The V L Value for Network Games

Hendrickx, R.L.P.; Borm, P.E.M.; van den Brink, J.R.; Owen, G.

Publication date:
2005

CentER

Discussion Paper

No. 2005–65

THE *V L* VALUE FOR NETWORK GAMES

By Ruud Hendrickx, Peter Borm, René van den Brink,
Guillermo Owen

April 2005

ISSN 0924-7815

TILBURG UNIVERSITY

The VL value for network games

Ruud Hendrickx[1,2] Peter Borm[1] René van den Brink[3]
Guillermo Owen[4]

Abstract

In this paper we consider a proper Shapley value (the VL value) for cooperative network games. This value turns out to have a nice interpretation. We compute the VL value for various kinds of networks and relate this value to optimal strategies in an associated matrix game.

Key words: undirected graph, network game, proper Shapley value, matrix game, optimal strategy
JEL classification number: C71

1 Introduction

Van den Brink et al. (2005) study symmetric networks (undirected graphs) by means of associated network games. In a network game, the value of a coalition of nodes represents its "control" in the network. The Shapley value of such a game then reflects in some sense the relative power of the nodes in the network. Van den Brink et al. (2005) characterise this value in terms of the underlying graph.

Vorob'ev and Liapounov (1998) introduce the concept of proper Shapley value. A proper Shapley value of a transferable utility game is defined as a fixed point of the so-called corresponding Shapley mapping. They furthermore provide a constrained maximisation problem, whose unique solution (henceforth, the VL value) is shown to be a proper Shapley value.

In this paper, we consider the VL value of network games. We examine the VL maximisation problem in terms of the underlying graph rather than the associated

[1]CentER and Department of Econometrics and Operations Research, Tilburg University.
[2]Corresponding author: P.O. Box 90153, 5000 LE Tilburg, The Netherlands. E-mail: ruud@uvt.nl.
[3]Department of Econometrics and Tinbergen Institute, Free University, Amsterdam.
[4]Department of Mathematics, Naval Postgraduate School, Monterey, CA.

game. The shape of the graph gives some clues about what the VL value looks like. For some special classes of graphs, we provide a full description of the VL value.

Instead of taking an axiomatic approach, as is done in Van den Brink et al. (2005), we focus on the interpretation of the VL maximisation problem and hence, of the VL value. We argue that the VL value can be seen as an optimal resource allocation in the graph when one wants to find a hidden substance in each of the nodes and the probability of finding this substance in a particular node is proportional to the amount of resources in this and all adjacent nodes.

When the object to be found in the graph is hidden in just one single node in such a way as to minimise the probability of finding it, another resource allocation problem arises. We present this non-cooperative problem in terms of an associated matrix game and show that for some classes of graphs, the two search problems have the same solution.

This paper is organised as follows. In section 2, we define some basic notions of networks, network games and proper Shapley values. The VL value is defined and analysed for various types of graphs in section 3. The interpretation of the VL value in terms of search problems is the subject of section 4, while section 5 deals with the associated matrix games.

2 Network games

In this section we first discuss some graph and game theoretic preliminaries before defining network games. A *(symmetric) network* or *undirected graph* is a pair (N, G), where $N = \{1, \ldots, n\}$ is a set of nodes and $G \subset \{\{i, j\} | i, j \in N, i \neq j\}$ is a set of edges, or links, between these nodes. Note that we restrict ourselves to irreflexive graphs, *ie*, $(i, i) \notin G$ for all $i \in N$. When no confusion arises, we represent the graph (N, G) by G.

We restrict ourselves to *connected* networks, *ie*, we assume that for every pair of nodes $i, j \in N, i \neq j$, there exists a sequence $h_1, \ldots, h_p$ such that $h_1 = i$, $h_p = j$, and $\{h_k, h_{k+1}\} \in G$ for all $k \in \{1, \ldots, p-1\}$. We denote the class of all connected networks on N by $\mathcal{G}^N$.

If $\{i, j\} \in G$ or $i = j$, then nodes i and j are called *relatives*[1]. By $R_G(i)$ we

[1] Van den Brink et al. (2005) allow for loops to be part of the graph and consequently have a different notion of relative.

denote the set of all relatives of node $i \in N$ in network G, *ie*, $R_G(i) = \{i\} \cup \{j \in N \,|\, \{i,j\} \in G\}$. For a set $S \subset N$ of nodes we denote $R_G(S) = \bigcup_{i \in S} R_G(i)$. A *pending node* is a node with exactly two relatives.

