Stable lexicographic rules for shortest path games

Eric Bahel∗ Christian Trudeau†

March 2014

Abstract

For the class of shortest path games, we propose a family of new cost sharing rules satisfying core selection. These rules allocate cost shares to the players according to some lexicographic preference relation. The average of all such lexicographic rules is shown to satisfy many desirable properties (core selection, symmetry, demand additivity,...). Our method relates to what Tijs et. al (2011) refer to as the Alexia value. We propose a procedure allowing to compute these lexicographic allocations for any shortest path game.

JEL classification: C71, D85

Keywords: shortest path, core, algorithm, lexicographic minima.

∗Department of Economics, Virginia Polytechnic Institute and State University, Blacksburg, VA 24061, USA. Email: erbahel@vt.edu.
†Department of Economics, University of Windsor, 401 Sunset Avenue, Windsor, Ontario, Canada. Email: trudeau@uwindsor.ca.


1 Introduction

In this paper we consider shortest path problems, which arise in networks where the cost of shipping demand units through each arc is linear. This framework, which is natural and appealing by its simplicity, may be used to model public transportation systems (connecting several urban areas) among other applications.

In a shortest path problem, each agent has to ship their demand from the source to their location, using the cost-minimizing route. In doing so, they might have to use the locations of other agents. This generates a cooperative game where the players have to share the total cost of shipping all demands through the optimal network. There are two underlying issues in this framework: (i) computing the shortest path network and (ii) designing sensible cost allocations for the resulting cooperative game. The issue (i) has been addressed by Dijkstra (1959), who proposed an algorithm allowing to compute the shortest path to any location in a network. In this paper, we focus on the issue (ii).

Shortest Path problems are part of the larger family of network (and source-connection) models.\(^1\) It is known from Rosenthal (2013), among others, that: (a) the core of a shortest path game is nonempty and (b) the Shapley value may produce an allocation that is not stable. A trivial core allocation is the one that charges the cost of their shortest path to the demander of each unit. This method does not remunerate at all the agents whose cooperation helps the demanders reduce their cost to connect to the source. This issue (of compensating agents who allow to reduce the costs) is present namely in the study of minimum cost spanning tree (mcst) problems, where all agents need to connect to a source and the cost to use an edge is the same regardless of the number of users of that edge. See for example Bird (1976) and Bergantinos and Vidal-Puga (2007).

Surprisingly, in the case of shortest path problems, the literature has not proposed stable cost sharing methods that are more sensible than the aforementioned rule.\(^2\) The main objective of the present paper is to address this deficiency. Our approach is in the vein of cost sharing problems with technological cooperation introduced by Bahel and Trudeau (2013 a,b). Indeed, in a shortest path problem,

\(^1\)See Sharkey (1995) for a review of network models and source connection problems.

\(^2\)Trudeau (2014) proposed an interesting family of rules for a class of problems that includes shortest path and mcst problems. However, these rules do not satisfy core selection (which is our key requirement in the present work).
adding agents to a given coalition typically increases its joint demand, but it also provides more connections through which units of demand can be shipped. As a result, each coalition has a specific technology (or network configuration in this case); and the total cost of shipping the demands may well decrease as a coalition gets larger.

We propose a family of methods that minimize the cost shares according to some lexicographic ordering of the agents. These methods are shown to satisfy core selection: they produce an extreme allocation in the core of every shortest path problem. In this sense our results relate to those of Tijs et. al (2011), who show the existence of these lexicographic extrema in the core of balanced games. For the case of shortest path games, we propose an algorithm allowing to compute these allocations. We then define a single-valued solution as the average of these lexicographic rules (obtained for all possible permutations of the set of players). In addition to core selection and the fact that it can be computed using our procedure, this solution concept satisfies many desirable properties.

2 Preliminaries

Let us consider a fixed point $s$ from which agents (residing at various locations) need to ship their respective demands of some homogeneous goods — $s$ is called the source. A Shortest Path Problem (SPP) is a tuple $(N, c, x)$, where (i) $N$ is the set of agents (or vertices) that need to connect to the source $s$; (ii) $c = \{c(i, j) | i, j \in N \cup \{s\}, i \neq j\}$ is a collection (of positive numbers) giving the unit cost of shipping demands through every edge $(i, j)$ s.t. $i \neq j$; (iii) $x \in \mathbb{N}^N$ is the demand profile: each agent $i$ has $x_i \in \mathbb{N}$ units of demand to ship from the source to her location. Note that we do not view the source $s$ as a player. Also observe that the unit costs $c(i, j)$ need not be symmetric: we may have $c(i, j) \neq c(j, i)$ for some $i$ and $j$. If instead $c(i, j) = c(j, i)$ for any distinct $i, j \in N \cup \{s\}$, we will say that the SPP has symmetric arcs.

For the rest of this section, we consider a fixed set of agents $N$, and a fixed cost structure $c$. Only the demand profile $x$ may vary from one problem to the other.

