EURO Journal on Computational Optimization manuscript No.
(will be inserted by the editor)

Generalized Minimum Spanning Tree Games

**Phuoc Hoang Le · Tri-Dung Nguyen ·
Tolga Bektaş**

Submitted: 8 August 2014

Abstract The minimum-cost spanning tree game is a special class of cooperative games defined on a graph with a set of vertices and a set of edges, where each player owns a vertex. Solutions of the game represent ways to distribute the total cost of a minimum-cost spanning tree among all the players. When the graph is partitioned into clusters, the generalized minimum spanning tree problem is to determine a minimum-cost tree including exactly one vertex from each cluster. This paper introduces the generalized minimum spanning tree game and studies some properties of this game. The paper also describes a constraint generation algorithm to calculate a stable payoff distribution and presents computational results obtained using the proposed algorithm.

Keywords Generalized minimum spanning tree game · cost allocation · cooperative games · the core · stability

1 Introduction

The *minimum-cost spanning tree problem* (MSTP) is a well-known problem in network optimization, with a wide range of applications in communication, transportation, and computer networks. The standard MSTP is stated as follows: Given a weighted graph whose vertices might represent cities and whose edges serve as possible communication links with edge weights representing the cost of building a link or the length of the link, the aim is to select a set of communication links that would connect all the vertices such that the tree has the minimum total weight

P.H. Le (✉) · T.D. Nguyen · T. Bektaş
Southampton Business School and Centre for Operational Research, Management Science and Information Systems (CORMSIS), University of Southampton, Southampton, UK, SO17 1BJ
E-mail: P.H.Le@soton.ac.uk

T.D. Nguyen
E-mail: T.D.Nguyen@soton.ac.uk

T. Bektaş
E-mail: T.Bektas@soton.ac.uk

(Horowitz and Sahni 1978). We refer the reader to the work of Kruskal (1956) and Prim (1957) for further details and efficient algorithms to solve the problem.

The *minimum-cost spanning tree game* (MSTG) is defined on an undirected graph $G = (V, E)$ where V is the set of vertices and E is the set of edges. Each player owns a vertex and the payoff for any group of players is defined as the cost of a minimum-cost spanning tree of the subgraph corresponding to the group. Minimum-cost spanning tree games arise in cost allocation problems in which a joint enterprise can be represented as a tree that connects agents to a common source (e.g. Claus, 1973; Bird, 1976). MSTGs are cost sharing games where players need to be connected to a certain service supplier and form *coalitions*, (i.e., subsets of users/vertices) to share the cost of this service. When the cost incurred to any coalition is known, the question arises as how the total cost would be allocated among the users.

A *cost allocation problem* has the following features: there is a set $N = \{1, \dots, n\}$ of users (e.g., residents, companies, divisions in an organization, etc.) who cooperate in the context of a joint venture (e.g., Internet cable network, emergency systems, etc.). The problem is to allocate the cost of the joint venture among all the users in a way that satisfies criteria such as fairness, stability, efficiency, etc. Recently, cooperative game theory has been used to model various cost allocation problems (see, for example, Frisk et al. 2010; Fiestras-Janeiro et al. 2011). Cooperative game theory analyses the potential grouping of players to form coalitions. It also provides mathematical tools for calculating stable cost allocations in the sense that a stable share prevents the collapse of the grand coalition. We will later give a brief introduction to cooperative game theory, and refer the interested readers to the books of Chalkiadakis et al. (2011) and Maschler et al. (2013) for a more detailed exposition.

Let F be the set of all 2^n coalitions where n is the number of players. A *cooperative game in characteristic function form* is represented by a pair $(N; c)$, where and $c: F \to \mathbb{R}$ is a characteristic function with $c(R)$ being the cost incurred for a given coalition $R \subseteq N$. If the set of feasible coalitions is a strict subcollection of F such as when coalitions are the connected subgraphs of a communication graph (Myerson 1977) or when a hierarchy exist on the set of players (Faigle and Kern 1997), we refer the readers to Grabisch (2013) for some further results of these games. For each subset $R \subseteq N$, let $G(R) := (V(R), E(R))$ be the subgraph connecting all the vertices in the set $R \cup \{0\}$. The value $c(R)$ of a coalition $R \subseteq N$ is defined as the cost of a minimum-cost spanning tree on the subgraph $G(R)$, and the cost of the empty coalition is zero.

