

# Power and Stability in Connectivity Games

Yoram Bachrach      Jeffrey S. Rosenschein  
School of Engineering and Computer Science  
The Hebrew University of Jerusalem, Israel  
{yori,jeff}@cs.huji.ac.il

Ely Porat  
Department of Computer Science  
Bar-Ilan University, Ramat-Gan, Israel  
porately@cs.biu.ac.il

## ABSTRACT

We consider computational aspects of a game theoretic approach to network reliability. Consider a network where failure of one node may disrupt communication between two other nodes. We model this network as a simple coalitional game, called the vertex Connectivity Game (CG). In this game, each agent owns a vertex, and controls all the edges going to and from that vertex. A coalition of agents wins if it fully connects a certain subset of vertices in the graph, called the *primary* vertices.

We show that power indices, which express an agent's ability to affect the outcome of the vertex connectivity game, can be used to identify significant possible points of failure in the communication network, and can thus be used to increase network reliability. We show that in general graphs, calculating the Banzhaf power index is  $\#P$ -complete, but suggest a polynomial algorithm for calculating this index in trees. We also show a polynomial algorithm for computing the core of a CG, which allows a stable division of payments to coalition agents.

## Categories and Subject Descriptors

F.2 [Theory of Computation]: Analysis of Algorithms and Problem Complexity;

I.2.11 [Artificial Intelligence]: Distributed Artificial Intelligence—*Multiagent Systems*;

J.4 [Computer Applications]: Social and Behavioral Sciences—*Economics*

## General Terms

Algorithms, Theory, Economics

## Keywords

Computational complexity, Network Reliability, Power indices

## 1. INTRODUCTION

Making joint decisions and cooperating are key features of multiagent systems. Cooperative game theory treats many aspects of these issues, and can serve as a foundation for

**Cite as:** Power and Stability in Connectivity Games, Yoram Bachrach, Jeffrey S. Rosenschein and Ely Porat, *Proc. of 7th Int. Conf. on Autonomous Agents and Multiagent Systems (AAMAS 2008)*, Padgham, Parkes, Müller and Parsons (eds.), May, 12-16., 2008, Estoril, Portugal, pp. 993-1000.

Copyright © 2008, International Foundation for Autonomous Agents and Multiagent Systems (www.ifaamas.org). All rights reserved.

analyzing such systems. One specific topic which has been studied extensively in game theory, as well as in other fields such as political science and economics, is that of voting. Voting is used in a variety of multiagent systems, typically to enable a team of agents to make a group decision.

An interesting measure applied to voting scenarios is that of power indices; such indices measure the control a voter has over decisions of a larger group [5]. One popular power index is the Banzhaf power index [2]. While it has mostly been used for measuring power in weighted voting systems, it can easily be adapted for other domains as well.

In this paper, we consider the use of the Banzhaf index to find key points of failure in a communication network. We model a communication network composed of servers and network links between them as a vertex connectivity game. The network is modeled as a graph, where the servers are the vertices, and the network links are the edges. A certain subset of the servers (vertices) are *primary*—a failure to send information between any two of them would constitute a major system failure. Another subset of the servers are always available (backbone servers).

In the vertex *Connectivity Game* (CG) that we introduce, each agent controls a different vertex in the graph. A coalition of agents can use any of the vertices controlled by the coalition members or the backbone vertices, and may send information between them. The coalition wins if it connects all the primary vertices, so that it can send information between any two of them. The Banzhaf index of an agent in this game reflects its criticality in maintaining this connectivity. In communication networks, this index could thus enable an administrator to identify potential critical points of failure in the network.

We consider the computational complexity of calculating the Banzhaf power index in this domain. We show that in general graphs, this problem is  $\#P$ -complete. Despite this negative result, we provide a polynomial algorithm for the restricted case where the graph is a tree. Many networks, including parts of the internet's backbone, are constructed as trees when the construction of a communication line is expensive, so this algorithm can analyze important real-world domains.

We also consider another game theoretic solution concept, the core, and show that it can be computed in polynomial time in CGs. The core of the CG indicates which payoff vectors are stable. When a coalition in the CG manages to connect all the primary vertices it wins, and gains a certain profit. This profit should be divided among the members of the coalition. Choosing a payoff vector in the core guaran-

tees that no subcoalition would choose to split from the main coalition, and attempt to establish its own network. Thus, the core can be used to allocate the gains of a coalition in a CG domain in a stable way.

The paper proceeds as follows. In Section 2, we provide background information regarding coalitional games and the Banzhaf power index, and fully define a vertex connectivity game. In Section 3, we discuss the Banzhaf power index in vertex connectivity games, and present the hardness result for the general case. Section 4 discusses vertex connectivity games in trees, and provides a polynomial algorithm for this case. Section 5 shows that the core of CGs can be computed in polynomial time. Section 6 discusses important related work, and we conclude in Section 7.

## 2. PRELIMINARIES

A coalitional game is composed of a set of  $n$  agents,  $I = (a_1, \dots, a_n)$ , and a function mapping any subset (coalition) of the agents to a real value  $v : 2^I \rightarrow \mathbb{R}$ . The function  $v$  is called the *coalitional function* (or sometimes the *characteristic function*) of the game. In a *simple* coalitional game,  $v$  only gets values of 0 or 1, so  $v : 2^I \rightarrow \{0, 1\}$ . A coalition  $C \subseteq I$  *wins* if  $v(C) = 1$ , and *loses* if  $v(C) = 0$ . The set of all winning coalitions is denoted  $W(v) = \{C \subseteq 2^I | v(C) = 1\}$ . An agent  $a_i$  is *critical* in a winning coalition  $C$  if the agent's removal from that coalition would make the coalition lose:  $v(C) = 1$  but  $v(C \setminus \{i\}) = 0$ . Thus, an agent can only be critical in a coalition that contains him.