**Definition 1** Let $i \in N$. We call path (of length $K$) to $i$ any sequence $p \equiv (p_k)_{k=1,...,K}$ such that: (i) $p_k \in N \cup \{s\}$, for $k = 0, 1, ..., K$; (ii) $p_0 = s$ and $p_K = i$; (iii) $p_k \neq p_{k'}$ for any distinct $k, k'$. 


Note from the definition that all paths $p$ originate from the source $s$ and cross any location $p_k$ only once. Thus, the length of each path and the number of paths to any given $i \in N$ are both finite. We denote by $P(i)$ the set containing all paths to $i$. For any path $p$ of length $K$, let $[p]$ refer to the set of players in the range of $p$, that is: $[p] \equiv \{ i \in N | p_k = i \text{ for some } k = 1, \ldots, K \}$. For any subset $M \subseteq N$ and any path $p$ (of length $K$) such that $M \subseteq [p]$, we will write $p \setminus M$ to refer to the unique path (of length $K - |M|$) where the agents of $M$ have been deleted and the remaining agents (of $[p]$) appear in the same order as in $p$. To ease on notation, we will often write $i$ instead of $\{i\}$ and $p \setminus i$ instead of $p \setminus \{i\}$, for any $i \in [p]$. Finally, for any $M \subseteq N$ we will write $\Pi(M)$ to refer to the set containing all permutations of $M$.

Given an SPP $(N, c, x)$, one can extend the cost function $c$ (initially defined on arcs) to paths in the following way: for any path $p$ (of length $K$) to $i$,

$$c(p) = \sum_{k=1}^{K} c(p_{k-1}, p_k).$$

In words, $c(p)$ stands for the cost of shipping one unit of demand from the source to agent $i$ via the path $p$. For any $i \in N$, we call shortest path to $i$ any path $p^* \in P(i)$ that solves the problem $\min_{p \in P(i)} c(p)$. In essence, a shortest path is one that minimizes the cost of shipping a unit of demand from the source to agent $i$. Note that there exists a shortest path to any $i \in N$ — since the set $P(i)$ is nonempty and finite; but it may not be unique.

**Example 1** Consider the SPP (with symmetric arcs) given by $P = (N, c, x)$, where $N = \{1, 2, 3\}$, $x = (2, 0, 1)$ and the cost structure is depicted by Figure 1. For example, we have $c(s, 1) = 200$, $c(3, 1) = c(1, 3) = 50$ and $c(2, 1) = c(1, 2) = 10$.

One can see that there are 5 paths to agent 1, $(s, 1), (s, 2, 1), (s, 3, 1), (s, 2, 3, 1), (s, 3, 2, 1)$; and the (unique) shortest path to 1 is $(s, 2, 1)$, with cost $c(s, 2, 1) = 80 + 10 = 90$. For agents 2 and 3, the costs of the respective shortest paths are $c(s, 2) = 80$ and $c(s, 3) = 100$.

Let us introduce some additional notation. For any vector $y \in \mathbb{R}^N$ and any subset $S \subseteq N$, let $y(S) \equiv \sum_{i \in S} y_i$. In addition, we will use the notation $S(y) \equiv \{ i \in S \text{ s.t. } y_i > 0 \}$. It is not difficult to see that there is a natural way to formulate the (transferable cost) cooperative game associated with $P$. 

3
Define the cost of any nonempty coalition $S \subseteq N$ as follows:

$$C_P(S) \equiv \min \left\{ \sum_{j \in S} x_j c(p^j) | p^j \in \mathcal{P}(j) \text{ and } [p^j] \subseteq S, \forall j \in S \right\}.$$  

Equation (1) gives the minimum possible cost of shipping (from the source) the respective demands of the members of $S$ when using only the connections available in $S$. Note in particular that $C_P(S) = 0$ for any $S \subset N$ such that $S(x) = \emptyset$ (the cost is null if there is no demand to ship). We also adopt the usual convention that $C_P(\emptyset) = 0$. As an illustration, for the problem $P$ depicted in Example 1, note that we have $C_P(N) = 2 \times c(s, 2, 1) + 0 \times c(s, 2) + 1 \times c(s, 3) = 180 + 100 = 280$.

**Definition 2** Given a shortest path problem $P = (N, c, x)$, we have the following.

(i) An allocation is a profile of cost shares, $y \in \mathbb{R}^N$, such that $y(N) = C_P(N)$. Let $\mathcal{A}(P)$ be the set containing all cost allocations.

(ii) The core of $P$ is the set

$$\text{Core}(P) \equiv \{ y \in \mathcal{A}(P) | y(S) \leq C_P(S), \text{ for any } S \text{ s.t. } \emptyset \neq S \subset N \}.$$ 

An allocation $y$ will be called stable if $y \in \text{Core}(P)$.

Definition 2-(i) says that a cost allocation is a way of “splitting” the cost $C_P(S)$ of the coalition $S$ between its members. Note that we allow for negative cost shares, which are desirable if we have (for instance) agents who demand zero while providing the
others with a cheap connection to the source. It is natural in such cases to subsidize these agents. Definition 2-(ii) is the standard notion of stability: an allocation \( y \) is stable if every coalition \( S \) of players jointly pays less than its stand-alone cost \( C_P(S) \).