Among several solution concepts of cooperative game theory, the core is a set of cost allocation where no group of players has an incentive to deviate. The core consists of all allocation vectors $x = \{x_j\}_{j \in N}$ such that (i) $\sum_{j \in N} x_j = c(N)$, and (ii) $x(R) := \sum_{j \in R} x_j \leq c(R)$ for every coalition $R \subseteq N$. Vectors in the core are natural candidates for stable cost allocations in the sense that no subset of users has an incentive to leave the grand coalition. A game may have an empty core, but even if not, generating solutions in the core may be computationally very difficult. Granot and Huberman (1981) prove that the core of a minimum-cost spanning tree game is non-empty. Furthermore, a point in the core can be calculated directly from the minimum-cost spanning tree in the problem. Later, Faigle and Kern (1997) show that, checking if a given payoff distribution is a member of the core and computing the least core for the MSTG are NP-hard problems. This was done by transforming the separation problem: "Given the vector $x \in \mathbb{R}^n$ with $x(N) = c(N)$ decide whether there exists a coalition R such that $x(R) \geq c(R)$" into the exact cover by 3-Sets problem, which is NP-complete . We will provide the formal definition of the core and the least core in Section 2.2.

The *generalized minimum spanning tree problem* (GMSTP) is an extension of the MSTP and it was introduced by Myung et al. (1995). Given an undirected graph whose vertices are partitioned into a number of subsets (clusters), the GMSTP is to find a minimum-cost tree which includes exactly one vertex from each cluster. Myung et al. (1995) show that the GMSTP is strongly NP-hard by using a reduction from the vertex cover problem. The authors also present four integer linear programming formulations and a branch-and-bound algorithm to solve instances of up to 100 vertices. Feremans et al. (2002) and Pop (2009) describe twelve different formulations for GMSTP and study the relationships between the polytopes of their linear relaxations.

This paper introduces the *generalized minimum spanning tree game* (GMSTG) and proposes the computational methods for calculating its cost allocation. The GMSTG is related to the GMSTP in the same way as the MSTG is related to the MSTP. Our study of the GMSTG is motivated by the potential applications that this game finds in practice, two examples are given below:

- **Designing local area networks:** In this application, the aim is to connect a number of local networks via transmission links such as optical fibres as might be the case in metropolitan area networks (Gerla and Frata 1988) or regional area network (Prisco 1986). In this case we are seeking a minimum-cost spanning tree which selects exactly one vertex from each local network. Then, finding an optimal network design with the least cost is equivalent to solving a GMSTP while calculating the shared cost among the local areas is the GMSTG.
- **Water-supply distribution:** Consider the example shown in Fig. 1 in which a company supplying water in a particular region wishes to establish distribution hubs in different cities shown by A–G within the region. The hubs are connected via edges shown in bold representing the water flow, each with a fixed installation cost. There are several potential locations shown by vertices labelled 1–18 on which to place distribution hubs, but only one hub will be chosen in each city (player) depending on its power, capacity and position.

The contributions of this paper are: (i) to introduce GMSTG as a generalization of the MSTG and study its properties, and (ii) to describe a constraint generation algorithm to calculate stable payoffs. The rest of the paper is organized as follows. Section 2 formally defines the GMSTG and discusses some properties of the game. A solution algorithm based on constraint generation is described in Section 3. Section 4 presents computational results obtained with the proposed algorithm. Conclusions are stated in Section 5.

2 Generalized Minimum Spanning Tree Game

In this section, we first provide a formal definition of the GMSTG, following which we define the core and the least core.

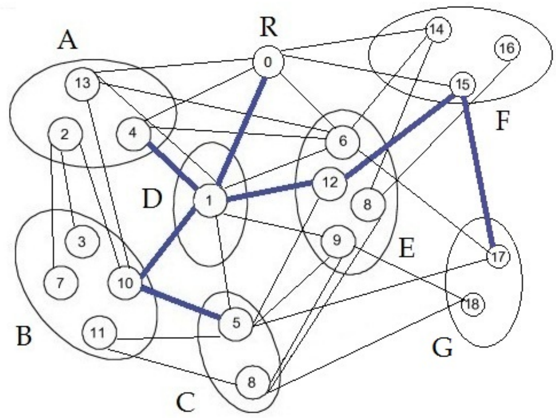

Fig. 1 An example of a generalized water-supply network graph and a solution of the GMSTP shown by bold lines

2.1 Definition

Let $G = (V, E)$ represent an undirected network. The vertex set $V = \{1, \ldots, m\}$ is partitioned into n clusters V_k with $k \in N = \{1, \ldots, n\}$ and the source vertex $\{0\} \notin V_k$ for all $k \in N$. The players in the GMSTG correspond to the n clusters $\{V_1, \ldots V_n\}$, which is in contrast with other cooperative games (e.g., Granot and Huberman 1981; Bergantinos and Gómez - Rúa 2015) where each vertex is a player. All players are assumed to be connected to the source.

