数理解析研究所講究録
947巻 1996年 143-149

A Cost Allocation Game Related to a Search Problem

富山大学　菊田健作（Kensaku KIKUTA）

1. The Problem

There are facilities which are connected by a communication-network. A company manages the network. The network system may often break down by a damage of even only one point. Whenever the system breaks down the company must seek the damaged places, checking points one by one, and must repair them. This kind of maintenauce of the system is done daily and in the long run. The cost of maintenaunce is paid by all facilities jointly. Then this joint cost must be allocated among them. How much must each facility pay its share of the cost ?

In this paper the facilities with communication network are expressed by a graph. We assume the graph is a rooted tree. At the root there is the company. The other vertices are facilities. The expected search cost of a damaged place is calculated, following the minimax criterion. This means that we imagine someone (called the hider) damages the system deliberately, and for the company (called the searcher) it costs to look for the place, and the hider and the searcher do a zero-sum game in which the payoff to the hider is the search cost. Then the expected search cost is the value of the game. For each subset of the facilities we can calculate the expected search cost in the same way. So we have a cost function defined on the subsets of the facilities. Then we apply the cooperative game theory in order to allocate the joint cost. The kernel of a cooperative game is adopted as a solution. It was defined in Davis/Maschler[2]. Since the resulting cooperative game is seen to be convex, the definition of the kernel becomes simple.

About applications of cooperative game theory to cost allocations on the network, the reader can find sources in the references of Granot/Granot[4]. About search theory, or search games, see Ahlswede/Wegener[1], Gal[3], Nakai[10], and Ruckle[11].

2. The Model

A *graph* (or *undirected graph*) G is an ordered pair (V,E) in which $V \equiv \{0,1,...,n\}$ is a finite set of *vertices*, and E is a finite set of pairs of different vertices, (i,j), called *edges*. If $(i,j) \in E$, we say i and j are *adjacent*. A *path* between Vertices i_0 and i_s is a finite sequence of distinct edges of the form $(i_0,i_1),(i_1,i_2),...,(i_{s-1},i_s)$. If $i_0 = i_s$ then this path is called a *cycle*. A *simple* path between i and j is a path between i and j with no repeated vertices. G is said to be *connected* if for any $i,j \in V$, there is a path between i and j.

Throughout this paper we assume a graph $G \equiv (V,E)$ is a *rooted tree*, i.e., it is connected, it has n edges, it has no cycle, and Vertex 0 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 and adjacent to i. 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 $D \equiv D_0$. Define the set of *leaves* by $L \equiv \{i \in V : K_i = \varnothing\}$. 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 $D_{(i;Y)} \equiv Y \cup \{j \in D_i : j \in A_y \text{ for some } y \in Y\} = \bigcup_{y \in Y} [i,y]$. Define a tree with i as its root by $G_{(i;Y)} \equiv (D_{(i;Y)} \cup \{i\}, \{(a(j),j) \in E : j \in D_{(i;Y)}\})$. In this paper for a nonnegative-valued function g on D, we let $g(Y) \equiv \sum_{i \in Y} g(i)$, where $Y \subseteq D$. We let

$g(Y) = 0$ if $Y = \varnothing$. For a finite set X, $|X|$ is the cardinality of X. Each edge $(a(j), j)$ $(j \in D)$ 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, abbreviated as **H**) hides among one of all vertices in D, and stays there. Player 2 (the searcher, abbreviated as **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 $d(i, j) > 0$ 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. For convenience, 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. For convenience we let $\sigma(n+1) = \sigma(0) = 0$.

For simplicity we assume in this paper

(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 :

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

Letting payoffs for **H** and **S** be $f(i,\sigma)$ and $-f(i,\sigma)$ respectively, 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. The elements of P and Q are expressed as $p = (p(1),...,p(n)) \in P$ and $q = \{q(\sigma)\} \in Q$, where $p(i)$ is the probability that **H** chooses $i \in D$, and $q(\sigma)$ is the probability that **S** chooses $\sigma \in \Sigma$. Thus

(2.3) $p(D) = 1, p(i) \geq 0$ for all $i \in D$, and $q(\Sigma) = 1, q(\sigma) \geq 0$ for all $\sigma \in \Sigma$.

For a strategy pair $(p,q) \in P \times Q$, $f(p,q)$ is the expected cost of finding **H**. In the same way we can define a game on $G_{(i;Y)}$ $(i \in V, Y \subseteq D_i)$, in which at the beginning of the search **S** is at i, and Y is the set of pure strategies of **H**. Call the mixed extension of this game a game $G_{(i;Y)}$.