**Example 2** Recall the SPP of Example 1. Stability requires that \( x_2 \leq 0 = C_P(\{2\}) \) and \( x_1 \leq 400 = C_P(\{1\}) \). In fact, a further examination shows that

\[
\text{Core}(P) = \{ x \in \mathbb{R}^3 | x_3 = 100, 180 \leq x_1 \leq 300 \text{ and } x_2 = 180 - x_1 \}.
\]

Note that any cost allocation s.t. \( x_1 > 300 \) and \( x_3 = 100 \) is blocked by the coalition \( \tilde{S} = \{1, 3\} \), since \( C_P(\tilde{S}) = 400 \).

The following section provides a procedure allowing to compute a (nonempty) subset of the core for any shortest path problem.

### 3 Analysis

In the remainder of the paper, unless otherwise specified, we consider an arbitrary but fixed SPP given by \( P = (N, c, x) \), where \( 0^N \neq x \in \mathbb{N}^N \).

#### 3.1 Decomposition

Since there are no externalities, congestion is not an issue in shortest path games: shipping one unit to a given agent does not affect the cost of shipping the next unit (to any agent). Using this observation, we first study "elementary" SPP, which have the property that only one agent has a (unitary) demand.

For every \( j \in N \), denote by \( e^j \in \mathbb{R}^N \) the vector characterized by \( e^j_j = 1 \) and \( e^j_i = 0 \), if \( i \in N \setminus j \). Let \( A, B \subset \mathbb{R}^N \) and \( \alpha \in \mathbb{R} \). We use the following conventions: \( A + B \equiv \{ a + b | a \in A \text{ and } b \in B \} \); \( \alpha \cdot A \equiv \{ \alpha a | a \in A \} \). This notation allows to write the result hereafter.

**Lemma 1** Given the problem \( P = (N, c, x) \), we have

\[
\mathcal{A}(P) = \sum_{j \in N(x)} x_j \cdot \mathcal{A}(P^j) \quad \sum_{j \in N(x)} x_j \cdot \text{Core}(P^j) \subseteq \text{Core}(P),
\]
where $P^j \equiv (N, c, e^j)$. The proof is omitted. It trivially follows from Equation (1) and Definition 2.

### 3.2 Algorithm

Let us now focus on elementary SPP of the form $P^j \equiv (N, c, e^j)$, with $j \in N$. Let $c^j = c(s, j)$ be the cost of the direct connection of agent $j$ to the source. Let us pick (one of) the shortest path(s) to $j$, $p^m$, and choose $\bar{p} \in \arg \min \{c(p) \mid p \in \mathcal{P}(j) \text{ s.t. } c(p) \leq c^j \text{ and } [p] \cap [p^m] = j\}$. Note that $\bar{p}$ always exists (but need not be unique) since the above set is nonempty and finite.\(^3\) Observe that, in any stable allocation, agent $j$’s share cannot exceed $c(\bar{p})$. We are now set to describe the algorithm allowing to compute $y^\pi$, the extreme point of the core of $P^j$ associated with any (given) permutation $\pi \in \Pi([p^m])$.

Fix a permutation $\pi$ of $[p^m]$. If $\pi(j) = \min_{i \in [p^m]} \pi(i)$, simply assign the shares $y^\pi(P^j) = c(p^m)e^j + 0^{N \setminus j} = c(p^m)e^j$.

Otherwise, we can write the following (without loss of generality):

$$\emptyset \neq \{i \in [p^m] \mid \pi(j) < \pi(i)\} = \{i_1, ..., i_l\},$$

where $l \geq 1$ and $\pi(i_1) < ... < \pi(i_l)$. The procedure described below then applies.

**Stage 1.**

Define $\mathcal{P}^1 \equiv \{p \in \mathcal{P}(j) \mid c(p) \leq c(\bar{p})\}$. Assign the following shares.

- $y^\pi_{i_1}(P^j) = 0$ if $i \notin \{i_1, ..., i_l, j\}$.
- $y^\pi_{i_1}(P^j) = c(p^m) - c(\bar{p}_{i_1})$, where $\bar{p}_{i_1} \in \arg \min \{c(p) \mid p \in \mathcal{P}^1 \text{ s.t. } i_1 \notin [p]\}$.\(^4\)

Let $\mathcal{P}^1$ be the set containing all paths $p$ (to $j$) that can be written as $p = p' \setminus i_1$, with $p' \in \mathcal{P}^1$; and then define

$$c_1(p) = \min(c(p), c(p') - y^\pi_{i_1}), \forall p \in \mathcal{P}^1;$$

$$c_1(p) = c(p), \forall p \in \mathcal{P}^1 \setminus \mathcal{P}^1.$$  \(2\)

Moreover, let $\bar{c}_1(\pi) \equiv \min \{c_1(p) \mid p \in \mathcal{P}^1 \cup \mathcal{P}^1 \text{ s.t. } [p] \cap \{i_1, ..., i_l, j\} = j\}$.

