경제학석사학위논문

Strong Demand Operator and the Dutta-Kar Rule for Minimum Cost Spanning Tree Problems

2013년 2월

서울대학교 대학원
경제학부 경제학 전공
한 창 용

Strong Demand Operator and the Dutta-Kar Rule for Minimum Cost Spanning Tree Problems

지도교수 전 영 섭

이 논문을 경제학석사 학위논문으로 제출함.

2012년 10월

서울대학교 대학원

경제학부 경제학 전공

한 창 용

한창용의 경제학석사 학위논문을 인준함.

2012년 12월

위 원 장 ______ 주 병 기 ______(인)

부위원장 ______ 전 영 섭 ______(인)

위 원 ______ 김 진 우 ______(인)

Abstract

Strong Demand Operator and the Dutta-Kar Rule for Minimum Cost Spanning Tree Problems

We study the *strong demand operator* introduced by Granot and Huberman (1984) for *minimum cost spanning tree problems*. First, we review the strong demand operator. Next, we study the irreducible minimum cost spanning tree games and the irreducible core. Finally, we define a procedure with tie-breaking rule which generates an allocation from given initial allocation. In our procedure, a cost matrix is changed to its irreducible matrix before the operator is applied. We show that the Dutta-Kar allocation is obtained by applying the *strong demand operator* from any allocation in irreducible core.

Keywords: minimum cost spanning tree problems, strong demand operator, irreducible matrix, irreducible core, Dutta-Kar rule, Prim algorithm.

Student Number: 2011-20197

Contents

1 Introduction **1**

2 Preliminaries **2**
2.1 Minimum cost spanning tree problem 2
2.2 Prim algorithm 3
2.3 Rules 4

3 Strong demand operator **5**

4 Irreducible minimum cost spanning tree problem **7**
4.1 Irreducible matrix 7
4.2 Partition by irreducible matrix 7

5 Main results **10**
5.1 Procedure with tie-breaking rule 10
5.2 Separability 11
5.3 Coincidence 12

6 Concluding remark **17**

1 Introduction

In this paper, we study *minimum cost spanning tree problems* (*mcstp*). Consider the situation there is a common supplier (denoted by 0) which can supply anyone who is connected to it. Demanding agents (denoted by $\mathcal{N}$) are located at distinct geographical places. Each agents will be served through connections which entail some cost. They do not bother whether they are connected directly or indirectly with the source. They are going to cooperate to minimize their total cost to construct a network which connects every agent to the source.

In this situation, two big questions arise. First, how can we construct the efficient network? And how to allocate the total cost? For the first question, Prim (1957) introduced prominent algorithms. For the second question, there are many studies about *mcstp* after Claus and Kleitman (1973) initiated the problem and Bird (1976) adopted cooperative game theoretic approach (Claus and Kleitman, 1973; Bird, 1976; Feltkamp et al., 1994; Kar, 2002; Dutta and Kar, 2004; Bergantinos and Vidal-Puga, 2007; Chun and Lee, 2012).

We investigate the allocation problem using strong demand operator introduced by Granot and Huberman (1984). First, we review the strong demand operator. Next, we study the *irreducible minimum cost spanning tree problem* and the *irreducible core*. Finally, we define a procedure with tie-breaking rule which generates an allocation from given initial allocation. In our procedure, a cost matrix is changed to its irreducible matrix before the operator is applied. We show that the Dutta-Kar allocation is obtained by applying the *strong demand operator* from any allocation in irreducible core.

Bird (1976) considers the *irreducible matrix* and the irreducible core. The irreducible matrix is the minimal cost matrix without reducing the total cost of the network. To understand this idea, suppose that a physical network is given. In the network formation model, costs of all possible link play important role. But in real world, a link cost of outside-network may have no meaning after a network is constructed. Thus one may insist that an allocation of a minimum cost spanning tree should only depend on the costs on the minimum cost spanning tree and that is the Fair allocation rule (Bergantinos and Vidal-Puga (2007)) which is the Shapley value based on irreducible matrix.

The *irreducible core* is a set of cost allocations such that each coalition has non-negative excess in terms of irreducible matrix. Irreducible core is convex hull of some extreme points. Aarts and Driessen (1993) and Tijs et al. (2004) show how to determine these extreme points.

Bird (1976) introduced an allocation rule for *minimum cost spanning tree* *problems*, now called the Bird rule. With the Bird rule, each agent connects sequentially to the source by using the Prim algorithm and pays the additional cost. We call a rule a *core selection* if no coalition of agents can be better off by building their own network. The Bird rule is a *core selection* but fails to satisfy *cost monotonicity*. *Cost monotonicity* requires that the cost allocated to agent i does not increase if the cost of a link involving i goes down, nothing else changing. Dutta and Kar (2004) suggested DK rule which is a core selection and also satisfies cost monotonicity.

This paper proceeds as follows. In section 2, we introduce minimum cost spanning tree problems, allocation rules for the problems. In section 3, we review the *strong demand operator*. In section 4, we study *irreducible minimum cost spanning tree problem*. We introduce *partition by irreducible matrix* in order to study the irreducible core. In section 5, we suggest iteration of the strong demand operator. We show the coincidence between the DK allocation and the iteration the strong demand operator from any allocation in the irreducible core.

2 Preliminaries

2.1 Minimum cost spanning tree problem

Let $\mathbb{N} = \{1, 2, \cdots\}$ be a (finite or infinite) universe of all potential agents and $\mathcal{N}$ be the collection of non-empty, finite subsets of $\mathbb{N}$. A typical element of $\mathcal{N}$ is denoted by $N \equiv \{1, ..., n\}$ and 0 is a special node called the *source*. We call each element of $N_0 \equiv N \cup \{0\}$ a node, and $\mathcal{N}_0 \equiv \{N_0 | N \in \mathcal{N}\}$.

Given $N_0 \in \mathcal{N}_0$, a *cost matrix* $C = (c_{ij})_{i,j \in N_0}$ represents the cost of direct links between any pair of nodes. For all $i, j \in N_0$, we assume that $c_{ij} \geq 0$ if $i \neq j$ and $c_{ij} = 0$ if $i = j$. Also, we assume that for all $i, j \in N_0$, $c_{ij} = c_{ji}$. The set of all cost matrices for N_0 is denoted by $\mathcal{C}_{N_0}$ and $\mathcal{C} \equiv \cup_{N_0 \in \mathcal{N}_0} \mathcal{C}_{N_0}$.

A *minimum cost spanning tree problem (mcstp)* is a pair (N_0, C) where $N \in \mathcal{N}$ is a finite set of agents, 0 is the source, and $C \in \mathcal{C}_{N_0}$ is the cost matrix.

A *network* g over N_0 is a subset of a complete graph $\mathcal{G}_{N_0} \equiv \{(i, j) | \forall i, j \in N_0, i \neq j\}$, whose element is an *arc*. Given a cost matrix C, we define the *cost* associated with g as $c(C, g) \equiv \sum_{(i,j) \in g} c_{ij}$.

