for the second term of R.H.S, note that for any $v \in S_{2m+1}$, for any edge $u \xrightarrow{e} v \in In(v)$, if $f_e < 0$, then u also belongs to S_{2m+1}. Therefore, that particular $|f_e|$ also appears as a contributing factor when we consider outgoing edges $Out(u)$ of u. That means this

quantity gets completely subsumed by the second term of the L.H.S. More specifically, we can rewrite the above as:

$$P + (F_{2m+1} + Q + R) = F_{2m+2} + Q$$

where

$$P = \sum_{\substack{e \in In(v), f_e > 0 \\ v \in S_{2m+1}}} |f_e| > 0$$

$$Q = \sum_{\substack{e \in In(v), f_e < 0 \\ v \in S_{2m+1}}} |f_e| = \sum_{\substack{v \xrightarrow{e} v' \in Out(v), f_e < 0 \\ v \in S_{2m+1}, v' \in S_{2m+1}}} |f_e|$$

$$R = \sum_{\substack{v \xrightarrow{e} v' \in Out(v), f_e < 0 \\ v \in S_{2m+1}, v' \notin S_{2m+1}}} |f_e|$$

Clearly, $F_{2m+2} > F_{2m+1}$.

The remaining case is when $l = 2m + 2$. Again we consider (6.9) with substituting l by $2m + 2$:

$$\sum_{\substack{e \in In(v), f_e > 0 \\ v \in S_{2m+2}}} |f_e| + \sum_{\substack{e \in Out(v), f_e < 0 \\ v \in S_{2m+2}}} |f_e| = \sum_{\substack{e \in Out(v), f_e > 0 \\ v \in S_{2m+2}}} |f_e| + \sum_{\substack{e \in In(v), f_e < 0 \\ v \in S_{2m+2}}} |f_e|$$

This case is similar to S_0. Here the first term of the L.H.S is at least F_{2m+2}, the second term is positive just like it has been argued in case of S_0. On the other hand, the first term of the R.H.S takes care of the internal edges of S_{2m+2}, and hence the whole terms gets subsumed by the first term of the L.H.S. Finally, the second term of the R.H.S is exactly F_{2m+3} because for any vertex v of S_{2m+2}, by definition for any negative incoming edge $u \xrightarrow{e} v$, e is in S_{2m+3}. Hence, here we can rewrite the above as:

$$(F_{2m+2} + Q' + R') + P' = Q' + F_{2m+3}$$

where

$$Q' = \sum_{\substack{e \in Out(v), f_e > 0 \\ v \in S_{2m+2}}} |f_e| = \sum_{\substack{v \xrightarrow{e} v' \in In(v), f_e < 0 \\ v \in S_{2m+2}, v' \in S_{2m+2}}} |f_e|$$

$$P' = \sum_{\substack{e \in Out(v), f_e < 0 \\ v \in S_{2m+2}}} |f_e| > 0$$

$$R' = \sum_{\substack{v \xrightarrow{e} v' \in In(v), f_e < 0 \\ v \in S_{2m+2}, v' \notin S_{2m+2}}} |f_e|$$

Therefore, $F_{2m+3} > F_{2m+2}$. □

As we mentioned briefly earlier, F_l for each $l \geq 0$ belongs to a finite set. In particular, it is easy to see that, $F_l \in \{ \sum_{e \in E'} |f_e| : E' \subseteq E\}$, which is a finite set. Therefore, an infinitely-increasing sequence $\langle F_l \rangle_{l \geq 0}$ cannot be possible.

Hence, our initial assumption that there exist edges e_i, e_j with $f_{e_i} > 0$ and $f_{e_j} < 0$ is wrong.

So, we can conclude from the analyses of the above three cases, whenever $M_{\mathcal{G}} \cdot \vec{f} = \vec{0}$, $\vec{f}$ must also be $\vec{0}$. In other words, $M_{\mathcal{G}}$ is invertible. □

Co-linearity of the period vectors. Here, we will show that all the period vectors for a $\mathfrak{N}_{\mathcal{G}}$ are co-linear.

As $M_{\mathcal{G}}$ is invertible, we have a unique parameterized solution for $M_{\mathcal{G}} \cdot X = b$, and that is:

$$X = M_{\mathcal{G}}^{-1} \begin{pmatrix} \vec{0} \\ \vec{\kappa_x} \end{pmatrix}$$

That is two period vectors differ from each other due to their corresponding value taken for the parameter κ_x.

Suppose $\vec{d}$ and $\vec{d'}$ are the two period vectors, and $c_{\vec{d}}$ and $c_{\vec{d'}}$ are the values taken for the parameter respectively. We have $M_{\mathcal{G}}(c_{\vec{d'}}.\vec{d} - c_{\vec{d}}.\vec{d'}) = \vec{0}$. Moreover, because $M_{\mathcal{G}}$ is invertible, we have $\vec{d'} = \frac{c_{\vec{d}}}{c_{\vec{d'}}}\vec{d}$. And this is true for any two period vectors $\vec{d}, \vec{d'}$.