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

An outline of the proof is given in Section 5. For each $S \subseteq D$, $C(0;S)$ is the expected search cost for the company. And in turn, $C(0;S)$ is the joint cost in which the facilities in S must pay when they consider the maintenaunce cooperatively. For each $S \subseteq D$, define

$$(2.4) \qquad v(S) \equiv \sum_{i \in S} C(0;\{i\}) - C(0;S) = \sum_{i \in S} d(0,i) - d(D_{(0;S)}) + \frac{|S|-1}{2}c,$$

and $v(S) = 0$ if $S = \varnothing$. $v(S)$ is the saving of the cost which is obtained by considering jointly the maintenaunce of the network of S. Our purpose is to consider how the total saving $v(N)$ should be reallocated to each company. In order to analyse this, we apply the solution-concepts in cooperative games in characteristic-function form. In the next section we analyse a cooperative game (D,v).

3. Analysis of the Model.

In this section we show that the characteristic-function v defined in (2.4) is convex. Then we calculate the kernel of the game (D,v). About the computational complexity of solution-concepts in cooperative games related to cost allocation, see Meggido [9].

Proposition 3.1. (i) v is convex, i.e., $v(S) + v(T) \le v(S \cup T) + v(S \cap T)$ for all $S,T \subseteq D$.

(ii) If $j \in K_i$, then j is more desirable than i in (D,v). I.e.,

$$v(S \cup \{j\}) \ge v(S \cup \{i\}) \text{ whenever } i,j \notin S.$$

(iii) If $j \in K_i \cap L$, then i and j are symmetric. I.e.,

$$v(S \cup \{j\}) = v(S \cup \{i\}) \text{ whenever } i,j \notin S.$$

Outline of the proof: (i) By (2.4), for all $S,T \subseteq D$,

$$B \equiv v(S \cup T) + v(S \cap T) - v(S) - v(T)$$
$$= -d(D_{(0;S \cup T)}) - d(D_{(0;S \cap T)}) + d(D_{(0;S)}) + d(D_{(0;T)}).$$

It suffices to prove $B \ge 0$. By induction on the number of the vertices.

(ii)(iii) Assume $j \in K_i$. Whenever $i,j \notin S.$,

$$E \equiv v(S \cup \{j\}) - v(S \cup \{i\}) = d(j) + d(D_{(0;S \cup \{i\})}) - d(D_{(0;S \cup \{j\})}).$$

It suffices to prove $E \ge 0$ in (ii) and $E = 0$ in (iii). □

By Proposition 3.1(i) we see that the core of (D,v) is nonempty. Furthermore the nucleolus, the prekernel, and the kernel coincide and consists of a single point. For these facts, see Maschler/Peleg/Shapley[8] and Shapley[12]. So we identify the kernel as a point and it can be defined as follows. Define the set of pre-imputations by $\mathbf{X}^* = \{x \in R^n : x(D) = v(D)\}$, where the i-th component of x is denoted by $x(i)$, and it corresponds to $i \in D$. For each $x \in \mathbf{X}^*$ and $i,j \in D$, $i \ne j$, we define the maximum surplus of i over j by $s_{ij}(x) \equiv \max\{e(S,x) : S \in \sigma_{ij}\}$, where

$\sigma_{ij} \equiv \{S \subseteq D : i \in S \text{ and } j \notin S\}$ and $e(S,x) \equiv v(S) - x(S)$. The *kernel* of (D,v) is a point $x \in \mathbf{X}^*$ such that

$$s_{ij}(x) = s_{ji}(x) \text{ for all } i,j \in D, i \neq j. \tag{3.1}$$

Again, by Proposition 3.1(i), the kernel is included in the core, i.e., $e(S,x) \leq 0$ for all $S \subseteq D$ if x is the kernel point. By Proposition 3.1(ii) and (iii) and by Theorem 4.8 of Maschler [7], we have the next property of the kernel point.

Proposition 3.2. Assume x is the kernel point. If $j \in K_i$, then $x(j) \geq x(i)$. If $j \in K_i \cap L$, then $x(j) = x(i)$.

In order to calculate the kernel, first we must consider the maximum surplus, and we have the next proposition.

