

Finding a Nash Equilibrium and an optimal sharing policy for multi-agent network expansion game

Nadia Chaabane, Alessandro Agnetis, Cyril Briand, Marie-José Huguet

▶ **To cite this version:**

Nadia Chaabane, Alessandro Agnetis, Cyril Briand, Marie-José Huguet. Finding a Nash Equilibrium and an optimal sharing policy for multi-agent network expansion game. Networks, 2016, 69 (1), pp.94-109. 10.1002/net.21711 . hal-01383580

HAL Id: hal-01383580
https://hal.science/hal-01383580

Submitted on 18 Oct 2016

HAL is a multi-disciplinary open access archive for the deposit and dissemination of scientific research documents, whether they are published or not. The documents may come from teaching and research institutions in France or abroad, or from public or private research centers.

L'archive ouverte pluridisciplinaire **HAL**, est destinée au dépôt et à la diffusion de documents scientifiques de niveau recherche, publiés ou non, émanant des établissements d'enseignement et de recherche français ou étrangers, des laboratoires publics ou privés.

Finding a Nash Equilibrium and an optimal sharing policy for multi-agent network expansion game

Nadia Chaabane[1], Alessandro Agnetis[2], Cyril Briand[1], and Marie-José Huguet[1]

[1] LAAS-CNRS, Université de Toulouse, CNRS, INSA, UPS, Toulouse, France
[2] Università di Siena, via Roma 56, 53100 Siena, Italy.

Abstract. In this work, a multi-agent network flow problem is addressed, aiming at characterizing the properties of stable flows and allowing their computation. Two types of agents are considered: *transportation-agents*, that carry a flow of products on a given network and another agent, either a *producer* or a *customer*, who is willing to ship (receive, respectively) products. Every transportation-agent controls a set of arcs, each having a capacity that can be increased up to a certain point at a given cost. The other agent (i.e., the customer/producer) is interested in maximizing the flow transshipped through the network. To this aim, we assume it offers the transportation-agents a reward that is proportional to the realized flow value. This particular multi-agent framework is referred to as a Multi-Agent network expansion game. We characterize and find particular stable strategies (i.e., Nash equilibria) that are of interest for this game. We particularly focus on the problem of finding a Nash Equilibrium and a sharing policy that maximize the value of the total flow. We prove that this problem is NP-hard in the strong sense and show how such a strategy can be characterized considering paths in specific auxiliary graphs. We also provide a mixed integer linear programming (MILP) formulation to solve the problem. Computational experiments are provided to prove the effectiveness of our approach and derive some insights for practitioners.

Keywords: Multi-Agent Network flow, Nash Equilibria, Complexity, Network Expansion Game, MILP.

1 Introduction

In the vast body of network flow literature, the capacity expansion problem has received considerable attention. This problem aims at deciding where to increase arc capacity so as to expand the network transportation capability while reducing the overall routing cost. Many research papers addressed this problem under various assumptions (see e.g., [14], [18] or [19] for the design of transportation networks; [20] for the management of electric utilities or [17] and [24] for building up robust solutions in the case demand is uncertain). In all these works, it is basically assumed that a single agent is able to control the capacity of the

entire network and the focus is put on the satisfaction of its own interest, which is obviously a limitation, as many transportation problems are more and more cooperative in essence.

Contrastingly, in this paper, a network flow involving a set of *transportation-agents* is considered, each owning a part of the network routes. Every transportation-agent can vary the capacities of its arcs at a given cost. A *producer* (or a *customer*) is willing to ship (to receive, respectively) a product flow as high as possible through the network. For each additional transshipped product unit, the customer offers a reward as incentive to the transportation-agents, and it decides how this reward is distributed among them. This framework defines a non-cooperative network flow game where the resulting flow eventually depends on the individual strategies of all agents. The agents' individual strategies are locally *stable* if there is no incentive for any agent to modify its own strategy in order to improve its profit, which corresponds to the definition of a Nash equilibrium [21]. This stability concept is important since, in the face of a stable strategy, although the agents are self-interested, none of them will be interested in individually changing its mind, and therefore the flow value remains unchanged. From the customer viewpoint, the problem of finding a Nash equilibrium that maximizes the flow value is relevant, because its solution gives the highest flow that can be reached for the customer, provided that the organization remains stable.

For illustration, let us consider the network displayed in Fig. 1 with two transportation-agents A_1 and A_2 and a customer-agent willing to transport the largest possible flow from node A to node D. We assume it offers a per-unit flow reward $\pi = 120$, which is shared between A_1 and A_2 according to the policy $W = (w_1, w_2)$ (with $w_1 + w_2 = 1$). The arcs belonging to each transportation-agent are $E_1 = \{b = (A, C), c = (B, C), d = (B, D)\}$ and $E_2 = \{a = (A, B), e = (C, D)\}$, represented with plain and dotted arcs, respectively. The intervals of normal and maximum capacities ($[\underline{q}_{i,j}, \overline{q}_{i,j}]$) and the expansion cost ($c_{i,j}$) are displayed on each arc. For instance, for arc a from A to B, the interval of capacity is $[0, 2]$ and the cost for increasing capacity by one unit equals 50.

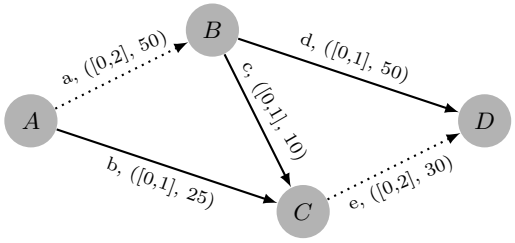


Fig. 1. Example of a network expansion game

Network flow games define an interesting research area with links to many application fields such as transportation networks, supply chain management, web services, production or energy management (see e.g., [8], [25]). In cooperative flow games, arcs are owned by different players, and in their most basic version the problem is to distribute the revenues related to a maximum flow. It was shown [16] that core allocations correspond to minimum cuts on the network. In [3, 27] a cooperative network flow game is studied and the cost of stability of coalitions is analyzed. A non-cooperative network flow game is the *path player game* [26], in which each players owns a set of arcs forming a path on the network, and paths are not necessarily disjoint. This feature models systems in which paths are owned by decision makers (*network providers*), as in public transportation, energy or information networks. The decision makers offer a certain bandwidth (*capacity*) to be used by the flow, like a bandwidth of electricity, or a certain daily frequency of trains. However, capacity is not associated with each network link, but rather there is a bound on total flow through the network. Equilibria in this model describe a stable market situation among competing path owners. This is perhaps the closest model to ours in the literature.

Other closely related works are [4] and [1], dealing with the Multi-Agent Project Scheduling, even if it is a different network problem. In this work, each agent controls a subset of activities that make up a project, each activity having a nominal duration and a minimum duration. The customer offers a fixed daily reward to the agents for completing a project sooner than its nominal length and the sharing policy is known. The problem is to find a duration for the activities so that the resulting solution is stable, and among all stable solutions, the one having the smallest makespan.

In this study, similarly to those cited above, we assume that the transportation network and the agents' parameters are known and that the decision is centralized. This situation can be viewed as unrealistic since, in many real-life contexts, agents make their own decisions without perfect knowledge of all parameters. However, our purpose is to show that, even with this strong assumption, some problems remain computationally difficult. Moreover, we believe that the concepts introduced here can be helpful for designing more realistic, stable cooperation mechanisms.

As we already mentioned, this paper considers the problem of finding a Nash equilibrium that maximizes the value of network flow. In contrast with [1] and [7], in this paper we assume that the customer is able to decide how to share the reward among the transportation-agents, hence a somewhat more general setting. The key contributions are the following. First, we show that the resulting problem is NP-hard. Second, a necessary and sufficient condition of stability is provided that allows the characterization of Nash equilibria in terms of paths in a residual graph. In addition, a mixed integer linear programming (MILP) model is given that allows finding both an optimal sharing policy and a stable agents' strategy that maximizes the maximum flow. Last, on the basis of experimental studies, we analyze various reward sharing policies and derive use-

ful insights which should help practitioners to properly deal with multi-agent capacity expansion problems.

The paper is organized as follows. Section 2 defines formally our Network Expansion Game and introduces some basic notations and properties. Thereafter, Section 3 focuses on the single transportation-agent case and shows how profitable strategies can be characterized in terms of paths in a residual graph. These results are generalized in Section 4, in which a necessary and sufficient condition of stability is stated and the complexity of finding a maximum-flow Nash equilibrium is analyzed. Section 5 and 6 are dedicated to the presentation of a MILP formulation and the analysis of our computational experiments. Finally, conclusions and future research directions are discussed in Section 7.

2 Problem Statement

This section defines more formally the so-called *Network Expansion Game* (`NEG`). The main focus of our analysis is on finding *stable* strategies or *Nash equilibria*, i.e., such that no *agent* has any incentive to unilaterally modify its strategy.

2.1 Network Expansion Game

Network expansion games are formally defined in this section with respect to a set of transportation-agents and a customer-agent. The producer case is quite symmetric as it only requires to reverse the arc direction on the network. We assume here that the sharing policy is free and part of the customer-agent's strategy, which differentiates this work from [7] and [6]. The concepts of a multi-agent network and a strategy vector are introduced first.

Definition 1. Multi-agent network
A Multi-agent network (`MAN`) is defined by a tuple $< G, \underline{\mathcal{Q}}, \overline{\mathcal{Q}}, \mathcal{C}, \mathcal{A}, \pi >$ *such that:*

- $G = (\mathcal{V}, \mathcal{E})$ *is a network composed of a set* $\mathcal{V}$ *of nodes* ($s, t \in \mathcal{V}$ *referring to the source and the sink nodes of the network, respectively) and a set* $\mathcal{E}$ *of arcs, an arc* $e \in \mathcal{E}$ *from node* i *to node* j *being denoted* (i, j)*;*
- $\underline{\mathcal{Q}} = (\underline{q}_{i,j})_{(i,j)\in\mathcal{E}}$ *and* $\overline{\mathcal{Q}} = (\overline{q}_{i,j})_{(i,j)\in\mathcal{E}}$ *are the vectors of normal (i.e., minimum) and maximum arc capacities, respectively;*
- $\mathcal{C} = (c_{i,j})_{(i,j)\in\mathcal{E}}$ *is the vector of unitary expansion costs ;*
- $\mathcal{A} = \{A_1, \ldots, A_u, \ldots, A_m, A_{m+1}\}$ *is the set of* $m+1$ *agents where* A_{m+1} *is the customer-agent and* A_u, $u \leq m$, *is transportation-agent* u *owning a set of* m_u *arcs, denoted* $\mathcal{E}_u$ *(with* $\mathcal{E}_{m+1} = \emptyset$ *and* $\mathcal{E}_u \cap \mathcal{E}_v = \emptyset$, $\forall (A_u, A_v) \in \mathcal{A}^2$, $u \neq v$*) ;*
- π *is the reward given by customer-agent* A_{m+1} *to the transportation-agents for every additional circulating unit of flow.*

Definition 2. Strategy Vector
Given a `MAN`, a strategy vector $\mathcal{S} = (S_1, \ldots, S_m, S_{m+1})$ *gathers the individual agents' strategies so that:*

- *if* $u \leq m$, *then* $S_u = (q_{i,j})_{(i,j)\in\mathcal{E}_u}$ *corresponds to the strategy of transportation-agent* A_u, $q_{i,j} \in [\underline{q}_{i,j}, \overline{q}_{i,j}]$ *being the arc capacity chosen for* (i,j) *by* A_u*;*
- *if* $u = m+1$, *then* $S_{m+1} = (w_u)_{u=1,\ldots,m}$ *is the strategy of customer-agent* A_{m+1}, w_u *being the part of reward* π *allocated by* A_{m+1} *to* A_u, *with* $\sum_{u=1}^{m} w_u = 1$.