Given a network g over N_0 and $i, j \in N_0$ such that $i \neq j$, a *path* from i to j in g is a sequence of different arcs $\{(i_{k-1}, i_k)\}_{k=1}^{K}$ that satisfies $(i_{k-1}, i_k) \in g$ for all $k \in \{1, 2, \cdots, K\}$, $i = i_0$ and $j = i_K$. Two distinct nodes i and $j \in N_0$ are *connected in* g if there exists a path from i to j. A network g over N_0 is *connected* if for all $i, j \in N_0$, i and j are connected in g.

A *tree* is a network with a unique path from any node to another node. We denote the set of all trees over N_0 as $\mathcal{T}_{N_0}$ and $\mathcal{T} \equiv \cup_{N_0 \in \mathcal{N}_0} \mathcal{T}_{N_0}$. For any $t \in \mathcal{T}_{N_0}$, let t_{ij} be the unique path from i to j in t.

Given $t \in \mathcal{T}_{N_0}$ and $i, j \in N_0$, i is a *predecessor* of j in t if there exist $k \in N_0$ such that $(i, k) \in t_{0j}$. Let $P(j|t)$ be the set of all predecessors of j in t. Agent i is the *immediate predecessor* of j in t if $(i, j) \in t_{0j}$. Let $p(j|t)$ be the immediate predecessor of j in t. j is a *follower* of i in t if $i \in Pre(j|t)$. Let $F(i|t)$ be the set of all followers of i in t. Agent j is an *immediate follower* of i if $p(j|t) = i$. Let $f(i|t)$ be the set of all immediate followers of i in t.

For all $N_0 \in \mathcal{N}_0$ and all $C \in \mathcal{C}_{N_0}$, a *minimum cost spanning tree (mcst)* over N_0, denoted by t_{N_0}, is defined to be $\text{argmin}_{t \in \mathcal{T}_{N_0}} \sum_{(i,j) \in t} c_{ij}$.

Let $m(N_0, C)$ be the minimum cost for the *mcstp* (N_0, C). That is, $m(N_0, C) \equiv \sum_{(i,j) \in t} c_{ij}$, where t is an *mcst* for the *mcstp* (N_0, C).

Let $C_{|S_0}$ be the restriction of the cost matrix C to the coalition $S_0 \subseteq N_0$. Bird (1976) associated a cooperative game (N, c) with each *mcstp* (N_0, C) where $c(S)$ = $m(S_0, C_{|S_0})$ for each $S \subset N$.

When there is no ambiguity, we use $P(i)$, $p(i)$, $F(i)$, $f(i)$, t, and (S_0, C) instead of $P(i|t)$, $p(i|t)$, $F(i|t)$, $f(i|t)$, t_{N_0}, and $(S_0, C_{|S_0})$, respectively.

The *core* of the cooperative game (N, c) is defined by

$$Core(N, c) \equiv \Big\{ x \in \mathbb{R}^N \Big| \sum_{i \in N} x_i = c(N) \text{ and } \sum_{i \in S} x_i \leq c(S), \forall S \subset N \Big\}.$$

2.2 Prim algorithm

To find an *mcst* t, we can use the the Prim algorithm (Prim (1957)) defined as follows:

Step 0 : Let $A^0 \equiv \{0\}$ and $g^0 \equiv \emptyset$.
Step 1 : Choose an ordered pair (a^1, b^1) such that

$$(a^1, b^1) = \underset{(i,j) \in A^0 \times (A^0)^c}{\text{argmin}} c_{ij},$$

where $(A^0)^c \equiv N \setminus A^0$. Let $A^1 \equiv A^0 \cup \{b^1\}$ and $g^1 \equiv g^0 \cup \{(a^1, b^1)\}$.
Step k : Choose an ordered pair (a^k, b^k) such that

$$(a^k, b^k) = \underset{(i,j) \in A^{k-1} \times (A^{k-1})^c}{\text{argmin}} c_{ij}.$$

The algorithm terminates at step n. Then, the *mcst* t for the *mcstp* (N_0, C) is g^n.

From now on, for a given *mcstp*, we assume that all agents are named by the Prim algorithm, i.e., agent 1 is chosen in the first step of the Prim algorithm and agent i is chosen in the i^{th} step. In fact, the Prim algorithm does not choose a unique agent in each step generally. Later, we will control this problem by restricting the domain of the cost matrices.

2.3 Rules

For each $N \in \mathcal{N}$, a *cost allocation rule*, or a *rule*, is a function ψ such that $\psi : \mathcal{C}_{N_0} \to \mathbb{R}^N_+$. Let $\mathcal{R}$ be the family of all rules. Given $N_0 \in \mathcal{N}_0$, $C \in \mathcal{C}_{N_0}$, and $\psi \in \mathcal{R}$, the i^{th} element of $\psi(C)$, $\psi_i(C)$, is the *cost allocation* to agent i.

We introduce two rules for *mcstp*, the Bird rule and the Dutta-Kar rule (DK rule). Before we study these rules, we impose a domain restriction on the permissible cost matrices.

$$\mathcal{C}^3_{N_0} \equiv \{C \in \mathcal{C}_{N_0} | C \text{ induces a unique agent in each step of the Prim algorithm}\},$$
$$\mathcal{C}^3 \equiv \bigcup_{N_0 \in \mathcal{N}_0} \mathcal{C}^3_{N_0}.$$

An *mcst* t does not have to be unique on $\mathcal{C}^3$. For example, let $N = \{1, 2, 3\}$ and $c_{01} = 6, c_{12} = 2, c_{13} = c_{23} = 3$ and other costs be larger than 6. Note that *mcst* t is not unique but a unique agent can be chosen in each step of the Prim algorithm.

The Bird rule charges each agent the additional cost incurred by his inclusion in the network.

Bird rule, ψ^B : For all $N_0 \in \mathcal{N}_0$, all $C \in \mathcal{C}^3_{N_0}$, and all $i \in N$, $\psi^B_i(N_0, C) = c_{p(i)i}$.

With the Bird rule, each agent connects sequentially to the source by using the Prim algorithm and pays the additional cost. We call a rule a *core selection* if no coalition of agents can be better off by building their own network. The Bird rule is a *core selection* but fails to satisfy *cost monotonicity*. *Cost monotonicity* requires that the cost allocated to agent i does not increase if the cost of a link involving i goes down, nothing else changing.

Dutta and Kar (2004) proposed the DK rule which is a *core selection* and also satisfies *cost monotonicity*. Dutta and Kar (2004) defined the DK rule, ψ^{DK}, by using the following algorithm.

Step 0 : Let $A^0 \equiv \{0\}, g^0 \equiv \emptyset, t^0 \equiv 0$.
Step 1 : Choose the ordered pair (a^1, b^1) such that

$$(a^1, b^1) = \underset{(i,j)\in A^0 \times A^0_c}{\text{argmin}} \; c_{ij}$$

Define

$$t^1 \equiv \max(t^0, c_{a^1b^1}), \; A^1 \equiv A^0 \cup \{b^1\}, \; g^1 \equiv g^0 \cup \{(a^1, b^1)\}.$$

Step k : Choose the ordered pair

$$(a^k, b^k) = \underset{(i,j)\in A^{k-1} \times A^{k-1}_c}{\text{argmin}} \; c_{ij}$$

Define

$$t^k \equiv \max(t^{k-1}, c_{a^kb^k}), \; A^k \equiv A^{k-1}, \cup\{b^k\}, g^k \equiv g^{k-1} \cup \{(a^k, b^k)\}$$
$$\psi^{DK}_{k-1} \equiv \min(t^{k-1}, c_{a^kb^k}).$$

The algorithm terminates at step n. Then, $\psi^{DK}_n \equiv t^n$.