The cost of connecting vertices i and j is denoted as d_{ij}, where $d_{ij} = 0$ for all $i, j \in V_k$ and $\forall k \in N$ (i.e., intra-cluster edges do not exist in the GMSTP)[1]. If the cost of an edge is large enough, then we expect that this edge will not stay in the graph. For a subgraph of some clusters (a coalition) $S \subseteq N$, consider the subnetwork $(V(S), E(S))$ of the source $\{0\}$ and other vertices in all clusters of S. Notice that coalition S must contain all vertices in a cluster and the shared cost $c(S)$ of such a coalition S is defined by the cost of a GMST of graph $(V(S), E(S))$.

[1] This paper supposes that intra-cluster edges do not exist and the players are different clusters of vertices. However, if intra-cluster edges do exist and that the cost must be allocated among the vertices of each cluster, it is possible to utilize the existing GMSTG model in a two-stage cost allocation scheme. Initially, the GMSTG is solved in the first stage to find the cost allocation to the clusters (e.g. cost allocation to the cities). In the second stage, the cost of each cluster is shared among the vertices (e.g. cost allocation to the subareas in each city). As the vertices in each cluster are connected, the cost of interconnecting these vertices is the same as the cost of the minimum spanning tree within that cluster, which is always a constant value no matter which vertex we choose in the cluster to form the GMST. Therefore, once the cost of the cluster has been calculated from the GMSTG, that cost can be distributed among the vertices by solving a MSTG on that cluster (with some scaling). The Bird rule (Bird 1976) could be used to efficiently find the cost allocation in the second stage.

2.2 The Core and Least Core of the GMSTG

In Section 2.3, we will show that the GMSTG has the super-additive property, i.e. a grand coalition is always formed. The question is how to share this total cost among all the players. One of the most popular solution concepts in cooperative game theory is the core, which can be given in the following form:

$$\mathbf{Core}(N,c) = \{x \in \mathbb{R}^n : \sum_{i\in N} x_i = c(N); \sum_{i\in S} x_i \leq c(S),\ \forall S \subset N\},$$

where $c(S)$ is the value of the GMSTP on subgraph $(V(S), E(S))$.

In the water-supply distribution example shown in Fig. 1, we consider a cost allocation vector x for seven different cities. For the cost allocation x to be in the core, any group of players $S \subseteq N$ must have no incentive to deviate to form a coalition themselves. In other words, the cost allocated $x(S)$ should be no larger than the cost incurred $c(S)$ by forming the coalition, i.e. $x(S) \leq c(S),\ \forall\ S \subseteq N$. When this condition is violated, the coalition is not happy with the cost allocation and the grand coalition is not stable.

The core, however, might be empty. If this happens, the condition of no sub-coalition can gain anything by deviating can be relaxed. Assuming that a sub-coalition will deviate from the grand coalition if the gain from the change is more than the cost of deviation, one can define the ϵ***-Core*** as follows:

$$\epsilon\text{-}\mathbf{Core}(N,c) = \{x \in \mathbb{R}^n : \sum_{i\in N} x_i = c(N); \sum_{i\in S} x_i \leq c(S) + \epsilon,\ \forall S \subset N\}.$$

Let $\epsilon^*(G) = \inf\{\epsilon > 0 : \epsilon\text{-}\mathbf{Core}$ of G is non-empty$\}$. The *least core* of G is its $\epsilon^*(G)$-Core and $\epsilon^*(G)$ is called the value of the least core of G. The least core can be found by solving the following linear programming problem

$$\begin{aligned} &\min_{\epsilon,x} \quad \epsilon \\ \text{s.t.} \quad &\epsilon \geq x(S) - c(S),\ \forall S \subset N, \\ &x(N) = c(N). \end{aligned} \tag{1}$$

One of the nice properties of the least core is non-emptiness and if the value of the least core problem (1) is zero then the core of the game exists and coincides with the least core.

For the MSTG, some heuristic methods have been proposed to calculate a cost allocation in the core such as the Bird rule (Bird 1976), the obligation rule (Tijs et al. 2006) and Folk solution (Bogomolnaia and Moulin 2010). The Bird rule of a MSTG on the graph G is described as follows. Initially, one has to find a minimum-cost spanning tree $\Gamma(G)$ by using Kruskal's or Prim's algorithms. Then the Bird allocation corresponding to $\Gamma(G)$ assigns for each player (vertex) the edge cost which connects that player with her immediate predecessor in $\Gamma(G)$. However, as we show in the example below, the Bird allocation cannot be applied to the GMSTG to find a cost allocation in the core.