A *cooperative game with transferable utility* (or simply TU game) is a pair (N, v) (again, often represented by v) with player set $N = \{1, \ldots, n\}$ and *characteristic function* $v : 2^N \to \mathbb{R}$ satisfying $v(\emptyset) = 0$.

For every coalition $T \subset N, T \neq \emptyset$, the *unanimity game* u_T is defined by $u_T(S) = 1$ if $S \subset T$ and $u_T(S) = 0$ otherwise. The class of unanimity games constitutes a basis for the space of all TU games:

$$v = \sum_{T \subset N, T \neq \emptyset} \Delta_v(T) u_T,$$

where the coefficients, called *dividends*, are defined by $\Delta_v(S) = v(S)$ if $|S| = 1$ and recursively $\Delta_v(S) = v(S) - \sum_{T \subsetneq S} \Delta_v(T)$ if $|S| \geq 2$.

A solution on a class $\mathcal{C}$ of TU games is a function $f : \mathcal{C} \to \mathbb{R}^N$, assigning an allocation $f(v) \in \mathbb{R}^N$ to every $v \in \mathcal{C}$. A well-known solution on the class of all TU games is the *Shapley value* (cf. Shapley (1953)), which for each coalition $S \subset N$ divides the dividend $\Delta_v(S)$ equally over all players in S:

$$\Phi_i^v = \sum_{\substack{S \subset N \\ i \in S}} \frac{\Delta_v(S)}{|S|}.$$

Given a system of positive weights $\omega \in \{x \in \mathbb{R}^N \,|\, \forall_{i \in N} : x_i > 0\}$, the *weighted Shapley value* $\Phi^v(\omega)$ of game v is given by $\Phi_i^v(\omega) = \sum_{\substack{S \subset N \\ i \in S}} \frac{\omega_i}{\omega(S)} \Delta_v(S)$ for all $i \in N$, where $\omega(S) = \sum_{i \in S} \omega_i$. Clearly, we obtain the Shapley value by taking equal weights. Of course, given a weight system ω, we get the same weighted Shapley value if we normalise the weights such that they add to one. Hence, we only consider weight systems taken from the unit simplex $\mathcal{S}^N = \{x \in \mathbb{R}^N \,|\, \forall_{i \in N} : x_i \geq 0, \sum_{i \in N} x_i = 1\}$.

Vorob'ev and Liapounov (1998) extend the definition of Φ^v to the whole unit simplex (and not just its interior) by taking its closure. The mapping $\Phi^v : \mathcal{S}^N \to \mathbb{R}^N$ that assigns to every weight system $\omega \in \mathcal{S}^N$ the weighted Shapley value $\Phi^v(\omega)$ of game v is called the *Shapley mapping* of game v. For a game v with nonnegative dividends and $v(N) = 1$, a fixed point of the Shapley mapping, *ie*, a vector $\overline{\omega} \in \mathcal{S}^N$ such that $\Phi^v(\overline{\omega}) = \overline{\omega}$, is a *proper Shapley value* (cf. Vorob'ev and Liapounov (1998)).

Van den Brink et al. (2005) define for each network (N, G) an associated conservative network power game (henceforth *network game*) (N, v_G), where we identify the nodes as players and the characteristic function (expressed in terms of our framework) is given by

$$v_G(S) = |\{i \in R_G(S) \mid R_G(i) \subset S\}|$$

for all $S \subset N$. That is, the worth of a coalition S of players equals the number of relatives of S that have no relatives outside S. The idea behind this network game is that it in some sense measures the "power" of the nodes in the network (cf. Van den Brink et al. (2005)). Because in our framework $i \in R_G(i)$ for all $i \in N$, only players inside S can be involved:

$$v_G(S) = |\{i \in S \mid R_G(i) \subset S\}|.$$

From the recursive formula for the dividends it is readily seen that

$$\Delta_{v_G}(S) = |\{i \in S \mid R_G(i) = S\}|$$

for all $S \subset N, S \neq \emptyset$ and hence,

$$\begin{aligned} v_G &= \sum_{T \subset N, T \neq \emptyset} \Delta_{v_G}(T) u_T \\ &= \sum_{T \subset N, T \neq \emptyset} |\{i \in T \mid R_G(i) = T\}| u_T \\ &= \sum_{i \in N} u_{R_G(i)}. \end{aligned}$$