If $C \notin \mathcal{C}^3$, the DK rule considers strict orderings over N which can be used as a tie-breaking rule. The DK rule takes the simple average of the cost allocation obtained for each ordering. But on the domain $\mathcal{C}^3$, we choose a unique agent in each step; therefore we do not need a tie-breaking rule.

Because we name the agent who is chosen in the i^{th} step i, the node b^i in each step of the Prim algorithm is the agent i, and the node a^i is the agent $p(i)$. Thus, $c_{a^ib^i} = c_{p(i)i}$ in each step. Therefore, on the domain $\mathcal{C}^3$, the DK rule can be rewritten as

$$\psi^{DK}_k = \min\{\max_{l\le k}\{c_{p(l)l}\}, c_{p(k+1)k+1}\}, \quad 1 \le k < n,$$

and ψ^{DK}_n is the remaining cost.

3 Strong demand operator

Granot and Huberman (1984) introduced *weak demand operator* (*wdo*) and *strong demand operator* (*sdo*). In this chapter, we deal with the *sdo*.[1]

[1] Kim (2011) deals with the weak demand operator case. He criticized the definition of the *wdo* and suggested the *modified weak demand operator*.

Let the *mcst* t be given. Let agent i be the immediate predecessor of agent j in t. Suppose they are assigned with the initial cost allocation y. Agent i wants to transfer some of his costs to j. In this case, how much can be transferred? If agent i transfers too much cost, agent j will disconnect the arc (i, j) and form his own tree. Therefore, agent i may transfer costs as long as it does not violate the participation constraints of agent j.

Before we formalize the *sdo*, we introduce a notation T_{R_1,R_2}. For $R_1, R_2 \subset N$, define a coalition set T_{R_1,R_2} as

$$T_{R_1,R_2} = \{S | R_1 \subseteq S, R_2 \cap S = \emptyset\}.$$

For convenience, if both R_1 and R_2 are singleton; say $R_1 = \{i\}$ and $R_2 = \{j\}$, we will use the notation $T_{i,j}$ for $T_{\{i\},\{j\}}$.

Strong Demand Operator (Granot and Huberman(1984)) : When agent $i \in N$ performs a *sdo* it gives a cost transfer to each j in his immediate follower set $f(i)$.

To find the optimal value of the cost transfer, agent i first solves the optimization problem,

$$\max\{\sum_{j \in f(i)} z_j\}$$

$$\begin{aligned}
\text{s.t. } & ex(R, z) \geq 0 \text{ for all } R \in T_{f(i),\{i\}} \cup (T_{S,f(i)\setminus S} : S \subset f(i)), \\
& z_k = y_k \text{ for all } k \notin \{i\} \cup f(i), \\
& \sum_{i \in N} z_i = \sum_{i \in N} y_i,
\end{aligned}$$

where $ex(R, z) \equiv c(R) - \sum_{i \in R} z_i$.

Then the *sdo* is defined by

$$sd_j^i(y) = \begin{cases} z_k & \text{if } j = k, k \in f(i); \\ y_i - \sum_{k \in f(i)} (z_k - y_k) & \text{if } j = i; \\ y_j & \text{otherwise.} \end{cases}$$

We call the first constraints of the optimization problem *participation constraints*. Note that if y is in the core then the constraints of the strong demand operator ensure that all cost allocations $x, x \in \{sd^i(y)\}$, are also contained in the core.

The *sdo* draws the maximal amount of transfers that agent i can afford as long as the remains in the problem.

4 Irreducible minimum cost spanning tree problem

4.1 Irreducible matrix

The *irreducible matrix* is the minimal cost matrix obtainable without reducing the total cost of the network. Bird (1976) introduced the minimal spanning network (N_0, C^*) associated with t as follows. For given *mcstp* (N_0, C) and its *mcst* t, $c^*_{ij} \equiv \max_{(k,l)\in t_{ij}}\{c_{kl}\}$. Bird (1976) used this minimal network to define the irreducible core of an *mcstp*, which is a subset of the core.

The characteristic function and the core of the irreducible games are similar to those of the *mcstp*. That is, $m(N_0, C^*) = \sum_{(i,j)\in t} c^*_{ij}$, where c^*_{ij} is each element of irreducible matrix C^* and t is a *mcst* of the *mcstp* (N_0, C^*). Let $C^*_{|S_0}$ be the restriction of the irreducible matrix C^* to the coalition $S_0 \subseteq N_0$. We use $c^*(S)$ as the characteristic function where $c^*(S) \equiv m(S_0, C^*_{|S_0})$.

When there is no ambiguity, we use (S_0, C^*) instead of $(S_0, C^*_{|S_0})$.

We denote the irreducible core of *mcstp* (N_0, C) as $Core(N, c^*)$ which is defined by

$$Core(N, c^*) = \left\{ x \in \mathbb{R}^N \middle| \sum_{i\in N} x_i = c^*(N) \text{ and } \sum_{i\in S} x_i \le c^*(S), \forall S \subset N \right\}.$$

4.2 Partition by irreducible matrix

We introduce a *partition by irreducible matrix (PIM)*, which is a partition of N. We call each element of the *PIM* a $cell$. To define the cells of the *PIM*, we define a *cell leader* and *cell followers*.

Given an irreducible matrix C^* of cost matrix C, we call an agent $l \in N$ a *cell leader* (CL) of C^* if $c^*_{0l} = c_{p(l)l}$. Given a CL l, we denote the set of all *cell followers* of l as $CF(l)$ which is defined by

$$CF(l) \equiv \{j \in N | c^*_{lj} < c^*_{0j},\ c^*_{0j} = c^*_{0l}\}.$$

Partition by Irreducible Matrix (PIM) : Given an irreducible matrix C^* of cost matrix C, the *PIM* of C^* is a partition of N, which is defined by

$$PIM(C^*) \equiv \left\{ \{l\} \cup CF(l) \middle| \, l \text{ is an } CL \text{ of } C^* \right\}.$$

From now on, we denote the *cell* which agent i belongs to as $cell(i)$ and the CL of $cell(i)$ as $\alpha(cell(i))$.

Next, we introduce *induced tree* which is a transformation of a *mcst*.

Induced tree : Given a tree t, an induced tree t^i is a tree transformed from t in which:

(i) Each CL directly connects to the source,

(ii) CF of each CL conserve their own tree structure.

Note that any $cell$ in t is an $cell$ in t^i.