We denote by $\underline{\mathcal{S}}$ any particular strategy for which $q_{i,j} = \underline{q}_{i,j}$, $\forall (i,j) \in \mathcal{E}$, whatever the strategy of customer-agent S_{m+1}.

Given a strategy $\mathcal{S}$, $\mathcal{F}(\mathcal{S}) = (f_{i,j})_{(i,j)\in\mathcal{E}}$ denotes a *maximum flow* vector, which respects the current values of capacities (i.e., $f_{i,j} \leq q_{i,j}$, $\forall (i,j) \in \mathcal{E}$). The maximum flow value is $F(\mathcal{S}) = \sum_{(s,j)\in\mathcal{E}} f_{s,j}$ (of course, considering one specific strategy $\mathcal{S}$, $\mathcal{F}(\mathcal{S})$ and $F(\mathcal{S})$ can be computed in polynomial time using the well-known Edmonds-Karp or Dinic algorithms [11, 12]). We further refer to $\underline{v}$ as the maximum flow value under a strategy $\underline{\mathcal{S}}$ (in other words, $\underline{v}$ is the largest possible flow value at zero cost).

Definition 3. Network Expansion Game
A `NEG` *is defined by a* `MAN`*, a strategy vector* $\mathcal{S}$ *and a profit vector* $Z(\mathcal{S}) = (Z_1(\mathcal{S}), \ldots, Z_m(\mathcal{S}), Z_{m+1}(\mathcal{S}))$ *, so that:*

- *if* $u \leq m$, $Z_u(\mathcal{S}) = w_u \pi(F(\mathcal{S}) - \underline{v}) - \sum_{(i,j)\in\mathcal{E}_u} c_{i,j}(q_{i,j} - \underline{q}_{i,j})$ *is the profit of transportation-agent* A_u*, which equals the difference between its part of reward and spending;*
- *if* $u = m+1$, *then* $Z_{m+1}(\mathcal{S}) = F(\mathcal{S})$ *is the profit of customer-agent* A_{m+1}*, which equals the value of the maximum flow.*

2.2 Nash equilibria and poor strategies

We next define some important concepts used throughout the paper. Given a strategy vector $\mathcal{S} = (S_1, \ldots, S_{m+1})$, we denote by $\mathcal{S}_{-u}$ the set of strategies played by the m agents except A_u, i.e., $\mathcal{S}_{-u} = (S_1, \ldots, S_{u-1}, S_{u+1}, \ldots S_{m+1})$. Focusing on a particular agent A_u, $\mathcal{S}$ can also be written as the pair $(S_u, \mathcal{S}_{-u})$.

A strategy vector is a *Nash equilibrium* and is said to be *stable* if there is no incentive for *any agent* to unilaterally modify its strategy in order to improve its profit (see [21], [22], [23] and [28]). More formally:

Definition 4. Nash equilibria
Given a `NEG`*, a strategy vector* $\mathcal{S} = (S_1, \ldots, S_m, S_{m+1})$ *is a Nash equilibrium if and only if for any agent* $A_u \in \mathcal{A}$*, choosing any another strategy* S'_u *would not increase its profit. That is:*

$$\forall A_u \in \mathcal{A}, \forall S'_u \neq S_u, \;\; Z_u(S_u, \mathcal{S}_{-u}) \geq Z_u(S'_u, \mathcal{S}_{-u}). \tag{1}$$

Clearly, the customer-agent is interested in finding the highest stable flow (hence, gain) it can expect for a given reward π. We can therefore define the problem addressed in this paper as follows.

Definition 5. Problem MaxNash
Given a `NEG`*, find a strategy vector* $\mathcal{S}^*$ *such that*

$$F(\mathcal{S}^*) = \max\{F(\mathcal{S}) : \mathcal{S} \text{ is a Nash equilibrium}\}.$$

We refer to $\mathcal{S}^N$ as the set of Nash equilibria. Let us now define another concept which turns out useful for properly characterizing Nash equilibria.

Definition 6. Poor strategy
A strategy vector $\mathcal{S}$ *with flow value* $F(\mathcal{S}) > \underline{v}$ *is said poor if there exists an agent* A_u *having an alternative strategy* S'_u *with flow value* $F(S'_u, \mathcal{S}_{-u}) = F(\mathcal{S})$ *such that* $Z_u(S'_u, \mathcal{S}_{-u}) > Z_u(\mathcal{S})$.

In other words, because the customer-agent cannot increase its profit as $F(\mathcal{S}') = F(\mathcal{S})$, a strategy vector is poor when one transportation-agent is able to find a better strategy, while keeping the network flow value unchanged. We denote the set of non-poor strategies as $\mathcal{S}^{np}$. Obviously, a poor strategy cannot be a Nash equilibrium, i.e., $\mathcal{S}^N \subseteq \mathcal{S}^{np}$.

Proposition 1. *Given a non-poor strategy vector* $\mathcal{S}$ *and a corresponding maximum flow* $\mathcal{F}(\mathcal{S})$*, for each arc* $(i,j) \in \mathcal{E}$*, either* $f_{i,j} = q_{i,j}$ *or* $q_{i,j} = \underline{q}_{i,j}$.

Proof. If in $\mathcal{F}(\mathcal{S})$ an arc (i,j) is not saturated and $q_{i,j} > \underline{q}_{i,j}$, the agent owning arc (i,j) can profitably decrease the arc capacity until it reaches $\max\{f_{i,j}, \underline{q}_{i,j}\}$, without affecting the flow value $F(\mathcal{S})$. This implies that $\mathcal{S}$ is a poor strategy. □

2.3 Mathematical model

Problem MaxNash can be formulated as the following mathematical program.

$$\begin{array}{ll}
Max\ F & \\
s.c. & \\
(i) & \sum_{(i,j)\in\mathcal{E}_i^+} f_{i,j} - \sum_{(j,i)\in\mathcal{E}_i^-} f_{j,i} = \begin{cases} 0\ , \forall i \neq s,t \\ F\ , i = s \\ -F\ , i = t \end{cases} \\
(ii) & 0 \leq f_{i,j} \leq q_{i,j},\ \forall (i,j) \in \mathcal{E} \\
(iii) & \underline{q}_{i,j} \leq q_{i,j} \leq \overline{q}_{i,j},\ \forall (i,j) \in \mathcal{E} \\
(iv) & \sum_{A_u \in \mathcal{A}} w_u = 1 \\
(v) & \mathcal{S} = \big((q_{i,j})_{(i,j)\in\mathcal{E}}, (w_u)_{u=1,\ldots,m}\big) \text{ is a Nash equilibrium} \\
 & q_{i,j} \in \mathbb{N},\ f_{i,j} \in \mathbb{N},\ \forall (i,j) \in \mathcal{E} \\
 & w_u \in \mathbb{R},\ \forall A_u \in \mathcal{A} \\
 & F \in \mathbb{N}
\end{array} \tag{2}$$

Constraints (i) and (ii) are classical flow conservation and capacity constraints, where $\mathcal{E}_i^+$ and $\mathcal{E}_i^-$ respectively denote the set of outgoing and ingoing

arcs from node i. Constraints (iii) impose restrictions on capacity values. Constraint (iv) enforces w_u to be a sharing policy. We observe that constraints $(i) - (iv)$ are linear. Eventually, the *complex* and non-explicit constraint (v) enforces the strategy $\mathcal{S}$ to be a Nash equilibrium.

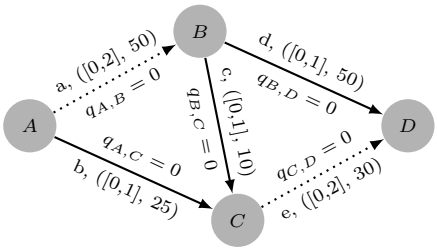
In Section 5, we show how the previous mathematical program can be modeled as a Mixed Integer Linear Program (MILP), replacing constraint (v) by a series of primal-dual linear constraints.

2.4 Illustrative example

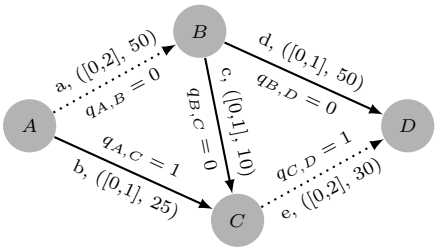
Let us get back to the previous example (see figure 1). We consider two different sharing policies $W^{(1)} = (\frac{1}{2}, \frac{1}{2})$ and $W^{(2)} = (\frac{1}{4}, \frac{3}{4})$. In this network, when all the capacities are normal ($\underline{q}_{i,j} = 0$), the maximum flow that can circulate is 0, which corresponds to the strategy vector $\underline{\mathcal{S}}$. Under such strategy, the value of the maximum flow is 3.

Let us consider the strategy vector $\mathcal{S}_0 = \underline{\mathcal{S}}$ (see figure 2(a)) and the first sharing policy $W^{(1)}$. The reward to each agent is 60 for each unit increase in the flow. It is possible to increase the arc capacities along the path $\{(A, C), (C, D)\}$, which leads to strategy $\mathcal{S}_1 = (0, 1, 0, 0, 1, W^{(1)})$ (see Figure 2(b)), leading to the maximum flow value $F(\mathcal{S}_1) = 1$ and $Z_1(\mathcal{S}_1) = 35$ and $Z_2(\mathcal{S}_1) = 30$. From this strategy, it is still possible to increase the arc capacities, hence the flow, along the path $\{(A, B), (B, D)\}$, which leads to strategy vector $\mathcal{S}_2 = (1, 1, 0, 1, 1, W^{(1)})$ (see Figure 2(c)), with maximum flow value $F(\mathcal{S}_2) = 2$ and profits $Z_1(\mathcal{S}_2) = 45$ and $Z_2(\mathcal{S}_2) = 40$. Nevertheless, we observe that strategy $\mathcal{S}_2$ is not stable since the capacities can be unilaterally decreased along path $\{(D, B), (B, C), (C, A)\}$ by agent A_1, leading to strategy vector $\mathcal{S}_3 = (1, 0, 1, 0, 1, W^{(1)})$ (see Figure 2(d)) with maximum flow value $F(\mathcal{S}_3) = 1$, which gives A_1 a higher profit (i.e., $Z_1(\mathcal{S}_3) = 50 > Z_1(\mathcal{S}_2)$), to the detriment of A_2 ($Z_2(\mathcal{S}_3) = -20$). Consequently, although with strategy $\mathcal{S}_2$ all agents (including the customer) get a better profit than under strategy $\mathcal{S}_1$, $\mathcal{S}_2$ is not a Nash equilibrium. Eventually, one can observe that only strategy $\mathcal{S}_1$ is a Nash equilibrium (under sharing policy $W^{(1)}$).

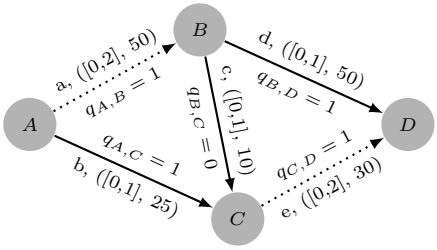
Let us consider now sharing policy $W^{(2)}$ and the initial strategy vector $\mathcal{S}_0$. It is possible to increase the flow by two units using respectively paths $\{(A, C), (C, D)\}$ and $\{(A, B), (B, C), (C, D)\}$, which leads to the strategies $\mathcal{S}'_1 = (0, 1, 0, 0, 1, W^{(2)})$ (see Figure 3(a)) and $\mathcal{S}'_2 = (1, 1, 1, 0, 2, W^{(2)})$ (see Figure 3(b)), respectively. The transportation-agent profits equal $Z_1(\mathcal{S}'_2) = 25$ and $Z_2(\mathcal{S}'_2) = 70$. Now, strategy $\mathcal{S}'_2$ is stable since no agent can unilaterally and profitably modify its arc capacities. Moreover, it is also a best Nash equilibrium with respect to the the flow value, as there does not exist obviously any other stable strategy vector providing a higher flow value. For the customer-agent, the sharing policy $W^{(2)}$ is preferable to $W^{(1)}$ since it allows him/her to reach a better Nash equilibrium.