3 The VL value for network games

Vorob'ev and Liapounov (1998) show that one particular proper Shapley value of v (henceforth the VL value) is given by the solution of the maximisation problem

$$\max_{x \in \mathcal{S}^N} \prod_{S \subset N} (\sum_{j \in S} x_j)^{\Delta_v(S)}.$$

In this section, we are going to consider this maximisation problem for network games. Each network game has nonnegative dividends, so the VL value of its 1-normalisation[2] is well-defined. Because $v_G(N) = n$ for each $G \in \mathcal{G}^N$, we can consider the maximisation problem without first normalising the game and still compare the

[2]The 1-normalisation w of a game v with $v(N) \neq 0$ is defined by $w(S) = \frac{v(S)}{v(N)}$ for all $S \subset N$.

results (which with slight abuse of terminology we refer to as VL values) across various networks on the same set of players.

For a network $G \in \mathcal{G}^N$, the VL value is given (using (2)) by the maximisation problem $\max_{x\in\mathcal{S}^N} \Pi_{S\subset N}(\sum_{j\in S} x_j)^{|\{i\in S\,|\,R_G(i)=S\}|}$. This problem can be rewritten as $\max_{x\in\mathcal{S}^N} \Pi_{i\in N} \sum_{j\in R_G(i)} x_j$, or equivalently,

$$\begin{array}{lll} \max & \prod_{i\in N} y_i & \\ \text{such that} & y_i = \sum_{j\in R_G(i)} x_j & \text{for all } i \in N, \\ & \sum_{j\in N} x_j = 1, & \\ & x_j \geq 0 & \text{for all } j \in N. \end{array} \tag{3.1}$$

Throughout the remainder, we use the vectors x and y as in program (3.1). If $n = 1$ or $n = 2$, then clearly every x is optimal, so from now on, assume $n \geq 3$. By taking $x = (\frac{1}{n}, \ldots, \frac{1}{n})$, we immediately conclude that the optimal value of (3.1), as well as the corresponding values for y, are positive.

In general, solving (3.1) can be quite complicated, depending on the form of the graph G. One obvious first step in finding the solution is presented in the following lemma.

Lemma 3.1 *Let $i \in N$ be a pending node in network G. Then it can never be optimal to have $x_i > 0$.*

Proof: The single node $j \in R_G(i)\backslash\{i\}$ must have at least two relatives, because G is connected and $n \geq 3$. Let $x \in \mathcal{S}^N$ be such that $x_i > 0$, then x' given by $x'_i = 0$, $x'_j = x_j + x_i$ and $x'_k = x_k$ for all other $k \in N$, gives an improvement, since it weakly increases y_k for all $k \in R_G(j)\backslash\{i, j\}$, at least one of them strictly, while otherwise leaving y unchanged. □

We are going to solve the maximisation program (3.1) for some special classes of networks. Clearly, the value $f(G)$ of the program (3.1) is positive and cannot be greater than one. This upper bound is obtained for networks that have a central node that is directly connected to all other nodes.

Proposition 3.2 *For every $G \in \mathcal{G}^N$ it holds that $f(G) \leq 1$. Moreover, $f(G) = 1$ if and only if there exists a node $i \in N$ such that $R_G(i) = N$.*

Proof: It is obvious that $f(G) \leq 1$. First, assume that $R_G(i) = N$ for some $i \in N$. Take $x \in \mathcal{S}^N$ such that $x_i = 1$ and $x_j = 0$ for all $j \in N\backslash\{i\}$. Since $i \in R_G(j)$ for all

$j \in N$, $y_j = 1$ for all $j \in N$ and $f(G) = 1$.
Next, assume that G is such that there is no $i \in N$ such that $R_G(i) = N$. Let $x \in \mathcal{S}^N$ and let $i \in N$ be such that $x_i > 0$. Take $j \in N \backslash R_G(i)$, which is possible by assumption. Then $y_j \leq 1 - x_i < 1$ and consequently, the objective value is smaller than 1. Since this holds for all $x \in \mathcal{S}^N$, $f(G) < 1$. □

In particular the maximal value $f(G)$ is obtained for a *star network*, *ie*, a network G such that there is an $i \in N$ with $R_G(i) = N$ and $R_G(j) = \{i, j\}$ for all $j \in N \backslash \{i\}$. Also, $f(G) = 1$ if G is the *complete graph*: $\{\{i, j\} \mid i, j \in N, i \neq j\}$.

