



**HAL**  
open science

## Finding Stable flows in multi-agent Networks under an equal-sharing policy

Nadia Chaabane, Cyril Briand, Marie-José Huguet

► **To cite this version:**

Nadia Chaabane, Cyril Briand, Marie-José Huguet. Finding Stable flows in multi-agent Networks under an equal-sharing policy. International Network Optimization Conference (INOC), May 2015, Warsaw, Poland. pp.157-164. hal-01152083v2

**HAL Id: hal-01152083**

**<https://hal.science/hal-01152083v2>**

Submitted on 21 Jun 2015

**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 Stable flows in multi-agent Networks under an equal-sharing policy

Nadia Chaabane<sup>b,2</sup>, Cyril Briand<sup>c</sup> and Marie-José Huguet<sup>b</sup>

<sup>a</sup> *CNRS-LAAS, 7 Avenue du Colonel Roche, F-31400 Toulouse, France*

<sup>b</sup> *CNRS, LAAS, University of Toulouse, INSA, F-31400 Toulouse, France*

<sup>c</sup> *CNRS, LAAS, University of Toulouse, UPS, F-31400 Toulouse, France*

---

## Abstract

A Multi-Agent network flow problem is addressed in this paper where a set of transportation-agents can control the capacities of a set of routes. Each agent incurs a cost proportional to the chosen capacity. A third-party agent, a customer, is interesting in maximizing the product flow transshipped from a source to a sink node through the network. He offers a reward proportional to the flow value, which is shared equally among the transportation-agents. We address the problem of finding a stable strategy (i.e., a Nash Equilibrium) which maximizes the network flow. We show that this problem, already proved to be NP-hard, can be modeled and solved using mixed integer linear programming (MILP).

*Keywords:* Multi-Agent Network flow, Nash Equilibria, Equal-sharing Policy.

---

## 1 Introduction

Multi-agent network games have become a promising interdisciplinary research area with important links to many application fields such as transportation networks, supply chain management, web services, production management, etc [1], [2]. This paper stands at the crossroad of two disciplines, namely multi-agent systems and social networks. The former ties in with the distributed resolution of multi-agent problems. The latter is connected to game theory, which formalizes the multi-agent optimization problem as a strategic

---

<sup>1</sup> This work was supported by the ANR project no. ANR-13-BS02-0006-01 named Athena.

<sup>2</sup> Email: [nadia.chaabane@laas.fr](mailto:nadia.chaabane@laas.fr)

game. In this paper, we consider a cooperative network flows problem involving a set of self-interested transportation-agents, each of them managing his proper set of network roads. Every agent is able to increase the capacity of the arcs belonging to him by gathering extra resources, at a given cost. A customer-agent is interested in increasing the flow circulating in the network. He offers a reward to be equally-shared among the agents. Therefore, the flow on the network depends on the strategies of all agents. Assuming an equal-sharing policy, we focus on the problem of finding a Nash equilibrium that maximizes the network flow.

The paper is organized as follows: Section 2 defines formally the problem. Thereafter, Section 3 shows how Nash equilibria can be characterized by means of augmenting or decreasing paths in a reduced network. In Sections 4 and 5 a mathematical formulation is proposed and experimental results are provided.

## 2 Problem Statement and Notations

More formally, the Multi-Agent Network-Flow with Controllable Capacities (MA-NFCC) can be defined as a tuple  $\langle G, \mathcal{A}, \underline{Q}, \overline{Q}, C, \pi, W \rangle$  where:

- $G = (V, E)$  is a network flow.  $V$  is the set of nodes and  $E$  is the set of arcs, each one having its capacity and receiving a flow.
- $\mathcal{A} = \{A_1, \dots, A_u, \dots, A_m\}$  is a set of  $m$  transportation-agents. Arcs are distributed among the agents. An agent  $A_u$  owns a set of  $m_u$  arcs, denoted  $E_u$ . Each arc  $(i, j)$  belongs to exactly one transportation-agent.
- $\underline{Q} = (\underline{q}_{i,j})_{(i,j) \in E}$  (resp.  $\overline{Q} = (\overline{q}_{i,j})_{(i,j) \in E}$ ) represents the vector of normal (resp. maximum) capacity for each arc  $(i, j) \in E$ .
- $C = (c_{i,j})_{(i,j) \in E}$  is the vector of unitary costs  $c_{i,j}$  incurred by agent  $A_u$ , for increasing  $q_{i,j}$  by one unit.
- $\pi$  refers to the reward given by the customer proportionally to the flow.
- $W = \{w_u\}$  defines the sharing policy of rewards among the agents. In this paper, it is assumed that  $w_u = \frac{1}{m}$ .