\(^3\)Remark that the path $(s, j)$ belongs to $\{c(p) \mid p \in \mathcal{P}(j) \text{ s.t. } c(p) \leq c^j \text{ and } [p] \cap [p^m] = j\}$. Also, in case there exist multiple such $\bar{p}$, which one is picked is irrelevant to the outcome.

\(^4\)Once again, such a $\bar{p}_{i_1}$ exists because the set $\{p \in \mathcal{P}^1 \text{ s.t. } i_1 \notin [p]\}$ is nonempty —it contains the path $\bar{p}$— and finite. It need not be unique, however.
Stage $k = 2, \ldots, l$.
Define $\mathcal{P}^k \equiv \{ p \in \mathcal{P}^{k-1} \cup \mathcal{P}^{k-1} | i_{k-1} \notin [p] \text{ and } c_{k-1}(p) \leq c_{k-1}^j(\pi) \}$ and assign the following share to player $i_k$:

- $y_{i_k}^\pi(P^j) = c_{k-1}(p_{i_{k-1}}^-) - c_{k-1}(p_{i_{k-1}}^+)$, where $p_{i_{k-1}}^- \in \arg\min\{ c_{k-1}(p) | p \in \mathcal{P}^k \text{ s.t. } i_k \notin [p] \}$.

Next, one needs to adjust the costs of the respective paths. Let $\mathcal{P}^k$ be the set containing all paths $p$ (to $j$) that can be written as $p = p' \setminus i_k$, where $p' \in \mathcal{P}^k$; and define

$$
c_k(p) = \min(c_{k-1}(p), c_{k-1}(p') - y_{i_k}^\pi), \forall p \in \mathcal{P}^k;
$$

$$
c_k(p) = c_{k-1}(p), \forall p \in \mathcal{P}^k \setminus \mathcal{P}^k.
$$

In addition, let $\overline{c}_k^j(\pi) \equiv \min\{ c_k(p) | p \in \mathcal{P}^k \cup \mathcal{P}^k \text{ s.t. } [p] \cap \{i_1, \ldots, i_l, j\} = j \}$.

- Finally, after all $l$ stages, assign the share $y_j^\pi(P^j) = c(p^m) - \sum_{k=1}^l y_{i_k}^\pi(P^j)$.

Note that all agents $i$ that do not belong to the optimal path (i.e., $i \notin [p^m]$) are assigned a cost share of zero in the first stage of the procedure.

Example 3 (a) Recall the Problem $P$ of Example 1 and consider the problem $P^1$, where only agent one has a (unit) demand. As already mentioned, the cost of shortest path to 1 is $p^m = (s, 2, 1)$, with cost 90. One can then see that we have $\bar{p} = (s, 2, 1)$, with cost 150. Let $\pi$ be the permutation of $[p^m] = \{1, 2\}$ s.t. $\pi(1) = 2$. At stage 1, we can define $\mathcal{P}^1 \equiv \{ p \in \mathcal{P}(1) | c(p) \leq c(\bar{p}) \} = \{(s, 3, 1), (s, 2, 1)\}$. Thus, one gets $y_3 = 0$ and $y_2^\pi(P^1) = 90 - 150 = -60$. The algorithm ends at this first stage (since $l = 1$) and we have to assign $y_j^\pi(P^1) = 90 - (-60) = 150$. The outcome is hence $y^\pi(P^1) = (150, -60, 0)$.

(b) Consider an elementary SPP (with symmetric arcs) $\hat{P}^1 = (N, c, e^1)$, where $N = \{1, 2, 3, 4\}$ and $c$ is depicted by Figure 2. One can see that the shortest path to 1 is $p^m = (s, 2, 4, 1)$, with cost 20 + 30 + 0 + 0 = 50. In this case, we find that $\bar{p} = (s, 1)$, with cost 100. This means that, at Stage 1, we have $\mathcal{P}^1 \equiv \{ p \in \mathcal{P}(1) | c(p) \leq 100 \} = \{(s, 3, 2, 4, 1), (s, 3, 1), (s, 2, 4, 1), (s, 2, 1), (s, 1)\}$. Using the ordering $\pi = 4321$ of $[p^m]$, one may then assign $y_j^\pi(\hat{P}^1) = 50 - c(s, 3, 1) = 50 - 70 = 20$.

---

5Observe that any path $p \in \mathcal{P}^{k-1}$ s.t. $i_{k-1} \notin [p]$ belongs to the domain of the function $c_{k-1}$ (defined at stage $k - 1$) and, therefore, $\mathcal{P}^k$ is well defined. Furthermore, we have $i_1, \ldots, i_{k-1} \notin [p]$ and $c_{k-1}(p_{i_{k-1}}) \leq c_{k-1}(p)$, for any $p \in \mathcal{P}^k$ and any $k \in \{2, \ldots, l\}$.

6It can be shown that $\overline{c}_k^j(\pi) = \overline{c}_{k-1}^j(\pi)$. 