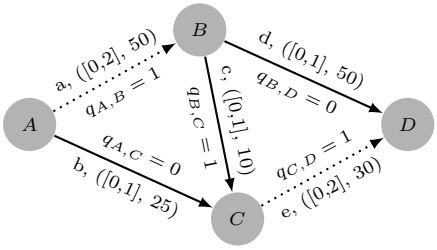
(a) Strategy $\mathcal{S}_0$



(b) Strategy $\mathcal{S}_1$

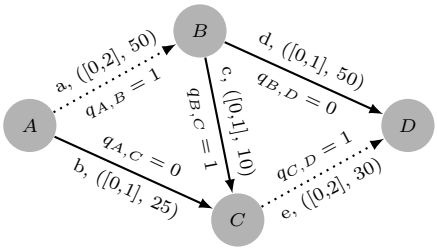


(c) Strategy $\mathcal{S}_2$

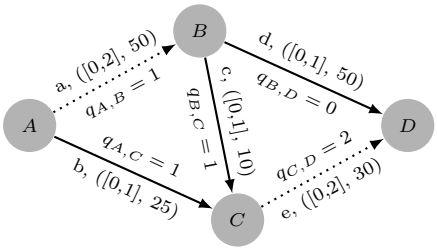


(d) Strategy $\mathcal{S}_3$

Fig. 2. Strategies with $W^{(1)}$



(a) Strategy $\mathcal{S}'_1$



(b) Strategy $\mathcal{S}'_2$

Fig. 3. Strategies with $W^{(2)}$

2.5 Basic properties

Proposition 2. *Given any strategy $\mathcal{S} \in S^{np}$ having flow value $F(\mathcal{S})$, the customer-agent A_{m+1} cannot unilaterally improve its profit.*

Proof. The only way for the customer-agent to *unilaterally* improve its profit is to increase the flow, which is impossible since $F(\mathcal{S})$ is maximum by definition. □

Proposition 2 allows to restrict Definition 4 of a Nash equilibrium as follows:

Corollary 1. *Given a* `NEG` *instance, a strategy $\mathcal{S} = (S_1, \ldots, S_m, S_{m+1})$ is a Nash equilibrium if and only if $\mathcal{S} \in S^{np}$ and, for any transportation-agent $A_u \in$*

$\mathcal{A}$, $u \leq m$, *with strategy* S_u, *the following inequation holds:*

$$Z_u(S_u, \mathcal{S}_{-u}) \geq Z_u(S'_u, \mathcal{S}_{-u}),\ \forall S'_u \neq S_u.$$

3 The Single Transportation-agent Case

This section focuses on the particular network expansion game involving a single transportation-agent and a single customer-agent. In this case, all the arcs belong to the same transportation-agent A_1, who receives the entire reward (i.e., $w_1 = 1$). We next consider how the total flow can be either increased or decreased profitably for the transportation-agent, using properly defined increasing and decreasing paths in terms of capacities. These notions will be generalized in Section 4 to the multi-transportation-agents case.

3.1 Residual graph

In order to characterize the flow variation in this single-agent network expansion game, we define below the notion of a residual graph. Arcs of the residual graph represent the possibility of varying the capacity and the flow through that arc, given a current strategy and the corresponding maximum flow vector.

Definition 7. Residual graph
Given a `NEG` *instance defined by a MAN* $< G, \underline{\mathcal{Q}}, \overline{\mathcal{Q}}, \mathcal{C}, \mathcal{A}, \pi >$ *and given a non-poor strategy vector* $\mathcal{S}$ *and a maximum flow vector* $\mathcal{F}(\mathcal{S}) = (f_{i,j})_{(i,j)\in\mathcal{E}}$ *of value* $F(\mathcal{S})$, *the corresponding* residual graph $G_r(\mathcal{F}(\mathcal{S})) = (\mathcal{V}, \mathcal{E}_F, \mathcal{E}_B)$ *is defined as follows: for each arc* $(i,j) \in \mathcal{E}$, *having a capacity* $q_{i,j}$ *and traversed by the flow* $f_{i,j}$ *in* G, *there are two arcs in* G_r *: a forward arc* $(i,j) \in \mathcal{E}_F$ *and a backward arc* $(j,i) \in \mathcal{E}_B$.

- *The forward arc* (i,j) *has cost* $d_F^{i,j}$ *and residual capacity* $r_F^{i,j} = \overline{q}_{i,j} - q_{i,j}$ *if* $f_{i,j} = q_{i,j} (\geq \underline{q}_{i,j})$ *or* $r_F^{i,j} = \underline{q}_{i,j} - f_{i,j}$ *if* $f_{i,j} < \underline{q}_{i,j}$.
- *The backward arc* (j,i) *has cost* $d_B^{j,i}$ *and residual capacity* $r_B^{j,i} = q_{i,j} - \underline{q}_{i,j}$ *if* $f_{i,j} = q_{i,j} (\geq \underline{q}_{i,j})$ *or* $r_B^{j,i} = f_{i,j}$ *if* $f_{i,j} < \underline{q}_{i,j}$.

The costs $d_F^{i,j}$ *and* $d_B^{j,i}$ *of forward and backward arc, respectively, in the residual graph* $G_r(\mathcal{F}(\mathcal{S}))$ *are defined as follows:*

$$d_F^{i,j} = \begin{cases} +\infty \text{ if } f_{i,j} = q_{i,j} = \overline{q}_{i,j} \\ c_{i,j} \text{ if } f_{i,j} = q_{i,j} < \overline{q}_{i,j} \\ 0 \text{ if } f_{i,j} < q_{i,j} = \underline{q}_{i,j} \end{cases} \tag{3}$$

$$d_B^{j,i} = \begin{cases} +\infty \text{ if } f_{i,j} = 0 \\ -c_{i,j} \text{ if } f_{i,j} = q_{i,j} > \underline{q}_{i,j} \\ 0 \text{ if } 0 < f_{i,j} \leq q_{i,j} = \underline{q}_{i,j} \end{cases} \tag{4}$$

Costs and residual capacities correspond to all possibilities of varying the flow through the arcs of the network, while maintaining the strategy vector non-poor.

A forward cost $d_F^{i,j}$ is incurred when we want to increase the flow on arc $(i,j) \in G$ by one unit. Recall that if $q_{i,j} > \underline{q}_{i,j}$, the arc is saturated (see Property 1). Hence, if arc $(i,j) \in G$ is already at its maximum capacity, no further flow increase is possible, so $r_F^{i,j} = 0$ and $d_F^{i,j} = +\infty$. If $q_{i,j} < \overline{q}_{i,j}$, flow can be increased by at most $r_F^{i,j} = \overline{q}_{i,j} - q_{i,j}$, by increasing the capacity at a unit cost of $d_F^{i,j} = c_{i,j}$.

If arc $(i,j) \in G$ is not saturated, it is possible to increase its flow at no cost until $\underline{q}_{i,j}$ is reached. Since $\mathcal{S}$ is non-poor, this can only occur when $q_{i,j} = \underline{q}_{i,j}$ (see Property 1), so the residual capacity is indeed $\underline{q}_{i,j} - f_{i,j}$.

Similarly, a backward cost $d_B^{i,j}$ is incurred when we want to decrease the flow on arc $(i,j) \in G$ by one unit. Obviously, if an arc $(i,j) \in G$ is empty, no decrease can occur, so $r_B^{i,j} = 0$ and $d_B^{i,j} = +\infty$. If $0 < f_{i,j} \leq q_{i,j}$, then from Property 1 $q_{i,j} = \underline{q}_{i,j}$, and hence the flow in the corresponding arc of G can be decreased (by at most $f_{i,j}$) but this brings no benefit ($d_B^{i,j} = 0$). Finally, if $(i,j) \in G$ is saturated and $q_{i,j} > \underline{q}_{i,j}$, the transportation-agent can save $c_{i,j}$ by decreasing the arc capacity, which is possible until the capacity reaches its normal value $\underline{q}_{i,j}$.

In what follows, with no loss of generality, we assume that arcs having 0 residual capacity are discarded from $G_r(\mathcal{F}(\mathcal{S}))$.

3.2 Modification of the Network flow

In this section, we address the following problem. Given a current strategy vector $\mathcal{S}$ and a current maximum flow vector $\mathcal{F}(\mathcal{S})$, what is the most profitable set of arcs' capacities to modify in order to increase or decrease the overall flow by one unit, given that all capacities are integers?

Let us consider the residual graph $G_r(\mathcal{F}(\mathcal{S}))$, with residual costs specified by equations (3)-(4). The problem of finding an optimal minimal modification of the network flow can be formulated as a linear program as follows:

$$
\begin{array}{ll}
\min \sum_{(i,j)\in\mathcal{E}} (a_{i,j} d_F^{i,j} + b_{i,j} d_B^{j,i}) & \\
s.t. & \\
(i) \quad \sum_{(i,j)\in\mathcal{E}_i^+} \varphi_{i,j} - \sum_{(j,i)\in\mathcal{E}_i^-} \varphi_{j,i} = \pm \begin{cases} 0 \; \forall i \neq s,t \\ 1 \; , \; i = s \\ -1 \; , \; i = t \end{cases} & \\
(ii) \quad \varphi_{i,j} = a_{i,j} - b_{i,j}, \; \forall (i,j) \in \mathcal{E} & \\
\qquad a_{i,j} \geq 0, \; \forall (i,j) \in \mathcal{E} & \\
\qquad b_{i,j} \geq 0, \; \forall (i,j) \in \mathcal{E} &
\end{array}
\tag{5}
$$

Variables $\varphi_{i,j}$ (for $(i,j) \in \mathcal{E}$) represent flow variation, with respect to the current value $f_{i,j}$ on the original graph G. Variables $a_{i,j}$ and $b_{i,j}$ denote the value of flow increase and decrease of arc (i,j) with respect to $f_{i,j}$, respectively. The objective

function expresses the cost of modifying the flow by one unit in the network. Recall that the information concerning the current strategy vector $\mathcal{S}$ (capacities and flows) is recorded into the parameters $d_F^{i,j}$ and $d_B^{i,j}$.

Constraints (i) express flow conservation at each node in the network when increasing (if "+" is selected in the right-hand-side of the constraint) or decreasing (if "−" is selected in the right-hand-side of the constraint) the overall flow by one unit.

Constraints (ii) express flow modification on each arc as the difference between increase and decrease.

A feasible solution to the linear program (5) defines a set of arcs such that, modifying their flows according to the values $a_{i,j}$ and $b_{i,j}$, the overall flow increases (if "+" is selected) or decreases (if "−" is selected).

Proposition 3. *Given a non-poor strategy vector $\mathcal{S}$, linear program (5) has either a finite optimal solution or no solution.*

Proof. We show the proof for the case in which "+" is selected in problem (5), the other case being perfectly symmetric.