Example 1 Consider a water supply network game with the root vertex $R = \{0\}$ and 3 players (cities) $N = \{A, B, C\}$, where the connection costs are shown in Fig.

2. Each player has one option for choosing the distribution hub, with the exception of player A who can choose between vertices $\{1\}$ or $\{2\}$. We assume that edges not shown in the graph do not exits, i.e., with infinity cost. The characteristic function for this game shown in Table 1.

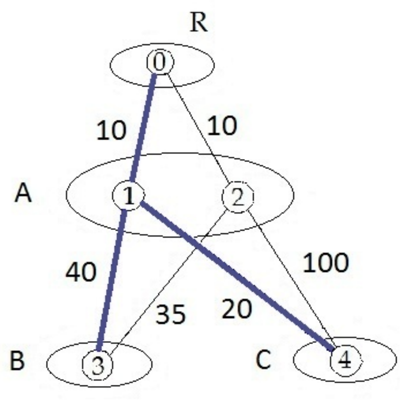

Fig. 2 A water distribution network with a solution of the GMSTP shown by bold lines

Coalition S	$\{\emptyset\}$	$\{A\}$	$\{B\}$	$\{C\}$	$\{A,B\}$	$\{A,C\}$	$\{B,C\}$	$\{A,B,C\}$
$c(S)$	0	10	45	30	45	30	70	70

Table 1 The characteristic function of water-supply network games with 3 players

The following set of inequalities fully describe the core of the GMSTG:

$$\begin{array}{lll} x_A \leq 10, & x_B \leq 45, & x_C \leq 30, \\ x_A + x_B \leq 45, & x_A + x_C \leq 30, & x_B + x_C \leq 70, \\ x_A, x_B, x_C \geq 0. & & \end{array} \tag{2}$$

Hence the core of this GMSTG is defined by

$$\textbf{Core}(N,c) := \{x \in \mathbb{R}^3 | x_A + x_B + x_C = 70 \text{ and inequalities (2) hold.}\}$$

We can quickly find a cost allocation such as $x = (5, 40, 25)$ which lies in the core. However, the solution $x = (10, 40, 20)$ yielded by the Bird rule is not in the core of the GMSTG, as it violates the constraint $x_A + x_B \leq 45$ in the inequality set (2).

One other important feature of the GMSTG is that, unlike the MSTG, its core might be empty. As an example, if we modify the game above such that the cost of the edge $\{2, 3\}$ that connects player A and player B is changed from 35 to 25, the core of the game becomes empty. This can be seen from the constraint set $x_A + x_B \leq 35$, $x_C \leq 30$ and $x_A + x_B + x_C = 70$, which does not have a solution.

Hence, the Bird algorithm for finding a good cost allocation in the MSTG does not apply to the GMSTG problem. Using the previous empty-core GMSTG in example 1, a similar result can be shown for other rules such as the obligation rule and Folk solution.

2.3 Properties of the GMSTG and its core

In this section, we present a number of properties of the GMSTG and its core. The first one concerns super-additivity.

Lemma 1 *GMSTG is a super-additive game, i.e.,* $c(S_1 \cup S_2) \leq c(S_1) + c(S_2)$ *for* $S_1, S_2 \subseteq N$ *and* $S_1 \cap S_2 = \{\emptyset\}$.

Proof Let S_1 and S_2 be two disjoint subsets of clusters and let $c(S_1), c(S_2), c(S_1 \cup S_2)$ be the values of the solutions of the GMSTP defined on graphs (V_{S_1}, E_{S_1}), (V_{S_2}, E_{S_2}) and $(V_{S_1 \cup S_2}, E_{S_1 \cup S_2})$ respectively. As we can always generate a spanning tree on $(V_{S_1 \cup S_2}, E_{S_1 \cup S_2})$ by combining the two spanning trees on two subgraphs (V_{S_1}, E_{S_1}) and (V_{S_2}, E_{S_2}) with a value equal to $c(S_1) + c(S_2)$, we have $c(S_1 \cup S_2) \leq c(S_1) + c(S_2)$ by the definition of minimum-cost spanning tree. □

Remark 1 Because the sum of the individual costs of the coalitions is no less than the cost of a union of disjoint coalitions, the grand coalition will be formed for the benefit of all players.

In the next part, we will define the optimal tree game (OTG) and show the relationship between its core and the core of GMSTG.