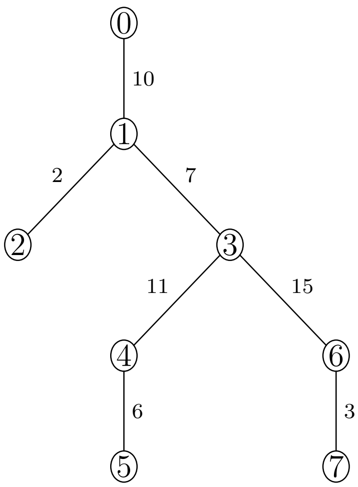


Figure 1.
Arbitrary t

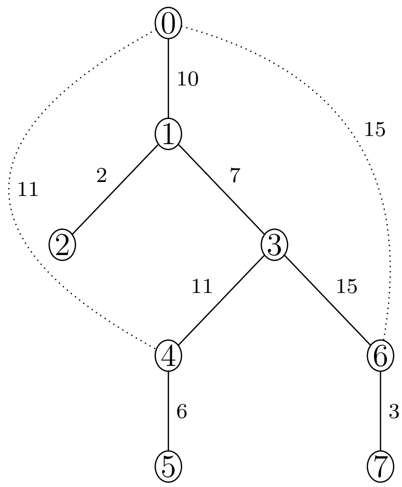


Figure 2. Transformation.

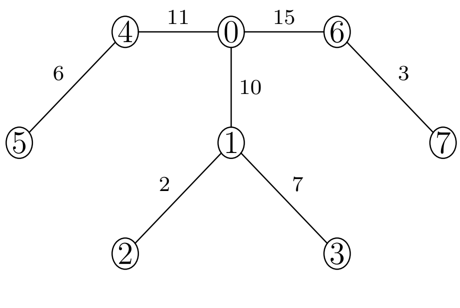


Figure 3. Induced tree t^i from t.

Lemma 1. *For all $N_0 \in \mathcal{N}_0$ and all $C \in \mathcal{C}^3_{N_0}$, the mcst t and its induced tree t^i has the same irreducible matrix.*

Proof. All we have to check is that for any agent i and j, they have the same irreducible cost c^*_{ij} in t and t^i.

• Case I. $(cell(i) = cell(j))$

Because each $IGFol$ conserves its structure in t^i, for any i and j in the same $cell$, t and t^i have the same unique paths from i to j. Therefore, from the definition of c^*_{ij}, $c^*_{ij}|_t = c^*_{ij}|_{t^i}$.

• Case II. $(cell(i) \neq cell(j))$

For any i and j with $cell(i) \neq cell(j)$, $c^*_{ij} = \max_{\{k,l\}\in t_{ij}} c_{kl}$ $= \max\{c_{0\alpha(IG(i))}, c_{0\alpha(IG(j))}\}$. We know that any IG in t is also IG in t^i. Therefore, $c^*_{ij}|_t = c^*_{ij}|_{t^i}$.

We can conclude that t and its induced tree t^i have the same irreducible matrix. Therefore, t^i is a *mcst* t for the irreducible form. □

Check that the induced tree is the *mcst* for irreducible game (N_0, C^*) with the maximum components, where the number of components of a tree is $|F(0)|$, and each component is a $cell$ of the PIM. From the definition of induced tree, it is easy to see that the partition $\{cell_1, \cdots, cell_p\}$ of N satisfies

$$m(N_0, C^*) = \sum_{i=1}^{p} m((cell_i)_0, C^*),$$

where p is the number of $cell$s of the $PIM(C^*)$.

Cell-wise efficiency : For all $N_0 \in \mathcal{N}_0$, all $C \in \mathcal{C}_{N_0}$ and all $y \in \mathcal{Y}$, an allocation y is *cell-wise efficient* if

$$\sum_{j\in cell_i} y_j = \sum_{j\in cell_i} c_{P(j)j}, \qquad \text{for all } 1 \leq i \leq p,$$

where p is the number of $cell$s of the $PIM(C^*)$.

Lemma 2. *If a cost allocation y is in $Core(N, c^*)$, it satisfies cell-wise efficiency.*

Proof. From the definition of induced tree and Lemma 1, $\sum_{j\in IG_i} c_{P(j)j}$ $= m((cell_i)_0, C^*) = c^*(cell_i)$ for all $1 \leq i \leq p$. Of course, $\sum_{j\in N} y_j = m(N_0, C^*)$

First, suppose that there exists an $cell$ such that $\sum_{j\in cell_i} y_j > \sum_{j\in cell_i} c_{P(j)j}$. It means that $\sum_{j\in cell_i} y_j > c^*(cell_i)$ thus violates the core constraint. Therefore, y is not in $Core(N, c^*)$.

Second, suppose that there exists an $cell$ such that $\sum_{j\in cell_i} y_j < \sum_{j\in cell_i} c_{P(j)j}$. It means that $\sum_{j\in IG_i} y_j < c^*(cell_i)$. In this case, since $\sum_{j\in N} y_j = m(N_0, C^*)$ and $m(N_0, C^*) = \sum_{i=1}^{p} m((cell_i)_0, C^*)$, there should be at least one $cell$ such that $\sum_{j\in cell_k} y_j > \sum_{j\in cell_k} c_{P(j)j}$. Therefore, y is not in $Core(N, c^*)$ for the same reason.

Consequently, any cost allocation y in $Core(N, c^*)$ satisfies cell-wise efficiency. □

We define component-wise efficiency, which will be used in our main results.

Component-wise efficiency For all $N_0 \in \mathcal{N}_0$, all $C \in \mathcal{C}_{N_0}$, all $i \in F(0)$, and all $y \in \mathcal{Y}$, an allocation y is component-wise efficient if

$$\sum_{j\in F(i)\cup\{i\}} y_j = \sum_{j\in F(i)\cup\{i\}} c_{P(j)j}.$$

Cell-wise efficiency implies *component-wise efficiency* from the definition of induced tree.

5 Main results

In this section, we define a procedure using the *sdo* and show that the outcome coincides with the DK rule.

5.1 Procedure with tie-breaking rule

In our procedure, we use irreducible matrix C^* instead of cost matrix C. This is based on two reasons. First, as mentioned, a link cost of outside-tree may have no meaning after a network is constructed. Second, if we use C, the *sdo* can generate a negative allocation for the operator. With these reasons, we use irreducible matrix C^* instead of cost matrix C.

Next we assume that the operator should leave the game after the *sdo* is applied to him so that the others cannot use that node. When the operator leaves the game, we need alternative tree for the remaining people. In this case, we need tie-breaking rule ρ because we do not have unique alternative tree since we use the irreducible matrix. We define the tie-breaking rule ρ as follows.

Tie-breaking rule, ρ **:** For all $N_0 \in \mathcal{N}_0$, all $C \in \mathcal{C}^3_{N_0}$, and all $i \in N$, choose the *alternative mcst* (t^a) for $(N_0 \setminus \{i\}, C^*)$ as follows.

- Connect $r \in F(i) \setminus \{i+1\}$ to agent $i+1$. If agent $i+1$ is not connected to the source, connect to the source.

Now we are ready to define a procedure using the strong demand operator. The procedure is defined as below.

Iteration of the strong demand operator on the irreducible matrix
Given $N_0 \in \mathcal{N}_0$ and $C \in \mathcal{C}^3_{N_0}$,

(i) Let y be the initial cost allocation.

(ii) Transform the cost matrix into irreducible matrix.

(iii) Apply the *sdo* with the tie-breaking rule ρ to each agent sequentially following the numbering of agents.

5.2 Separability

Separability : For all $N_0 \in \mathcal{N}_0$, all $C \in \mathcal{C}_{N_0}$, and all $S \subset N$ satisfying $m(N_0, C) = m(S_0, C) + m((N \setminus S)_0, C)$, a rule is *separable* if

$$\psi_i(N_0, C) = \begin{cases} \psi_i(S_0, C) & \text{if } i \in S, \\ \psi_i((N \backslash S)_0, C) & \text{if } i \notin S. \end{cases}$$

We show that DK-rule is separable.