Note that if no path exists from the source to the sink in $G_r(\mathcal{F}(\mathcal{S}))$, problem (5) is infeasible, which means that the flow cannot be increased anymore. Otherwise, (5) always admits a feasible solution. Suppose now by contradiction that problem (5) is unbounded. Then, there exists a set $\overline{E}$ of arcs such that either $a_{i,j}$ or $b_{i,j}$ goes to infinity. With no loss of generality, we can express these values as $a_{i,j} = \overline{a}_{i,j}M$ and $b_{i,j} = \overline{b}_{i,j}M$, where M is arbitrarily large. If the problem is unbounded, it means that

$$\sum_{(i,j)\in\overline{E}} \overline{a}_{i,j} d_F^{i,j} + \sum_{(i,j)\in\overline{E}} \overline{b}_{i,j} d_B^{j,i} < 0. \tag{6}$$

However, even if some variables go to infinity, the net flow increase from s to t is 1, due to constraints (i) written for s and t. As a consequence, there must be at least one $\mathcal{S}$ and $\mathcal{F}(\mathcal{S})$ on G that allow increasing by ε the flow on all arcs $(i,j) \in \mathcal{E}$ such that, $(i,j) \in \mathcal{E}_F \cap \overline{E}$ in G_r, and decrease by ε the flow on all arcs $(i,j) \in \mathcal{E}$ such that $(j,i) \in \mathcal{E}_B \cap \overline{E}$ in G_r. In this case one would get a new flow having the same value $F(\mathcal{S})$, but the transportation-agent would increase its profit, because of (6). However, this can only occur if $\mathcal{S}$ is a poor strategy. □

The dual of linear program (5) is:

$$\begin{array}{ll} \max \pm(\tau_t - \tau_s) & \\ s.t. & \\ (i) & \tau_j - \tau_i \le d_F^{i,j},\ \forall (i,j) \in \mathcal{E} \\ (ii) & \tau_i - \tau_j \le d_B^{j,i},\ \forall (i,j) \in \mathcal{E} \\ & \tau_i \in \mathbb{R},\ \forall i \in \mathcal{V} \end{array} \tag{7}$$

We note that the dual (7) can be interpreted as a shortest path problem *on the residual graph*. In fact, for each arc $(i,j) \in \mathcal{E}$, there is a pair of constraints

$(i) - (ii)$. Now associate with each $(i,j) \in \mathcal{E}$ a forward edge $(i,j) \in \mathcal{E}_F$ and a backward edge $(j,i) \in \mathcal{E}_B$, having weights $d_F^{i,j}$ and $d_B^{j,i}$, respectively. More precisely, if we are interested in *increasing* the network flow at minimum cost, the dual objective function is $\tau_t - \tau_s$, and hence, viewing variables τ_i as node potentials (see [9, p.26]), linear program (7) is exactly the problem of finding *the shortest path from s to t* on the residual graph. Viceversa, if we wish to *decrease* network flow maximizing the saving, then the dual objective function becomes $\tau_s - \tau_t$, and the problem is therefore to find *the shortest path from t to s* on the residual graph. In both cases, the optimal solution to (5) is specified by a path, and the length of the path on the residual graph equals the cost of modifying by one unit the flow throughout the network. This justifies the following definitions.

Definition 8. Increasing path
Given a non-poor strategy vector $\mathcal{S}$ and a maximum flow vector $\mathcal{F}(\mathcal{S})$, an increasing path *is a path P in $G_r(\mathcal{F}(\mathcal{S}))$ from the source node s to the sink node t.*

In fact, consider a strategy vector $\mathcal{S}$ and the corresponding maximum flow vector $\mathcal{F}(\mathcal{S})$. If one increases the flow on the arcs of G corresponding to forward arcs in G_r by the amount q, and decreases the flow on the arcs of G corresponding to backward arcs in G_r by the same amount q, we obtain a new strategy $\mathcal{S}'$ such that $F(\mathcal{S}') = F(\mathcal{S}) + q$. The greatest flow increase that can be achieved using a certain increasing path P is $\theta = \min\{r_F^{i,j}, r_B^{i,j} : (i,j) \in P\}$. Denoting forward and backward arcs in P as P^+ and P^- respectively, the cost $cost(P)$ of increasing the flow by one unit using an increasing path $P \in \mathcal{P}$ is given by:

$$cost(P) = \sum_{(i,j)\in P^+} d_F^{i,j} + \sum_{(i,j)\in P^-} d_B^{i,j} \tag{8}$$

Note that such increasing operation entails adjusting the capacities of the arcs so that the resulting strategy vector is also non-poor. Of course, if t is not reachable from s on G_r, then no increasing path exists and hence no flow increase can be carried out.

Symmetrical considerations hold if we want to decrease the flow on G.

Definition 9. Decreasing path
Given a non-poor strategy vector $\mathcal{S}$ and a maximum flow vector $\mathcal{F}(\mathcal{S})$, a decreasing path *is a path $\bar{P}$ in $G_r(\mathcal{F}(\mathcal{S}))$ from the the sink node t to the source node s.*

Again, decreasing by q the flow on the arcs of G corresponding to backward arcs in G_r, and increasing by the same amount q the flow on the arcs of G corresponding to forward arcs in G_r, we obtain a new strategy $\mathcal{S}'$ such that $F(\mathcal{S}') = F(\mathcal{S}) - q$. The greatest flow decrease that can be achieved using a path $\bar{P}$ is again given by $\bar{\theta} = \min\{r_F^{i,j}, r_B^{i,j} : (i,j) \in \bar{P}\}$.

Denoting forward and backward arcs in $\bar{P}$ as $\bar{P}^+$ and $\bar{P}^-$ respectively, the saving $sav(P)$ attained from decreasing the flow by one unit using a decreasing

path $\bar{P}$ is given by:

$$sav(\bar{P}) = -\left[\sum_{(i,j)\in\bar{P}^+} d_F^{i,j} + \sum_{(i,j)\in\bar{P}^-} d_B^{i,j}\right] \tag{9}$$

The decreasing operation entails adjusting the capacities of the arcs so that the resulting strategy is still non-poor. If s is not reachable from t on G_r, then no decreasing path exists and hence no flow decrease can be carried out, which obviously occurs only when $F(\mathcal{S}) = 0$.

In any case, finding the most profitable increasing or decreasing path consists in solving a shortest path problem, and can be therefore done in polynomial time. Moreover, notice that if P^* and $\bar{P}^*$ are the optimal increasing and decreasing paths respectively, one has that $cost(P^*) \geq 0$ and $sav(\bar{P}^*) \geq 0$.

3.3 Example

Consider the network $G(\mathcal{V}, \mathcal{E})$, displayed in Fig. 4, composed of the node set $\mathcal{V} = \{s, 1, 2, 3, 4, t\}$ and the arc set $\mathcal{E} = \{a, b, c, d, e, f, g\}$, belonging to a single transportation-agent. For each arc, the values $[\underline{q}_{i,j}, \overline{q}_{i,j}]$ and $c_{i,j}$ are reported. We assume a customer-agent's reward $\pi = 215$, smaller than the cost of a longest path, which equals 220. In order to increase the flow, an increasing path P such that $cost(P) \leq 215$ has to be found.

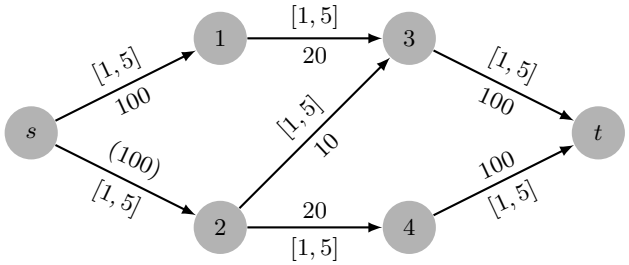
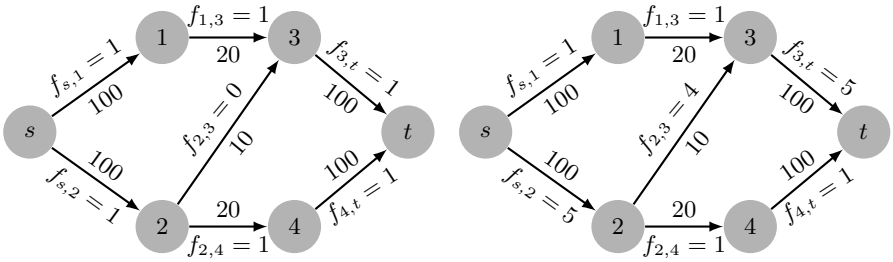


Fig. 4. A single transportation-agent network expansion game

Consider the initial strategy vector $\mathcal{S}_0 = \underline{\mathcal{S}}$ described in Fig. 5(a). The value of the maximum flow that circulates in the network at zero cost is $F(\mathcal{S}_0) = 2$. The only profitable way to increase the flow in the network is to use the increasing path $P = ((s, 2), (2, 3), (3, t))$ with cost $cost(P) = 210 \leq \pi$. The capacity of arcs $(s, 2)$ and $(3, t)$ is increased by 4 units, while the capacity of $(2, 3)$ is increased by 3 units. In fact, notice that for arc $(2, 3)$ the first unit increase is free (since $\underline{q}_{2,3} = 1$). It yields the new strategy $\mathcal{S}_1 = (1, 5, 1, 4, 1, 5, 1)$ (see Figure 5(b)) with $F(\mathcal{S}_1) = 6$ and cost 830. Let us remark that path $P' = (s, 2), (2, 4), (4, t)$ is not profitable as $cost(P') = 220 > \pi$.



(a) Strategy $\mathcal{S}_0$: Flow at zero cost (b) strategy $\mathcal{S}_1$

Fig. 5. Flow variation in the network of Fig.4

4 The Multi Transportation-agents Case

In what follows, we extend some of the concepts reviewed for the single-agent case to the multi-agent case. We first introduce a *multi-agent residual graph*, then use it to define agent-specific increasing and decreasing paths, and finally we characterize Nash equilibria.

In a NEG involving several transportation-agents, any agent A_u may increase (or decrease) its arc capacities if it improves its profit Z_u. Since A_{m+1} can never unilaterally improve its profit Z_{m+1} (see proposition 2), the focus is only on transportation-agents. Therefore, for simplicity in this section we use the term *agent* to refer to a transportation-agent, unless otherwise specified.

4.1 Multi-agent residual graph

In the multi-agent case, we define a multi-agent residual graph such that the arcs have the same residual capacities as in the single-agent case, but different residual costs, as they now depend on the considered agent A_u.

Definition 10. Multi-agent residual graph
Given a NEG instance defined by a MAN $< G, \underline{\mathcal{Q}}, \overline{\mathcal{Q}}, \mathcal{C}, \mathcal{A}, \pi >$*, a non-poor strategy vector* $\mathcal{S} \in \mathcal{S}^{np}$*, a flow vector* $\mathcal{F}(\mathcal{S})$*, a multi-agent residual graph* $G_r^{\mathcal{A}}(\mathcal{F}(\mathcal{S})) = (\mathcal{V}, \mathcal{E}_F^{\mathcal{A}}, \mathcal{E}_B^{\mathcal{A}})$ *is defined as follows. For each arc* $(i,j) \in \mathcal{E}$*, there are two arcs, namely a forward arc* $(i,j) \in \mathcal{E}_F^{\mathcal{A}}$ *and a backward arc* $(j,i) \in \mathcal{E}_B^{\mathcal{A}}$*. The forward arc* (i,j) *(backward arc* (j,i)*, resp.) is associated with a cost vector* $\Delta_F^{i,j} = (\delta_F^{i,j,u})_{u=1,\ldots,m}$ *(*$\Delta_B^{j,i} = (\delta_B^{j,i,u})_{u=1,\ldots,m}$*,resp.) and the residual capacity* $r_F^{i,j}$ *(*$r_B^{j,i}$*, resp.). The equations that give the residual capacities of forward and backward arcs are identical to those of the single-agent case, i.e.:*

- $r_F^{i,j} = \overline{q}_{i,j} - q_{i,j}$ *if* $f_{i,j} = q_{i,j} (\geq \underline{q}_{i,j})$ *or* $r_F^{i,j} = \underline{q}_{i,j} - f_{i,j}$ *if* $f_{i,j} < \underline{q}_{i,j}$*;*
- $r_B^{j,i} = q_{i,j} - \underline{q}_{i,j}$ *if* $f_{i,j} = q_{i,j} (\geq \underline{q}_{i,j})$ *or* $r_B^{j,i} = f_{i,j}$ *if* $f_{i,j} < \underline{q}_{i,j}$*.*

The cost vectors are calculated as follows:

- $\Delta_F^{i,j}$:
  - if $0 \leq f_{i,j} < \underline{q}_{i,j}$, then $\delta_F^{i,j,u} = 0$;
  - if $\underline{q}_{i,j} \leq q_{i,j} = f_{i,j} < \overline{q}_{i,j}$, then $\delta_F^{i,j,u} = c_{i,j}$ whether $(i,j) \in \mathcal{E}_u$, $+\infty$ otherwise;
  - if $f_{i,j} = \overline{q}_{i,j}$, then $\delta_F^{i,j,u} = +\infty$;
- $\Delta_B^{j,i}$:
  - if $f_{i,j} = 0$, then $\delta_B^{j,i,u} = +\infty$;
  - if $0 < f_{i,j} \leq \underline{q}_{i,j}$, then $\delta_B^{j,i,u} = 0$;
  - if $\underline{q}_{i,j} < q_{i,j} = f_{i,j} \leq \overline{q}_{i,j}$, then $\delta_B^{j,i,u} = -c_{i,j}$ whether $(i,j) \in \mathcal{E}_u$, 0 otherwise.

