Optimizing Nash Social Welfare in Semi-Competitive Intermediation Networks

Amelia Bădică
University of Craiova
Craiova, Romania
ameliabd@yahoo.com

Costin Bădică
University of Craiova
Craiova, Romania
cbadica@software.ucv.ro

Ion Buligiu
University of Craiova
Craiova, Romania
buligiu.ion@gmail.com

Liviu Ion Ciora
University of Craiova
Craiova, Romania
liviuciora2004@yahoo.com

Maria Ganzha
Warsaw University of Technology
Warsaw, Poland
Maria.Ganzha@ibspan.waw.pl

Mirjana Ivanović
University of Novi Sad
Novi Sad, Serbia
mira@dmi.uns.ac.rs

Marcin Paprzycki
Polish Academy of Sciences
Warsaw, Poland
marcin.paprzycki@ibspan.waw.pl

***Abstract*—We have recently proposed a mathematical model of collective profitability, in semi-competitive intermediation networks. In this work, we are interested in determining optimal pricing strategies of network participants. The optimization criterion is defined using the Nash social welfare function. We provide theoretical results of existence of such strategies, as well as computational experimental results, based on nonlinear convex mathematical optimization.**

***Index Terms*—semi-competitive intermediation, collective profitability, Nash social welfare, nonlinear convex optimization**

I. Introduction

The context of this work is related to our recent research efforts and results in the area of modeling and analysis of semi-competitive intermediation networks that serve complex business processes. They are motivated by current practices in the management of distribution activities, by engaging in multiple inter-related and concurrently operating distribution channels.

Our preliminary results on this subject were initially presented in recent papers [4], [5]. The main reported achievement was the proposal of a general process structure, based on directed acyclic graphs (DAG hereafter) and the establishment of necessary and sufficient conditions for the existence of profitable pricing strategies of all process participants.

Let us consider a "society", defined through all participants in an intermediation business process. Social choice theory postulates that a social choice can be defined using a social welfare function that maps each tuple of individually preferred outcomes of participants to the socially preferred outcome.

Authors of [8] formulated a set of four rationality conditions that should be satisfied by a reasonable social welfare function. Then they shown that there exists a unique social welfare function that satisfies these conditions. It is called Nash social welfare function, and it can be represented quantitatively using the Nash social welfare index.

In this contribution we assume that participants choices are represented by transaction prices and their social welfare can be quantitatively captured by the Nash social welfare index. The main achievement is the definition of optimal pricing strategies of transaction participants as a nonlinear convex optimization problem [6]. We provide theoretical results regarding the existence of optimal pricing strategies in a semi-competitive intermediation networks, as well as experimental results that support the feasibility of our approach.

II. Semi-Competitive Intermediation Networks

Let us start by reviewing the definition of collectively profitable intermediation DAG, previously introduced in [4], [5]. Nevertheless, we simplify the definition, by removing DAG annotations with information about exchanged products that is not relevant for the purpose of this paper.

Definition 1 (Intermediation DAG): Let $\mathcal{S}$, $\mathcal{B}$, and $\mathcal{I}$ be three finite, nonempty and pairwise disjoint sets of seller, buyer and intermediary agents. An *intermediation DAG* is defined by a triple $G = \langle \mathcal{V}, \mathcal{A}, \pi \rangle$ such that:

i) The set of vertices is defined by $\mathcal{V} = \mathcal{S} \cup \mathcal{B} \cup \mathcal{I}$.
ii) There are no incoming arcs to $\mathcal{S}$ and no outgoing arcs from $\mathcal{B}$.
iii) $\pi : \mathcal{S} \cup \mathcal{B} \cup A \to [0, +\infty)$ is the annotation function, to attach pricing information to components of G as follows:
  iii.1) If $s \in \mathcal{S}$ then $\pi(s) = \sigma_s > 0$.
  iii.2) If $b \in \mathcal{B}$ then $\pi(b) = \beta_b > 0$.
  iii.3) If $t = (u, v) \in \mathcal{A}$ is a transaction, then $\pi(t) = \pi_{u,v} > 0$ represents the transaction price, for which agent u sells some of its products to agent v.