7
−20. We then have to modify the costs of the following paths, as recommended by the algorithm.

\[ c_1(s, 2, 1) = \min \left( c(s, 2, 1), c(s, 2, 4, 1) - y_4^\pi(\hat{P}^1) \right) = \min(85, 60 - (-20)) = 80; \]

\[ c_1(s, 3, 1) = c(s, 3, 1) = 70 \text{ and } c_1(s, 1) = c(s, 1) = 100 = \bar{c}_1^j(\pi). \]

At stage 2, we have \( P^2 = \{(s, 3, 1), (s, 2, 1), (s, 1)\} \) and hence we may assign the share \( y_3^\pi(\hat{P}^1) = c_1(s, 3, 1) - c_1(s, 2, 1) = 70 - 80 = -10 \). Adjusting the cost of the remaining paths then gives \( c_2(s, 1) = \min (c_1(s, 1), c_1(s, 3, 1)) - y_3^\pi(\hat{P}^1) = 70 - (-10) = 80 \) and \( c_2(s, 2, 1) = c_1(s, 2, 1) = 80 \).

Hence, at Stage \( l = 3 \), we have \( y_2^\pi(\hat{P}^1) = c_2(s, 2, 1) - c_2(s, 1) = 80 - 80 = 0 \). Finally, after all three stages, one can write \( y_1^\pi(\hat{P}^1) = 50 - (-20) = 70 \). That is to say, \( y^\pi(\hat{P}^1) = (80, 0, -10, -20) \).

See the Appendix for the respective allocations obtained for all possible permutations.

### 3.3 Lexicographic core allocations

We first claim that the allocations generated by the algorithm are extreme points of the core of \( P^j \), which is stated in the following theorem.

**Theorem 1** Given \( P^j = (N, c, e^j) \), where \( j \in N \), and a permutation \( \pi \) of some optimal path \( p^m \) (to \( j \), the cost allocation \( y^\pi(P^j) \) is an extreme point of Core(\( P^j \)).
Proof. Fix $j \in N$ and let $p^m$ be a shortest path to $j$. In addition, consider a permutation $\pi$ of $[p^m]$. For notational simplicity, since $P^j$ is fixed, we will write $y^\pi$ [instead of $y^\pi(P^j)$] throughout this proof. Note that $y^\pi$ is a well-defined allocation for the problem $P^j$, since $y_j^\pi + \sum_{k=1}^iy_k^\pi = c(p^m)$ and $y_i = 0$ if $i \notin \{i_1, ..., i, j\}$.

(a) Let us first argue that $y^\pi$ is stable. We show that no coalition $S \subseteq N$ can pay less (on their own) than their joint share $y^\pi(S)$.

(a.1) It is obvious that no $S$ s.t. $j \notin S$ can improve on $y^\pi$ —since $y^\pi(S) \leq 0$.

(a.2) If $j \in S$ and $S \setminus \{j\} \subseteq N \setminus \{i_1, ..., i_l\}$, then we have:

$$y^\pi(S) = y_j^\pi = c(p^m) - y_{i_1}^\pi - ... - y_{i_l}^\pi = c_i(\pi) \leq c(p),$$

for any path $p \in \mathcal{P}(j)$ s.t. $[p] \subseteq N \setminus \{i_1, ..., i_l\}$.

(a.3) Finally, consider $S$ s.t. $j \in S$ and $(S \setminus \{j\}) \cap \{i_1, ..., i_l\} = \{k_1, ..., k_t\}$, with $1 \leq t \leq l$ and $\pi(k_1) < ... < \pi(k_t) < \pi(j)$. Then we have

$$y^\pi(S) = y_{k_1}^\pi + ... + y_{k_t}^\pi + y_j^\pi = y_{k_1}^\pi + ... + y_{k_t}^\pi + c_i(\pi).$$

By way of contradiction, assume that there exists a path $p \in \mathcal{P}(j)$ s.t. $[p] \subseteq S$ and

$$c(p) < y^\pi(S) = y_{k_1}^\pi + ... + y_{k_t}^\pi + c_i(\pi).$$

Let $n_{t'} = |i \in [p] \setminus j$ s.t. $\pi(i) < \pi(k_{t'})| + 1$, for $t' = 1, ..., t$. One can then write

$$c_{n_1}(p \setminus i_{k_1}) \equiv \underbrace{c_{n_1 - 1}(p)}_{\leq c(p)} - y_{k_1}^\pi \leq c(p) - y_{k_1}^\pi < y_{k_2}^\pi + ... + y_{k_t}^\pi + c_i(\pi).$$

Doing the same for $k_2$ gives

$$c_{n_2}(p \setminus i_{k_1}i_{k_2}) \leq c_{n_1}(p \setminus i_{k_1}) - y_{k_2}^\pi < y_{k_3}^\pi + ... + y_{k_t}^\pi + c_i(\pi).$$

After $t$ iterations of this procedure, one obtains

$$c_{n_t}(p \setminus i_{k_1}...i_{k_t}) \leq c_{n_{t-1}}(p \setminus i_{k_1}...i_{k_{t-1}}) - y_{k_t}^\pi < c_i^t(\pi). \quad (4)$$