Lemma 3. *For all $N_0 \in \mathcal{N}_0$ and all $C \in \mathcal{C}^3_{N_0}$, $\psi^{DK}(\cdot)$ is separable.*

Proof. Let $N_0 \in \mathcal{N}_0$, $C \in \mathcal{C}^3_{N_0}$. Let $|f(0)| = m$ and $G = \{G^1_1, ..., G^1_{q_1}, G^2_1, ..., G^2_{q_2}, G^m_1, ..., G^f_{q_f}\}$ be the partition of N where each G^i_j is an $cell$ and for all $i \in \{1, \cdots, m\}$, G^i_1 is directly connected to the source. Let $c_{0G^i_j} \equiv c_{0\alpha(G^i_j)}$.
By the definition of Dutta-Kar allocation rule, for all $i \in \{1, \cdots, m\}$ and all $l-1 \in N$,

$$\psi^{DK}_{l-1}(N_0, C) = \min\{\max_{k \leq l-1}\{c_{p(k)k}\}, c_{p(l)l}\}.$$

Since each G^i_j is an $cell$, for all G^i_1,

$$\psi^{DK}_{l-1}(N_0, C) = \min\{c_{0G^i_j}, c_{p(l)l}\}.$$

Thus,

$$\psi^{DK}_{l-1}(N_0, C) = \begin{cases} c_{p(l)l} & \text{if } l \in G^i_1 \\ c_{0G^i_j} & \text{if } l \notin G^i_1. \end{cases}$$

Let $c_{p(G^i_j)G^i_j} \equiv c_{p(\alpha(G^i_j))\alpha(G^i_j)}$. Similarly, for all $i \in \{1, \cdots, m\}$, all $j \in \{1, \cdots, q_i\}$, and all $l-1 \in G^i_j$,

$$\psi^{DK}_{l-1}(N_0, C) = \begin{cases} c_{p(l)l} & \text{if } l \in G^i_j \\ c_{p(G^i_j)G^i_j} & \text{if } l \notin G^i_j. \end{cases}$$

Thus, for all $i \in \{1, \cdots, m\}$ and all $l-1 \in S = \cup_{j \in \{1, \cdots, q_i\}} G^i_j$, $\psi^{DK}_{l-1}(N_0, C)$ assigns some internal link cost regardless of $N \backslash S$ which implies $\psi^{DK}_{l-1}(N_0, C) = \psi^{DK}_{l-1}(S_0, C)$. □

5.3 Coincidence

If an allocation y is contained in irreducible core, it satisfies cell-wise efficiency by Lemma 2. From the definition of induced tree, we know that cell-wise efficiency implies component-wise efficiency. Therefore, if an cost allocation y is in irreducible core, it satisfies component-wise efficiency.

Since the *sdo* is a transfer between an operator and his immediate followers, it is a operation within a component. Therefore, if y satisfies component-wise efficiency, $sd^i(y)$ also satisfies component-wise efficiency.

Lemma 4. *For all $N_0 \in \mathcal{N}_0$, all $C \in \mathcal{C}^3_{N_0}$, all $i \in N$, if y is component-wise efficient, then $sd^i(y)$ is component-wise efficient.*

Proof. It is easy to check from the definition of the strong demand operator. □

We check how much cost is allocated to agent 1, if the *sdo* is applied to agent 1.

Lemma 5. *For all $N_0 \in \mathcal{N}_0$, all $C \in \mathcal{C}^3_{N_0}$, if the cost allocation y is contained in $Core(N, c^*)$ and we use the irreducible matrix, then $sd^1_1(y) = \min\{c_{01}, c_{12}\}$.*

Proof. Let y^0 be the initial allocation and y^1 be the allocation after the *sdo* is applied to agent 1. That is, $y^1 \equiv sd^1(y^0) = (sd^1_1(y^0), sd^1_2(y^0), \cdots, sd^1_n(y^0))$. Since y^0 is contained in the irreducible core, we know that y^1 is also contained in the irreducible core from the definition of the *sdo*.

Check that $c^*(N \setminus \{1\}) = c^*(N) - \min\{c_{01}, c_{12}\}$.

First, suppose that $y_1^1 < \min\{c_{01}, c_{12}\}$.

$$
\begin{aligned}
ex(N \setminus \{1\}, y^1) &= c^*(N \setminus \{1\}) - \sum_{i \ inN\setminus\{1\}} y_i^1 \\
&= \big(c^*(N) - \min\{c_{01}, c_{12}\}\big) - (\sum_{i\in N} y_i^1 - y_1^1) \\
&= \big(c^*(N) - \min\{c_{01}, c_{12}\}\big) - (\sum_{i\in N} y_i^0 - y_1^1) \\
&= y_1^1 - \min\{c_{01}, c_{12}\} \\
&< 0 \quad (\because y_1^1 < \min\{c_{01}, c_{12}\} \text{ by assumption}).
\end{aligned}
$$

From the definition of the *sdo*, y^1 is a core allocation, therefore every coalition has non-negative excess under y^1. Contradiction.

Second, suppose that $y_1^1 > \min\{c_{01}, c_{12}\}$.

Let $d \equiv y_1^1 - \min\{c_{01}, c_{12}\} > 0$. Check that for all $S \subseteq N$, $ex(S, y^1) \geq 0$ holds since y^1 is a core allocation from the definition of the *sdo*.

We consider a new allocation $\hat{y}$ such that agent 1 transfers d to agent 2 from y^1. We divide $R \subset N$ into four cases, $R \cap \{1,2\} = \emptyset$, $R \cap \{1,2\} = \{1\}$, $R \cap \{1,2\} = \{2\}$ and $R \cap \{1,2\} = \{1,2\}$.

(i) For every coalition R such that $R \cap \{1,2\} = \emptyset$, $ex(R, \hat{y}) \geq 0$ since $\forall i \in N \setminus \{1,2\},\ \hat{y}_i = y_i^1$.

(ii) For every coalition R such that $R \cap \{1,2\} = \{1\}$, $ex(R, \hat{y}) \geq 0$ since $\forall i \in N \setminus \{1,2\}, \hat{y}_i = y_i^1$ and $\hat{y}_1 < y_1^1$.

(iii) For every coalition R such that $R \cap \{1,2\} = \{1,2\}$, $ex(R, \hat{y}) \geq 0$ since $\sum_{i\in R} \hat{y}_i = \sum_{i\in R} y_i^1$.