Now fix a period vector $\vec{d}$ and consider $g = \underset{e \in E}{gcd}\, d_e$. Let us define $\vec{\delta} = (\delta_e)_{e \in E}$ as $\delta_e = \frac{d_e}{g}\ \forall e \in E$. Then,

Lemma 6.21. *For any period vector $\vec{d'}$ of $\mathcal{G}$, there exists $m_{\vec{d'}} \in \mathbb{N}$ such that $\vec{d'} = m_{\vec{d'}}\vec{\delta}$*

Proof. Consider an arbitrary period vector $\vec{d'}$. Remember, we have already fixed a $\vec{d}$ from which we defined $\vec{\delta}$. Moreover, we had $\vec{d'} = \frac{c_{\vec{d'}}}{c_{\vec{d}}}.\vec{d}$, which we can write as $\vec{d'} = \frac{p}{q}.g.\vec{\delta}$, where $gcd(p, q) = 1$. Also note that $\underset{e \in E}{gcd}\, \delta_e = 1$.

Now, if we can show $q|g$, we are done. Let us define $g' = gcd(g, q)$, and take $\frac{m}{n} = \frac{g/g'}{q/g'}$, which implies $gcd(m, n) = 1$. From this, we can further write $\vec{d'} = \frac{p.m}{n}\vec{\delta}$ where $gcd(p.m, n) = 1$.

Suppose $n \neq 1$. In that case, from the fact $\vec{d'_e} \in \mathbb{N}$ for all $e \in E$ (by design), we have $n|\delta_e$ for all $e \in E$, which essentially means $\underset{e \in E}{gcd}\, \delta_e \geq n > 1$. Contradiction! That means n must be 1, and we are through. □

Therefore we have:

$$\mathfrak{N}_{\mathcal{G}} = \bigcup_{i \in I} L(b_i, \{m_{ij} \cdot \vec{\delta} : 1 \leq j \leq |P_i|, m_{ij} \in \mathbb{N}\}) \tag{6.10}$$

Here, $Base = \{b_i : i \in I\}$ is the finite set of basis for some index set I, with $|I| < \infty$. Note that, for any semi-linear set, the choice of basis and the set of period vectors may not necessarily be unique. As of yet, we have established a property which every period vectors necessarily satisfies.

6.4 Discussion

From (6.10), we have $m \cdot \vec{\delta}$ for some $m \in \mathbb{N}$ as a period vector for some base vector $\vec{b}$. Here, $\vec{b}$ itself is an edge NE which means there is a path NE corresponding to it. And for any path-strategy profile corresponding to $m \cdot \vec{\delta}$, we know that the costs across all paths are identical. With these, and using the series/parallel composition, we thus formulate the following conjecture:

Conjecture 6.22. *For series-parallel graph $\mathcal{G}$, $\vec{\delta}$ is a period vector for all the bases of $\mathfrak{N}_{\mathcal{G}}^{edge}$.*

Although, we do not have a proof for the above conjecture for the moment, which understandably makes the rest of the part incomplete. Therefore, for the rest, we show what the consequences are if it is indeed true.

First of all, then we can write:

$$\mathfrak{N}_{\mathcal{G}} = \bigcup_{i \in I} L(b_i; \{\vec{\delta}\}) \tag{6.11}$$

where $\vec{\delta} = M_{\mathcal{G}}^{-1} \begin{pmatrix} \vec{0} \\ \vec{c}_{\vec{\delta}} \end{pmatrix}$ in which $c_{\vec{\delta}}$ is the smallest integer such that $M_{\mathcal{G}}^{-1} \cdot \begin{pmatrix} \vec{0} \\ \vec{c}_{\vec{\delta}} \end{pmatrix}$ is a column matrix of natural numbers.

Definition 6.23. *For series-parallel graph $\mathcal{G}$, $\vec{\delta} = M_{\mathcal{G}}^{-1} \begin{pmatrix} \vec{0} \\ \vec{c}_{\vec{\delta}} \end{pmatrix}$ is called the* shift vector *for NE of $\mathcal{G}$.*

Note that, the sets of bases in (6.10) and (6.11) may not necessarily be identical. In the next section, we embark upon to compute the set of bases in the latter case, to draw a complete picture for the set of NE of an pNCG in a series-parallel graph (albeit the proof of conjecture).

6.4.1 Basis for the Set of all Edge NE profiles

Recall Problem 6.4 at the beginning of this Chapter. Our objective was to compute Nash equilibria for sufficiently large number of k. In fact, we will represent the set of all NE, $\mathfrak{N}_{\mathcal{G}}$, by a finite basis, and the sole period vector, from which the solution of the above problem comes as an easy consequence.

First, we show that if one component of an NE profile of a series-parallel network $\mathcal{G}$ is bounded, then so are all other components. In other words:

Lemma 6.24. *The set $BoundNE(e, M) = \{\vec{n} = (n_e)_{e \in E} \in \mathfrak{N}_{\mathcal{G}} : n_e < M\}$ is finite when $\mathcal{G}$ is a series-parallel network.*

Proof. In order to prove this, it is enough to show that for every edge $e' \in E$, there exists a function $\mathtt{f}_{e,e'} : \mathbb{N} \to \mathbb{N}$ such that for all NE $\vec{n}$ with $n_e < M$, $n_{e'} < \mathtt{f}_{e,e'}(M)$.