Each seller/buyer of an intermediation DAG is responsible with selling/buying a given set of products. We define the limit price σ_s of seller s for selling its set of products, meaning that s will agree to sell its products only at price $p \geq \sigma_S$. Similarly, we define the limit price β_b of buyer b for buying its set of products, meaning that b will agree to purchase its set of products only at price $p \leq \beta_b$.

Figure 1 presents an example of intermediation DAG. Note that in this example $\mathcal{S} = \{1, 2\}$, $\mathcal{B} = \{5, 7, 8\}$, $\pi(1) = \sigma_1$, $\pi(5) = \beta_5$, and $\pi((3, 6)) = \pi_{3,6}$.

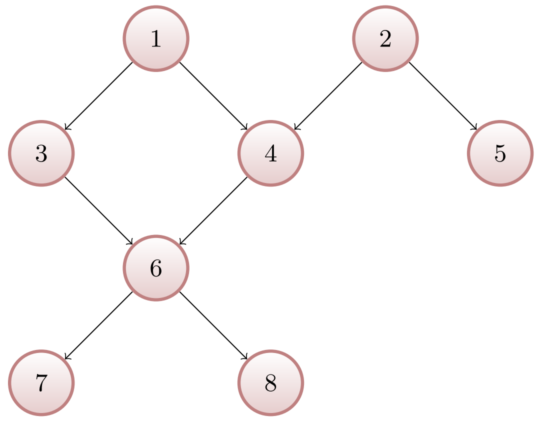

Fig. 1. DAG-based intermediation process.

A participant $v \in \mathcal{V}$ is profitable if and only if it gains a positive utility, i.e. $u(v) \geq 0$. Let us define functions in and out using equations (1).

$$\begin{array}{l} in : \mathcal{V} \rightarrow 2^{\mathcal{V}} \text{ defined as } in(v) = \{u | (u, v) \in \mathcal{A}\} \\ out : \mathcal{V} \rightarrow 2^{\mathcal{V}} \text{ defined as } out(v) = \{u | (v, u) \in \mathcal{A}\} \end{array} \tag{1}$$

We can now define the utility $u_v = u(v)$ of each node v using equations (2), (3), and (4).

i) The utility of a seller $s \in \mathcal{S}$ is defined by:

$$u(s) = -\sigma_s + \sum_{v \in out(s)} \pi_{s,v} \tag{2}$$

ii) The utility of a buyer $b \in \mathcal{B}$ is defined by:

$$u(b) = \beta_b - \sum_{v \in in(b)} \pi_{v,b} \tag{3}$$

iii) The utility of an intermediary $i \in \mathcal{I}$ is defined by:

$$u(i) = \sum_{v \in out(i)} \pi_{i,v} - \sum_{v \in in(i)} \pi_{v,i} \tag{4}$$

An intermediation DAG is collectively profitable if and only it can be annotated with positive transaction prices, such that for each $v \in \mathcal{V}$ we have $u(v) \geq 0$. In paper [5], we formulated necessary and sufficient collective profitability conditions that must be satisfied by limit prices $\boldsymbol{\sigma}$ and $\boldsymbol{\beta}$. In this paper, assuming that these conditions hold, i.e. our DAG is collectively profitable, we are interested in formulating the maximization of social welfare as a mathematical optimization problem.

III. Optimizing Nash Social Welfare

Let us consider a collectively profitable intermediation DAG G, i.e. $u_v \geq 0$ for each participant v. In what follows we denote with $n \geq 1$ the number of participants (i.e. number of elements of $\mathcal{V}$) and with $e \geq 1$ the number of transactions (i.e. number of elements of $\mathcal{A}$). Hence, participants are $1, 2, \ldots, n$ and transactions (numbered in lexicographical order) are $1, 2, \ldots, e$.

The Nash social welfare function for G is defined by equation (5).

$$U(\boldsymbol{\pi}) = \sum_{i=1}^{n} \log u_i \tag{5}$$

Note that if the intermediation DAG is collectively profitable then U is well defined by equation (5), as $u_i \geq 0$ for all $1 \leq i \leq n$ (we assume that $\log 0 = -\infty$). Let us consider a vector $\boldsymbol{b} \in \mathbb{R}^{n \times 1}$ defined by equation (6).

$$b_i = \begin{cases} -\sigma_i & i \in \mathcal{S} \\ \beta_i & i \in \mathcal{B} \\ 0 & i \in \mathcal{I} \end{cases} \tag{6}$$