(iv) For every coalition R such that $R \cap \{1,2\} = \{2\}$, let $R^{+1} \equiv R \cup \{1\}$. We know that $ex(R^{+1}, y^1) \geq 0$ since $y^1 = sd^1(y^0)$ is a core allocation. Check that $c^*(R) = c^*(R^{+1}) - \min\{c_{01}, c_{12}\}$.

$$
\begin{aligned}
ex(R, \hat{y}) &= c^*(R) - \sum_{i \in R} \hat{y}_i \\
&= c^*(R^{+1}) - \min\{c_{01}, c_{12}\} - \{\sum_{i \in R^{+1}} y_i^1 - \hat{y}_1\} \\
&= c^*(R^{+1}) - \sum_{i \in R^{+1}} y_i^1 - \min\{c_{01}, c_{12}\} + \hat{y}_1 \\
&\geq -\min\{c_{01}, c_{12}\} + \hat{y}_1 \quad (\because ex(R^{+1}, y^1) \geq 0) \\
&= -\min\{c_{01}, c_{12}\} + y_1^1 - d \\
&= -\min\{c_{01}, c_{12}\} + \min\{c_{01}, c_{12}\} = 0.
\end{aligned}
$$

Therefore agent 1 can transfer d to agent 2 without violating any participation constraint by checking $(i), (ii), (iii)$, and (iv). It means that y_1^1 is not an solution for the optimization problem of the *sdo*. Contradiction.

To sum up, $y_1^1 = sd_1^1(y^0) > \min\{c_{01}, c_{12}\}$ cannot happen, and $y_1^1 = sd_1^1(y^0) < \min\{c_{01}, c_{12}\}$ also cannot happen, as desired. □

Lemma 5 means that $sd_1^1(y)$ is always unique, whereas $sd^1(y)$ may be a set of allocations, as mentioned.

Since the other agents cannot use the node 1 after the *sdo* is applied to agent 1, we consider the *alternative mcstp* related with the alternative tree t^a which is made according to tie-breaking rule ρ after the *sdo* is applied to agent 1.

We denote the irreducible *mcstp* using the alternative tree t^a as $(N_0 \setminus \{i\}, C^{**})$.

Lemma 6. *For any $mcstp$ (N_0, C) and its irreducible $mcstp$ (N_0, C^*), $(N_0 \setminus \{1\}, C^*)$ and $(N_0 \setminus \{1\}, C^{**})$ are the same $mcstps$.*

Proof. We will show that C^* restricted on $N_0 \setminus \{1\}$ is equal to C^{**}.

• Case I. $f(1) = \{2\}$.

First, we check costs between agents. Since tie-breaking rule ρ requires that $N \setminus (\{1\} \cup f(1))$ keep their tree structure, for all $i, j \in N \setminus \{1\}, i \neq j$, $t_{ij} = t_{ij}^a$. Therefore for all $i, j \in N \setminus \{1\}, i \neq j$, $c_{ij}^* = c_{ij}^{**}$ from the definition of the irreducible matrix.

Next, we check costs between agents and the source. In case I, we know that $c_{02}^* = \max\{c_{01}, c_{12}\} = c_{02}^{**}$. Since $c_{0i}^* = \max_{(j,k) \in t_{0i}} c_{jk}$, $\max\{c_{01}, c_{12}, \cdots, c_{p(i)i}\} = \max\{\max\{c_{01}, c_{12}\}, \cdots, c_{p(i)i}\} = \max\{c_{02}^{**}, \cdots, c_{p(i)i}\} = c_{0i}^{**}$. Therefore for all $i \in N \setminus \{1\}$, $c_{0i}^* = c_{0i}^{**}$ from the definition of the irreducible matrix.

Therefore, C^* restricted on $N_0 \setminus \{1\}$ is equal to C^{**} in Case I.

• Case II. $\{2\} \subsetneq f(1)$.

Before we prove, we first need to show that for any $i \in f(1) \setminus \{2\}$, $c^*_{1i} = c^*_{2i} = c^{**}_{2i}$. Since Prim algorithm chooses i later than 2, we know that $c_{12} < c_{1i}$. And from the definition of irreducible matrix, $c^*_{2i} = \max\{c_{12}, c_{1i}\} = c_{1i} = c^*_{1i}$.

Tie-breaking rule ρ requires us that $c^{**}_{2i} = c^*_{2i}$. Therefore, $c^*_{1i} = c^*_{2i} = c^{**}_{2i}$.

First, we check costs between agents, that is, we want to show that for any $i, j \in N \setminus \{1\}$ and $i \neq j$, $c^*_{ij} = c^{**}_{ij}$

If agent 1 is not on the unique path t_{ij}, $c^*_{ij} = c^{**}_{ij}$, because tie-breaking rule ρ requires that $N \setminus (\{1\} \cup F(1))$ keep their tree structure.

If agent 1 is on the unique path t_{ij}, we define agent k, l as $k \in f(1)$, $i \in F(k)$ and $l \in f(1)$, $j \in F(l)$.[2] The only difference between t_{ij} and t^a_{ij} is that $\{(k,1),(1,l)\} \subset t_{ij}$ whereas $\{(k,2),(2,l)\} \subset t^a_{ij}$.[3] We already showed $c^*_{1k} = c^*_{2k}$ and $c^*_{1l} = c^*_{2l}$ which implies the costs on the path t_{ij} is equal to t^a_{ij} .[4] Therefore $c^*_{ij} = c^{**}_{ij}$ for any $i, j \in N \setminus \{1\}$ and $i \neq j$.

Next, we check costs between agents and source. Let $i \in N \setminus \{1\}$ and k be an agent such that $k \in f(1)$ and $i \in F(k)$. [5]

If $k = 2$ (that is, $i \in F(2)$), the tie-breaking rule ρ requires agent 2 connects to the source. In this case, the only difference between t_{0i} and t^a_{0i} is that $\{(0,1),(1,2)\} \subset t_{ij}$ whereas $\{(0,2)\} \subset t^a_{ij}$. We can check that $c^{**}_{02} = \max\{c_{01}, c_{02}\}$ $= c^*_{02}$. Therefore, $c^*_{0i} = c^{**}_{0i}$ if 2 is on t_{0i}.

If $k \neq 2$ (that is, $i \notin F(2)$), we have to check the unique paths t_{0i} and t^a_{0i}. The only difference between t_{0i} and t^a_{0i} is that $\{(0,1),(1,k)\} \subset t_{0i}$ whereas $\{(0,2),(2,k)\} \subset t^a_{ij}$. We already know that $c^*_{1k} = c^*_{2k} = c^{**}_{2k}$. We also know that $c^{**}_{02} = \max\{c^*_{01}, c^*_{12}\}$. Therefore, $\max\{c^*_{01}, c^*_{1k}\} = \max\{c^{**}_{02}, c^{**}_{0k}\}$. Thus, the maximum costs on the path t_{0i} and t^a_{0i} are the same. It implies that $c^*_{0i} = c^{**}_{0i}$ if 2 is not on the path t_{oi}.

---

[2]We define $k = i$ if $i \in F(1)$ and we define $l = j$ if $j \in F(1)$.

[3]If $k = 2$, $\{(k,1),(1,l)\} \subset t_{ij}$ whereas $\{(2,l)\} \subset t^a_{ij}$. It means that the length of the path changes, if we define the length of any path as the number of links on the path. We have the same result if $l = 2$.

[4]If $k = 2$, the $c_{k1} < c_{1l}$ for any l. It implies that $\max\{c_{k1}, c_{1l}\} = c_{1l} = c^*_{1l} = c^*_{2l}$. That is, even though the length of the path changes, we have the same maximum cost on the path t_{ij} and t^a_{ij}. We have the same result if $l = 2$.

[5]We define $k = i$ if $i \in f(1)$.

Therefore, C^* restricted on $N_0 \setminus \{1\}$ is equal to C^{**} in Case II.