For each set of vertices $V' \subset V$ that contains at least one vertex Q'_k from each cluster V_k, $\forall k \in M \subseteq N$, we consider the subgraph $G' := (V', E(V'))$ of $G = (V, E)$. For each coalition of players $S \subseteq M$, let $V'(S)$ be the set of vertices in the subgraph G' belonging to the coalition S. The value $\widetilde{c}(S, V')$ is set as the optimal cost the MSTP on the subgraph $G'(S) := (V'(S), E(V'(S)))$. We then define the GMSTG generated by $\{0\} \cup \{V'\}$ and the cost function $\widetilde{c}(S, V')$ as the *GMST-subgame* $(V', \widetilde{c})$.

Assume we solve the GMSTP on $G = (V, E)$ and let $V^* := \{Q^*_1, Q^*_2, \ldots, Q^*_n\}$ be the set of vertices appearing in an *optimal tree* (OT) solution. The GMST-subgame $(V^*, \widetilde{c})$ is called as the *optimal tree game* (OTG) $(V^*, \widetilde{c})$ of graph G. Let $\{x_B\}$ be the cost allocation generated by the 'Bird rule' on the optimal tree game (Bird, 1976). If we reduce the graph G to contain only vertices in V^*, the GMSTG becomes the MSTG and the 'Bird rule' algorithm could be adapted to find a stable cost allocation. In what follows, we show a relationship between the cores of the OTG and the GMSTG.

Proposition 1 *The core of GMSTG* (N, c) *is a subset of the core of the optimal tree game* $(V^*, \widetilde{c})$.

Proof Assume the GMSTG-core is non-empty, i.e., there is a cost allocation $x = \{x_1, \ldots, x_n\}$ that satisfies all constraints in the core problem below:

$$\begin{aligned} & x(S) \leq c(S), \quad \forall S \subset N, \\ & x(N) = c(N). \end{aligned} \tag{3}$$

Consider the optimal tree game generated by $\{0\}$ and $\{V^*\}$ as the solution of the GMSTP on $G = (V, E)$; it is obvious that the vector x also satisfies all the constraints in core problem of the optimal tree game $(V^*, \widetilde{c})$. The definition $c(S)$ as the minimum-cost spanning tree in the graph $G(S) = (V(S), E(V(S)))$ for an arbitrary $S \subset N$ leads to the following:

$$c(S) = \min_{\{Q'_k\}_{k=1}^n \in \prod_{k=1}^n V_k} \widetilde{c}(S, \{Q'_k\}_{k=1}^n) \leq \widetilde{c}(S, \{Q^*_k\}_{k=1}^n).$$

For an arbitrary group of players $S' \subset N$, $\widetilde{c}(S', \{Q^*_k\}_{k=1}^n)$ is the cost that a coalition S' has to pay in the optimal tree game of subgraph $(S^*, E(S^*))$. Hence, $x(S') \leq c(S') \leq \widetilde{c}(S', \{Q^*_k\}_{k=1}^n)$ for all coalition S' of the optimal tree game and therefore the cost allocations $\{x_1, \ldots, x_n\}$ belongs to the core of the optimal tree game $(V^*, \widetilde{c})$.

□

The demonstration of the proper subset property for these two core sets is shown in Example 1, where the Bird allocation $x_B = \{10, 40, 20\}$ is in the core of the OTG $(V^*, \widetilde{c})$, but not in the core of GMSTG.

When defining the least core, the stability constraints in the definition of the core are relaxed. An alternative approach is to relax the feasibility constraint. This happens when an external party wishes to stabilize the game, by offering a subsidy amount Δ to the grand coalition if all players collaborate as a large group. More formally, given a super-additive game (N, c) and $\Delta \geq 0$, let $G^{\Delta} = (N, c^{\Delta})$ be cooperative game over the set of players N with the characteristic function given by $c^{\Delta}(N) = c(N) - \Delta$ and $c^{\Delta}(S) = c(S)$ for all $S \subset N$. Then, the core set of cost allocation for the game G^{Δ} is defined as follows:

$$\begin{aligned} \mathbf{Core}(N, c^{\Delta}) : x(S) &\leq c(S),\ \forall S \subset N, \\ x(N) &= c^{\Delta}(N). \end{aligned} \tag{4}$$

In general, the characteristic function c is not a monotone function. Fig. 3 shows a subgraph of previous water-supply network in Fig. 1 where the solution of the GMSTP defined on a coalition $M \subset N$ is not a subtree of the solution of the GMSTP defined on N. Moreover, $c(M) \geq c(N)$ might happen for such coalition $M \subset N$.