We prove the above by structural induction on $\mathcal{G}$. The base case of this induction is when $\mathcal{G}$ is a single parallel network. Now: $w_{e'} \cdot n_{e'} \leq w_e \cdot (n_e + 1)$

$$\text{Thus, } \mathtt{f}_{e,e'}(M) = \frac{w_e}{w_{e'}}(M+1) \tag{6.12}$$

Series Composition. Suppose $\mathcal{G}_1$ and $\mathcal{G}_2$ are two series-parallel networks composed in series at vertex v, yielding a network $\mathcal{G}$. We also suppose that e is an edge of $\mathcal{G}_1$, and e' is an edge of $\mathcal{G}_2$. As an induction hypothesis, we assume that for every edge e'' of $\mathcal{G}$, there is a $\mathtt{f}_{e,e''}$. Then from flow equations, a crude upper-bound for $n_{e'}$ can be given as:

$$\mathtt{f}_{e,e'}(M) = \sum_{e'' \in In(v)} \mathtt{f}_{e,e''}(M)$$

Parallel Composition Suppose $\mathcal{G}_1$ and $\mathcal{G}_2$ are two series-parallel networks composed in parallel, yielding a network $\mathcal{G}$. We also suppose that e is an edge of $\mathcal{G}_1$ and e' is an edge of $\mathcal{G}_2$. As an induction hypothesis, we assume that for every edge e'' of $\mathcal{G}_1$, there is a $\mathtt{f}_{e,e''}$, then

$$\sum_{e \in E_2} w_e \cdot n_e \leq \sum_{e \in E_1} w_e \cdot (n_e + 1)$$

$$\text{Then, } \mathtt{f}_{e,e'}(M) = \frac{\max\limits_{e'' \in E_1} w_{e''} \mathtt{f}_{e,e''}(M)}{\min\limits_{\hat{e} \in E_2}}$$

Therefore, for all pair of edges e, e' in a series-parallel network $\mathtt{f}_{e,e'}$ exists, hence $BoundNE(e, M)$ is a finite set. □

Hence, we can rewrite (6.12)

$$\mathfrak{N}_{\mathcal{G}} = \sum_{e \in E} L(BoundNE(e, \delta_e)\,, \vec{\delta}) \tag{6.13}$$

From this, we can finally have a finite representation of $\mathsf{NE}(k)$ as follows. We denote $\kappa_{\vec{\delta}}$ to be the number of players in $\vec{\delta}$, then we take $m = \left\lfloor \frac{k}{\kappa_{\vec{\delta}}} \right\rfloor$. Then, among k players, for $m \cdot \kappa_\delta$ players, we consider $\vec{\delta}$. And because $k - m \cdot \kappa_{\vec{\delta}} < \kappa_{\vec{\delta}}$, for any edge NE $\vec{n}$ with $k - m \cdot \kappa_{\vec{\delta}}$ players, there has to be at least one $e \in E$ such that $n_e < \delta_e$, i.e, $\vec{n} \in BoundNE(e, \delta_e)$.

6.5 Conclusion

In this chapter, we considered the problem which asks for a finite representation of a function which maps an integer k to an edge NE profile with k players. We established that the set of edge NE profiles is a semi-linear set, and we obtained a parameterized system of equations, of which every period vectors of the semi-linear set is a solution. Then, we exploit the semantics of the system of equations to show that given a parameter, there is a unique solution to the system of equations. From this, because of the parameter, we established that all the period vectors of the semi-linear set

that characterized the set of all NE, are co-linear. Finally, we formulate a conjecture which says that not only all the period vectors are co-linear, there is a unique period vector which works for all bases. If this conjecture turns out to be true, then we would have a finite representations of the set of all NE of a pNCG, which easily answers the problem that we considered at the beginning of the chapter.

CONCLUSION

In this chapter, we summarize the contributions of this thesis, discuss the shortcomings and try to shed lights on what possible future directions can be taken from here.

Summary

Network Congestion Games with Fixed Number of Players

We defined a new model of network congestion games with two features: cost computation at synchronized congestion only and dynamic strategies

In this model, we obtained algorithms for three constrained optimal problems: Social optima (SO), Nash Equilibria (NE) and Subgame Perfect Equilibria (SPE) in Chapter 2, 3 and 4 respectively. In the following, we briefly recall how we addressed these problems, and what complexity bounds we obtained.

We started with the constrained social optimal problem, in which given an instance of a network congestion game and a value, we asked whether there is a strategy profile with social cost less than the given value. To solve this problem, a naive approach could to guess a play (aka a path in the configuration graph), step by step, while maintaining a counter to keep in check the social cost within the given threshold. This approach would give us a EXPSPACE complexity. To have a better upper bound complexity result, we exploited the graph structure and the cost computation model, and abstracted away exact configuration to their Parikh image. Then in that abstract configuration graph, we non-deterministically guess a play, and check whether its social cost is less than the given threshold. Because acyclic strategy profiles suffice, this yields a PSPACE-algorithm.

In case of Nash Equilibria (NE), we first showed the existence of NE in our model of network congestion games, by showing convergence of the *best-response* dynamics. Then, we studied a similar constrained problem for NE, where the input again is an instance of an NCG and a threshold. Our approach to tackle this problem was similar to what we did for SO, except here we verify two conditions. We non-deterministically, step by step, guess a play of the game, and at each step verify: whether the cumulative social cost up to current step is still less than the threshold, and whether it *is going to be* the outcome of an NE.

To check the first condition, just like the SO problem, we maintain a counter which stores the cumulative social cost, and verify at each step.. For the second condition, our algorithm makes use of a characterization of all NE outcomes. The characterization justifies that if a player were to deviate, there is a response from the coalition of other players which would make this deviating player's cost worse than what would have been if they didn't deviate.

Now, in the characterization itself, we made use of a two-player (zero-sum) total payoff game[35], and the value in that game. This game is actually between a deviating player and the coalition of all other players, and the value in that game is the worst cost that the deviating player may end

up paying due to how the coalition *punishes*. Our algorithm maintains a counter for each player, and verifies at each step whether the cost up to that step is less than the *values* of these two-player games for each deviating player and for their all possible unilateral deviating configurations.