The residual costs still reflect the possibility to use forward and backward arcs in order to increase or decrease the flow. An arc having residual cost equal to $+\infty$ can indeed be discarded from the residual network.

For forward arcs, a cost $\delta_F^{i,j,u} = c_{i,j}$ reflects the fact that the arc belonging to agent A_u can have its flow locally increased at cost $c_{i,j}$ by expanding $q_{i,j}$ (i.e., $q_{i,j} < \overline{q}_{i,j}$). A cost of $+\infty$ occurs if either the arc belongs to agent A_u and it is no longer possible to increase the flow (i.e., $f_{i,j} = \overline{q}_{i,j}$), or the arc does not belong to agent A_u and it is not possible to increase the flow at zero cost (i.e., $\underline{q}_{i,j} \leq f_{i,j} \leq \overline{q}_{i,j}$). A forward arc having cost equal to 0 models the fact that one can increase the flow with zero cost since the arc (i,j) is not saturated (i.e., $0 \leq f_{i,j} < q_{i,j}$), no matter if the arc belongs to the agent or not. For a non-poor strategy vector, this can only occur when $q_{i,j} = \underline{q}_{i,j}$.

Similarly, a backward arc having null cost indicates that the flow in the corresponding arc of G can be decreased but this brings no benefit, since the arc capacity cannot be decreased. If the arc belongs to agent A_u, this happens if one can decrease the flow ($f_{i,j} \neq 0$) with no cost (i.e., $0 < f_{i,j} \leq \underline{q}_{i,j}$). If the arc does not belong to agent A_u, this happens as long as we can decrease the flow ($f_{i,j} \neq 0$). A cost $-c_{i,j}$ reflects the fact that A_u can locally make a saving $c_{i,j}$ by decreasing $q_{i,j}$. This occurs when the arc (i,j) belongs to A_u, it is saturated (i.e., $f_{i,j} = q_{i,j}$) and the capacity can still be decreased (i.e., $q_{i,j} > \underline{q}_{i,j}$). Finally, setting the cost to ∞ models the impossibility of using this arc to decrease the flow (i.e., $f_{i,j} = 0$). As the definition of the residual capacities in $G_r^{\mathcal{A}}(\mathcal{F}(\mathcal{S}))$ is the same as for $G_r(\mathcal{F}(\mathcal{S})$, the notions of an increasing and decreasing path remain valid (see definitions (8) and (9)). An increasing or decreasing path still reflects a way to increase (decrease, resp.) the flow by modifying the capacity of the arcs of the path. However, as the arcs may now belong to different agents, their cost now depends on the agent considered.

4.2 Increasing the flow profitably

An augmenting path P from s to t in $G_r^{\mathcal{A}}(\mathcal{F}(\mathcal{S}))$ consists of a set of forward and backward arcs $P = \{P^+, P^-\}$ such that, by simultaneously increasing $q_{i,j}$ by

one unit $\forall(i,j) \in P^+$ and decreasing $q_{i,j}$ by one unit $\forall(i,j) \in P^-$, the overall flow increases by one unit. The cost of such modification is:

$$cost_u(P) = \sum_{(i,j)\in P^+} \delta_F^{i,j,u} + \sum_{(i,j)\in P^-} \delta_B^{i,j,u} \tag{10}$$

Let us now introduce the notion of a *profitable increasing path*.

Definition 11. Profitable increasing path
Given a non-poor strategy vector $\mathcal{S}$ and a maximum flow vector $\mathcal{F}(\mathcal{S})$, a profitable increasing path *is a path P in $G_r^{\mathcal{A}}(\mathcal{F}(\mathcal{S}))$* ***from the source node*** *s* ***to the sink node*** *t such that, for every A_u, the expansion cost is strictly lower than the payoff, i.e., $cost_u(P) < w_u\pi$.*

4.3 Decreasing the flow profitably

A decreasing path $\bar{P}$ in $G_r^{\mathcal{A}}(\mathcal{F}(\mathcal{S}))$ from t to s consists of a set of forward and backward arcs $\bar{P} = \{\bar{P}^+, \bar{P}^-\}$ such that by simultaneously increasing $q_{i,j}$ by one unit $\forall(i,j) \in \bar{P}^+$ and decreasing $q_{i,j}$ by one unit $\forall(i,j) \in \bar{P}^-$, it is the overall flow decreases by one unit. The saving induced by such a modification depends on the considered agent and can be defined as follows:

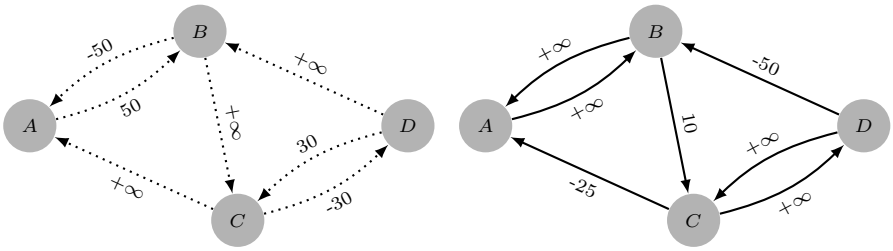
$$sav_u(\bar{P}) = -\left[\sum_{(i,j)\in\bar{P}^+} \delta_F^{i,j,u} + \sum_{(i,j)\in\bar{P}^-} \delta_B^{i,j,u}\right] \tag{11}$$

Considering a transportation-agent A_u, the saving $sav_u(\overline{P})$ generated by decreasing the flow by one unit through a decreasing path $\overline{P}$ is defined as follows.

Definition 12. Profitable decreasing path
Given a non-poor strategy vector $\mathcal{S}$ and a maximum flow vector $\mathcal{F}(\mathcal{S})$, a profitable decreasing path *is a path $\bar{P}$ in $G_r^{\mathcal{A}}(\mathcal{F}(\mathcal{S}))$* ***from the sink node*** *t* ***to the source node*** *s such that, for every agent A_u, the saving is greater than the reward loss, i.e.,*
$sav_u(\bar{P}) \geq w_u\pi$.

With no loss of generality we can make the realistic assumption that any agent A_u making no loss or gain (i.e., such that $sav_u(\bar{P}) = w_u\pi$) will not participate in any flow increase/decrease.

Let us consider the strategy depicted in the example of figure 2(c). The residual graphs of agents A_2 and A_1 corresponding to this strategy are given in figures 6(a) and 6(b), respectively. For A_2 (dotted arcs), the cost of the shortest path from D to A being equal to $+\infty$, there is obviously no existing profitable decreasing path. In contrast, for A_1 (plain arcs), the shortest path from D to A having a length of -65 (path D-B-C-A) and leading so to a saving 65 greater than 60 (part of reward allocated to A_1), there exists a decreasing profitable path (i.e., the flow is not stable).



(a) Residual graph for A_2 (b) Residual graph for A_1

Fig. 6. Residual graphs for strategy $\mathcal{S}_2$ of figure 2(c)

4.4 Stable strategies

We refer to the set of increasing and decreasing paths in $G_r^{\mathcal{A}}(\mathcal{F}(\mathcal{S}))$ as $\mathcal{P}^{\mathcal{A}}$ and $\bar{\mathcal{P}}^{\mathcal{A}}$, respectively.

Proposition 4. Nash Equilibrium strategy
Given a non-poor strategy vector $\mathcal{S}$ and the multi-agent reduced graph $G_r^{\mathcal{A}}(\mathcal{F}(\mathcal{S}))$, $\mathcal{S}$ is a Nash Equilibrium if and only if:

- $\forall P \in \mathcal{P}^{\mathcal{A}}$:
$$\nexists A_u \in \mathcal{A} \text{ such that } cost_u(P) < w_u\pi; \tag{12}$$
- *and* $\forall \bar{P} \in \bar{\mathcal{P}}^{\mathcal{A}}$:
$$\nexists A_u in \mathcal{A} \text{ such that } sav_u(\bar{P}) \geq w_u\pi. \tag{13}$$

Proof. From Proposition 2 and Corollary 1, the customer-agent A_{m+1} cannot unilaterally increase the value of the flow. Therefore, only the case of transportation-agents is of interest. Consider a strategy vector $\mathcal{S}$ and a transportation-agent A_u. If $\mathcal{S}$ is poor, then $\mathcal{S}$ is not a Nash equilibrium. If $\mathcal{S}$ is non-poor, A_u can only improve its situation by either increasing or decreasing the flow, and $\mathcal{S}$ is not stable. If it is possible, it means that there exists a profitable, either increasing (P) or decreasing ($\bar{P}$) path, in $G_r^{\mathcal{A}}(\mathcal{F}(\mathcal{S}))$. On the one hand, if such a profitable increasing path P exists, then it will mean that $cost_u(P) < w_u\pi$ which contradicts Condition (12). On the other hand, if a decreasing profitable path $\bar{P}$ exists, then $sav_u(\bar{P}) \geq w_u\pi$, which is in contradiction with Condition (13). □

4.5 Complexity issues

Finding a Nash equilibrium that maximizes the flow in the network in the special multi-agent case where each arc is managed by a specific agent was proved to be polynomial by Chaabane *et al.* in [7]. Finding a Nash equilibrium that maximizes the flow in the network when the profit sharing has been predefined is known to be NP-hard in the strong sense (see [6]). The proof was based on a reduction from

3-Partition, which assumed an equal sharing of the reward among the agents. We prove below that also the problem in which sharing coefficients have to be decided is NP-hard in the strong sense.