In such a network game, a *strategy*  $S$  is the vector that gather the individual strategies of all agents:  $S = (Q_1, \dots, Q_m)$  where  $Q_u = (q_{i,j}), \forall (i, j) \in E_u$ . Given a strategy  $S$ ,  $F(S)$  denotes the *flow* that can circulate on the network given the current values of capacities. For each arc  $(i, j)$ , the circulating flow  $f_{i,j}$  is such that  $\underline{q}_{i,j} \leq f_{i,j} \leq q_{i,j}$ . With respect to the above payment scheme, the total reward given by the customer-agent for a circulating flow  $F(S)$  under a strategy  $S$  is  $\pi \times F(S)$ . The profit  $Z_u(S)$  of transportation-

agent  $A_u$  under strategy  $S$  is equal to the difference between his reward and spending  $Z_u(S) = \frac{1}{m} \times \pi \times F(S) - \sum_{(i,j) \in E_u} c_{i,j} \times (q_{i,j} - \underline{q}_{i,j})$ .

Ideally, agents should choose a strategy which is both optimal, with respect to the flow value, and stable, with respect to the Nash equilibrium concept. Since such a strategy does not always exist, we search for a Nash equilibrium that is as efficient as possible with respect to the customer viewpoint.

### 2.1 Example of a MA-NFCC Problem

Let us consider a customer-agent willing to transport a flow of products in the network, displayed in Fig. 1, from the source node  $A$  to the sink node  $D$ . He offers a reward  $\pi = 120$  which is shared equally between two agents  $A_1$  and  $A_2$ , i.e.  $w_1 = w_2 = \frac{1}{2}$ . The set of arcs of each transportation-agent are  $E_1 = \{b, c, d\}$  and  $E_2 = \{a, e\}$ , which are represented with plain and dotted arcs, respectively. Each arc in the graph of Fig. 1 is valued by the interval of normal and maximum capacities  $([\underline{q}_{i,j}, \bar{q}_{i,j}])$ , and the unitary cost incurred when increasing the arc capacities  $(c_{i,j})$ . For the strategy  $S$ , where both

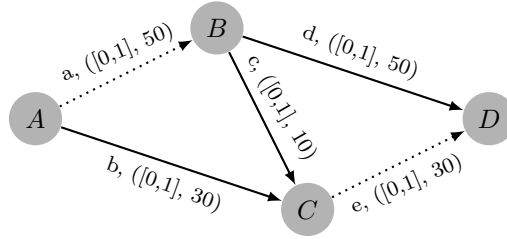


Fig. 1. Example of MA-NFCC problem

transportation-agents choose to open their respective routes  $b$  and  $e$  by one unit, the flow  $F(S)$  is equal to 1. Given the part of the equal-shared reward for each agent is equal to 60, the profit for both agents is  $Z_1 = Z_2 = 60 - 30 = 30$ , which means that this strategy is profitable for both agents.

## 3 Characterization of Nash Equilibrium

For sake of simplicity, it is assumed throughout this section, that  $\underline{q}_{i,j} = 0$ .

### 3.1 Multi-agent residual graphs

In order to explain how to increase or decrease the flow value, we need first to introduce the concept of a residual graph, which is built up with respect to one agent  $A_u$  (there are as many residual graphs as agents) .

**Definition 3.1** Given a network  $G = (V, E)$  and a strategy  $S$ , we define the residual graph  $G_f^u(S) = (V, E_r)$  of agent  $A_u$  as follows. . To each edge

$(i, j) \in E$ , we associate a forward and a backward edge. Each forward edge  $(i, j)$  of  $G_f^u$  has a residual capacity  $r_{i,j} = \bar{q}_{i,j} - q_{i,j}$  and cost  $\delta_F^{i,j,u}$ . Each backward edge  $(j, i)$  of  $G_f^u$  is with a residual capacity  $r_{j,i} = q_{i,j}$  and cost  $\delta_B^{i,j,u}$ .