For every coalition $M \subset N$, we define Δ_M such that $\Delta_M \leq \Gamma_M := c(N \setminus M) + c(M) - c(N)$ and $\Delta_M \geq \Lambda_M := \max\{c(M) - c(N), 0\}$. From the GMST-subgame (M, c) we define a corresponding game (M, c^{Δ_M}) in in a similar manner as the previous part. The characteristic function of this game satisfies $c^{\Delta_M}(M) = c(M) - \Delta_M \leq c(M) - \Lambda_M \leq c(N)$ and $c^{\Delta_M}(S) = c(S),\ \forall S \subset M$. We can define the core set of the game (M, c^{Δ_M}) of players in coalition M by the following model:

$$\begin{aligned} \mathbf{Core}(M, c^{\Delta_M}) : x(S) &\leq c^{\Delta_M}(S),\ \forall S \subset M, \\ x(M) &= c^{\Delta_M}(M). \end{aligned} \tag{5}$$

After calculating the core of this subgame (M, c^{Δ_M}), for each of its cost allocation $x_M^{\Delta_M}$ we could build a class $\{x\}_G^{\Delta_M}$ of correspondent payoff distributions for

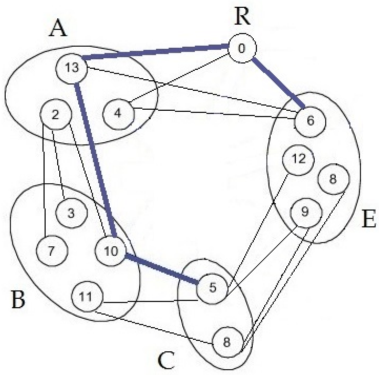

Fig. 3 An example of a subnetwork of the players M including clusters A, B, C, E reduced from the original graph shown in Fig. 1 and an optimal tree on this subgraph shown by the bold lines

the GMSTG (N, c) by using the amount $c(N) - c(M) + \Delta_M \geq 0$ divided arbitrarily to players in $N \setminus M$. This process give us a set of GMSTG cost allocations denoted by **C-set**(G, M). More formally,

$$\textbf{C-set}(G, M) := \bigcup_{\Gamma_M \geq \Delta_M \geq \Lambda_M} \{x\}_G^{\Delta_M}.$$

Remark 2 Notice that the core of GMSTG (N, c) is exactly the **C-set**(G, N).

Proposition 2 *For coalitions $S_2 \subset S_1 \subset N$, we have the relationship of **C-sets** as follows:*

$$\textbf{\textit{C-set}}(G, S_1) \subseteq \textbf{\textit{C-set}}(G, S_2).$$

Proof From the **C-set** definition, we have **C-set**$(G, S_1) = \bigcup_{\Delta_{S_1} \geq \Lambda_{S_1}} \{x\}_G^{\Delta_{S_1}}$. For each $\Delta_{S_1} \in [\Lambda_{S_1}, \Gamma_{S_1}]$, there exists a value Δ_{S_2} such that $\Gamma_{S_2} \geq \Delta_{S_2} \geq \Lambda_{S_2}$, and $\{x\}_G^{\Delta_{S_1}} \subseteq \{x\}_G^{\Delta_{S_2}}$.

To prove that, let x^* be the solution in $\{x\}_G^{\Delta_{S_1}}$, i.e., the subvector $x^*_{S_1} = \{x^*_i\}_{i \in S_1}$ is in core of the game $(S_1, c^{\Delta_{S_1}})$. Because $S_2 \subset S_1$, we can define $\Delta_{S_2} := c(S_2) - x^*(S_2) = c(S_2) - \sum_{i \in S_2} x^*_i$. For any coalition S that satisfies $S \subset S_2 \subset S_1$, we have $x^*(S) \leq c(S) = c^{\Delta_{S_2}}(S)$ and $x^*(S_2) = c(S_2) - \Delta_{S_2} = c^{\Delta_{S_2}}(S_2)$. Therefore, subvector $\{x^*_{S_2}\}$ is also in the core of the game $(S_2, c^{\Delta_{S_2}})$, i.e, x^* is the solution in $\{x\}_G^{\Delta_{S_2}}$. □

We now present another proposition showing the relationships between the GMSTG core and its related **C-sets** as follows:

Proposition 3 *The core of GMSTG (N, c) is the intersection of the **C-set**(G, M) where $M = N \setminus \{i\}$ for all clusters (players) $\{i\} \in N$, i.e.,*

$$\textbf{\textit{Core}}(N, c) = \bigcap_{i=1}^{n} \textbf{\textit{C-set}}(G, N \setminus \{i\}).$$