Proposition 5. *Problem MaxNash is strongly NP-hard.*

Proof. Let us consider the decision variant of MaxNash: is it possible to find a stable strategy vector $\mathcal{S}$ and a corresponding reward sharing policy such that $F(\mathcal{S}) > \varphi$, φ being an arbitrary integer value? The proof is based on a reduction from 3-Partition problem, which is known to be NP-complete in the strong sense [15]. A 3-partition problem consists in deciding whether a set $\zeta = \{a_0, \ldots, a_{K-1}\}$ of $K = 3k$ positive integers, such that $\sum_{l=0}^{K-1} a_l = kB$ and $a_l \in]B/4, B/2]$, can be partitioned into k subsets $\zeta_1, \ldots, \zeta_k$ such that the sum of integers in each subset equals B (note that, due to the bounds on the integer values, $|\zeta_i| = 3$ in any YES-instance). Obviously, MaxNash is in NP since, given a strategy vector $\mathcal{S}$, $F(\mathcal{S})$ can be determined in polynomial time using classical Max-Flow algorithms. From a 3-partition problem instance, we build up a network G with kK arcs and $K+1$ nodes. The network considers k agents. An agent $A_u \in \mathcal{A} = \{A_1, \ldots, A_k\}$ owns K arcs. The tail of an arc e_i is $V_{i \operatorname{div} K}$, its head is $V_{(i \operatorname{div} K)+1}$. Between any nodes $V_{i \operatorname{div} K}$ and $V_{i \operatorname{div} K+1}$, there are k parallel arcs, indexed from i to $(i+K)$ step k, each of them belonging to a specific agent: arc e_i belongs to $A_{i \operatorname{div} K}$. The cost of arc e_i is $c_{e_i} = a_{i \bmod K}$. In other words, for each positive integer $a_l \in \zeta$, we define k parallel arcs with same head and tail, maximum capacity $\overline{q}_{e_i} = 1$ and cost a_l. The total reward is set to $\pi = (B + \varepsilon)k$, ε being an arbitrary small positive value. In addition, an extra path from source node V_0 to sink node V_K is considered. It consists of k arcs starting from node V_0, each arc belonging to a specific agent $A_u \in \mathcal{A}$, having capacity $q_{i,j} \in [0, 1]$ and cost $c_{i,j} = B$. An illustration of this reduction is provided in Figure 7 with $\zeta = \{7, 7, 7, 7, 8, 8, 9, 9, 10\}$, $k = 3$ and $B = 24$. We want to establish whether a stable strategy exists having flow value greater or equal to $\varphi = k + 1$. We first show that a feasible MaxNash solution can be associated with a YES-instance of 3-Partition. First, it should be observed that the maximum flow value in the MaxNash instance is $k + 1$ (one unit of flow traverses every arc). Consequently, in any feasible solution, the extra path is also traversed by one unit of flow and, since this path must be profitable, it enforces $w_u = 1/m$. One should now find k profitable increasing paths using the remaining part of the network, respecting the constraint that $cost_u(P) < B + \epsilon$, $\forall P$. Therefore, any profitable path will necessarily traverse 3 arcs per agent such that the sum of the expansion costs of these arcs equals B, which specifies the solution to the 3-Partition instance. Let us consider now a NO-MaxNash Instance. Clearly, as the maximum flow equals $k + 1$, there is not any Nash equilibrium with such a flow value. It means that there does not exist any path P such that $cost_u(P) < B + \epsilon$, which implies the infeasibility of the 3-Partition instance. □

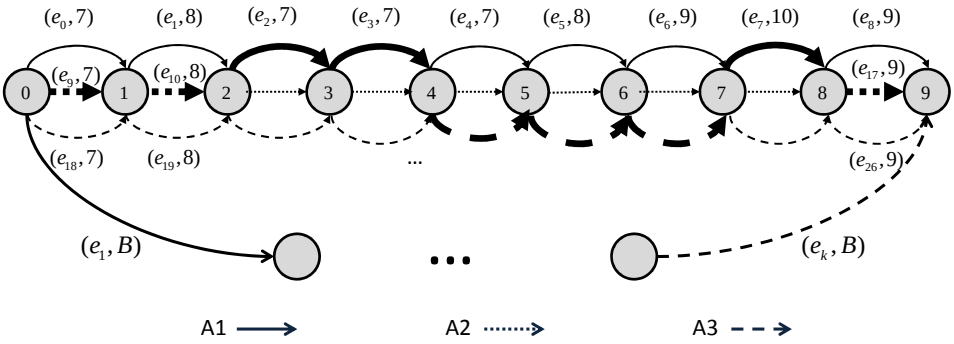


Fig. 7. Reduction from 3-PARTITION with $\zeta = \{7, 7, 7, 7, 8, 8, 9, 9, 10\}$, $k = 3$ and $B = 24$

5 Mathematical formulation

In this section, we show how Proposition 4 can be exploited to solve the problem. We resume mathematical formulation (2) of section 2, but we are now going to explicitly specify the constraints that a feasible solution must be a Nash equilibrium. As in (2), variables $q_{i,j}$ indicate the capacity and $f_{i,j}$ the flow in arc (i, j) respectively, while F is the value of the flow. The first four constraint sets $(i) - -(iv)$ are identical to (2), and express flow conservation and capacity limitations. Thereafter, we must add the constraints that the feasible solution be a Nash equilibrium. To this aim, from Proposition 4 we require that *no profitable decreasing path* $\bar{P}$ exists in $G_r^{\mathcal{A}}(\mathcal{F}(\mathcal{S}))$, as dictated by condition (13) of proposition 4. In other words, for all decreasing paths $\bar{P}$ on the residual graph, the saving $sav_u(\bar{P})$ expressed by (11) must be smaller than the missed reward $w_u\pi$:

$$
\begin{array}{ll}
Max\ F - \frac{\sum_{\forall (i,j)\in\mathcal{E}} c_{i,j}(q_{i,j} - \underline{q}_{i,j})}{1+\sum_{\forall (i,j)\in\mathcal{E}} c_{i,j}(\overline{q}_{i,j} - \underline{q}_{i,j})} & \\
\text{s.}c. & \\
(i) & \sum_{(i,j)\in\mathcal{E}_i^+} f_{i,j} - \sum_{(j,i)\in\mathcal{E}_i^-} f_{j,i} = \begin{cases} 0\ \forall i \neq s,t \\ F\ ,\ i = s \\ -F\ ,\ i = t \end{cases} \\
(ii) & 0 \leq f_{i,j} \leq q_{i,j},\ \forall (i,j) \in \mathcal{E} \\
(iii) & \underline{q}_{i,j} \leq q_{i,j} \leq \overline{q}_{i,j},\ \forall (i,j) \in \mathcal{E} \\
(iv) & \sum_{A_u\in\mathcal{A}} w_u = 1 \\
(v) & sav_u(\bar{P}) < w_u\pi,\ \forall A_u \in \mathcal{A},\ \forall \bar{P} \in G_r^{\mathcal{A}}(\mathcal{F}(\mathcal{S})) \\
 & q_{i,j} \in \mathbb{N},\ f_{i,j} \in \mathbb{N},\ \forall (i,j) \in \mathcal{E} \\
 & w_u \in \mathbb{R},\ \forall A_u \in \mathcal{A} \\
 & F \in \mathbb{N}
\end{array}
\tag{14}
$$

Note that we do not consider condition (12) in the above formulation. Indeed, it is implicitly taken into account in the optimization verse of the objective function. As the flow is *maximized*, the graph $G_r^{\mathcal{A}}(\mathcal{F}(\mathcal{S}^*))$ associated to an optimal strategy $\mathcal{S}^*$ cannot contain any profitable increasing path without contradiction.

In the objective function, the second positive term (which is strictly lower than 1) enforces a strategy vector to be non-poor as it minimizes the cost of the additional capacity. Consequently, all the conditions that make $\mathcal{S}$ stable (see proposition 4) are met.

Formulation (14) cannot be straightforwardly used, since the number of possible decreasing paths in the multi-agent residual graph can grow exponentially. Moreover, since $\bar{\mathcal{P}}^{\mathcal{A}}$ depends on the strategy, constraints (v) are clearly non-linear. The remainder of this section focuses on the linearization of these constraints. More precisely, we show how to express $sav_u(\bar{P})$ using a finite number of primal-dual constraints.

First of all, let us introduce the new binary variables $x_{i,j}$ and $y_{i,j}$ such that $x_{i,j} = 1$ ($y_{i,j} = 1$, resp.) if the capacity $q_{i,j}$ of an arc $(i,j) \in \mathcal{E}$ can be increased (decreased resp.) and 0 otherwise. We also introduce the binary variables $\alpha_{i,j}$ and $\beta_{i,j}$ such that $\alpha_{i,j} = 1$ ($\beta_{i,j} = 1$, resp.) if flow on $(i,j) \in \mathcal{E}$ can be increased at zero cost (decreased at zero saving, resp.), 0 otherwise.

Variables $x_{i,j}$ and $y_{i,j}$ are linked with the other variables through the following linear constraints, to be added into the MILP formulation:

$$x_{i,j} \leq \overline{q}_{i,j} - q_{i,j} \leq (\overline{q}_{i,j} - \underline{q}_{i,j})x_{i,j},\ \forall (i,j) \in \mathcal{E} \tag{15}$$

$$y_{i,j} \leq q_{i,j} - \underline{q}_{i,j} \leq (\overline{q}_{i,j} - \underline{q}_{i,j})y_{i,j},\ \forall (i,j) \in \mathcal{E} \tag{16}$$

Let us now consider variables $\alpha_{i,j}$ and $\beta_{i,j}$. Since $\alpha_{i,j} = 1$ must imply that the flow on arc (i,j) can be increased at zero cost, the following constraints are stated:

$$(x_{i,j} - \alpha_{i,j})\underline{q}_{i,j} + (1 - x_{i,j})\overline{q}_{i,j} \leq f_{i,j} \leq (1 - \alpha_{i,j})\overline{q}_{i,j} - \varepsilon x_{i,j} + \underline{q}_{i,j}\alpha_{i,j},\ \forall (i,j) \in \mathcal{E} \tag{17}$$

where $\varepsilon > 0$ is a suitable small integer. Similarly, variables $\beta_{i,j}$ are linked with the other variables by the following constraints:

$$\beta_{i,j}\varepsilon + (\underline{q}_{i,j} + \varepsilon)y_{i,j} \leq f_{i,j}f_{i,j} \leq \beta_{i,j}\underline{q}_{i,j} + y_{i,j}\overline{q}_{i,j}, \forall (i,j) \in \mathcal{E} \tag{18}$$

As mentioned above, only non-poor strategies are of interest. For this reason, we add the constraints:

$$\alpha_{i,j} \leq x_{i,j}, \forall (i,j) \in \mathcal{E} \tag{19}$$

Note that (19) forbids the combination $\alpha_{i,j} = 1$ and $x_{i,j} = 0$, which corresponds to the case where $0 \leq f_{i,j} < \underline{q}_{i,j}$, and $q_{i,j} = \overline{q}_{i,j}$, which is a poor strategy since arc (i,j) is not full ($f_{i,j} < \underline{q}_{i,j}$) while the capacity is set to its maximum value.

$$y_{i,j} + \beta_{i,j} \leq 1, \forall (i,j) \in \mathcal{E} \tag{20}$$

Similarly, we forbid the combination $\beta_{i,j} = 1$ and $y_{i,j} = 1$. This case corresponds to $0 < f_{i,j} \leq \underline{q}_{i,j}$ and $q_{i,j} > \underline{q}_{i,j}$, which means that the arc (i,j) is not full while its capacity is greater than its minimal value (a poor strategy).

We now want to express the cost vectors as a function of the actual strategy. To this aim, we let:

$$\delta_F^{i,j,u} = c_{i,j} - (c_{i,j} - M)(1 - x_{i,j}) - c_{i,j}\alpha_{i,j} \ , \forall (i,j) \in \mathcal{E} \cap \mathcal{E}_u, \forall A_u \in \mathcal{A} \quad (21)$$

$$\delta_F^{i,j,u} = M(2 - x_{i,j} - \alpha_{i,j}) \ , \forall (i,j) \in \mathcal{E} \setminus \mathcal{E}_u, \forall A_u \in \mathcal{A} \quad (22)$$

$$\delta_B^{j,i,u} = (1 - \beta_{i,j} - y_{i,j})M - c_{i,j}y_{i,j} \ , \forall (i,j) \in \mathcal{E} \cap \mathcal{E}_u, \forall A_u \in \mathcal{A} \quad (23)$$

$$\delta_B^{j,i,u} = (1 - \beta_{i,j} - y_{i,j})M \ , \forall (i,j) \in \mathcal{E} \setminus \mathcal{E}_u, \forall A_u \in \mathcal{A} \quad (24)$$

The main idea of our MILP model lies in the determination of $sav_u(\overline{P})$ in equation (v) by computing the shortest decreasing path in terms of cost in the residual network $G_r^{\mathcal{A}}(\mathcal{F}(\mathcal{S}))$.