Because computing the value in the above-mentioned two player games can be done in EXPTIME, our algorithm has an EXPSPACE complexity.

Finally, we turned our focus to *subgame perfect equilibria* (SPE) - another solution concept for stability, but more relevant in the dynamic setting. Unlike NE, in our model of NCG we have neither a proof of existence of SPE, nor we have an example where we can show that SPE do not exist.

Nonetheless, we considered the constrained SPE problem similar to that of SO and NE. Not surprisingly this is technically more challenging than the earlier two. Again our basic approach remains the same as it were for NE: we guess a play in the game, and verify whether it is an outcome of an SPE with social cost less than the threshold given in the decision problem. To verify whether the play is indeed an outcome of an SPE, we make use of a characterization of outcomes of SPE.

The main idea behind the characterization is to define a function which maps an edge to a so called λ-value such that any SPE outcome starting from that edge must have cost less than that λ-value, and then showing that any play is an outcome of an SPE iff it is so-called λ-*consistent*. In fact, we build this λ-mapping recursively till it reaches a fix-point: the i^{th} λ-value is defined using the $(i-1)^{th}$ λ-consistent plays, and so on. Initially, all plays are λ-consistent, at each step when we update the λ-values, and subsequently the set of λ-consistent plays gets refined, which finally, when reaches a fixed-point, gives exactly the set of outcomes of all SPE. One way to obtain a negative instance on the existence of SPE problem (if exists) could be to see if there is an NCG, for which at some step of updating λ-value, the set of λ-consistent plays became empty. Nonetheless, at the end, we check whether the set corresponding to the fix-point contains a play with social cost less than the bound given in question. This gives us EXPSPACE-membership of constrained SPE problem.

Parameterized Network Congestion Games

In this second part of the thesis, we consider parametric network congestion game (pNCG) where the number of players is a parameter. Our aim in this part has been to compute optimal strategy profiles (NE, SO etc) for arbitrary large number of players from their counterpart in the same network with small number of players.

We first introduced a necessary framework to study pNCG for series-parallel network. It is noteworthy that for this part, we re-adjust the model to only consider path strategies and non-synchronous path computation. As per our knowledge, it has not been studied how NE profiles are related with each other when we vary the number of players in an instance of congestion games, in any model. Therefore, we considered the classical model for the network congestion games , and started studying the same on it; of course our future goal would be to extend it into our model of NCG, mainly to incorporate the dynamic strategy structure.

Then we started studying NE for pNCG in series-parallel graph. The set of all NE is a semi-linear set in the current framework, and we obtained a pattern on the set of period vectors of the semi-linear set. At this point, we made a conjecture which says, in fact, there is a unique period vector for a pNCG. If the conjecture turns out to be true, then we can compute each and every

NE of the pNCG by a finite set of NE (for small number of players) and the unique period vector (which we call the *shift vector* of pNCG). This will enable us to comment on best/worst NE in the pNCG, how they evolve as the number of players change in an NCG.

Future Work

In this section, we first go through some of the follow-up questions that have come up in the earlier chapters, and then we discuss some research directions that our work may embark upon.

Immediate follow ups

Matching complexity lower bound This thesis leaves several complexity questions open. We have NP-hardness for Social optimal, but there too remains a gap with the PSPACE membership. Hence, an immediate series of follow-up questions would be to find hardness complexity results for NE and SPE problem, and/or to improve the upper bound complexity results to close the gaps.

Existence of SPE problem Even though we obtained an algorithm for constraint SPE problem, we haven't managed to neither positively nor negatively answer whether SPE exists in our model. In another dynamic model of congestion games, namely dynamic resource-allocation games[8], there does not always exist an SPE. But in that game, the non-synchronous cost computation model has been a major contributing factor in designing an instance where SPE doesn't exist, which has no immediate consequences in our setting. Therefore, it remains an open problem in our model.

Parametrized Congestion games Regarding parametrized congestion games, one way to go ahead is to prove the conjecture that we formulate, which says, there is an unique period vector for a pNCG. As mentioned earlier, this completes studying the problem for NE, and an immediate follow-up would be if similar results can be obtained for SO, which in turn might show how Price of Anarchy (PoA)/Price of Stability(PoS) evolves as the number of players grows in an NCG.

Even though some of the results on this part have been proven only for series-parallel graphs, some other results, notably the most technically challenging result on this part have been shown for any network arena whatsoever. This opens up a possibility to look at the problem for larger class of arenas, and possibly extending to dynamic strategies.

Long-term Objectives

Congestion games with Imperfect information. In our model of NCG, the players when they make their next move from a vertex, know where all the other players are, and take their decision accordingly. This may not always be realistic. There might be scenario where one or more player get to know only partial information about other players' position, which should leads us to study *imperfect information games* in the context of network congestion games. Both in turn-based [10, 14, 24] and concurrent [19, 28, 29, 55] settings, imperfect information games

have been studied extensively for different objectives. But, to the best of our knowledge, imperfect information have not yet been studied in congestion games. To give a basic framework, we can start with the concurrent game structure that we have associated with NCG, where a configuration of the NCG represents a state of the concurrent game. We declare two states of the concurrent game *equivalent* from the point of view of a player if in the respective configurations, the number of players that they observe is the same. We have used similar information graph in our model for solving the decision problem related to social optima. But for the equilibria we have used full observations of a player, therefore we believe this is an interesting framework to investigate what kind of stable profiles can be achieved when the information set is partial, and how well/bad it might perform with respect to a profile which optimizes social cost.