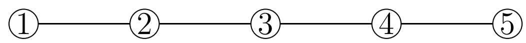

Figure 1: Line network

Next, we consider the case of a line, defined as a network G consisting of the edges $\{\{i, i+1\} \mid i \in \{1, n-1\}\}$, as depicted for $n = 5$ in Figure 1. Using Lemma 3.1, an optimal x must satisfy $x_1 = x_n = 0$. For all other i, the variable x_i contributes to three of the coordinates of y, and we conclude that in the optimum we must have

$$\sum_{i \in N} y_i = 3. \tag{3.2}$$

Hence, the arithmetic mean of the variables y_i must be equal to $3/n$. By a well-known inequality, the geometric mean cannot be greater than this. But our objective function is precisely the n-th power of the geometric mean, so for our program we obtain the theoretical upper bound

$$\max \prod_{i \in N} y_i \leq (3/n)^n.$$

Note, however, that this is an upper bound, not necessarily the maximum. To obtain this upper bound, all of the variables y_i (not just their arithmetic mean) must be equal to $3/n$. This cannot always be done, as is easily checked by considering the case $n = 4$. For $n = 6$, however, the assignment

$$x = (0, \frac{1}{2}, 0, 0, \frac{1}{2}, 0)$$

will clearly do the job. In fact, this can be done whenever n is a multiple of 3. For, in such a case, let x be given by

$$x_i = \begin{cases} \frac{3}{n} & \text{if } i \equiv 2, \\ 0 & \text{if } i \equiv 0, 1, \end{cases}$$

where all equivalences $\equiv$ are mod 3. It is easily checked that this will give the desired value $\frac{3}{n}$ for each coordinate of y.

For other n (*ie*, not multiples of 3), the situation is slightly more complicated. First, we consider the case $n \equiv 1$, so let $n = 3q + 1$ with $q \in \mathbb{N}$. Given (3.2) and because $\sum_{i \in N: i \equiv 1} y_i = \sum_{i \in N} x_i = 1$, the following program is a relaxation of (3.1) for this line (*ie*, this program provides an upper bound for the original program and its solution might or might not be feasible there):

$$\begin{array}{rll} \max & \prod_{i \in N} y_i & \\ \text{such that} & \sum_{i \in N: i \equiv 1} y_i = 1, & \\ & \sum_{i \in N: i \equiv 0,2} y_i = 2, & \\ & y_i \geq 0 & \text{for all } i \in N. \end{array}$$

The optimal solution to this problem is given by

$$y_i = \begin{cases} \frac{1}{q+1} & \text{if } i \equiv 1, \\ \frac{1}{q} & \text{if } i \equiv 0, 2. \end{cases}$$

If we can find a vector $x \in \mathbb{R}^N$ solving

$$\begin{cases} y_i = \sum_{j \in N: |i-j| \leq 1} x_j & \text{for all } i \in N, \\ \sum_{j \in N} x_j = 1, & \\ x_j \geq 0 & \text{for all } j \in N, \end{cases}$$

then this x maximises the original program (3.1). For all n (not necessarily $n \equiv 1$) we have to solve

$$\begin{cases} M_n x = y, \\ \sum_{i \in N} x_i = 1, \\ x \geq 0 \end{cases} \tag{3.3}$$

with

$$M_n = \begin{bmatrix} 1 & 1 & 0 & 0 & & \dots & 0 \\ 1 & 1 & 1 & 0 & & \dots & 0 \\ 0 & 1 & 1 & 1 & & \dots & 0 \\ & \vdots & & \ddots & & \vdots & \\ 0 & & \dots & & 1 & 1 & 1 \\ 0 & & \dots & & 0 & 1 & 1 \end{bmatrix}.$$

By Lemma 3.1 we have that $x_1 = 0$. From the first row of M_n it then follows that x_2 is uniquely determined. But then the third row uniquely determines x_3, and so on. Hence, the solution for x is unique.