To compute $sav_u(\overline{P})$, we use primal-dual constraints. Given a residual graph $G_r^{\mathcal{A}}(\mathcal{F}(\mathcal{S}))$, the primal constraints make use of *node potentials* t_j^u. Since the dual of a shortest path is the problem of sending one unit of flow at minimum cost (i.e., maximum saving), we let $\varphi_{i,j}^u$ denote a binary variable such that $\varphi_{i,j}^u = 1$ if one unit of flow circulates in the residual network associated to agent A_u from t to s. Hence, we can write the following constraints.

$$t_j^u - t_i^u \leq \delta_F^{i,j,u}, \forall (i,j) \in \mathcal{E}_F^{\mathcal{A}}, \forall A_u \in \mathcal{A} \quad (25)$$

$$t_j^u - t_i^u \leq \delta_B^{i,j,u}, \forall (i,j) \in \mathcal{E}_B^{\mathcal{A}}, \forall A_u \in \mathcal{A} \quad (26)$$

$$\sum_{(i,j) \in \mathcal{E}_F^{\mathcal{A}} \cup \mathcal{E}_B^{\mathcal{A}}} \varphi_{i,j}^u - \sum_{(i,j) \in \mathcal{E}_F^{\mathcal{A}} \cup \mathcal{E}_B^{\mathcal{A}}} \varphi_{j,i}^u = \begin{cases} 0 \ \forall i \neq s,t \\ -1 \ , \ i = s \\ 1 \ , \ i = t \end{cases} \ , \forall i \in \mathcal{V}, \forall A_u \in \mathcal{A} \quad (27)$$

From duality, the length of a path in the residual graph is always greater or equal to the min-cost flow, so we obtain the following equations (where 0 and $n+1$ stand for the source and sink vertices, respectively):

$$t_0^u - t_{n+1}^u \geq \sum_{(i,j) \in \mathcal{E}_F^{\mathcal{A}}} \varphi_{i,j}^u \delta_F^{i,j,u} + \sum_{(i,j) \in \mathcal{E}_B^{\mathcal{A}}} \varphi_{i,j}^u \delta_B^{j,i,u}, \forall A_u \in \mathcal{A} \quad (28)$$

The right-hand-side of (28) contains products of two variables, namely $\varphi_{i,j}^u \delta_F^{i,j,u}$ and $\varphi_{i,j}^u \delta_B^{j,i,u}$. Fortunately, since $\varphi_{i,j}^u$ is a binary variable, both products can be easily linearized using standard techniques.

Finally, as $t_0^u - t_{n+1}^u$ is equal to $-sav_u(\overline{P})$, the absence of profitable decreasing paths can be expressed as:

$$t_{n+1}^u - t_0^u < w_u \pi, \forall A_u \in \mathcal{A} \quad (29)$$

The complete MILP model involves a polynomial number of variables and constraints with respect to the number of nodes and the number of transportation-agents. Although the model is rather huge, medium instances can be efficiently solved using a standard MILP solver, as discussed in the next Section.

6 Experimental results

The previous MILP formulation was implemented using the C++ API for GUROBI Optimization 6.0.0. The algorithm performance was evaluated on a PC with Linux Ubuntu server 12.04, 8 GB of RAM and a Xeon E5-1650 processor. In the following sections, we focus on the generation of the instances and provide an analysis of the experimental results.

6.1 Benchmark instances

Since no standard benchmark instance exists for our problem, we built our test set using the RanGen1 generator [10].

The first and the last nodes of the generated networks are considered as the source and sink nodes, respectively. In the generated networks, the number of nodes n varies in $\{10, 50, 70\}$ and the number of arcs is produced by fixing the value of the Order Strength (OS) parameter of the generator. Recall that the OS parameter represents the number of precedence relations divided by the theoretical maximum number of precedence relations in the network. For each value of n, our instances were generated with an Order Strength value set to 0.5.

The number of arcs varies between 20 and 450 in our instances, depending on the number of vertices. Each arc is randomly assigned to one transportation-agent. For each instance, two cases were considered, either $m = 2$ or $m = 5$. For each arc (i, j), minimal capacities were set to 0 and maximal capacities to $\overline{q}_{i,j} = \underline{q}_{i,j} + rand(20)$, where $rand(20)$ is a uniformly random integer number between 0 and 20. The expansion costs $c_{i,j}$ were computed as random integers in the interval $[5, 30]$.

The reward π given by the customer agent is determined with respect to the longest path length in G. It is a fraction α (with $\alpha \in \{0.1, 0.3, 0.5, 0.7, 0.9\}$) of the cost of the longest path from s to t in G when expansion costs (for the agent who owns the arc) are taken as arc costs. Such length (expressed by equation (10)) represents the theoretically most expensive path that can be used to increase the flow.

Hence, 6 sets of benchmark scenarios were obtained by combining the number of nodes $n \in \{10, 50, 70\}$ and the number of transportation-agents $m \in \{2, 5\}$. For each scenario, 7 network instances have been generated and, for each of them, the five different reward values (specified by parameter α) were considered, which gives us a total of 210 instances of MaxNash.

In our experiments, for each network instance we subsequently compute the optimal solutions for the different values of α, from $\alpha = 0.1$ to $\alpha = 0.9$, using the optimal solution obtained for a certain α as an initial feasible solution for the

next step. This allows considerable time savings (see Section 6.2). As minimal capacities are set to 0, we consider the zero flow as an initial feasible solution for the first step ($\alpha = 0.1$). The time limit to solve each instance of MaxNash has been fixed to 1800 seconds.

In our model, the sharing policy w is itself part of the decision. Anyhow, the same MILP can obviously handle the special case where a sharing policy is predefined, simply fixing the m variables w_u. This allows an interesting comparison between such fixed sharing policies and the policy prescribed by the solution to MaxNash, called *MaxNash policy*.

In our experimental study, we consider two 'reasonable' sharing policies. The first is the *equitable* policy, that consists in equally splitting the reward among the m agents, i.e., $w_u = 1/m$. The second policy is based on the agents' costs and expansion capability, and is denoted as *cost-weighted* policy. It consists in defining the share coefficient of agent A_u as $w_u = \frac{\sum_{(i,j)\in\mathcal{E}_u} c_{i,j}\times(\overline{q}_{i,j}-\underline{q}_{i,j})}{\sum_{(i,j)\in\mathcal{E}} c_{i,j}\times(\overline{q}_{i,j}-\underline{q}_{i,j})}$. Such policy prioritizes the transportation-agents having both high expansion cost and high expansion capacities.

6.2 Results

In what follows, we let F^* denote the value of the optimal solution to MaxNash, and F_{max} the value of the *overall* maximum flow , attained when all capacities are at their maximum value.

The ratio F^*/F_{max} is linked to the *price of stability* [2]. More precisely, F^*/F_{max} is the *reciprocal* of the price of stability, since ours is a maximization problem. We preferred to report F^*/F_{max} rather than its reciprocal, since it allows a more effective comparison among the various scenarios. Notice that this ratio attains values ranging between 0 (unlimited price of stability) and 1 (no price to be paid, i.e., there is a strategy having flow value $F_{\max}$ which is also a Nash equilibrium).

In all forthcoming tables, the figures represent averages over the 7 network instances of each scenario. Tables 1 and 2 report the results obtained for 2 and 5 transportation-agents respectively, for each of the three sharing policies considered. For each set of instances, we report the average ratio F^*/F_{max} and the *cumulative* average CPU time (in seconds), i.e., for each value of α, it is the time required to solve all instances of MaxNash problem up to that value of α.

A number of interesting facts can be observed from Tables 1 and 2. Correctly, F^* increases with α, since as the reward grows, the agents have more incentive to increase the flow. However, it may be noted that even by giving a high reward, in most cases the Nash flow will *not* reach the maximum flow. Moreover, with the MaxNash policy, a high value of flow can be attained with a small reward. The value of F^* sharply grows (for all sharing policies) when α exceeds 0.3. In

Instances			*Equitable*		*Cost-weighted*		*MaxNash*	
n	m	α	F^*/F_{max}	Cumul CPU	F^*/F_{max}	Cumul CPU	F^*/F_{max}	Cumul CPU
10	2	0.1	0	0.28	0.000	0.22	0.000	0.33
		0.3	0.025	0.52	0.220	0.38	0.478	0.87
		0.5	0.623	0.67	0.683	0.56	0.888	0.97
		0.7	0.837	0.75	0.861	0.65	0.968	1.01
		0.9	0.946	0.82	0.928	0.72	0.986	1.07
50	2	0.1	0.011	93.48	0.019	228.28	0.034	883.82
		0.3	0.607	1231.83	0.631	1447.04	0.767	1977.52
		0.5	0.869	1800.07	0.989	2293.33	0.991	2805.71
		0.7	0.899	2167.24	0.993	2621.11	0.998	3135.31
		0.9	0.998	2339.22	0.993	2816.74	0.999	3299.27
70	2	0.1	0.000	87.09	0.000	98.27	0.000	1046.80
		0.3	0.029	1009.95	0.341	1056.75	0.540	2846.81
		0.5	0.397	1986.36	0.599	1964.17	0.872	4113.68
		0.7	0.400	2499.54	0.724	2567.12	0.998	5104.31
		0.9	0.772	3127.01	0.808	3295.96	0.999	5163.11

Table 1. Ratio F^*/F_{max} and cumulative CPU time - 2 transportation-agents.

Instances			*Equitable*		*Cost-weighted*		*MaxNash*	
n	m	α	F^*/F_{max}	Cumul CPU	F^*/F_{max}	Cumul CPU	F^*/F_{max}	Cumul CPU
10	5	0.1	0.000	0.13	0.000	0.26	0.286	0.14
		0.3	0.386	0.26	0.588	0.5	0.942	0.26
		0.5	0.885	0.34	0.928	0.59	1.000	0.33
		0.7	0.934	0.40	0.928	0.62	1.000	0.41
		0.9	1.000	0.43	0.949	0.65	1.000	0.48
50	5	0.1	0.007	922.40	0.001	1012.53	0.123	1395.50
		0.3	0.759	2229.88	0.737	2467.51	0.984	2195.99
		0.5	0.995	2708.25	0.920	2840.55	1.000	2206.38
		0.7	0.998	2934.40	0.997	2892.73	1.000	2206.69
		0.9	0.998	3159.71	1.000	2937.28	1.000	2207.01
70	5	0.1	0.000	1652.41	0.000	1296.60	0.056	1461.89
		0.3	0.292	3245.49	0.340	2698.96	0.577	2872.53
		0.5	0.609	4187.90	0.690	3473.13	0.999	3349.28
		0.7	0.723	4578.43	0.855	3949.06	1.000	3350.34
		0.9	0.800	4905.73	0.999	4087.26	1.000	3350.86

Table 2. Ratio F^*/F_{max} and cumulative CPU time - 5 transportation-agents.

particular, for the MaxNash policy, the value of F^* already exceeds 80% of the maximum flow value as $\alpha \geq 0.5$.

In terms of cumulative average CPU time, we observe that the time needed to solve the same problem for growing values of α is increasingly smaller, for all considered sharing policies (i.e., the cumulative CPU time appears to be a concave function of α). This is apparently due to the fact that the optimal solution for the previous value of α is used as an initial feasible solution for the current value of α, which significantly speeds up computation especially when the previous optimal flow value is already very close to the optimal value for the current value of α. This phenomenon is especially apparent for $m = 5$.

An interesting insight for the customer-agent is that it is not necessary to pay the highest reward in order to get a large value of flow. In fact, depending on the sharing policies, endowing 50% to 70% of the maximum reward may be enough to get a flow larger than 80% of the overall maximum flow $F_{\max}$. Moreover, as already observed, the reward needed to reach a given value of flow is smaller for the MaxNash sharing policy than for the other policies.