Dynamic Timed Congestion games. In our model of NCG, there is no separate temporal component. Similar to any other possible resources affected due to congestion, like fuel cost in case of road-traffic, and connection/broadcast quality in case of internet, time here is also abstracted as a part of the cost. A separate temporal component had been added and studied in the context of congestion games in [6, 7, 31]. Avni et al. [6] compute congestion cost depending on how much time (continuous) a player spend with another player together at some vertex. Cost functions are associated with vertices, and there are conditions associated with edges which make players stay at a vertex for some time. Because they consider continuous time-points, this framework raises an uncountable strategy space for players. The setting is more general than what we have considered here except they do not consider dynamic strategies. They have studied the existence of NE, and then following the classical approach they comment on bounds of PoA/PoS for their model. Hence, what we can propose for a future work is to study our constraint problems(SO, NE, and SPE) in the timed setting with dynamic strategies. Notably, existence of SPE also needs to be addressed when we extend timed setting into dynamic strategy space.

Both the above mentioned long-term objectives are related to our model described in Part I. This is because, having only some preliminary results in Part II, we believe there are lots of gaps to fill in here on its own, and then only we could provide a convincing argument to go forward on that line of research. Nonetheless, we believe computing solution concepts for instances with many players from instances with fewer players could be, in general, an interesting topic to investigate in several contexts.

Bibliography

1. A. Akella, S. Seshan, R. Karp, S. Shenker, and C. Papadimitriou. "Selfish Behavior and Stability of the Internet: A Game-Theoretic Analysis of TCP". *SIGCOMM Comput. Commun. Rev.* 32:4, 2002, pp. 117–130. ISSN: 0146-4833. DOI: 10.1145/964725.633037. URL: https://doi.org/10.1145/964725.633037.
2. R. Alur, T. A. Henzinger, and O. Kupferman. "Alternating-time temporal logic". *J. ACM* 49:5, 2002, pp. 672–713. DOI: 10.1145/585265.585270. URL: https://doi.org/10.1145/585265.585270.
3. E. Anshelevich, A. Dasgupta, J. Kleinberg, É. Tardos, T. Wexler, and T. Roughgarden. "The Price of Stability for Network Design with Fair Cost Allocation". *SIAM Journal on Computing* 38:4, 2008, pp. 1602–1623. DOI: 10.1137/070680096.
4. I. Ashlagi, D. Monderer, and M. Tennenholtz. "Resource selection games with unknown number of players". In: *Proceedings of the fifth international joint conference on Autonomous agents and multiagent systems*. 2006, pp. 819–825.
5. R. J. Aumann. "16. Acceptable Points in General Cooperative n-Person Games". In: *Contributions to the Theory of Games (AM-40), Volume IV*. Princeton University Press, 2016, pp. 287–324.
6. G. Avni, S. Guha, and O. Kupferman. "Timed Network Games". In: *Proceedings of the 42nd International Symposium on Mathematical Foundations of Computer Science (MFCS'17)*. Ed. by K. G. Larsen, H. L. Bodlaender, and J.-F. Raskin. Vol. 84. Leibniz International Proceedings in Informatics. Leibniz-Zentrum für Informatik, 2017, 37:1–37:16. DOI: 10.4230/LIPIcs.MFCS.2017.37.
7. G. Avni, S. Guha, and O. Kupferman. "Timed Network Games with Clocks". In: *Proceedings of the 43rd International Symposium on Mathematical Foundations of Computer Science (MFCS'18)*. Ed. by I. Potapov, P. G. Spirakis, and J. Worrell. Vol. 117. Leibniz International Proceedings in Informatics. Leibniz-Zentrum für Informatik, 2018, 23:1–23:18. DOI: 10.4230/LIPIcs.MFCS.2018.23.
8. G. Avni, T. A. Henzinger, and O. Kupferman. "Dynamic Resource Allocation Games". *Theoretical Computer Science* 807, 2020, pp. 42–55. DOI: 10.1016/j.tcs.2019.06.031.
9. B. Awerbuch, Y. Azar, and A. Epstein. "the price of routing unsplittable flow". In: *Proceedings of the 37th Annual ACM Symposium on the Theory of Computing (STOC'05)*. Ed. by H. N. Gabow and R. Fagin. ACM Press, 2005, pp. 57–66. DOI: 10.1145/1060590.1060599.
10. N. Bertrand, B. Genest, and H. Gimbert. "Qualitative Determinacy and Decidability of Stochastic Games with Signals". *Journal of the ACM (JACM)* 64:5, 2017, 33:1–33:48. DOI: 10.1145/3107926. URL: https://hal.inria.fr/hal-01635127.