$$\delta_F^{i,j,u} = \begin{cases} -\infty & \text{if } (i, j) \in \mathcal{E}_u \wedge q_{i,j} = \bar{q}_{i,j} \\ -c_{i,j} & \text{if } (i, j) \in \mathcal{E}_u \wedge q_{i,j} < \bar{q}_{i,j} \\ -\infty & \text{if } (i, j) \notin \mathcal{E}_u \wedge q_{i,j} = \bar{q}_{i,j} \\ -\infty & \text{if } (i, j) \notin \mathcal{E}_u \wedge q_{i,j} < \bar{q}_{i,j} \end{cases} \quad (1)$$

$$\delta_B^{j,i,u} = \begin{cases} -\infty & \text{if } (i, j) \in \mathcal{E}_u \wedge q_{i,j} = \underline{q}_{i,j} \\ c_{i,j} & \text{if } (i, j) \in \mathcal{E}_u \wedge q_{i,j} > \underline{q}_{i,j} \\ -\infty & \text{if } (i, j) \notin \mathcal{E}_u \wedge q_{i,j} = \underline{q}_{i,j} \\ 0 & \text{if } (i, j) \notin \mathcal{E}_u \wedge q_{i,j} > \underline{q}_{i,j} \end{cases} \quad (2)$$

A forward arc having the cost  $-c_{i,j}$  indicates that  $A_u$  can possibly make a profit by increasing the capacity of  $(i, j)$  since  $q_{i,j} < \bar{q}_{i,j}$ . A forward arc of cost  $-\infty$  cannot be used to increase the flow in the network (i.e., it is already at its maximum capacity or the arc belongs to another agent). In a similar way, a backward arc valued by 0 indicates that the arc does not matter in the computation of the cost of agent  $A_u$ . An arc valued by the cost  $c_{i,j}$  reflects the fact that the agent can make a saving by decreasing his arcs capacities (i.e.,  $q_{i,j} > \underline{q}_{i,j}$ ). Finally, a  $-\infty$  cost on a backward arc guarantees the impossibility of using the corresponding arc to decrease the flow (i.e.,  $q_{i,j} = \underline{q}_{i,j}$ ).

### 3.2 Increasing the flow

In the multi-agent context, an augmenting path  $P$ , going from the source node to the sink one, is composed by a set of forward and backward arcs, i.e.  $P = \{P^+, P^-\}$  and allows to increase the flow value by one unit at least. The cost of an augmenting path for agent  $A_u$  is as follows:

$$cost_u(P) = \sum_{(i,j) \in P^+ \cap E_u} c_{i,j} - \sum_{(i,j) \in P^- \cap E_u} c_{i,j} \quad (3)$$

**Definition 3.2** An augmenting path  $P \in \mathcal{P}$  is said profitable for all agents if, for every agent  $A_u$  involved in  $P$ ,  $cost_u(P) < w_u \times \pi$ .

In other words, increasing the flow by one unit, is profitable for all the agents owning the arcs of the path.

### 3.3 Decreasing the flow

A decreasing path  $\bar{P} = \{\bar{P}^+, \bar{P}^-\}$ , going from the sink node to the source one, is composed of forward and backward arcs. The saving of agent  $A_u$  when decreasing capacity by one unit through a decreasing path is defined as follows:

$$sav_u(\bar{P}) = \sum_{(i,j) \in \bar{P}^+ \cap E_u} c_{i,j} - \sum_{(i,j) \in \bar{P}^- \cap E_u} c_{i,j} \quad (4)$$

**Definition 3.3** *Profitable decreasing path.* A decreasing path  $\bar{P} \in \bar{\mathcal{P}}$  is profitable if there is one agent  $A_u$  such that  $sav_u(\bar{P}) > w_u \times \pi$ .

**Proposition 3.4** *Nash Equilibrium.* For a given non-poor strategy profile  $S$  and its corresponding residual graph  $G_u^f(S)$ ,  $S$  is a Nash Equilibrium if and only if:

- $cost_u(P) > w_u \times \pi, \forall A_u \in \mathcal{A}, \forall P \in \mathcal{P}$  such that  $(i,j) \in E_u$
- $sav_u(\bar{P}) \leq w_u \times \pi, \forall A_u \in \mathcal{A}, \forall \bar{P} \in \bar{\mathcal{P}}$

**Proof.** See detailed proof in [5]. □

## 4 A Nash-Max-Flow MILP for MA-NFCC

In this section, a mathematical formulation is given for solving the problem of finding a Nash equilibrium that maximizes  $F(S)$ , which was proved to be NP-Hard in [5]. This Nash-Max Flow problem is modeled in (5) where constraints (i), (ii) and (iii) represent the constraints of arcs capacities and flow conservation, respectively. The non-explicit constraints (iv) impose that no decreasing path  $\bar{P}$  can exist in solution  $S$  having a saving  $sav_u(\bar{P}) \geq w_u \times \pi$ . In other words, it enforces any solution to be Nash stable.