MaxNash		$m = 2$		$m = 5$				
n	α	w_1	w_2	w_1	w_2	w_3	w_4	w_5
10	0.1	0.50	0.50	0.31	0.36	0.19	0.03	0.11
	0.3	0.70	0.30	0.15	0.20	0.34	0.12	0.19
	0.5	0.57	0.43	0.08	0.10	0.42	0.19	0.22
	0.7	0.61	0.39	0.12	0.10	0.13	0.18	0.47
	0.9	0.53	0.47	0.04	0.17	0.22	0.06	0.51
50	0.1	0.50	0.50	0.14	0.17	0.18	0.15	0.36
	0.3	0.50	0.50	0.18	0.19	0.20	0.21	0.22
	0.5	0.59	0.41	0.11	0.11	0.12	0.12	0.54
	0.7	0.61	0.39	0.08	0.08	0.08	0.09	0.67
	0.9	0.63	0.37	0.06	0.06	0.07	0.07	0.74
70	0.1	0.50	0.50	0.08	0.16	0.14	0.10	0.53
	0.3	0.50	0.50	0.14	0.16	0.13	0.13	0.44
	0.5	0.44	0.56	0.16	0.13	0.16	0.13	0.43
	0.7	0.67	0.33	0.09	0.09	0.12	0.09	0.61
	0.9	0.61	0.39	0.07	0.07	0.06	0.07	0.73
Cost-weighted		$m = 2$		$m = 5$				
10		0.51	0.49	0.13	0.22	0.32	0.16	0.17
50		0.49	0.51	0.17	0.21	0.23	0.19	0.20
70		0.52	0.48	0.21	0.20	0.20	0.19	0.20
Equitable		$m = 2$		$m = 5$				
all n		0.5	0.5	0.2	0.2	0.2	0.2	0.2

Table 3. Reward coefficients for each transportation-agent.

Table 3 shows the optimum sharing coefficients obtained with the various sharing policies. One may observe that in MaxNash, different agents get very

different shares of the whole reward. In particular, when $m = 5$, there is often one agent who gets a very large share of the reward.

Table 3 also shows that the sharing coefficients obtained with the cost-weighted policy are not far from the equitable coefficients. In this sense, the equitable or cost-weighted policies may be viewed as enforcing a higher degree of *fairness* with respect to MaxNash. However, Table 4 shows that when considering absolute profits attained by the agents, the superiority of MaxNash is apparent. In fact, even if the problem objective does not directly address the maximization of the agents' profits, one observes that in most cases the MaxNash sharing policy leads to a significantly larger total profit for transportation-agents than the other sharing policies. This feature is more apparent when the network is larger. Moreover, from Tables 3 and 4 we also notice that even if equitable and cost-weighted policies have similar sharing coefficients, their turnout in terms of agents' profit is very different. This suggests that the agents' profit is quite sensitive to reward coefficients, and hence, from a practical viewpoint, it is worth to optimally determine such coefficients, i.e., solve problem (14).

7 Conclusions

In this paper, a Multi-Agent Network expansion game has been considered where various transportation-agents control the capacity of the arcs at a given cost, and where a customer-agent shares a reward among the transportation-agents, which is proportional to the transshipped flow. We show that finding a Nash Equilibrium and a sharing policy that jointly maximize the flow is NP-Hard in the strong sense. To address this problem, we provide a MILP formulation that solves the problem, exploiting a graph-theoretic characterization of Nash equilibria. The formulation uses a polynomial number of primal-dual constraints. Although the MILP model is not trivial and quite large, we could solve instances with up to 70 nodes, 200 arcs, and 5 agents in reasonable time. Our experiments show that when the customer-agent is free to allocate the rewards among the agents, she can attain a large value of transshipped flow with a relatively small value of total reward.

Additional research efforts are needed for investigating particular cases with remarkable graph topology or agent structure. Moreover, taking advantage of the notions of profitable increasing or decreasing paths, more effective exact algorithms might be designed. Finally, as the transportation-agent satisfaction is not considered in this research, it might also be interesting to analyze other optimization problems, as for instance finding the most profitable stable strategy for one specific agent under the constraint that it provides a given flow value.

8 Acknowledgments

This work was supported by the ANR projects no. ANR-13-BS02-0006-01 named Athena and ANR-12-INOV-0002 named EASY.

Instances			*Equitable*		*Cost-weighted*		*MaxNash*	
n	m	α	F^*/F_{max}	Z	F^*/F_{max}	Z	F^*/F_{max}	Z
10	2	0.1	0.000	0.00	0.000	0.00	0.000	0.00
		0.3	0.025	3.64	0.220	29.88	0.478	37.16
		0.5	0.623	181.40	0.683	210.20	0.888	221.60
		0.7	0.837	443.52	0.861	494.76	0.968	489.32
		0.9	0.946	753.32	0.928	739.04	0.986	759.92
average				276.38		294.78		301.60
50	2	0.1	0.011	11.09	0.019	15.84	0.034	21.18
		0.3	0.607	1595.76	0.631	1629.47	0.767	2134.13
		0.5	0.869	5957.73	0.989	6757.45	0.991	6701.18
		0.7	0.899	9765.09	0.993	10890.51	0.998	10897.93
		0.9	0.998	15040.56	0.993	15030.39	0.999	15078.72
average				6474.05		6864.73		6966.63
70	2	0.1	0.000	0.00	0.000	0.00	0.000	0.00
		0.3	0.029	213.60	0.341	1905.12	0.540	3177.32
		0.5	0.397	4668.40	0.599	7332.40	0.872	11924.80
		0.7	0.400	7587.08	0.724	15277.48	0.998	21435.22
		0.9	0.772	21035.40	0.808	23707.72	0.999	29475.05
average				6700.90		9644.54		13202.48
10	5	0.1	0.000	0.00	0.000	0.00	0.286	34.25
		0.3	0.386	401.67	0.588	534.67	0.942	753.83
		0.5	0.885	1477.92	0.928	1516.67	1.000	1598.17
		0.7	0.934	2304.83	0.928	2281.33	1.000	2406.67
		0.9	1.000	3200.50	0.949	3100.42	1.000	3224.67
average				1476.98		1486.62		1603.52
50	5	0.1	0.007	35.16	0.001	2.91	0.123	267.88
		0.3	0.759	8927.31	0.737	9153.91	0.984	12135.56
		0.5	0.995	23514.34	0.920	22434.35	1.000	23497.47
		0.7	0.998	34673.50	0.997	34610.50	1.000	34647.66
		0.9	0.998	45806.69	1.000	45858.22	1.000	45797.85
average				22591.40		22411.98		23269.28
70	5	0.1	0.000	0.00	0.000	0.00	0.056	254.03
		0.3	0.292	7147.53	0.340	7921.59	0.577	13125.59
		0.5	0.609	26269.13	0.690	29454.50	0.999	43403.97
		0.7	0.723	46064.66	0.855	54803.50	1.000	63849.28
		0.9	0.800	67561.38	0.999	84097.32	1.000	84271.28
average				29408.54		35255.38		40980.83

Table 4. Total agents' profit.

References

1. A. Agnetis and C. Briand and J-C. Billaut and P. Sucha: Nash equilibria for the multi-agent project scheduling problem with controllable processing times. Journal of Scheduling Volume 18, Issue 1, pp 15-27 (2015).
2. E. Anshelevich, A. Dasgupta, J. Kleinberg, E. Tardos, T. Wexler, T. Roughgarden: The Price of Stability for Network Design with Fair Cost Allocation, SIAM Journal on Computing, 38(4), 16021623 (2008).
3. Y. Bachrach and J. S. Rosenschein: Power in threshold network flow games. Journal of Autonomous Agents and Multi-Agent Systems, 18(1):106132, 2009.
4. C. Briand and A. Agnetis and J.C. Billaut: The multiagent project scheduling problem : complexity of finding an optimal Nash equilibrium, in E. Demeulemeester and W. Herroelen (eds.), Proceedings of 13th Conference on Project management and Scheduling (PMS), Leuven, pp. 106–109, (2012).
5. G.P. Cachon and M.A. Lariviere: Supply Chain Coordination with Revenue-Sharing Contracts: Strengths and Limitations. International Journal of Management Science. vol. 51(1), pp. 30–44. (2005)
6. N. Chaabane Fakhfakh , C. Briand and M-J. Huguet. A Multi-Agent Min-Cost Flow problem with Controllable Capacities - Complexity of Finding a Maximum-flow Nash equilibrium. Proceedings of International Conference on Operations Research and Enterprise Systems (ICORES 2014), pp. 27–34 (2014).
7. N. Chaabane Fakhfakh , C. Briand and M-J. Huguet. Nash equilibria for Multi-agent Network Flow with Controllable Capacities. Operations Research and Enterprise Systems, Lecture notes in Communications in Computer and Information Science (CCIS), Springer-Verlag. Volume 509, pages 47-62 (2015).
8. B. Chen and H.H. Cheng: A Review of the Applications of Agent Technology in Traffic and Transportation Systems. IEEE transactions of intelligent transportation systems. vol. 11(2), pp. 485–497 (2010)
9. W.J. Cook, W. H. Cunningham, W. R. Pulleyblank, A. Schrijver. Combinatorial Optimization, Wiley (1998).
10. E. Demeulemeester and M. Vanhoucke and W. Herroelen: Rangen: A random network generator for activity on-the-node networks. Journal of Scheduling, vol. 6, pp. 17–38 (2003).
11. E. A. Dinic: Algorithm for solution of a problem of maximum flow in a network with power estimation. Soviet Math. Doklady, Doklady Nauk SSSR, vol. 11, 1970, pp. 1277-1280 (1970).
12. J. Edmonds and R.M. Karp. Theoretical improvements in algorithmic efficiency for network flow problems. Journal of the ACM, Association for Computing Machinery (ACM), vol. 19, no 2, pp. 248–264 (1972).
13. M.A. Estevez Fernandez: A game theoretical approach to sharing penalties and rewards in projects. European Journal of Operational Research. vol. 216(3), pp. 647–657 (2012)
14. D.R. Fulkerson: Increasing the Capacity of a Network: The Parametric Budget Problem. Management Science. vol. 5(4), pp. 472–483 (1959)
15. M.R. Garey and D.S. Johnson: Computers and Intractability: A Guide to the Theory of NP-Completeness. W.H. Freeman & Co. New York, USA (1979).
16. E. Kalai and E. Zemel, Totally balanced games and games of flow, mathematics of Operations Research, 7, pp.476–478 (1982).
17. M. Laguna. Applying robust optimization to capacity expansion of one location in telecommunications with demand uncertainty. Management Science, 44(11):101110 (1998).

18. T. Magnanti and R. Wong: Network design and transportation planning: Models and algorithms. Transportation Science, 18(1):155, 1984.
19. M. Minoux. Network synthesis and optimum network design problems: Models, solution methods, and applications. Networks, 19:313360, 1989.
20. F. Murphy and H. Weiss. An approach to modeling electric utility capacity expansion planning. Naval Research Logistics, 37:827845, 1990.
21. J. Nash: Equilibrium points in n-person games. Proceedings of the National Academy of Sciences. Vol. 36(1), pp. 48–49 (1950).
22. J. Nash: Non-Cooperative Games. The Annals of Mathematics. Vol. 54(2), pp. 286–295 (1951).
23. N. Nisan and T. Roughgarden and E. Tardos and V.V. Vazirani: Algorithmic Game Theory. Cambridge University Press (2007).
24. F. Ordonez and J. Zhao: Robust Capacity Expansion of Network flow. Networks. vol. 50(2), pp. 136–145. Wiley-Interscience (2007) s, ISBN 978-0-521-89943-7 (2009).
25. M. Pechoucek and V. Marik: Industrial deployment of multi-agent technologies: review and selected case studies. Autonomous Agents and Multi-Agent Systems. Vol. 17 (3), pp. 397–431 (2008)
26. J. Puerto, A. Schöbel and S. Schwarze, The path player game: A network game from the point of view of the network providers. Mathematical Methods of Operations Research, 68, 1–20 (2008).
27. E. Resnick, Y. Bachrach, R. Meir, JJ S. Rosenschein: The Cost of Stability in Network Flow Games . Mathematical Foundations of Computer Science, Volume 5734 of the series Lecture Notes in Computer Science pp 636-650 (2009).
28. Y. Shoham and K. Leyton-Brown: Multiagent Systems: Algorithmic, Game-Theoretic, and Logical Foundations. New York: Cambridge University P