11. N. Bertrand, N. Markey, S. Sadhukhan, and O. Sankur. "Dynamic Network Congestion Games". *CoRR* abs/2009.13632, 2020. arXiv: 2009.13632. URL: https://arxiv.org/abs/2009.13632.
12. D. P. Bertsekas and J. N. Tsitsiklis. *Parallel and distributed computation: numerical methods*. Prentice Hall, 1989.
13. U. Bhaskar, L. Fleischer, and E. Anshelevich. "A Stackelberg strategy for routing flow over time". *Games and Economic Behavior* 92, 2015, pp. 232–247. DOI: 10.1016/j.geb.2013.09.004.
14. B. Bosansky, C. Kiekintveld, V. Lisy, and M. Pechoucek. "An exact double-oracle algorithm for zero-sum extensive-form games with imperfect information". *Journal of Artificial Intelligence Research* 51, 2014, pp. 829–866.
15. T. Brihaye, V. Bruyère, A. Goeminne, J.-F. Raskin, and M. Van den Bogaard. "The Complexity of Subgame Perfect Equilibria in Quantitative Reachability Games". In: *Proceedings of the 30th International Conference on Concurrency Theory (CONCUR'19)*. Ed. by W. J. Fokkink and R. van Glabbeek. Vol. 140. Leibniz International Proceedings in Informatics. Leibniz-Zentrum für Informatik, 2019, 13:1–13:16. DOI: 10.4230/LIPIcs.CONCUR.2019.13.
16. T. Brihaye, V. Bruyère, N. Meunier, and J.-F. Raskin. "Weak Subgame Perfect Equilibria and their Application to Quantitative Reachability". In: *Proceedings of the 24th EACSL Annual Conference on Computer Science Logic (CSL'15)*. Ed. by S. Kreutzer. Vol. 41. Leibniz International Proceedings in Informatics. Leibniz-Zentrum für Informatik, 2015, pp. 504–518. DOI: 10.4230/LIPIcs.CSL.2015.504.
17. T. Brihaye, V. Bruyère, J. D. Pril, and H. Gimbert. "On Subgame Perfection in Quantitative Reachability Games". *Log. Methods Comput. Sci.* 9, 2012.
18. T. Brihaye, F. Laroussinie, N. Markey, and G. Oreiby. "Timed Concurrent Game Structures". In: *CONCUR 2007 - Concurrency Theory, 18th International Conference, CONCUR 2007, Lisbon, Portugal, September 3-8, 2007, Proceedings*. Ed. by L. Caires and V. T. Vasconcelos. Vol. 4703. Lecture Notes in Computer Science. Springer, 2007, pp. 445–459. DOI: 10.1007/978-3-540-74407-8_30. URL: https://doi.org/10.1007/978-3-540-74407-8%5C_30.
19. K. Chatterjee and T. A. Henzinger. "Semiperfect-information games". In: *International Conference on Foundations of Software Technology and Theoretical Computer Science*. Springer. 2005, pp. 1–18.
20. R. Cominetti, M. Scarsini, M. Schröder, and N. E. S. Moses. "Convergence of Large Atomic Congestion Games". *CoRR* abs/2001.02797, 2020. arXiv: 2001.02797. URL: http://arxiv.org/abs/2001.02797.
21. R. Cominetti, M. Scarsini, M. Schröder, and N. E. S. Moses. "Price of Anarchy in Stochastic Atomic Congestion Games with Affine Costs". *CoRR* abs/1903.03309, 2019. arXiv: 1903.03309. URL: http://arxiv.org/abs/1903.03309.
22. J. Correa, R. Hoeksma, and M. Schröder. "Network congestion games are robust to variable demand". *Transportation Research Part B: Methodological* 119, 2019, pp. 69–78.

23. M. Dowson. "The Ariane 5 software failure". *ACM SIGSOFT Software Engineering Notes* 22:2, 1997, p. 84.

24. L. Doyen and J.-F. Raskin. *Games with imperfect information: theory and algorithms.* 2011.

25. L. Epstein, M. Feldman, T. Tamir, Ł. Witkowski, and M. Witkowski. "Approximate strong equilibria in job scheduling games with two uniformly related machines". *Discrete Applied Mathematics* 161:13, 2013, pp. 1843–1858. ISSN: 0166-218X. DOI: https://doi.org/10.1016/j.dam.2013.02.035. URL: https://www.sciencedirect.com/science/article/pii/S0166218X13001224.

26. D. Fotakis. "Stackelberg Strategies for Atomic Congestion Games". In: *Algorithms – ESA 2007*. Ed. by L. Arge, M. Hoffmann, and E. Welzl. Springer Berlin Heidelberg, Berlin, Heidelberg, 2007, pp. 299–310. ISBN: 978-3-540-75520-3.

27. S. Ginsburg and E. H. Spanier. "Semigroups, Presburger formulas, and languages." *Pacific Journal of Mathematics* 16:2, 1966, pp. 285–296. DOI: pjm/1102994974. URL: https://doi.org/.

28. V. Gripon and O. Serre. "Qualitative Concurrent Games with Imperfect Information". *CoRR* abs/0902.2108, 2011. arXiv: 0902.2108. URL: http://arxiv.org/abs/0902.2108.

29. V. Gripon and O. Serre. "Qualitative concurrent stochastic games with imperfect information". In: *International Colloquium on Automata, Languages, and Programming*. Springer. 2009, pp. 200–211.

30. K. M. Gurumurthy, K. M. Kockelman, and M. D. Simoni. "Benefits and Costs of Ride-Sharing in Shared Automated Vehicles across Austin, Texas: Opportunities for Congestion Pricing". *Transportation Research Record* 2673:6, 2019, pp. 548–556. DOI: 10.1177/0361198119850785. eprint: https://doi.org/10.1177/0361198119850785. URL: https://doi.org/10.1177/0361198119850785.

31. M. Hoefer, V. S. Mirrokni, H. Röglin, and S.-H. Teng. "Competitive routing over time". *Theoretical Computer Science* 412:39, 2011, pp. 5420–5432. DOI: 10.1016/j.tcs.2011.05.055.

32. V. Jacobson. "Congestion Avoidance and Control". In: *Symposium Proceedings on Communications Architectures and Protocols*. SIGCOMM '88. Association for Computing Machinery, Stanford, California, USA, 1988, pp. 314–329. ISBN: 0897912799. DOI: 10.1145/52324.52356. URL: https://doi.org/10.1145/52324.52356.

