A COST ALLOCATION GAME RELATED TO A SEARCH PROBLEM

01105053　富山大学　菊田健作　KIKUTA Kensaku

1. Introduction

A cooperative game is defined via a cost allocation problem. We calculate the Shapley value of the game and interpret it to be an index of importance of each vertex.

2. The Model

A (*undirected*) *graph* G is an ordered pair (V,E) in which $V \equiv \{0,1,2,...,n\}$ is a finite set of *vertices*, and E is a finite set of pairs of different vertices, (i,j), called *edges*.. A *path* between $i_0, i_s \in V$ is a finite sequence of distinct edges of the form $(i_0,i_1),...,(i_{s-1},i_s)$. This is denoted as $(i_0,i_1,...,i_s)$. A *simple* path between i and j is a path between i and j with no repeated vertices.

We assume $G \equiv (V,E)$ is a *rooted tree*, and $0 \in V$ is designated the *root*. It is well-known that for any $i,j \in V$, there is uniquely a simple path between i and j. The set of vertices on this path is denoted by $[i,j]$. For $i,j \in V$ such that $i \neq j$, i is called an *ancestor* of j if $i \in [0,j]$. j is called a *descendant* of i if i is an ancestor of j. j is called a *child* of i if j is a descendant of i and $(i,j) \in E$. For $i \in V$, let D_i, K_i, and A_i for $i \neq 0$ be the sets of all descendants, all children and all ancestors of i respectively. We let $D \equiv D_0$. For any $j \in D$, there is uniquely $a(j) \in A_j$ such that $j \in K_{a(j)}$. Let $V_i \equiv \{i\} \cup D_i$. For $i \in V$ and $Y \subseteq D_i$, define $= \bigcup_{y \in Y}[i,y]$. Define a tree with i as its root by $G_{(i;Y)} \equiv (D_{(i;Y)} \cup \{i\}, \{(a(j),j) \in E : j \in D_{(i;Y)}\})$. For a nonnegative-valued function g on D, we let $g(Y) \equiv \sum_{i \in Y} g(i)$ for $Y \subseteq D$. We let $g(Y) = 0$ if $Y = \varnothing$. For a finite set X, $|X|$ is the cardinality of X. Each edge $(a(j),j)$ is associated with a positive number $d(j)$, called the *weight* of $(a(j),j)$. The *length* of a path is the sum of the weights of all the edges in the path. For $i,j \in V$, we define $d(i,j)$ by the length of the simple path between i and j. Clearly $d(a(j),j) = d(j)$ for $j \in D$.

Define a game on G. Player 1 (the hider, or **H**) hides among one of all vertices in D, and stays there. Player 2 (the searcher, or **S**) examines each vertex until **S** finds **H**, traveling along edges. It is assumed that at the beginning of the search **S** is at 0, and that **S** travels along the simple path between i and j when $(i,j) \notin E$ and **S** examines i after having examined j. Associated with the examination of $i \in D$ is the examination cost that consists of two parts:(I) a traveling cost $d(j,i) > 0$ of examining i after having examined j, and (II) an examination cost $c > 0$. There is not a probability of overlooking **H**, given that the right vertex is searched. We let $d(i,i) = 0$ for all $i \in D$. Before searching (hiding resp.), **S** (**H** resp.) must determine a strategy so as to make the cost of finding **H** as small (large resp.) as possible. A (*pure*) *strategy* for **H** is expressed by an element, say i, of D, which means **H** determines on hiding in i. D is the set of all strategies for **H**. A strategy for **S** is a permutation on D. Thus under a permutation σ, **S** examines Vertices $\sigma(1),\sigma(2),...,\sigma(n)$ in this order. We let $\sigma(n+1) = \sigma(0) = 0$. We assume

(2.1)　　**S** travels along each edge in E at most twice in his search.

Intuitively it is efficient in distance for **S** to choose a permutation which indicates a search procedure satisfying (2.1). For $Y \subseteq D$, let $\Sigma(Y)$ be the set of all permutations on Y which satisfies (2.1). We let $\Sigma \equiv \Sigma(D)$. For a strategy pair $(i,\sigma) \in D \times \Sigma$, the cost of finding **H**, written as $f(i,\sigma)$, is :

$$f(i,\sigma) = \sum_{x=1}^{\sigma^{-1}(i)} d(\sigma(x),\sigma(x-1)) + \sigma^{-1}(i)c. \tag{2.2}$$

Letting the payoff for **H** be $f(i,\sigma)$, we have a finite, two-person zero-sum game, denoted by $(f;D,\Sigma)$. Let $(f;P,Q)$ be the mixed extension of $(f;D,\Sigma)$ and we call it a game G just as we denote the graph. Similarly we can define a game on $G_{(i;Y)}(i \in V, Y \subseteq D_i)$, where at the beginning **S** is at i, and Y is the set of pure strategies of **H**. Call the mixed extension of it a game $G_{(i;Y)}$.

Theorem 2.1. The value of the game $G_{(i;Y)}$ is $C(i;Y) \equiv d(D_{(i;Y)}) + \frac{|Y|+1}{2}c$.

For each $S \subseteq D$, $C(S) \equiv C(0;S)$ is the expected search cost for the company. And in turn, $C(S)$ is the joint cost in which the facilities in S must pay when they consider the maintenance cooperatively. Define $v(S) = 0$ if $S = \emptyset$. For $S(\neq \emptyset) \subseteq D$, define $v(S) \equiv \sum_{i \in S} C(0;\{i\}) - C(0;S)$. $v(S)$ is the saving of the cost which is obtained by considering jointly the maintenance of the network of S. Our purpose is to consider how the total saving $v(N)$ should be reallocated to each company, applying the solution-concepts in cooperative games in characteristic-function form.

3. The Shapley Value of the Game

The *Shapley value* of a game (D,v) is defined to be : For $i \in D$,

$$\varphi_i(v) \equiv d(0,i) + c - c(i)$$

where $c(i) \equiv \sum_{S|i \notin S} \frac{(n-s-1)!s!}{n!}\{C(S \cup \{i\}) - C(S)\}.$

Theorem 3.1. For $i \in D$, let $(i_0, i_1, \ldots, i_m)$ be the simple path between $i_0 = 0$ and $i_m = i$. Then

$$c(i) = (1 + \frac{1}{n})\frac{c}{2} + \frac{d(i_1)}{|V_{i_1}|} + \ldots + \frac{d(i)}{|V_i|}.$$

References. [1]Granot,D.and Granot,F.:On Some Network Flow Games. *Mathematics of Operations Research* **1 7**(1992), 792-841.

[2] K. Kikuta: A Cost Allocation Game Related to a Search Problem I:The Kernel. Working Paper No.153, Fac. of Econ., Toyama Univ., September 1995.