We are interested in measuring the influence a given agent has on the result of the game. A common interpretation for this is the probability that this agent would significantly affect the outcome of the game.

One common power index is the Banzhaf power index [2]. It has been widely used for measuring political power in weighted voting systems, but its definition does not rely on the specific features of a weighted voting game. The index depends on the number of coalitions in which an agent is critical. The Banzhaf index of agent  $a_i$  is the proportion of all winning coalitions where  $a_i$  is critical, out of all winning coalitions that contain  $a_i$ .

DEFINITION 1. *The Banzhaf index is given by  $\beta(v) = (\beta_1(v), \dots, \beta_n(v))$  where*

$$\beta_i(v) = \frac{1}{2^{n-1}} \sum_{S \subseteq N | i \in S} [v(S) - v(S \setminus \{i\})].$$

When the agents are independent in their choices, so every coalition has an equal probability of occurring, the Banzhaf index measures the probability of an agent being critical; for the winning coalitions that contain  $a_i$ , the index counts in how many of them it is critical (i.e., its removal from the winning coalition would make the coalition lose). Other power indices reflect other assumptions. For example, the Shapley-Shubik index assumes the model that agents are randomly added to a coalition, so every *ordering* of the agents is equally probable. The index then measures in how many of the orderings  $a_i$  is the first agent that makes the coalition win (i.e., the first agent that, when added to the previous agents, makes the coalition win).<sup>1</sup> In this paper, we consider server failures in a network. Since each server

<sup>1</sup>Although the Banzhaf and Shapley-Shubik indices are sim-

ilar, the Banzhaf index considers all possible *subsets* of agents, whereas the Shapley-Shubik index is defined over all *orderings* (or permutations) of the agents, so the two indices can be computationally quite different.

may fail independently from the others, the Banzhaf power index is more appropriate for this case. In simple coalitional games, the coalitional function  $v$  only gets values of 0 or 1. In general coalitional games, this function may have any value in  $\mathbb{R}^+$ , and we say this function determines the total gains of a coalition. The coalitional function only defines the utility a coalition can achieve, but does not define how the agents distribute this utility among themselves. A payoff vector  $(p_1, \dots, p_n)$  is a division of the gains of the grand coalition  $I$  among the agents, where  $p_i \in \mathbb{R}$ , and  $\sum_{i=1}^n p_i = v(I)$ . We call  $p_i$  the payoff of agent  $a_i$ , and denote the payoff of a coalition  $C$  as  $p(C) = \sum_{i \in \{j | a_j \in C\}} p_i$ . By assumption, agents are rational and attempt to maximize their own share of the utility. Game theory offers several solution concepts, determining which payoff vectors are possible when agents act rationally.

A simple requirement for the payoff vector is *individual-rationality*: for all agents  $a_i \in C$ , we have  $p_i \geq v(\{a_i\})$ , otherwise that agent is better off on its own. Similarly, we say a coalition  $B$  *blocks* the payoff vector  $(p_1, \dots, p_n)$  if  $p(B) < v(B)$ , since the members of  $B$  would split off from the coalition and gain more utility without the rest of the agents. If a blocked payoff vector is chosen, the coalition is unstable. A solution concept that emphasizes such stability is the core [8].

DEFINITION 2. *The Core of a coalitional game is the set of all payment vectors  $(p_1, \dots, p_n)$  that are not blocked by any coalition, so for any coalition  $C$  we have  $p(C) \geq v(C)$ .*

A value distribution in the core makes sure that no subset of the agents would split off, so the coalition is stable. In general the core can be empty, so every possible value division is blocked by some coalition. In this work, we give results regarding computing the core in vertex Connectivity Games. Another solution concept, which contains only a single possible payment vector, is the Shapley value, whose application to simple coalitional games is the Shapley-Shubik power index, discussed above. However, the core focuses on stability, and the Shapley value focuses on certain fairness conditions.

Our hardness result for calculating the Banzhaf power index in CGs considers the class #P. #P is the set of integer-valued functions that express the number of accepting computations of a nondeterministic Turing machine of polynomial time complexity. Let  $\Sigma$  be the finite input and output alphabet for Turing machines.

DEFINITION 3. *#P is the class consisting of the functions  $f : \Sigma^* \rightarrow \mathbb{N}$  such that there exists a non-deterministic polynomial time Turing machine  $M$  that for all inputs  $x \in \Sigma^*$ ,  $f(x)$  is the number of accepting paths of  $M$ .*

The complexity classes #P and #P-complete were introduced by Valiant [20]. These classes express the hardness of problems that “count the number of solutions”.<sup>2</sup>

ilar, the Banzhaf index considers all possible *subsets* of agents, whereas the Shapley-Shubik index is defined over all *orderings* (or permutations) of the agents, so the two indices can be computationally quite different.

<sup>2</sup>Informally, NP and NP-hardness deal with checking if at least one solution to a combinatorial problem exists, while #P and #P-hardness deal with calculating the *number* of solutions to a combinatorial problem. Counting the number of solutions to a problem is at least as hard as determining