For $n \equiv 1$, the solution is given by

$$x = \frac{1}{q(q+1)}(0, q, 1, 0, q-1, 2, 0, q-2, 3, 0 \ldots, 0, 2, q-1, 0, 1, q, 0).$$

Next, we consider the case $n \equiv 2$, so let $n = 3q+2$ with $q \in \mathbb{N}$. Then the original program (3.1) can be relaxed to:

$$\begin{array}{ll} \max & \prod_{i \in N} y_i \\ \text{such that} & \sum_{i \in N: i \equiv 1,2} y_i = 2, \\ & \sum_{i \in N: i \equiv 0} y_i = 1, \\ & y_i \geq 0 \qquad\qquad \text{for all } i \in N. \end{array}$$

This maximisation problem has the following solution:

$$y_i = \begin{cases} \frac{1}{q+1} & \text{if } i \equiv 1, 2, \\ \frac{1}{q} & \text{if } i \equiv 0. \end{cases}$$

Again, we solve $M_n x = y$, which with Lemma 3.1 results in the following unique solution:

$$x = \frac{1}{q(q+1)}(0, q, 0, 1, q-1, 0, 2, q-2, 0 \ldots, 0, q-1, 1, 0, q, 0).$$

The solution of (3.1) in case G is a line is summarised in the following proposition.

Proposition 3.3 *Let $G \in \mathcal{G}^N$ be a line and let $n = 3q + r$ with $q \in \mathbb{N}, r < 3$. The solution of (3.1) is given by the following three cases, depending on r (all equivalences are mod 3):*

a. $r = 0$. *Then* $y_i = \frac{1}{q}$ *for all* $i \in N$ *and* $f(G) = (\frac{1}{q})^n$.

b. $r = 1$. *Then* $y_i = \begin{cases} \frac{1}{q+1} & \text{if } i \equiv 1, \\ \frac{1}{q} & \text{if } i \equiv 0, 2 \end{cases}$ *and* $f(G) = (\frac{1}{q+1})^{(q+1)}(\frac{1}{q})^{(2q)}$.

c. $r = 2$. *Then* $y_i = \begin{cases} \frac{1}{q+1} & \text{if } i \equiv 1, 2, \\ \frac{1}{q} & \text{if } i \equiv 0 \end{cases}$ *and* $f(G) = (\frac{1}{q+1})^{2(q+1)}(\frac{1}{q})^q$.

The corresponding value of x is unique and determined by (3.3).

One natural question is whether for a given set N of nodes, the minimum value of $f(G)$ over all networks G on N is obtained when G is a line. This turns out not to be the case, as is shown in the next example.

Example 3.1 Consider $N = \{1, \ldots, 7\}$. For the line network G^ℓ on N, we use Proposition 3.3 to get

$$f(G^\ell) = (\frac{1}{3})^3(\frac{1}{2})^4 = \frac{1}{432}.$$

Next, consider the network G depicted in Figure 2.

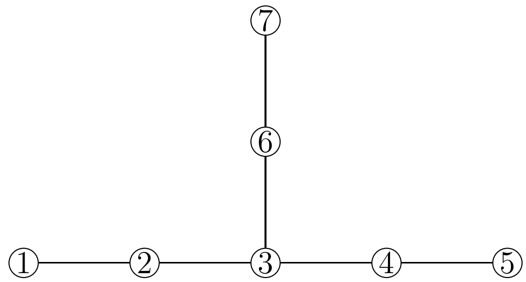

Figure 2: Network G

As can be easily seen, the maximum of (3.1) for this network is obtained in $x = (0, \frac{1}{3}, 0, \frac{1}{3}, 0, \frac{1}{3}, 0)$, so

$$f(G) = (\frac{1}{3})^6 \cdot 1 = \frac{1}{729}.$$

Hence, $f(G^\ell) > f(G)$. ◁

A last interesting type of network that we consider is a cycle, defined as a graph G consisting of the line and the extra link $\{n, 1\}$, as depicted for $n = 5$ in Figure 3. Again, for all $x \in \mathcal{S}^N$, we have (3.2) for the corresponding optimal y. Contrary to the line case, however, we can always find an x such that $y_i = \frac{3}{n}$ for all $i \in N$.

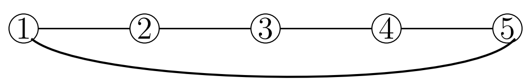

Figure 3: Cycle network

Proposition 3.4 *Let $G \in \mathcal{G}^N$ be a cycle. Then (3.1) is solved by $y_i = \frac{3}{n}$ for all $i \in N$, supported by the vector $x \in \mathcal{S}^N$ with $x_j = \frac{1}{n}$ for all $j \in N$. The optimal value equals $f(G) = (\frac{3}{n})^n$.*

4 Interpretation of the VL value

In this section, we provide an interpretation of the VL value for network games studied in the previous paragraph. For this, we interpret the variables x and y in program (3.1).