Max  $F$

s.c.

$$(i) \quad f_{i,j} \leq q_{i,j}, \forall (i,j) \in E$$

$$(ii) \quad \sum_{j \in P^+(i)} f_{i,j} - \sum_{j \in P^-(i)} f_{j,i} = \begin{cases} 0 & \forall i \neq s, t \\ F & , i = s \\ -F & , i = t \end{cases} \quad (5)$$

$$(iii) \quad \underline{q}_{i,j} \leq q_{i,j} \leq \bar{q}_{i,j}, \forall (i,j) \in E$$

$$(iv) \quad sav_u(\bar{P}) < w_u \times \pi, \forall \bar{P} \in \bar{G}_f$$

$$f_{i,j} \geq 0, \forall (i,j) \in E$$

We focus on the linearization of the non-linear constraints (iv) of the mathematical model (5). We show how it can be replaced by a finite number of primal-dual constraints.

**Arcs forming decreasing paths:** In order to determine whether an arc capacity increase or decrease impacts the value of the flow in the network, we introduce two new binary variables  $x_{i,j} \in \{0,1\}$  and  $y_{i,j} \in \{0,1\}$  such that  $x_{i,j} = 1$  (resp.  $y_{i,j} = 1$ ) if the capacity  $q_{i,j}$  can be increased (resp. decreased) and 0 otherwise. **Longest path in the residual Network:** The main idea relies on determining  $sav_u(\bar{P})$  in equation (iv) by computing the longest decreasing path in the residual network  $G_f^u(S)$  using primal-dual constraints. Assuming residual graph  $G_f^u(S)$ , equations (6)-(7) define the primal constraints of the longest path problem where  $t_i^u \geq 0$ . Equations (8) define the dual constraints of the problem of unitary flow variation at minimum cost using the binary variables  $\varphi_{i,j}^u \in \{0,1\}$ .  $\varphi_{i,j}^u = 1$  if one unit of flow circulates on arc  $(i,j)$ , 0 otherwise.

$$t_j^u - t_i^u \geq \delta_F^{i,j,u}, \forall (i,j) \in E, \forall A_u \in \mathcal{A} \quad (6)$$

$$t_i^u - t_j^u \geq \delta_B^{j,i,u}, \forall (i,j) \in E, \forall A_u \in \mathcal{A} \quad (7)$$

Expression  $t_j^u - t_i^u$  represents the potential associated with the arc  $(i,j)$ . With respect to the Complementary Slackness Theorem, which imposes, at optimality, that the cost of a path in  $G_f^u(S)$  is equal to the saving of the units of flows circulating in the network, we obtain the following constraints:

$$\sum_{(i,j) \in E \cup E_r} \varphi_{i,j}^u - \sum_{(i,j) \in E \cup E_r} \varphi_{j,i}^u = \begin{cases} 0 & \forall i \neq s, t \\ -1, & i = s, \forall i \in V, \forall A_u \in \mathcal{A} \\ 1, & i = t \end{cases} \quad (8)$$

$$t_{n+1}^u \geq \sum_{(i,j) \in E} \varphi_{i,j}^u \times \delta_F^{i,j,u} + \sum_{(i,j) \in E_r} \varphi_{i,j}^u \times \delta_B^{j,i,u}, \forall A_u \in \mathcal{A} \quad (9)$$

The right-hand-side of (9) contains multiplication of two variables. Fortunately, since  $\varphi_{i,j}^u$  is binary, the multiplication can be easily linearized by introducing variables  $\psi_{1,i,j}^u = \varphi_{i,j}^u \times \delta_F^{i,j,u}$  and  $\psi_{2,i,j}^u = \varphi_{i,j}^u \times \delta_B^{j,i,u}$ . Finally, since  $t_{n+1}^u$  is equal to  $sav_u(\bar{P})$ , constraints (iv) can be replaced in the MILP model by  $t_{n+1}^u < w_u \times \pi, \forall A_u \in \mathcal{A}$ . The detailed MILP model is given in Appendix ??.