Given that $i_q \notin S$, for any $i_q \in [p^m]$ s.t. $\pi(i_q) > \pi(k_t)$, we can claim that $c_q(p \setminus i_{k_1}...i_{k_t}) = c_{n_t}(p \setminus i_{k_1}...i_{k_t})$, for any $q = n_t, ..., l$. In particular, it follows that $c_t(p \setminus i_{k_1}...i_{k_t}) = c_{n_t}(p \setminus i_{k_1}...i_{k_t})$. Plugging this into (4) then gives

$$c_t(p \setminus i_{k_1}...i_{k_t}) < c_i^t(\pi),$$

9
which is a contradiction since $c_l$ is (by definition) the minimized value of the cost function $c_l(\cdot)$ on the set of $p' \in \mathcal{P}(j)$ such that $[p'] \cap \{i_1, ..., i_l, j\} = j$ —a condition satisfied by the path $p \setminus i_{k_1} ... i_{k_l}$, since $[p] \subseteq S$ and $(S \setminus \{j\}) \cap \{i_1, ..., i_l\} = \{k_1, ..., k_l\}$.

(b) By construction, $y^\pi$ lexicographically-minimizes the vector of cost shares (in the core) with respect to any ordering $\pi'$ of $N$ whose restriction to $[p^m]$ coincides with $\pi$. It is an extreme point (vertex) of the core because, for any $y \in \mathbb{R}^N$, we have the following: $y <_{\pi'} y^\pi \Rightarrow y \notin \text{Core}(N, c, x)$ —where $<_{\pi'}$ stands for the lexicographic order associated with $\pi'$.

The allocations constructed by the procedure described in Subsection 3.2 are thus stable and extreme in the core. A noteworthy observation is the fact that different permutations (of the optimal path to $j$) may well lead to the same allocation.\footnote{For instance, the reader can check that taking $\pi = 4231$ in Example 3-(b) results in the allocation $y^\pi(\hat{P}^1) = (80, 0, -10, -20)$, which is identical to the one we found using $\pi = 4321$.}

For any finite set $X \subset \mathbb{R}^N$, let $\text{Conv}(X)$ denote the convex hull of $X$. Since the core of any TU game is convex, the following result easily obtains.

**Theorem 2** For any elementary problem $P^j = (N, c, e^j)$ and any shortest path (to $j$) $p^m$, we have

$$\text{Conv}\left\{(y^\pi(P^j)|\pi \in \Pi([p^m]))\right\} \subseteq \text{Core}(P^j).$$

Our algorithm thus allows to generate a nonempty subset of the core when one varies the permutation $\pi$ of the shortest path to $j$. Recall the observation that there may exist multiple shortest paths. The following lemma states that, in such a case, the outcome of the procedure remains the same no matter what the optimal path used in the algorithm (as long as the chosen permutations coincide on the intersection of these shortest paths).

**Lemma 2** Suppose there exist two shortest paths to $j \in N$, $\bar{p}$ and $\tilde{p}$. Then for any $\bar{\pi} \in \Pi([\bar{p}])$ and $\tilde{\pi} \in \Pi([\tilde{p}])$ that coincide on the set $[\bar{p}] \cap [\tilde{p}]$, we have $y^{\bar{\pi}}(P^j) = y^{\tilde{\pi}}(P^j)$.

**Proof.** Consider two shortest paths to $j \in N$, $\bar{p}$ and $\tilde{p}$, and fix $\bar{\pi} \in \Pi([\bar{p}])$ and $\tilde{\pi} \in \Pi([\tilde{p}])$ s.t. $\bar{\pi}(i) = \tilde{\pi}(i)$, for all $i \in [\bar{p}] \cap [\tilde{p}]$. Recall from the algorithm that $y^{\bar{\pi}}_i(P^j) = 0$, for any $i \notin [\bar{p}]$. In addition, by efficiency, we can write:

$$\sum_{i \in [\bar{p}] \cap [\tilde{p}]} y^{\bar{\pi}}_i(P^j) + \sum_{i \in [\bar{p}] \setminus [\tilde{p}]} y^{\bar{\pi}}_i(P^j) = c(\bar{p}) = c(\tilde{p})$$

(5)
Suppose now (by contradiction) that \( \exists k \in [\bar{p}] \setminus [\tilde{p}] \) s.t. \( y^\pi_k(P_j) < 0 \). Since \( y^\pi_i(P_j) \leq 0 \) for any \( i \in N \setminus j \) (by individual rationality), we then have \( \sum_{i \in [\bar{p}] \setminus [\tilde{p}]} y^\pi_i(P_j) < 0 \). It hence follows from (5) that

\[
\sum_{i \in [\bar{p}] \setminus [\tilde{p}]} y^\pi_i(P_j) = \sum_{i \in [p] \cap [\tilde{p}]} y^\pi_i(P_j) + \sum_{i \in [\bar{p}] \setminus [\tilde{p}]} y^\pi_i(P_j) = \sum_{i \in [\bar{p}]} y^\pi_i(P_j) > c(\tilde{p}).
\]