Proposition 3.3. Assume x is the kernel point.
(i) $s_{ij}(x) = \max\{e(D \setminus V_j, x), e(D \setminus \{j\}, x)\}$ and $s_{ji}(x) = e(D \setminus \{i\}, x)$ for $j \notin L$ and $j \in K_i$, $i \neq 0$.
(ii) $s_{ij}(x) = e(D \setminus \{j\}, x)$ and $s_{ji}(x) = e(D \setminus \{i\}, x)$ for $j \in L$ and $j \in K_i$.
(iii) $s_{ij}(x) = \max\{e(D \setminus V_j, x), e(D \setminus \{j\}, x)\}$ for $i,j \in K_0$.

Define $z(j) \equiv \max\{x(j) - d(j), x(V_j) - v(V_j)\}$ for all $j \in D$.

Theorem 3.4. Suppose $x \in \mathbf{X}^*(v)$. x is the kernel point if and only if :
(i) For i, j such that $j \in K_i$ and $i \neq 0$, $x(i) = z(j)$.
(ii) For $i,j \in K_0$, $z(i) = z(j)$.

Outline of the proof: Assume x is the kernel point. By (3.1) and Proposition 3.3, we have

$$\max\{e(D \setminus V_j, x), e(D \setminus \{j\}, x)\} = e(D \setminus \{i\}, x) \text{ for } j \notin L \text{ and } j \in K_i, \tag{3.2}$$

$$e(D \setminus \{j\}, x) = e(D \setminus \{i\}, x) \text{ for } j \in L \text{ and } j \in K_i, \tag{3.3}$$

$$\max\{e(D \setminus V_i, x), e(D \setminus \{i\}, x)\} = \max\{e(D \setminus V_j, x), e(D \setminus \{j\}, x)\} \text{ for } i,j \in K_0. \tag{3.4}$$

From these, (2.4) and (3.1), we have (i) and (ii). Assume $y \in \mathbf{X}^*(v)$ satisfies (i) and (ii). Suppose $i \in K_0$, and $j \in L \cap V_i$. By (i), for every point $s \in V_i$, $y(s)$ is expressed by $y(j)$, considering the simple path between s and j. Next suppose $k \in K_0$, and suppose for every point $s \in V_k$, $y(s)$ is expressed by $y(r)$, $r \in L \cap V_k$. In particular, $y(i)$ is expressed by $y(j)$. $y(k)$ is expressed by $y(r)$. By (ii) we have a relation between $y(i)$ and $y(k)$. Consequently we have a relation between $y(j)$ and $y(r)$. So, for every

point $s \in D$, $y(s)$ is expressed by $y(j)$. Then by the equation : $y(D) = v(D)$, $y(j)$ is determined uniquely. Hence we must have $y = x$. □

Applying Theorem 3.4 we can calculate the kernel inductively.

4. A Numerical Example

The next example illustrates the calculation of the kernel point by applying previous results.

Example 4.1. Let $V = \{0,1,...,9\}$ and
$E = \{(0,1),(1,2),(1,3),(3,4),(3,5),(0,6),(6,7),(6,8),(6,9)\}$.

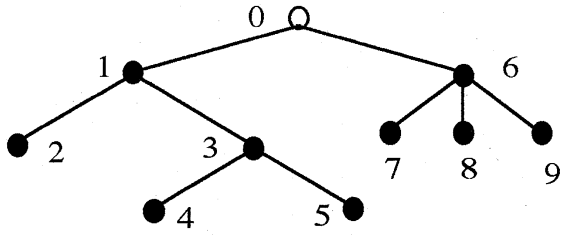

Figure 1.

By Theorem 3.4, we have $x(1) = x(2), x(3) = x(4) = x(5)$ and $x(6) = x(7) = x(8) = x(9)$. Furthermore, $x(1) = \max\{x(3) - d(3), x(345) - v(345)\}$, and
$\max\{x(12345) - v(12345), x(1) - d(1)\} = \max\{x(6789) - v(6789), x(6) - d(6)\}$.
Suppose $d(j) = 1$ for all $j \in D$, then we have

$$x(1) = 3x(3) - 4 - c,\ x(3) = \frac{14}{9} + \frac{17c}{36},\ x(6) = \frac{3}{4} + \frac{7c}{16} \text{ if } c \le \frac{4}{3},$$

$$x(1) = 3x(3) - 4 - c,\ x(3) = \frac{23}{15} + \frac{22c}{45},\ x(6) = \frac{4}{5} + \frac{2c}{5} \text{ if } \frac{4}{3} \le c \le 3,$$

$$x(1) = x(3) - 1,\ x(3) = x(6) + 1,\ x(6) = \frac{2}{3} + \frac{4c}{9} \text{ if } c \ge 3.$$

If c is large enough then $x(i) \approx \frac{4c}{9}$ for all i.