Assume that there is some noxious substance hidden within *each* of the nodes in a network. A searcher wishes to find and uproot this substance in all nodes. He does this by distributing some fixed amount of resources (normalised to 1) among the several nodes.

It is assumed that the probability of finding and uprooting the substance which is located at some particular node, over a small time interval, equals the amount of resources placed at that node and at all adjacent nodes. The probability of finding the substance at any one node is independent of what happens at any of the other nodes. It is necessary, however, to destroy all of the substance within a small time interval; otherwise, if any substance survives at any of the nodes, it can reproduce and will be able to reoccupy all of the nodes. Thus, if the process does not end, it begins anew at each time interval.

Assume that the searcher assigns resources equal to $x_i \geq 0$ at node $i \in N$ and that in total he has one unit of this resource available, so $\sum_{i \in N} x_i = 1$. Then the probability of locating the substance at i is $y_i = \sum_{j \in R_G(i)} x_j$.

Since it is necessary to locate the substance at all n nodes simultaneously (and these probabilities are independent by assumption) we find that the searcher must maximise the joint probability $F(x) = \prod_{i \in N} y_i = \prod_{i \in N} [\sum_{j \in R_G(i)} x_j]$ subject to the constraint that the quantities x_j must be non-negative and add to 1. In this way we are faced with the maximisation problem given by (3.1). Hence, the VL value for network games studied in the previous section is the resource allocation that maximises the probability of uprooting the noxious substance.

A different problem, which will be analysed in the next section, is in terms of terrorists. Instead of a noxious substance located within all nodes, a single terrorist hides himself in one particular node. The objective now is to maximise the probability of finding this terrorist before he carries out an attack. Assuming that the terrorist chooses his hiding location intelligently (*ie*, minimising the probability of being found), this problem can be described in terms of a matrix game. So, instead of maximising the product of the finding probabilities y_i, the searcher has to maximise their minimum.

5 A relation with optimal strategies in matrix games

For a network $G \in \mathcal{G}^N$, we denote by A^G the $N \times N$ *adjacency matrix* of G, so A^G is a symmetric matrix with

$$A^G_{ij} = \begin{cases} 1 & \text{if } \{i,j\} \in G, \\ 0 & \text{if } \{i,j\} \notin G. \end{cases}$$

This matrix A^G gives rise to a matrix game[3], where both players' strategy space is $\mathcal{S}^N$.

This matrix game describes the terrorist search mentioned in the previous section. Both player 1 (the searcher) and player 2 (the terrorist) place probabilities on the nodes. The searcher does this in such a way that he maximises his probability (given the terrorist's strategy choice) of ending up in a "1", *ie*, of finishing at the terrorist's or adjacent node. The terrorist's objective is to evade the searcher and maximise the probability of ending up in a "0".

Player 1's set of optimal strategies in the matrix game A^G is defined by

$$O_1(A^G) = \{x \in \mathcal{S}^N \mid \min_{z \in \mathcal{S}^N} x^\top A^G z = \max_{x' \in \mathcal{S}^N} \min_{z \in \mathcal{S}^N} (x')^\top A^G z\}.$$

In this section, we study the relationship between the VL value for the special types of graphs considered in Section 3 and player 1's optimal strategies in the matrix games arising from the corresponding adjacency matrices.

If $i \in N$ is such that $R_G(i) = N$, then the row corresponding to i in the adjacency matrix A^G contains only ones. It is immediately clear that then the vector $x \in \mathcal{S}^N$ with $x_i = 1$ and $x_j = 0$ for all $j \in N \backslash \{i\}$ is an optimal strategy for player 1 in the matrix game A^G.

Proposition 5.1 *Let $G \in \mathcal{G}^N$ be a network such that there is an $i \in N$ with $R_G(i) = N$. Then for the vector $x \in \mathcal{S}^N$ with $x_i = 1$ and $x_j = 0$ for all $j \in N \backslash \{i\}$ we have $x \in O_1(A^G)$.*

---

[3]Bapat and Tijs (1996) study another associated matrix game, in which player 1 puts probabilities on the nodes and player 2 puts probabilities on the edges.