This contradicts the fact that \( y^\pi(P_j) \) is a stable cost allocation for \( P_j \). Therefore, \( y^\pi_i(P_j) = 0 \) for any \( i \in [\bar{p}] \setminus [\tilde{p}] \). Likewise, \( y^\pi_i(P_j) = 0 \) for any \( i \in [\tilde{p}] \setminus [\bar{p}] \). Since \( \bar{\pi} \) and \( \tilde{\pi} \) coincide on the set \([\bar{p}] \cap [\tilde{p}]\), it is then easy to see that the algorithm will give \( y^\pi_i(P_j) = y^\pi_i(P_j) \) for any \( i \in [\bar{p}] \cap [\tilde{p}] \). Hence, \( y^\pi_i(P_j) \) and \( y^\pi_i(P_j) \) are identical. \( \square \)

This result is useful in the sense that it allows to define a unique cost allocation (associated with any given permutation of the set of players) in the following way. For any \( j \in N \), let \( p^m_j \) be (any of) the shortest path(s) to \( j \in N \); and consider a permutation \( \sigma \in \Pi(N) \). In the problem \( P = (N, c, x) \), define the following cost allocation:\(^8\)

\[
y^\sigma(P) \equiv \sum_{j \in N(x)} x_j y^\sigma[p^m_j](P_j),
\]

where \( \sigma[p^m_j] \) is the restriction of \( \sigma \) to \([p^m_j]\). We may now state the following result.

**Theorem 3** Let \( P = (N, c, x) \) be a shortest path problem. Then we have

\[
\text{Conv} \left( \{ y^\sigma(P) | \sigma \in \Pi(N) \} \right) \subseteq \text{Core}(P).
\]

**Proof.** The result follows from the combination of Lemma 1 and Theorem 2.

The share profile \( y^\sigma \) gives an allocation that minimizes the share of agent \( \sigma(1) \) in the core of \( P \). Furthermore, \( y^\sigma \) minimizes agent \( \sigma(2) \)'s share among all stable allocations that give agent \( \sigma(1) \) her lowest share in the core, and so forth. These \( y^\sigma \) (for all \( \sigma \in \Pi(N) \)) correspond to the stable “lexicographic allocations” studied by Tijs et al. (2011): for the case of shortest path games, our algorithm allows to

\(^8\)Note that, if the result of Lemma 2 did not hold, \( y^\sigma \) would not be well defined in the case of multiple shortest paths. Lemma 2 tells us that only \( \sigma \) (and not the optimal path chosen to compute the shares) matters in such a case.
compute these core allocations.\(^9\) As shown by Tijs et al. (2011), these lexicographic allocations are in general not the unique extreme points of the core; this is so even for the subclass of shortest path games.

### 3.4 Average lexicographic allocation

We call (cost sharing) rule any mapping \(y\) which, to every shortest path problem \(P = (N, c, x)\), assigns an allocation of \(P\), \(y(P)\). Interestingly, a natural (single-valued) solution to any shortest path problem \(P\) is the average lexicographic allocation [referred to as the Alexia by Tijs et al. (2011)], which we define here as

\[
y^*(P) = \frac{1}{|N|!} \sum_{\sigma \in \Pi(N)} y^\sigma(P).
\]

The Alexia of an SPP is obviously stable as a convex combination of lexicographic allocations (Theorem 3); and it may be computed using our algorithm. As an illustration, for the problem \(P\) introduced in Example 1, it is not difficult to see that:\(^{10}\)

\[
y^*(P) = \frac{1}{6} (6 \cdot (0, 0, 100) + 3 \cdot [2 \cdot (90, 0, 0) + 2 \cdot (150, -60, 0)]) = (240, -60, 100).
\]

See Appendix A for the computation of the Alexia of the SPP in Example 3-(b).

The mapping \(y^*\), which we will refer to as the Alexia rule, satisfies many desirable properties, as stated by the next result. This is in addition to the fact that it may be computed using the procedure described in this work.

**Definition 3**

We say that a cost sharing rule \(y\) satisfies

(a) core selection if: \(y(P) \in Core(P)\), for any shortest path problem \(P\);

(b) symmetry if: \(y_i(P) = y_j(P)\), for any \(P = (N, c, x)\) and any \(i, j \in N\) s.t. (i) \(x_i = x_j\) and (ii) \(c_{ik} = c_{jk}, \forall k \in \{s\} \cup N \setminus ij\);

\(^9\)Tijs et al. (2011) prove the existence of these extreme allocations in the core of balanced games; they do not provide a computational method. We should also point out that the authors define these lexicographic allocations for value games; but it is readily checked that their concept carries through to cost games (such as our SPP).