33. R. Karp, E. Koutsoupias, C. Papadimitriou, and S. Shenker. "Optimization problems in congestion control". In: *Proceedings 41st Annual Symposium on Foundations of Computer Science*. 2000, pp. 66–74. DOI: 10.1109/SFCS.2000.892066.

34. A. Kesselman, S. Leonardi, and V. Bonifaci. "Game-Theoretic Analysis of Internet Switching with Selfish Users". In: *Internet and Network Economics*. Ed. by X. Deng and Y. Ye. Springer Berlin Heidelberg, Berlin, Heidelberg, 2005, pp. 236–245. ISBN: 978-3-540-32293-1.

35. L. Khachiyan, E. Boros, K. Borys, K. Elbassioni, V. Gurvich, G. Rudolf, and J. Zhao. "On short paths interdiction problems: Total and node-wise limited interdiction". *Theory of Computing Systems* 43:2, 2008, pp. 204–233. DOI: 10.1007/s00224-007-9025-6.

36. M. Klimm and P. Warode. "Complexity and parametric computation of equilibria in atomic splittable congestion games via weighted block Laplacians". In: *Proceedings of the Fourteenth Annual ACM-SIAM Symposium on Discrete Algorithms*. SIAM. 2020, pp. 2728–2747.
37. R. Koch and M. Skutella. "Nash equilibria and the price of anarchy for flows over time". *Theory of Computing Systems* 49:1, 2011, pp. 71–97. DOI: 10.1007/s00224-010-9299-y.
38. R. E. Korf. "A complete anytime algorithm for number partitioning". *Artificial Intelligence* 106:2, 1998, pp. 181–203.
39. E. Koutsoupias and C. H. Papadimitriou. "Worst-case equilibria". *Computer Science Review* 3:2, 2009, pp. 65–69. DOI: 10.1016/j.cosrev.2009.04.003.
40. E. Koutsoupias and K. Papakonstantinopoulou. "Contention Issues in Congestion Games". In: *Proceedings of the 39th International Colloquium on Automata, Languages and Programming (ICALP'12) – Part II*. Ed. by A. Czumaj, K. Mehlhorn, A. Pitts, and R. Wattenhofer. Vol. 7392. Lecture Notes in Computer Science. Springer-Verlag, 2012, pp. 623–635. DOI: 10.1007/978-3-642-31585-5_55.
41. D. C. Kozen. *Automata and computability*. Springer Science & Business Media, 2012.
42. H. W. Kuhn. "11. Extensive Games and the Problem of Information". In: 1953.
43. *Lectures in Game Theory for Computer Scientists*. Cambridge University Press, 2011. DOI: 10.1017/CBO9780511973468.
44. N. Leveson and C. Turner. "An investigation of the Therac-25 accidents". *Computer* 26:7, 1993, pp. 18–41. DOI: 10.1109/MC.1993.274940.
45. L. Lopez, A. Fernandez, and V. Cholvi. "A game theoretic analysis of protocols based on fountain codes". In: *10th IEEE Symposium on Computers and Communications (ISCC'05)*. 2005, pp. 625–630. DOI: 10.1109/ISCC.2005.11.
46. S. Malik, M. A. Khan, and H. El-Sayed. "Collaborative Autonomous Driving—A Survey of Solution Approaches and Future Challenges". *Sensors* 21:11, 2021. DOI: 10.3390/s21113783. URL: https://www.mdpi.com/1424-8220/21/11/3783.
47. C. A. Meyers and A. S. Schulz. "The complexity of welfare maximization in congestion games". *Networks* 59:2, 2012, pp. 252–260. DOI: 10.1002/net.20439.
48. D. Monderer and L. S. Shapley. "Potential Games". *Games and Economic Behavior* 14:1, 1996, pp. 124–143. DOI: 10.1006/game.1996.0044.
49. R. B. Myerson. "Population uncertainty and Poisson games". *International Journal of Game Theory* 27:3, 1998, pp. 375–392.
50. J. von Neumann and O. Morgenstern. *Theory of Games and Economic Behavior (60th Anniversary Commemorative Edition)*. Princeton University Press, 2007. ISBN: 9781400829460. DOI: doi:10.1515/9781400829460. URL: https://doi.org/10.1515/9781400829460.
51. M. J. Osborne et al. *An introduction to game theory*. Vol. 3. 3. Oxford university press New York, 2004.
52. R. Parikh. "On Context-Free Languages". *J. ACM* 13, 1966, pp. 570–581.

53. M. Penn, M. Polukarov, and M. Tennenholtz. "Random Order Congestion Games". *Mathematics of Operations Research* 34:3, 2009, pp. 706–725. DOI: 10.1287/moor.1090.0394.

54. A. Pnueli. "The temporal logic of programs". In: *18th Annual Symposium on Foundations of Computer Science (sfcs 1977)*. 1977, pp. 46–57. DOI: 10.1109/SFCS.1977.32.

55. J.-F. Raskin, T. A. Henzinger, L. Doyen, and Chattere. "Algorithms for omega-regular games with imperfect information".

56. R. W. Rosenthal. "A class of games possessing pure-strategy Nash equilibria". *International Journal of Game Theory* 2:1, 1973, pp. 65–67. DOI: 10.1007/BF01737559.