Let also denote with $D \in \mathbb{R}^{n \times e}$ the incidence matrix [1] of the intermediation DAG, defined according to equation (7).

$$D_{i,j} = \begin{cases} 1 & \text{if node } i \text{ is the head of arc } j \\ -1 & \text{if node } i \text{ is the tail of arc } j \\ 0 & \text{otherwise} \end{cases} \tag{7}$$

Then, the vector $\boldsymbol{u}$ of participants' utilities can be defined using equation (8).

$$\boldsymbol{u} = D\boldsymbol{\pi} + \boldsymbol{b} \tag{8}$$

Now, coupling collective profitability condition $\boldsymbol{u} \succeq 0$, with positivity condition of transaction prices $\boldsymbol{\pi} \succeq 0$, produces inequalities (9) that define the domain of the social welfare function U.

$$\begin{aligned} -D\boldsymbol{\pi} &\preceq \boldsymbol{b} \\ -\boldsymbol{\pi} &\preceq 0 \end{aligned} \tag{9}$$

Note that inequalities (9) define a finite intersection of half-spaces, i.e. a polyhedron. Moreover, positivity conditions of $\boldsymbol{\pi}$ define a lower bound. Finally, moving up in the intermediation DAG, starting from the bottom vertices denoting buyers (see also Figure 1), we can recursively define finite upper bounds for each transaction price. Thus, it easily follows that the domain of U is a bounded polyhedron, i.e. a polytope [10].

We now study the convexity of U. First observe that:

$$\begin{aligned} \frac{\partial U}{\partial \pi_{i,j}} &= \frac{1}{u_i} - \frac{1}{u_j} \\ \frac{\partial^2 U}{\partial \pi_{i,j}^2} &= -(\frac{1}{u_i^2} + \frac{1}{u_j^2}) \\ \frac{\partial^2 U}{\partial \pi_{i,j} \partial \pi_{i,k}} = \frac{\partial^2 U}{\partial \pi_{j,i} \partial \pi_{k,i}} &= -\frac{1}{u_i^2} \\ \frac{\partial^2 U}{\partial \pi_{i,j} \partial \pi_{j,k}} = \frac{\partial^2 U}{\partial \pi_{j,k} \partial \pi_{i,j}} &= \frac{1}{u_j^2} \end{aligned} \tag{10}$$

According to equations (10), the Jacobian and Hessian matrices of U are $1 \times e$ and, respectively, $e \times e$ matrices defined by equations (11).

$$\begin{aligned} \boldsymbol{J}_U(\boldsymbol{\pi}) &= \left[\frac{1}{\boldsymbol{u}}\right]^T D \\ \boldsymbol{H}_U(\boldsymbol{\pi}) &= D^T diag\left(\left[\frac{-1}{\boldsymbol{u}^2}\right]\right) D \end{aligned} \tag{11}$$

$$\boldsymbol{b} = \begin{bmatrix} -\sigma_1 \\ -\sigma_2 \\ 0 \\ 0 \\ \beta_5 \\ 0 \\ \beta_7 \\ \beta_8 \end{bmatrix} \tag{12}$$

Matrices $\boldsymbol{b}$ and D, for the example intermediation DAG from Figure 1, are defined by equations (12) and (13). Moreover, Jacobian and Hessian matrices of the intermediation DAG, from Figure 1, are defined by equations (14).

$$D = \begin{bmatrix} 1 & 1 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 1 & 0 & 0 & 0 & 0 \\ -1 & 0 & 0 & 0 & 1 & 0 & 0 & 0 \\ 0 & -1 & -1 & 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & -1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & -1 & -1 & 1 & 1 \\ 0 & 0 & 0 & 0 & 0 & 0 & -1 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & -1 \end{bmatrix} \tag{13}$$

Note that the Hessian matrix is useful for understanding why the social welfare function U is a concave function, while both matrices will be also useful for the implementation of the nonlinear convex optimization problem.