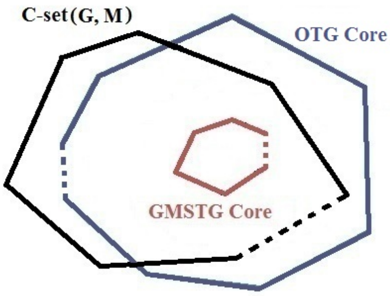

Fig. 4 The relationship among the GMSTG core, the OTG core and the C-set

Proof If there exists a cost allocation $\{x\}$ in the intersection of $\textbf{C-set}(G, N \setminus \{i\})$ with all $i \in N$, then $\{x\} \in \textbf{C-set}(G, M)$ for arbitrary coalition $M \subset N$ because of Proposition 2. This payoff distribution will satisfy constraints $x(M) = c^{\Delta_M}(M) \leq c(M)$ for arbitrary subsets $M \subset N$. Therefore, $x(S) \leq c(S)$ for all $S \subset N$ and $x(N) = c(N)$, i.e, $x \in \textbf{Core}(N, c)$.

We now assume $\{x'\}$ is a cost allocation belonging to the core of GMSTG (N, c). For arbitrary set $M \subset N$, we will prove that $x' \in \textbf{C-set}(G, M)$. Because $x'(M) \leq x'(N) = c(N)$ and $x'(M) \leq c(M)$, we set $\Delta'_M := c(M) - x'(M) \geq \max\{c(M) - c(N), 0\}$. For arbitrary $S \subset M \subset M$, $x'(S) \leq c(S) = c^{\Delta'_M}(S)$ and $x'(M) = c(M) - \Delta'_M = c^{\Delta'_M}(M)$. Therefore $x' \in \{x\}_G^{\Delta'_M} \subset \textbf{C-set}(G, M)$.

□

Remark 3 If there exists one coalition $M \subset N$ of players such that the $\textbf{C-set}(G, M)$ is empty, then the core of GMSTG (N, c) is also empty.

By solving the GMSTP, each general graph $G = (V, E)$ can always generate an optimal tree denoted as $\mathrm{OT}(G)$. We now consider the class $\Omega(N)$ of networks G such that each network has n clusters and the $\mathrm{OT}(G)$ has $p \geq 2$ edges incident to the source vertex 0 (denote as the p branches property of the optimal tree). Afterwards, p GMST-subgames $(N^i, \widetilde{c})$ are constructed for $i = \{1, \dots, p\}$. Note that as $\widetilde{c}(S) = c(S)$ for all $S \subseteq N^i$, the notation c is adopted instead of $\widetilde{c}$. We will prove that the core of the GMSTG (N, c) is a proper subset of the Cartesian product of the core of these p GMST-subgames.

The condition that a network $G = (V, E)$ with $\mathrm{OT}(G)$ has more than one edges incident to the source vertex 0 is equivalent to the existence of an partition of N clusters into subsets of clusters $\{N^1, \dots, N^p\}$ such that $c(N) = \sum_{i=1}^{p} c(N^i)$ (i.e., efficient coalition structure). An example of such a network partition is shown as follows:

For each branch $i \in \{1, \dots, p\}$ of the $\mathrm{OT}(G)$, let denote (N^i, c) as the GMST-subgame defined by the set of clusters $\{0\} \cup N^i$ and cost function $c(S, N^i) = c(S),\ \forall S \subset N^i$. The core of the GMST-subgame (N^i, c) is presented in following models:

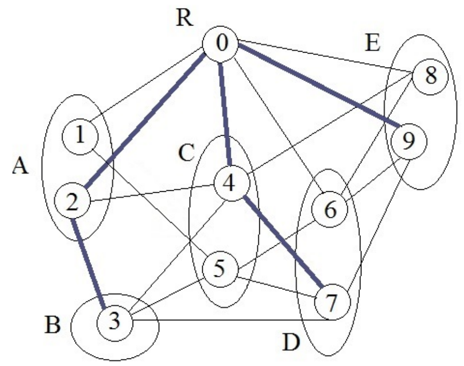

Fig. 5 A simple network in $\Omega(N)$ with OT(G) having $p = 3$ branches and an optimal tree shown by bold lines

$$\begin{aligned}\textbf{Core}(N^i, c) : x^i(S) &\le c(S),\ \forall S \subset N^i,\\ x^i(N^i) &= c(N^i),\end{aligned} \tag{6}$$

where n_i is the number of clusters in N^i and $x^i = \{x^i_j\}_{j=1}^{n_i}$.