So, for a star graph and a complete graph, finding an $x \in \mathcal{S}^N$ that maximises (3.1) boils down to finding an optimal strategy for player 1 in the associated matrix game A^G. Also for cycle networks, this phenomenon is immediate.

Proposition 5.2 *Let $G \in \mathcal{G}^N$ be a cycle network. Then $x \in \mathcal{S}^N$ with $x_i = \frac{1}{n}$ for all $i \in N$ is an element of $O_1(A^G)$.*

For line networks, the VL value is also an optimal strategy for player 1, as is shown in the following proposition.

Proposition 5.3 *Let $G \in \mathcal{G}^N$ be a line and let x be the optimal vector as given in Proposition 3.3. Then $x \in O_1(A^G)$.*

Proof: Let $n = 3q + r$ with $q \in \mathbb{N}, r < 3$. We first prove the statement for $r = 1$. Using Proposition 3.3, it suffices to show that for all $x' \in \mathcal{S}^N$,

$$\min_{i \in N}((x')^\top A^G)_i \leq \frac{1}{q+1}.$$

But this holds, since

$$\begin{aligned}
\min_{i \in N}((x')^\top A^G)_i &\leq \min_{i \in N: i \equiv 1}((x')^\top A^G)_i && (5.1)\\
&= \min_{i \in N: i \equiv 1} \sum_{j \in R_G(i)} x'_j \\
&\leq \frac{1}{q+1} \sum_{j \in N} x'_j && (5.2)\\
&= \frac{1}{q+1},
\end{aligned}$$

where (5.2) follows from the observation that the minimum of the numbers $\sum_{j \in R_G(i)} x'_j$ cannot exceed their average.
For $r = 0$, we use the same construction to obtain

$$\begin{aligned}
\min_{i \in N}((x')^\top A^G)_i &\leq \min_{i \in N: i \equiv 2} \sum_{j \in R_G(i)} x'_j \\
&\leq \frac{1}{q} \sum_{j \in N} x'_j \\
&= \frac{1}{q}
\end{aligned}$$

and for $r = 2$ we have

$$\begin{aligned}\min_{i\in N}((x')^\top A^G)_i &\leq \min_{i\in N: i\equiv 1,2} \sum_{j\in R_G(i)} x'_j \\ &\leq \frac{1}{2(q+1)} \sum_{j\in N} 2x'_j \\ &= \frac{1}{q+1}.\end{aligned}$$

□

Apparently, in case the graph is a star, line or cycle, the optimal searching strategies for finding the noxious substance and the terrorist are the same. In general, however, the shape of the graph does make a difference, as is shown in the following example.

Example 5.1 Consider the network G depicted in Figure 4. The optimal $x \in \mathcal{S}^N$ for this network is $x = (0, \frac{2}{3}, 0, \frac{1}{3}, 0)$. This vector, however, is not an optimal strategy for player 1, since

$$\min_{i\in N}(x^\top A^G)_i = \frac{1}{3},$$

while for $x' = (0, \frac{1}{2}, 0, \frac{1}{2}, 0)$,

$$\min_{i\in N}((x')^\top A^G)_i = \frac{1}{2}.$$

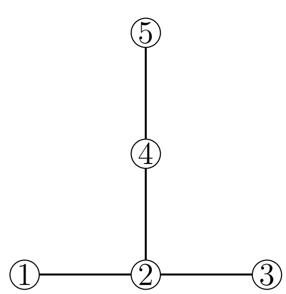

Figure 4: Network G

◁

References

Bapat, R. and S. Tijs (1996). Incidence matrix games. FEW Research Memorandum 716, Tilburg University, Tilburg, The Netherlands.

Brink, R. van den, P. Borm, R. Hendrickx, and G. Owen (2005). Characterisations of network power measures. Mimeo.

Shapley, L. (1953). A value for n-person games. In: H. Kuhn and A. Tucker (Eds.), *Contributions to the theory of games II*, Volume 28 of *Annals of Mathematics Studies*, pp. 307–317. Princeton: Princeton University Press.

Vorob'ev, N. and A. Liapounov (1998). The proper Shapley value. In: L. Petrosjan and M. Mazalov (Eds.), *Game Theory and Applications*, Volume 4, pp. 155–159. New York: Nova Science.