$$\boldsymbol{J}_U(\boldsymbol{\pi}) = \begin{bmatrix} \frac{1}{u_1} - \frac{1}{u_3} & \frac{1}{u_1} - \frac{1}{u_4} & \frac{1}{u_2} - \frac{1}{u_4} & \frac{1}{u_2} - \frac{1}{u_5} & \frac{1}{u_3} - \frac{1}{u_6} & \frac{1}{u_4} - \frac{1}{u_6} & \frac{1}{u_6} - \frac{1}{u_7} & \frac{1}{u_6} - \frac{1}{u_8} \end{bmatrix}$$

$$\boldsymbol{H}_U(\boldsymbol{\pi}) = \begin{bmatrix} -\frac{1}{u_1^2} - \frac{1}{u_3^2} & -\frac{1}{u_1^2} & 0 & 0 & \frac{1}{u_3^2} & 0 & 0 & 0 \\ -\frac{1}{u_1^2} & -\frac{1}{u_1^2} - \frac{1}{u_4^2} & -\frac{1}{u_4^2} & 0 & 0 & \frac{1}{u_4^2} & 0 & 0 \\ 0 & -\frac{1}{u_4^2} & -\frac{1}{u_2^2} - \frac{1}{u_4^2} & -\frac{1}{u_2^2} & 0 & \frac{1}{u_4^2} & 0 & 0 \\ 0 & 0 & -\frac{1}{u_2^2} & -\frac{1}{u_2^2} - \frac{1}{u_5^2} & 0 & 0 & 0 & 0 \\ \frac{1}{u_3^2} & 0 & 0 & 0 & -\frac{1}{u_3^2} - \frac{1}{u_6^2} & -\frac{1}{u_6^2} & \frac{1}{u_6^2} & 0 \\ 0 & \frac{1}{u_4^2} & \frac{1}{u_4^2} & 0 & -\frac{1}{u_6^2} & -\frac{1}{u_4^2} - \frac{1}{u_6^2} & \frac{1}{u_6^2} & 0 \\ 0 & 0 & 0 & 0 & \frac{1}{u_6^2} & \frac{1}{u_6^2} & -\frac{1}{u_6^2} - \frac{1}{u_7^2} & \frac{1}{u_7^2} \\ 0 & 0 & 0 & 0 & 0 & 0 & \frac{1}{u_7^2} & -\frac{1}{u_7^2} - \frac{1}{u_8^2} \end{bmatrix} \tag{14}$$

We analyze the convexity of U by using an adaptation of Sylvester's criterion for semi-definite positive/negative matrices. Basically, we show that the Hessian matrix of U, i.e. $\boldsymbol{H}_U(\boldsymbol{\pi})$, is negative semi-definite by showing that all its principal minors of order k have sign $(-1)^k$ [9]. This means that function U is concave.

Let us consider a principal minor $\Delta_{i_1,i_2,\ldots,i_k}$, of the Hessian matrix $\boldsymbol{H}_U(\boldsymbol{\pi})$, such that $1 \leq i_1 < i_2 < \cdots < i_k \leq n$ are k arbitrary arcs of the intermediation DAG G. Let us consider the undirected subgraph G', obtained by discarding arc orientation of the subgraph of G, induced by the subset of $\{i_1, i_2, \ldots, i_k\}$ of arcs. Let V' be the set of vertices of G'.

If G' does contain cycles, then it is not difficult to observe that $\Delta_{i_1,i_2,\ldots,i_k} = 0$. If we consider a cycle (together with its orientation), we can assign an arbitrary orientation to each of its arcs. Then, the algebraic sum of the Hessian columns corresponding to the arcs of this cycle, signed according to the arc orientation with respect to the cycle orientation, will be obviously zero, as the arcs determine a cycle in the graph. For example, let us consider the cycle $(1,3), (3,6), (6,4), (4,1)$ of the DAG from Figure 1. Using arc indices, and assigning $+$ or $-$ to each arc according to its orientation with respect to the cycle orientation, we obtain the cycle $1, 5, -6, -2$. Now it is very easy to check that adding columns 1, and 5 and then subtracting columns 2 and 6, of $\boldsymbol{H}_U(\boldsymbol{\pi})$, we obtain zero.