57. T. Roughgarden. "Routing games". In: *Algorithmic Game Theory*. Ed. by N. Nisan, T. Roughgarden, É. Tardos, and V. V. Vazirani. Cambridge University Press, 2007. Chap. 18, pp. 461–486.

58. W. J. Savitch. "Relationships between nondeterministic and deterministic tape complexities". *Journal of Computer and System Sciences* 4:2, 1970, pp. 177–192. ISSN: 0022-0000. DOI: https://doi.org/10.1016/S0022-0000(70)80006-X. URL: https://www.sciencedirect.com/science/article/pii/S002200007080006X.

59. M. Simaan and J. B. Cruz. "On the Stackelberg strategy in nonzero-sum games". *Journal of Optimization Theory and Applications* 11:5, 1973, pp. 533–555.

60. S. Suri, C. D. Tóth, and Y. Zhou. "Selfish Load Balancing and Atomic Congestion Games". *Algorithmica* 47:1, 2007, pp. 79–96. DOI: 10.1007/s00453-006-1211-4.

61. G. Thandavarayan, M. Sepulcre, and J. Gozalvez. "Cooperative Perception for Connected and Automated Vehicles: Evaluation and Impact of Congestion Control". *IEEE Access* 8, 2020, pp. 197665–197683. DOI: 10.1109/ACCESS.2020.3035119.

62. C. Wang, X. V. Doan, and B. Chen. "Atomic congestion games with random players: network equilibrium and the price of anarchy". *Journal of Combinatorial Optimization*, 2020, pp. 1–20.

63. C. Wang, X. V. Doan, and B. Chen. "Price of anarchy for non-atomic congestion games with stochastic demands". *Transportation Research Part B: Methodological* 70, 2014, pp. 90–111.

DOCTORAT BRETAGNE LOIRE / MATHSTIC

UNIVERSITÉ DE RENNES 1

Titre : Les jeux de congestion dans les réseaux sous l'angle de la vérification

Mot clés : Jeux de congestion, équilibres de Nash, équilibres parfaits en sous-jeux, jeux de congestion paramétrés, théorie algorithmique des jeux

Résumé : Les jeux de congestion sont un domaine de recherche bien étudié ; dans ce domaine, les jeux de congestion dans les réseaux premettent de représenter la congestion des réseaux de distribution, et d'étudier à quel point un modèle de réseau est bon ou mauvais en termes de coût total lorsque chaque joueur joue de façon égoïste, cherchant uniquement à optimiser son propre coût ; Nous considérons ces jeux de congestion du point de vue des méthodes formelles, cherchant à vérifier par exemple que, dans un réseaux fixé, il existe un profil de stratégies optimal qui satisfasse une propriété donnée.

Nous définissons un modèle de jeux de congestion avec deux particularités : d'une part, le calcul du coût d'une transition dépend du nombre de joueurs utilisant simultanément une arête ; d'autre part, les joueurs choisissent leur chemin de façon dynamique en fonction des choix des autres joueurs. Nous montrons que dans ce modèle les équilibres de Nash existent toujours en montrant la convergence de la dynamique de meilleure réponse. Nous étudions ensuite le problème de vérification mentionné ci-dessus, résolvant le problème de l'existence d'un équilibre social, d'un équilibre de Nash ou d'un équilibre parfait en sous-jeux ayant un côut borné.

Dans une deuxième partie, nous étudions les jeux de congestion paramétrés, dans lesquels le nombre de joueurs est un paramètre. Nous nous intéressons à l'évolution des équilibres de Nash en fonction du nombre de joueurs : notre objectif est de calculer efficacement l'ensemble des équilibres de Nash pour un nombre arbitrairement grand de joueurs, à partir des équilibres de Nash pour de petits nombres de joueurs. Nos premiers résultats portent sur les réseaux *série-parall'ele*, sans les particularités ci-dessus. Nous conjecturons que ces résultats s'étendent à l'ensemble des graphes, ce qui donnerait lieu à un calcul efficace de tous les équilibres de Nash, quel que soit le nombre de joueurs.

Title: A Verification Viewpoint to Network Congestion Games

Keywords: Congestion games, Nash Equilibria, Subgame perfect Equilibria, Parametrized congestion games, Algorithmic game theory

Abstract: Congestion games are a well-studied area of research, and Network congestion games (NCG) model the problem of congestion in flow networks. The most common problem is to study, broadly speaking, how well or how bad a model of NCG is in terms of total cost for all players when each player plays selfishly. We view network congestion games from a formal-methods standpoint, in which we are interested in problem like: given an instance (network and number of players fixed) of a chosen model of NCG and given a specification, does there exist an optimal profile satisfying the specification?

We define a model of network congestion games with two pecularities: first, the players bear congestion effect on their cost for an edge only if they use that edge with other players simultaneously; second, players can choose their path dynamically, at each step of their route, which differs from the classical setting where players choose their path at the beginning. We show that in this model Nash Equilibria always exist by showing convergence of best-response dynamics. Then we study three decision problems on Social optima, Nash Equilibria and Subgame perfect Equilibria, each of which asks whether, given an instance of our model and a bound, there is a corresponding strategy profile with bounded social cost.

In the second part of the thesis, we study parameterized network congestion games, where the number of players is left as a parameter. Our main problem here is to compute Nash Equilibria for instances with many players, from instances with fewer players, instead of computing them from scratch. Here, we started solving the problem for the classical model of NCG, without the above mentioned two pecularities, and with the arena restricted to series-parallel graph. We obtained some preliminary results on this problem, and formulated a conjecture about how to efficiently compute all Nash equilibria for any number of players.