We can conclude that $(N_0 \setminus \{1\}, C^*)$ and $(N_0 \setminus \{1\}, C^{**})$ are the same $mcstp$s. $\square$

The interpretation of Lemma 6 is that t^a is (one of) *mcst*s for $(N_0 \setminus \{1\}, C^*)$.

Theorem 1. *For all $N_0 \in \mathcal{N}_0$, and all $C \in \mathcal{C}^3_{N_0}$, if an initial allocation y is in $Core(N, c^*)$, the allocation obtained by iterating the sdo using the irreducible matrix from y coincides with the DK allocation.*

Proof. Let $N_0 \in N_0$, $C \in \mathcal{C}^3_{N_0}$ and y^0 be an allocation in $Core(N, c^*)$.

Because ψ^{DK} satisfies separability by Lemma 3, it suffices to consider the case that $|f(0)| = 1$. Since each agent is numbered by the Prim algorithm, $f(0) = \{1\}$.

At 1^{st} stage, agent 1 applies the *sdo* and gets $y^1_1 = \min\{c_{01}, c_{12}\}$ by Lemma 5. After that, $N \setminus \{1\}$ form an alternative tree according to the tie-breaking rule ρ such that the costs of the alternative trees are $c_{02} = \max\{c_{01}, c_{12}\}$ and for all $j \in F(1) \setminus \{2\}$, $c_{2j} = c_{P(j)j}$. From the definition of the *sdo*, we know that y^1 is in $Core(N, c^*)$. $y^1_{|N\setminus\{1\}}$ is the projection of y^1 onto $\mathbb{R}^{N\setminus\{1\}}$.

We need to show that $y^1_{|N\setminus\{1\}}$ is in $Core(N \setminus \{1\}, c^{**})$.

We first check the excess conditions. Since y^0 is in $Core(N_0, c^*)$, the *sdo* ensures that y^1 is in $Core(N_0, c^*)$. So, we know that $ex(R, y^1_{|N\setminus\{1\}}) \geq 0$ for all $R \subseteq N \setminus \{1\}$ in the *mcstp* $(N_0 \setminus \{1\}, C^*)$. By Lemma 6, we know that $(N_0 \setminus \{1\}, C^*) = (N_0 \setminus \{1\}, C^{**})$. Therefore, in the *mcstp* $(N_0 \setminus \{1\}, C^{**})$, every coalition has non-negative excess under cost allocation $y^1_{|N\setminus\{1\}}$.

Next we check the efficiency condition. Since y^0 satisfies component-wise efficiency and *sdo* is also efficient by Lemma 4, we know that that $m(N_0, C^*) = \sum_{(i,j)\in t} c^*_{ij} = \sum_{i\in N} y^1$. From the tie-breaking rule ρ, $\sum_{(i,j)\in t} c^*_{ij} = \sum_{(i,j)\in t^a} c^{**}_{ij} + \min\{c_{01}, c_{12}\}$ and $\sum_{i\in N} y^1 = \sum_{i\in N\setminus\{1\}} y^1_{|N\setminus\{1\}} + y^1_1$
$= \sum_{i\in N\setminus\{1\}} y^1_{|N\setminus\{1\}} + \min\{c_{01}, c_{12}\}$ by Lemma 5.
Thus, $\sum_{(i,j)\in t^a} c^{**}_{ij} = \sum_{i\in N\setminus\{1\}} y^1(N \setminus \{1\})$ which implies the efficiency condition. Therefore, $y^1_{|N\setminus\{1\}}$ is in $Core(N \setminus \{1\}, c^{**})$=$Core(N \setminus \{1\}, c^*)$ by excess conditions and efficiency condition.

Next stage, the *sdo* is applied to agent 2 and same thing happens. That is, $y^2_2 = \min\{c_{02}, c_{23}\}$ and $y^2_{|N\setminus\{1,2\}}$ is in $Core(N \setminus \{1, 2\}, c^*)$.

Check that $c_{03} = \max\{c_{02}, c_{23}\} = \max\{c_{01}, c_{12}, c_{p(3)3}\}$.

At k^{th} stage $(k < n)$, the *sdo* is applied to agent k and he gets $y_k^k = min\{c_{0k}, c_{k(k+1)}\}$.

Check that $c_{0k} = \max\{c_{01}, c_{12}, \cdots, c_{P(k)k}\}$=$\max_{j \leq k}\{c_{P(j)j}\}$ and $y^k_{|N\setminus\{1,\cdots,k\}}$ is in $Core(N \setminus \{1, \cdots, k\}, c^*)$.

Therefore, in each stage, the operator $k < n$ gets $y_k^k = \min\{\max_{l \leq k}\{c_{P(l)l}\}, c_{P(k+1)k+1}\}$ which coincides with ψ_k^{DK}.

□

6 Concluding remark

In this paper we study the relation between the *sdo* and the DK rule. Our main result shows the coincidence between our procedure using the strong demand operator and the DK rule.

Granot and Huberman (1984) also suggested the weak demand operator. Kim (2011) criticized weak demand operator and suggested a modified weak demand operator.

Modified weak demand operator is related with follower's opportunity cost. Compared to modified weak demand operator, strong demand operator usually transfers more than modified weak demand operator does. But in irreducible form, the amount of the transfer is limited since participation constraints of irreducible matrix C^* are more tighter than the participation constraints of cost matrix C.

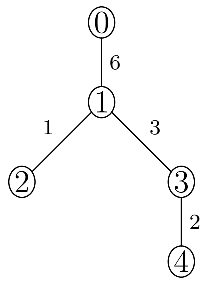
More precisely, if the *sdo* is applied to agent 1 in irreducible form, the participation constraint for coalition $N\setminus\{1\}$ coincides with the constraint of the modified weak demand operator, so we face more constraints compared with modified weak demand operator. Therefore if we proceed with weak demand operator, we have more initial allocation that can possibly generate DK allocation. Kim (2011) uses modified weak demand operator and shows that procedure using modified weak demand operator generates DK allocation if the initial allocation is component-wise efficient. Irreducible core is subset of component-wise efficient allocations, therefore the condition of modified weak demand operator is weaker than our condition.

Appendix: An example

Example 1. Let $N = \{1, 2, 3, 4\}$ and the cost matrix C be as below.

$$C = \begin{pmatrix} 0 & 6 & 7 & 8 & 10 \\ 6 & 0 & 1 & 3 & 6 \\ 7 & 1 & 0 & 7 & 4 \\ 8 & 3 & 7 & 0 & 2 \\ 10 & 6 & 4 & 2 & 0 \end{pmatrix}$$

The unique *mcst* t is illustrated as follows.



The irreducible matrix C^* is

$$C^* = \begin{pmatrix} 0 & 6 & 6 & 6 & 6 \\ 6 & 0 & 1 & 3 & 3 \\ 6 & 1 & 0 & 3 & 3 \\ 6 & 3 & 3 & 0 & 2 \\ 6 & 3 & 3 & 2 & 0 \end{pmatrix}$$

Let the initial cost allocation be $y^0 = \psi^B = (6, 1, 3, 2)$.