If G' does not contain any cycles, then let us consider the connected components G'_j of G', $1 \leq j \leq p$, such that G'_j contains k_j arcs, $\sum_{j=1}^{p} k_i = k$ and has vertices V'_j. Then, minor $\Delta_{i_1,i_2,\ldots,i_k}$ is defined by equation (15).

$$\Delta_{i_1,i_2,\ldots,i_k} = (-1)^k \prod_{j=1}^{p} \sum_{\substack{S_j \subseteq V'_j \\ |S_j| = k_j}} \prod_{i \in S_j} \frac{1}{u_i^2} \tag{15}$$

Equation (15) clearly shows that $(-1)^k \Delta_{i_1,i_2,\ldots,i_k} > 0$. So, finally, we can conclude that social welfare function U is concave. As U is defined on a polytope domain, i.e. a bounded convex set, U has a local maximum and the local maximum is also a global maximum, thus proving the existence of an optimal pricing strategy of the network participants that maximizes the Nash social welfare.

We close this section with an example, showing the value of minor $\Delta_{1,3,4}$ of the intermediation DAG from Figure 1. Note that arc 1 is $(1,3)$, arc 3 is $(2,4)$, and arc 4 is $(2,5)$, so graph G' has 5 vertices $\{1,2,3,4,5\}$ and $p = 2$ connected components defined by subsets of vertices $\{1,3\}$ and $\{2,4,5\}$. Moreover $k_1 = 1$ and $k_2 = 2$. Graph G' does not contain cycles, so $\Delta_{1,3,4}$ is defined by equation (16).

$$\Delta_{1,3,4} = -\left(\frac{1}{u_1^2} + \frac{1}{u_3^2}\right)\left(\frac{1}{u_2^2 u_4^2} + \frac{1}{u_2^2 u_5^2} + \frac{1}{u_4^2 u_5^2}\right) \tag{16}$$

IV. Computational Experiments

A. Experimental Setting

We have performed computational experiments, for assessing the correctness and feasibility of our approach using the CVXOPT package for convex optimization [3]. We have used the 64-bit (AMD64) version of Python 3.7.3 on an *x64*-based PC with a 2 cores/4 threads Intel© Core™i7-5500U CPU at 2.40 GHz, running Windows 10.

Our problem was formulated as a nonlinear convex optimization problem, suitable CVXOPT. We introduce the matrices G of size $(e+n) \times e$ and $\boldsymbol{h}$ of size $(n+e) \times 1$ using equations (17).

$$\begin{aligned} G &= \left[\begin{array}{c} -D \\ -I_e \end{array} \right] \\ \boldsymbol{h} &= \left[\begin{array}{c} \boldsymbol{b} \\ 0 \end{array} \right] \end{aligned} \tag{17}$$

Now, the nonlinear convex optimization problem, prepared for CVXOPT, can be described using equations (18).

$$\begin{array}{ll} \text{minimize} & -U(\boldsymbol{\pi}) = -\sum_{i=1}^{n} \log(D\boldsymbol{\pi} + \boldsymbol{b})_i \\ \text{subject to} & G\boldsymbol{\pi} \preceq \boldsymbol{h} \end{array} \tag{18}$$

The implementation using CVXOPT using solver `cvxopt.solvers.cp` assumes the following steps [3]:

i) Definition of a Python function `U` that evaluates the objective and the constraint functions.
ii) Implementation of Python function `U` makes use of the Jacobian and Hessian matrices of the objective function. They must be implemented using the CVXOPT-specific data type `cvxopt.matrix`.
iii) Function `U` also makes use of a point `x0`, inside the domain of the objective function, so we had to compute this point before calling the solver. More details are given below.
iv) Prepare a series of data sets such that one data set represents a single problem defined by a specific intermediation DAG. More details about these data sets are provided below.
v) Define a main Python script that loads data sets, configures CVXOPT parameters, calls the solver function `cvxopt.solvers.cp`, and then extracts the solution.

We analyzed two options to determine one point inside the polytope domain:

i) To convert the polytope from halfspace to vertex representation [10], and then sample its interior using a uniform probability distribution.
ii) To directly chose one point using the halfspace representation. For that purpose we could its Chebyshev center representing the deepest point inside the polytope [6].

We performed some tests with both approaches, using the PYPOMAN package for polyhedral manipulations [7]. The first approach was not usable, because the vertex computation took too much computing time, probably because of the high number of generated polytope vertices. For example, for an intermediation DAG with 15 vertices and 35 arcs (i.e. 35 dimensions of our polytope), the resulting polytope had 58795 vertices. So we used the second approach in our experiments. Nevertheless, it was good to have two different methods to generate points inside the polytope, at least for smaller problem sizes, to be able to check the convergence of the method with initial points generated by distinct methods.