\(^{10}\)Recall from Example 3-(a) that \(y^{\Pi_1}(P^1) = (150, -60, 0)\). In addition, note that there are 3 permutations \(\sigma\) of \(N = \{1, 2, 3\}\) whose restrictions to \(\sigma[1] = 2, \sigma[2] = 1\) is \(\pi = \Pi_1\). Also recall that the demands are \(x_1 = 2, x_2 = 0\) and \(x_3 = 1\). Finally, recall that the shortest path to 3 is \((s, 3)\), which means that, for each of the 6 permutations of \(N\), agent 3 pays 100 to ship her unit (with the others receiving no subsidies).
(c) demand additivity if: for any 3 SPP of the form $\tilde{P} = (N, c, \tilde{x}), \hat{P} = (N, c, \hat{x})$ and $P = (N, c, \tilde{x} + \hat{x})$, we have $y_i(P) = y_i(\tilde{P}) + y_i(\hat{P}), \forall i \in N$;

(d) continuity if: for any sequence $\{P_k = (N, c_k, x_k)\}_{k \in \mathbb{N}}$ which converges to $P = (N, c, x)$, we have $\lim_{k \to \infty} y(P_k) = y(P)$.

It is readily checked that the following result about the Alexia rule holds.

**Theorem 4** On the set of shortest path problems, the Alexia rule satisfies core selection, symmetry, demand additivity, continuity.

It is worth noting that the Alexia rule is not the unique rule satisfying these requirements. For example, it can be checked that the nucleolus rule (which is much harder to compute, especially in large SPP) also meets the properties of Theorem 4.

## 4 Concluding comments

We study shortest path problems, that is, network problems for which the cost of each edge is linear in the demand crossing it. For this class of problems, we show that core allocation may always be constructed by considering arbitrary lexicographic preferences. Our procedure is first defined for elementary problems (where a single agent has a unit demand); and then extended in a unique way to general shortest path problems.

Taking the average of all lexicographic allocations obtained with the algorithm, we define a single-valued solution that coincides with the Alexia value of Tijs et al. (2011). Our algorithm thus allows to compute this solution for shortest path problems. In addition, the Alexia rule satisfies many appealing properties in our context (core selection, symmetry, demand additivity,...). It is hence a natural and computable solution for these shortest path games.

An interesting open question for future research is the axiomatization of the Alexia rule in shortest path games or, more generally, in cooperative games with transferable utility.
References


A Appendix

Consider the problem \( \hat{P}^1 \) introduced in Example 3-(b). Taking all possible permutations of \( [p^m] = \{1, 2, 3, 4\} \), one obtains the following table.\(^\text{11}\) Recall that, given a permutation \( \pi \) of \( [p^m] \), we have \( l = |\{i \in [p^m] \mid \pi(i) < \pi(1)\}| \).

<table>
<thead>
<tr>
<th>( \pi ) of ( [p^m] )</th>
<th>number of ( \pi )</th>
<th>value of ( l )</th>
<th>( y^\pi(P^1) )</th>
</tr>
</thead>
<tbody>
<tr>
<td>1----</td>
<td>6</td>
<td>0</td>
<td>(50, 0, 0, 0)</td>
</tr>
<tr>
<td>21--</td>
<td>2</td>
<td>1</td>
<td>(70, -20, 0, 0)</td>
</tr>
<tr>
<td>31--</td>
<td>2</td>
<td>1</td>
<td>(60, 0, -10, 0)</td>
</tr>
<tr>
<td>41--</td>
<td>2</td>
<td>1</td>
<td>(70, 0, 0, -20)</td>
</tr>
<tr>
<td>--14</td>
<td>2</td>
<td>2</td>
<td>(80, -20, -10, 0)</td>
</tr>
<tr>
<td>2413</td>
<td>1</td>
<td>2</td>
<td>(70, -20, 0, 0)</td>
</tr>
<tr>
<td>4213</td>
<td>1</td>
<td>2</td>
<td>(70, 0, 0, -20)</td>
</tr>
<tr>
<td>--12</td>
<td>2</td>
<td>2</td>
<td>(80, 0, -10, -20)</td>
</tr>
<tr>
<td>4--1</td>
<td>2</td>
<td>3</td>
<td>(80, 0, -10, -20)</td>
</tr>
<tr>
<td>3421</td>
<td>1</td>
<td>3</td>
<td>(80, 0, -10, -20)</td>
</tr>
<tr>
<td>3241</td>
<td>1</td>
<td>3</td>
<td>(80, -20, -10, 0)</td>
</tr>
<tr>
<td>2--1</td>
<td>2</td>
<td>3</td>
<td>(80, -20, -10, 0)</td>
</tr>
<tr>
<td>Alexia</td>
<td>24</td>
<td></td>
<td>(68(\frac{1}{3}), -6(\frac{2}{3}), -5, -6(\frac{2}{3}))</td>
</tr>
</tbody>
</table>

The average lexicographic allocation is thus \( y^*(\hat{P}^1) = (68\frac{1}{3}, -6\frac{2}{3}, -5, -6\frac{2}{3}) \).

\(\text{11}\) The blanks in the column of the permutations mean that we may arrange the players that are not listed in any way we want without affecting the outcome. This allows to group permutations resulting in the same outcome (and exhibiting the same \( l \)).