## 5 Experimental results

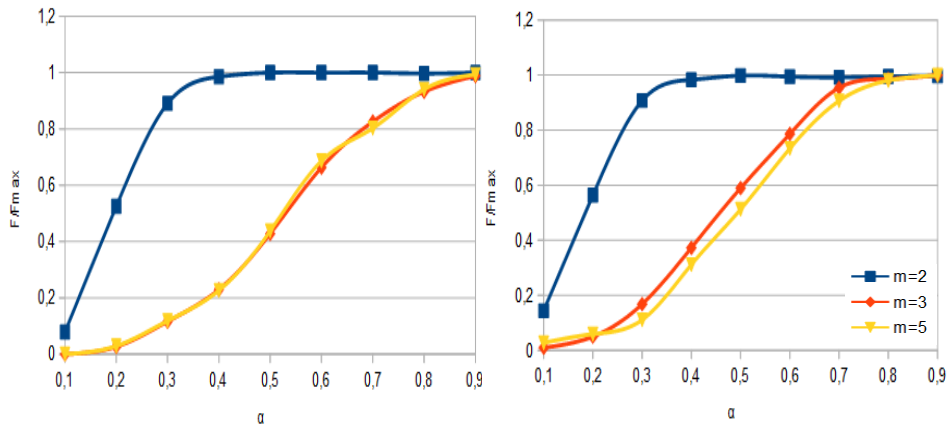
The MILP model was implemented in the Gurobi solver using C++ API for GUROBI Optimization 5.6.3. The algorithm performance was evaluated on a PC with Linux Ubuntu server 12.04, 8 GO of RAM. Since no standard benchmark instances exist for our problem, the instances were built up using the RanGen1 generator [8]. For each problem size, 100 instances were generated with an Order Strength (OS) value = 0.5.

To illustrate the efficiency of the MILP we consider 6 sets of benchmark instances where  $n = \{50, 100\}$  and  $m = \{2, 3, 5\}$ . The reward  $\pi = \alpha \times m \times LP$  where  $LP$  is the value of the longest path from  $s$  to  $t$  in terms of costs and  $\alpha$  is a percentage of the reward. Because problem instances are generated from activity-on-nodes networks, the number of edges in the resulting network flow varies significantly from an instance to another. Therefore, the number of edges is significantly larger than the number of vertices. The results shows that the solving phase remains fast since we solve instances with 100 nodes (almost 800 edges) in less than 1400s when having 5 agents. To compare the solution and to analyze the influence of the number of agents and the value of  $\alpha$ , we varied the value of  $\alpha \in \{0.1, 0.2, \dots, 0.9\}$ . Figure 6 shows the average relative flow increasing with respect to the maximum flow (i.e.  $\frac{F}{F_{max}}$ ) in function of the percentage of the reward  $\alpha$ . It can be expected that the larger  $\pi$  (i.e. the higher  $\alpha$ ), the larger the Nash-Flow (i.e. the larger  $\frac{F}{F_{max}}$ ). However, the customer may not want to waste money by paying more than necessary to obtain a given flow, or by paying more without any flow increase. Figure 6 shows the evolution of the Nash-Flow increasing in function of the customer reward for all considered instances. It is important to note that (i) the correlation between flow increase and the reward is not linear, (ii) until  $\alpha = 0.4$  every additional reward results into significant flow increase after that almost no improvement is obtained for  $m = 2$ , (iii) for  $m$  greater than 2 this stability can be obtained from  $\alpha = 0.8$ .

## 6 Conclusion

In this paper, we particularly point out the notions of efficiency and stability of a strategy and we introduce the notion of profitable augmenting or decreasing paths. We show that the problem of finding a Nash Equilibrium that maximizes the flow, already proved to be NP-hard, can be modeled and solved using a linear model and we presents preliminary results to show the efficiency of our MILP.





(a)  $n = 50$

(b)  $n = 100$

## References

- [1] Chen, B., and H.H. Cheng, "A Review of the Applications of Agent Technology in Traffic and Transportation Systems", *IEEE transactions of intelligent transportation systems*, **11(2)** (2010), 485–497.
- [2] Pechoucek, M., and V. Marik, *Industrial deployment of multi-agent technologies: review and selected case studies*, *Autonomous Agents and Multi-Agent Systems*, **17 (3)** (2008), 397–431.
- [3] Agnetis, A., B. Cyril, J-C. Billaut and P. Scha, *Nash equilibria for the multi-agent project scheduling problem with controllable processing times*, *J. Scheduling* (2014).
- [4] Cachon, G.P., and M.A. Lariviere, *Supply Chain Coordination with Revenue-Sharing Contracts: Strengths and Limitations*, *J. of Management Science*, **51(1)** (2005), 30–44.
- [5] Chaabane, N., 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 ICORES* (2014), 27–34.
- [6] Ehrgott, M., "Multicriteria Optimization", *Algorithmic Game Theory*, Springer (2005), 487–516.
- [7] Nash, J., "Equilibrium points in n-person games", *Proceedings of the National Academy of Sciences*, **36(1)** (1950), 48–49.
- [8] Demeulemeester, E., M. Vanhoucke and W. Herroelen, *Rangen: A random network generator for activity on-the-node networks*, *J. Scheduling*, **6** (2003), 17–38.