B. Data Sets

We randomly generated a number of DAGs of increasing sizes, representing intermediation networks. We used the following parameters:

i) Number n of graph nodes, an element of the set $\{5, 6, 7, 8, 9, 10, 15, 20, 30, 40, 50\}$.
ii) The density factor f of the graph. The higher is this factor the more arcs are in the graph. Value of f is given as a percentage in the set $\{10, 20, 30, 40, 50, 60, 70, 80, 90\}$.
iii) Number ng of generated graphs for each value of n and f. We have chosen $ng = 10$.

It follows that a total number of $11 \times 9 \times 10 = 990$ DAGs were generated. Each DAG is represented by its adjacency matrix A defined according to equation (19), and then converted into its incidence matrix representation D for the optimization purpose.

$$A_{i,j} = \begin{cases} 1 & \text{if there is an arc from node } i \text{ to node } j \\ 0 & \text{otherwise} \end{cases} \tag{19}$$

The adjacency matrix of each DAG was stored into a separate text file. The name of this file included the generating parameters. For example, the 5-th DAG with 15 vertices and density factor 40% is stored into a text file with name `graf-15-40-5.txt`. This information is used in description of experimental results.

Before closing this section, we follow with two remarks regarding the random DAGs generating process:

i) The first problem that had to be solved, during the generation process, was to assure that the resulting directed graph is a DAG. Simply generating random graphs with a pre-imposed density factor is not enough. This constraint was fulfilled by generating only adjacency matrices in upper triangular form. On the one hand, this assures that the resulting directed graph is a DAG. Conversely, the process is not restrictive, as the adjacency matrix of any DAG can be converted into upper triangular form by renumbering the graph nodes according to the topological sorting of the graph.
ii) During the generation process, we had to ignore (as not relevant) those DAGs that contain "singular" nodes, i.e. nodes without incoming and outgoing arcs, as these graphs do not satisfy the requirements set for intermediation DAGs.

Finally, we had to choose values for the limit prices of buyers and sellers. Note that, according to the results from [5], a necessary condition for the collective profitability of an intermediation DAG is defined by equation (20).

$$\sum_{b \in \mathcal{B}} \beta_b \geq \sum_{s \in \mathcal{S}} \sigma_s \tag{20}$$

Let $n_b = |\mathcal{B}|$ and $n_s = |\mathcal{S}|$. We have chosen $\boldsymbol{\sigma}$ and $\boldsymbol{\beta}$ according to equations (21). Note that this choice automatically assures that condition (20) is fulfilled. Nevertheless, this condition is only necessary, but not sufficient. This means that for few data sets, the optimization procedure might fail to produce a solution, as such a solution does not exist (this will be highlighted by the experimental results presented in the next section).

$$\begin{array}{llll} \sigma_s & = \sigma & = 100 & \text{for all } s \in \mathcal{S} \\ \beta_b & = \beta & = \sigma \times (\lceil \frac{n_b}{n_s} \rceil + 1) & \text{for all } b \in \mathcal{B} \end{array} \tag{21}$$

For example, using equations (21), if there are 3 buyers and 2 sellers we obtain limit price 100 of each seller and limit price 300 of each buyer.

C. Results and Discussion

We present, in Table I, information about the size of our data sets. We have evaluated the minimum and maximum number of arcs for each DAG in the subset of DAGs with the same number of nodes. Moreover, we have compared the maximum number of arcs, in our data sets, with the maximum number of arcs of a DAG with a given number n of nodes. Note that this value is obtained when the adjacency matrix has only values in the upper triangle, i.e. the total number of arcs is $\frac{n(n-1)}{2}$.

TABLE I
SIZE OF DATA SETS

# nodes	**# arcs** e		
n	***Min. (data set)***	***Max. (data set)***	***Max.*** $\frac{n(n-1)}{2}$
5	3	15	15
6	3	15	15
7	4	21	21
8	4	28	28
9	5	36	36
10	7	45	45
15	12	99	105
20	19	177	190
30	39	401	435
40	67	721	780
50	107	1111	1225

We can observe that for smaller number of nodes $n \in \{5, 6, 7, 8, 9, 10\}$ our data sets contain graphs with maximum number of arcs. However, for larger values of n this is not true. For example, for graphs with $n = 40$ nodes, the maximum number of arcs in our data set is 721.