At 1^{st} step, *sdo* is applied to agent 1. Let $y^1 \equiv (10 - x - y, x, y, 2)$ be the allocation after the *sdo* is applied to agent 1. In this case, coalition set to be considered is $T_{\{2,3\},\{1\}} \cup T_{2,3} \cup T_{3,2} = \{\{2, 3\}, \{2, 3, 4\}\} \cup \{\{2\}, \{1, 2\}, \{2, 4\}, \{1, 2, 4\}\} \cup \{\{3\}, \{1, 3\}, \{3, 4\}, \{1, 3, 4\}\}$.

coalition (S)	cost ($c^*(S)$)	allocation	participation constraint
$\{2,3\}$	9	$x+y$	$9-x-y \geq 0$
$\{2,3,4\}$	11	$x+y+2$	$9-x-y \geq 0$
$\{2\}$	6	x	$6-x \geq 0$
$\{1,2\}$	7	$10-y$	$y-3 \geq 0$
$\{2,4\}$	9	$x+2$	$7-x \geq 0$
$\{1,2,4\}$	10	$12-y$	$y-2 \geq 0$
$\{3\}$	6	y	$6-y \geq 0$
$\{1,3\}$	9	$10-x$	$x-1 \geq 0$
$\{3,4\}$	8	$y+2$	$6-y \geq 0$
$\{1,3,4\}$	11	$12-x$	$x-1 \geq 0$

Agent 1 chooses maximum $x+y$ under these participation constraints. Therefore, $x+y=9$ and $sd^1(y^0)=(1,9-t,t,2)$ where $3 \leq t \leq 6$. In this case, $sd^1(y^0)$ is a set of allocations and agent 1 can choose any allocation in this set. For example, suppose that agent 1 chooses $t=5$, then $y^1=(1,4,5,2)$.

At 2^{nd} step, *sdo* is applied to agent 2. Let $y^2 \equiv (1,9-z,z,2)$ be the allocation after the *sdo* is applied to agent 2. In this case, coalition set to be considered is $T_{3,2}=\{\{3\},\{1,3\},\{3,4\},\{1,3,4\}\}$.

coalition (S)	cost ($c^*(S)$)	allocation	participation constraint
$\{3\}$	6	z	$6-z \geq 0$
$\{1,3\}$	9	$1+z$	$8-z \geq 0$
$\{3,4\}$	8	$z+2$	$6-z \geq 0$
$\{1,3,4\}$	11	$3+z$	$8-z \geq 0$

Agent 2 chooses maximum z under these participation constraints. Therefore, $z=6$ and $y^2=(1,3,6,2)$.

At 3^{rd} step, *sdo* is applied to agent 3. Let $y^3 \equiv (1,3,8-w,w)$ be the allocation after the *sdo* is applied to agent 3. In this case, coalition set to be considered is $T_{4,3}=\{\{4\},\{1,4\},\{2,4\},\{1,2,4\}\}$.

coalition (S)	cost ($c^*(S)$)	allocation	participation constraint
$\{4\}$	6	w	$6-w \geq 0$
$\{1,4\}$	9	$1+w$	$8-w \geq 0$
$\{2,4\}$	9	$3+w$	$6-w \geq 0$
$\{1,2,4\}$	10	$4+w$	$6-w \geq 0$

Agent 3 chooses maximum w under these participation constraints. Therefore, $w=6$ and $y^3=(1,3,2,6)$.

At 4^{th} step, the *sdo* is applied to agent 4, but in this time nothing changes since $f(4) = \emptyset$. Therefore, $y^4 = (1, 3, 2, 6)$.

Each figure below shows the alternative tree which is made when *sdo* is applied to agent 1 to agent 4.

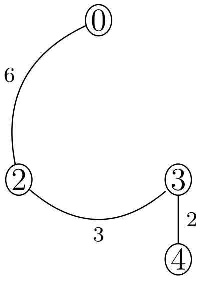


Figure 1-1.

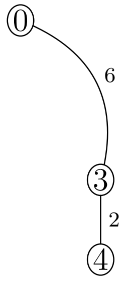


Figure 1-2.

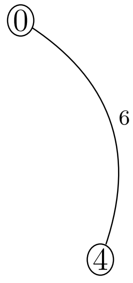


Figure 1-3.

Figure 1-4.

References

Aarts H. and T. Driessen, The irreducible core of a minimum cost spanning tree game, *ZOR - Methods and Models of Operations Research* **38** (1993), 163-174.

Bergantinos G. and J.J. Vidal-Puga, A fair rule in minimum cost spanning tree problems, *Journal of Economic Theory* **137** (2007), 326-352.

Bird C.G., On cost allocation for a spanning tree: A game theoretic approach, *Networks* **6** (1976), 335-350.

Chun Y. and J. Lee, Sequential contributions rules for minimum cost spanning tree problems, *Mathematical Social Sciences* **64** (2012), 136–143.

Claus A. and D. Kleitman, Cost allocation for a spanning tree, *Networks* **3** (1973), 289-304.

Dutta B. and A. Kar, Cost monotonicity, consistency and minimum cost spanning tree games, *Games and Economic Behavior* **48** (2004), 223-248.

Feltkamp V., S. Tijs, and S. Muto, On the irreducible core and the equal remaining obligations rule of minimum cost spanning extension problems, *Center for Economic Research Discussion Paper* **106** (1994), Tilburg University.

Granot D. and G. Huberman, On the core and nucleolus of minimum cost spanning tree games, *Mathematical Programming* **29** (1984), 323-347.

Kar A., Axiomatization of the Shapley value on minimum cost spanning tree games, *Games and economic behavior* **38** (2002), 265–277.

Kim B., Weak demand operator and the Dutta-Kar rule for minimum cost spanning tree problems, *mimeo* (2011), Seoul National University.

Prim R. C., Shortest connection networks and some generalizations, *Bell Systems Techn, J.* **36** (1957), 1389-1401.

Tijs S., S. Moretti, R. Branzei, and H. Norde, The Bird Core for Minimum Cost Spanning Tree Problems Revisited: Monotonicity and Additivity Aspects, *Center for Economic Research Discussion Paper No. 2005-03.*, Tilburg University.

국문초록

최소신장가지문제에서의 강요구연산자와 두타-카 규칙

한 창 용
경제학부 경제학 전공
서울대학교 대학원

본 논문은 최소신장가지문제(Minimum Cost Spanning Tree Problem)에서의 비용 배분에 대해 논의한다. 그 중에서도 버드 규칙 (Bird Rule)의 (i) 분배상태에 대한 비판을 하고 있는 강요구연산자 (Strong Demand Operator)와 (ii) 특성에 대한 비판을 하고 있는 두타-카 규칙 (Dutta-Kar Rule)간의 관계를 연구한다. 우선, 최소신장가지문제가 무엇인지에 대해 정의한다. 다음으로 기존에 정의된 강요구연산자를 복습한다. 다음으로 강요구연산자를 적용시키는 새로운 과정을 정의한다. 최종적으로 초기 비용 배분상태가 축약불가코어에 속해있다면, 이 새로운 과정의 결과는 두타-카 규칙과 일치한다는 것을 확인하여, 강요구연산자와 두타-카 규칙간의 관계를 밝힌다.

주요어: 최소신장가지문제, 강요구연산자, 축약불가행렬, 축약불가코어, 두타-카 규칙, 프림 알고리듬

학번: 2011-20197