5. Outline of the proof of Theorem 2.1.

For $j \in D$, we write as

$$w(j) \equiv 2d(j) + c. \tag{5.1}$$

$w(j)$ is the cost of the return trip which starts at $a(j)$ and examines j. For $i \in V$ and $Y \subseteq D_i$, we let $w(i;Y) \equiv |Y|c + 2d(D_{(i;Y)})$. This is the cost of the most efficient return trip which starts at i and examines all vertices in Y.

First we define a strategy for **H** and analyse its properties in the game $G \equiv (f;P,Q)$. Define $p^* \in P$ inductively as follows: For $j \in D \setminus K_0$, let

$$\frac{p^*(a(j))}{c} = \frac{p^*(V_j)}{w(V_j)} \tag{5.2}$$

For $j,k \in K_0$, let

$$\frac{p^*(V_j)}{w(V_j)} = \frac{p^*(V_k)}{w(V_k)}. \tag{5.3}$$

Suppose **S** is at i. The left hand side of (5.2) is the probability per unit cost when he examines i, while the right hand side is the probability per unit cost when he returns to i after examining all vertices in V_j. By (2.3), (5.2) and (5.3), $p^* \in P$ is defined completely and uniquely. The following proposition gives basic properties of p^*.

Lemma 5.1. (i) For $i \in D$, $\frac{p^*(i)}{c} = \frac{p^*(D_i)}{w(D_i)}$.

(ii) For $i \in K_0$, $p^*(V_i) = \frac{w(V)}{w(D)}$.

For any $\sigma \in \Sigma$, σ^* is defined to be a mixed strategy such that **S** chooses σ and its reverse with probability 1/2 respectively.

Lemma 5.2. Suupose $\sigma \in \Sigma$. Then $f(i;\sigma) = C(0;D)$ for all $i \in D$.

Lemma 5.3. For any $\sigma \in \Sigma$, $f(p^*,\sigma) = C(0;D)$.

From Lemmas 5.2 and 5.3 we see in the game G, p^* and σ^* are optimal strategies for **H** and **S** respectively. The value of the game is $C(0;D)$. For the game $G_{(i;Y)}$ ($i \in V$ and $Y \subseteq D_i$), we can give a similar argument as in the case of the game G because of the inductive structure of the tree and in ductive definition of p^*. Arguments in detail is omitted here. This completes the proof of the theorem.

References

[1]Ahlswede, R. and Wegener, I.: *Search Problems.* John Wiley & Sons, Chichester, 1987. esp. pp. 264-265.

[2] Davis,M. and Maschler,M.:The Kernel of a Cooperative Game. *Naval Res. Logist. Quart.* **12**(1965), 223-259.

[3] Gal, S.: *Search Games.* Math. in Sci. and Eng., **149**, Academic Press. 1980. esp. pp. 17-33.

[4] Granot,D.and Granot,F.:On Some Network Flow Games. *Mathematics of Operations Research* **17**(1992), 792-841.

[5]Kikuta,K.: A Search Game with Traveling Cost on a Tree. *Journal of the Operations Research Society of Japan* **38** (1995). 70-88.

[6] -----: A Cost Allocation Game Related to a Search Problem. mimeo. 1996.

[7]Maschler,M.: The Bargaining Set, Kernel, and Nucleolus. Ch.18 of *Handbook of Game Theory*, Vol.1, Edited by R.J.Aumann and S.Hart. Elsevier Science Publishers 1992.

[8]Maschler,M. Peleg,B. and Shapley, L.S.:The Kernel and Bargaining Set for Convex Games. *International Journal of Game Theory* **1**(1972) 73-93.
[9] Megiddo,N.:Computational Complexity of the Game Theory Approach to Cost Allocation for a Tree. *Mathematics of Operations Research* **3**(1978), 189-196.
[10] Nakai, T.: *Tansaku Riron Tenbo* (In Japanese) mimeo. 1986.
[11] Ruckle, W.H.: *Geometric Games and Their Applications.* Pitman. Mass. 1983. esp. pp. 148-155.
[12] Shapley,L.S.:Cores of Convex Games. *International Journal of Game Theory* **1**(1971) 11-26.