Note that the number of nodes and arcs of the DAG provides a useful information about the "size" of the optimization problem. Here, n gives the number of network participants and it also represents the number of terms of the sum that defines the Nash social welfare utility function (see equation (5)). The e is the number of transaction prices, and also represents the number of decision variables of our optimization problem. According to Table I, our largest optimization problem had 1111 decision variables. However, the largest problem that we were able to solve successfully, was `graph-50-90-2` of size 1102.

We have developed a Python script for running the convex optimization solver on each DAG in our data set. The solver was configured as follows:

- `'maxiters'` parameter, denoting the maximum number of iterations, was set to `30`.
- `'refinement'` parameter, denoting the number of iterative refinement steps when solving KKT equations, was set to `2`.
- `'show_progress'` parameter, for turning on the output of the optimization progress, was set to `True`.
- `'abstol'`, `'reltol'`, and `'feastol'` parameters, were set to their default values `1e-7`, `1e-6`, and `1e-7`.

Note that the maximum number of iterations for successfully solving an optimization problem was 29 and it was achieved for data set `graph-40-10-8`.

We also recorded the execution times required for running the optimization solver, for each set of graphs with a given number of nodes. The results are recorded in Table II. Note that the time required for running the solver, for the problems with 50 nodes, was significantly larger than for the rest of the problems. This was also partly influenced by the fact that we did not obtain convergence for 13 problems in total, all of them for graphs with 50 nodes.

TABLE II
COMPUTATION TIME

# nodes n	**Time [sec.]**
5	1.875
6	1.875
7	1.953
8	2.062
9	2.906
10	3.437
15	6.390
20	9.890
30	38.843
40	134.109
50	1578.890
Total	1782.230

Before discussing the results of the optimization process, first note that, as we already pointed out in our previous paper [4], the sum of utilities of all the nodes of an intermediation DAG is constant, i.e. equality (22) holds.

$$\sum_{i=1}^{n} u_i = \sum_{b \in \mathcal{B}} \beta_b - \sum_{s \in \mathcal{S}} \sigma_s \tag{22}$$

So, denoting with u_δ the increment utility defined by equation (23), we can immediately apply the classic inequality between arithmetic and geometric means, to obtain the upper bound of the Nash social utility function given by equation (24).

$$u_\delta = (\sum_{b \in \mathcal{B}} \beta_b - \sum_{s \in \mathcal{S}} \sigma_s)/n \tag{23}$$

$$U(\boldsymbol{\pi}) \leq n \log u_\delta \tag{24}$$

Note, however that the upper bound stated by equation (24), is achieved if and only if the system of linear equations (25) has positive solutions.

$$(D\boldsymbol{\pi} + \boldsymbol{b})_i = u_\delta \quad \text{for all } i = 1, \ldots, n \tag{25}$$

Based on this observation, we can distinguish between two situations, when our solver returns an optimal solution:

- The problem has an optimal solution for which utilities of all participants are equal. In this case the solution can be obtained by solving linear system (25), to produce positive values representing feasible transaction prices that determine equal utilities for all participants. In what follows we will call this case "trivial" (for obvious reasons).
- Linear system (25) does not have positive solutions. In this case we must run the optimization solver to produce a solution that maximizes that Nash social welfare, but utilities of participants are not equal. In what follows we will call this case "non-trivial" (for obvious reasons).

Moreover, there are two situations when the optimization process does not produce solutions:

- The optimization process terminates after reaching the maximum number of iterations, without reaching convergence. This situation is labeled "unknown", meaning that the optimality status of the solution is not known.
- The optimization process terminates abruptly when the solver raises an exception. This situation is labeled "exception". One possible cause for this could be that the intermediation DAG is not collectively profitable. Recall that we generated our data sets by assuring only the necessary condition of collective profitability, so it is possible that our data sets contains intermediation DAGs, which are not collectively profitable.

Table III presents the optimization results, for each set of graphs with a given number of nodes, by distinguishing each of the four possible optimization outcomes: "trivial", "non-trivial", "exception", and "unknown". Note that in most cases, the solver obtained the trivial solution. This is not a surprise. However, for a significant number of cases between 17.5% (for $n = 40$) and 44.2% (for $n = 7$), the solver obtained a non-trivial solution. Actually these values clearly depend both on the DAG structure, as well as on the values of limit prices $\boldsymbol{\sigma}$ and $\boldsymbol{\beta}$. An example of non-trivial solution is obtained for problem `graph-50-20-9`. The problem has 50 nodes, 216 arcs, and the convergence was obtained after 22 iterations.

TABLE III
OPTIMIZATION RESULTS

# nodes	**# arcs e**		**# arcs e**	
n	***Trivial***	***Non-trivial***	***Exception***	***Unknown***
5	68	22	0	0
6	68	22	0	0
7	61	27	2	0
8	72	15	3	0
9	65	20	5	0
10	68	21	1	0
15	72	14	4	0
20	61	27	2	0
30	80	9	1	0
40	82	7	1	0
50	67	9	1	13

Note also that the solver terminated for at least one case with exception (most probably meaning that the DAG is not collectively profitable), for all the values on n between 7 and 50. A closer analysis revealed that in all these situations (there are 20 in total, see column labeled *Exception* of Table III) the density factor of the graph was at most 40%, while in 17 of these cases the density factor of the graph was at most 20% – i.e. the graph was sparse.

Finally, note that for 9 problems of 50 nodes the optimization solver terminated neither with exception, nor with convergence (see column labeled *Unknown* of Table III). These situations were probably caused by the internal functionality of the solver. It is interesting to observe that this happened only for dense graphs, with density factor at least 70%.

V. Conclusions and Future Works

In this paper we formulated the problem of determining optimal pricing strategies, for maximizing the Nash social welfare of participants to an intermediation network in semi-competitive environments, as a nonlinear convex optimization problem. We have theoretically proven that, if the network is collectively profitable, then there exists a globally optimal pricing strategy of the participants that maximizes their the Nash social welfare. We have also presented results of computational experiments using a nonlinear convex optimization package, to support our conclusions, and illustrate the feasibility of model.

Acknowledgment

This paper is a part of the Serbia-Romania-Poland collaboration within multilateral agreement on "Agent systems and applications" and Romania-Poland collaboration within bilateral project "Novel methods for development of distributed systems".

References

[1] R. K. Ahuja, T. L. Magnanti, J. B. Orlin, Network Flows: Theory, Algorithms, and Applications, Pearson, 1993.

[2] M. Andersen, J. Dahl, Z. Liu, and L. Vandenberghe, "Interior-point methods for large-scale cone programming," in Optimization for Machine Learning. S. Sra, S. Nowozin, and S. J. Wright, Eds. MIT Press, 2011, pp. 1–26.

[3] M. Andersen, J. Dahl, and L. Vandenberghe, "CVXOPT User's Guide," Release 1.2.3 – February 5, 2019. https://cvxopt.org/.

[4] A. Bădică, C. Bădică, M. Ivanović, and I. Buligiu, "Collective Profitability and Welfare in Selling-Buying Intermediation Processes," in Computational Collective Intelligence. ICCCI 2016. Part II. N. Nguyen, L. Iliadis, Y. Manolopoulos, and B. Trawiński, Eds. Lecture Notes in Computer Science, 9876, Springer, Cham, 2016, pp. 14–24.

[5] A. Bădică, C. Bădică, M. Ivanović, and D. Logofătu, "Collective Profitability of DAG-Based Selling-Buying Intermediation Processes," in Intelligent Distributed Computing XII. J. Del Ser, E. Osaba, M. N. Bilbao, J. J. Sánchez-Medina, M. Vecchio, and X.-S. Yang, Eds. Studies in Computational Intelligence, 798, Springer, Cham, 2018, pp. 414–424.

[6] S. Boyd and L. Vandenberghe, Convex Optimization, Cambridge University Press, 2004.

[7] S. Caron, "PYthon module for POlyhedral MANipulations – PYPOMAN,", version 0.5.4, 2018. https://scaron.info/doc/pypoman/

[8] M. Kaneko and K. Nakamura, "The Nash Social Welfare Function," Econometrica, vol. 47, no. 2, 1979, pp. 423–435.

[9] J. E. Prussing, "The Principal Minor Test for Semidefinite Matrices", Journal of Guidance, Control, and Dynamics, vol. 9, no. 1, 1986, pp. 121—122.

[10] R. R. Thomas, Lectures in Geometric Combinatorics, American Mathematical Society, 2006.