Proposition 4 *If $\{N^1, \ldots, N^p\}$ is an efficient coalition structure of the GMSTG (N, c) with $p \ge 2$, the core of the GMSTG (N, c) is a proper subset of the Cartesian product of the cores of the GMST-subgame (N^i, c) for $i = 1, \ldots, p$, i.e.,*

$$\textbf{\textit{Core}}(N, c) \subset \prod_{i=1}^{p} \textbf{\textit{Core}}(N^i, c) \subset \textbf{\textit{C-set}}(G, N^i).$$

Proof Let $x^* = \{x^*_1, \ldots, x^*_n\}$ be a payoff vector in the core of GMSTG (N, c). We will prove that $x^* = \prod_{i=1}^{p} x^{*i}$, where $x^{*i} := \{x^*_j\}_{j\in N^i}$ is a cost payoff of the GMST-subgame (N^i, c). Because $\{N^1, \ldots, N^p\}$ is an efficient coalition structure of the GMSTG (N, c), then $c(N) = \sum_{i=1}^{p} c(N^i)$. Moreover, the definition of the core has the condition $x^*(N^i) \le c(N^i)$ for all $i = 1, \ldots, p$ and $x^*(N) = c(N)$. Hence, the inequality in sequence $x^*(N) = \sum_{i=1}^{p} x^*(N^i) \le \sum_{i=1}^{p} c(N^i) = c(N)$ becomes equality, and $x^*(N^i) = c(N^i)$ for all $i = 1, \ldots, p$. In particular, x^{*i} satisfies all constraints in the core formulation (6). Therefore, x^* belongs to the Cartesian product of the cores of the GMST-subgame (N^i, c) for $i = 1, \ldots, p$. To show that the inclusion is strict, we provide an example in Section 4.1, where $\textbf{Core}(N, c) \ne \prod_{i=1}^{p} \textbf{Core}(N^i, c)$.

Let y is a payoff vector in $\prod_{i=1}^{p} \textbf{Core}(N^i, c)$. For any $i \in \{1, \ldots, p\}$, we have $y(S) \le c(S),\ \forall S \subset N^i$ and $y(N^i) = c(N^i)$. Because of the efficient coalition structure property, $\sum_{i=1}^{p} y(N^i) = \sum_{i=1}^{p} c(N^i) = c(N)$. Combined with the definition of **C-set**, y is also a payoff vector in **C-set**(G, N^i). □

Remark 4 If a network G has an OT(G) such that there exists $i \in \{1, \ldots, p\}$ where $\textbf{Core}(N^i, c) = \emptyset$, then the core of the GMSTG (N, c) is empty. This property provides an effective way to check the emptiness of the core for large-sized GMSTGs.

3 Computational Methods for Finding the Core and the Least Core

The problem of finding a feasible cost allocation in the least core is difficult because its mathematical programming formulation has 2^n constraints, which would necessitate the solution of 2^n GMSTPs just for the purpose of obtaining the input of the resulting optimization problem. Therefore, brute-force techniques that attempt to solve 2^n GMSTPs and then solve an LP with 2^n constraints would be impractical to solve GMSTG instances when the number of clusters is $n \geq 8$. To overcome this difficulty, we now present a constraint generation algorithm to solve the GMSTG, and start by describing a formulation of the GMSTP that is used within the algorithm.

3.1 A multi-commodity GMSTP Formulation

Among the four existing GMSTP formulations presented by Myung et al. (1995), four formulations by Feremans et al. (2002) and four others in Pop (2009), we consider a multi-commodity flow formulation defined on a directed graph $D = (V, A)$ of the first paper. The motivation behind the choice of this particular model is its compact form (with a polynomial number of constraints) compared to other models (with an exponential number of constraints). Moreover, the formulation yields the best linear relaxation among others because of its polytope structure. Therefore it arises as a natural candidate for the construction of the exact algorithm for the separation problem in Section 3.2.

This model has a total supply $\sum_{i\in N} y_i$ at a source $\{0\}$ and individual demands y_i at each sink vertex $\{i\} \in V \setminus \{0\}$. As variables y_i are defined on vertices of V, we denote the total demands $\sum_{i\in V_k} y_i$ of vertices in a cluster V_k by $y(V_k)$. A binary variable w_{ij} is defined which equals 1 if and only if vertices i and j are connected, and 0 otherwise. Continuous variables f_{ij}^k are used as the amount of flow on arc (i,j) travelling from the source $\{0\}$ to the cluster V_k, for all $k \in N$.

This multi-commodity network flow formulation has a polynomial number of constraints and is shown in the following:

$$\mathbf{P_1}: \quad \min_{w_{ij}, y_i, f_{ij}^k} \sum_{(i,j)\in A} c_{ij} w_{ij} \tag{7a}$$

$$\text{s.t.} \quad \sum_{i\in V_k} y_i = 1, \ \forall k \in N \cup \{0\}, \tag{7b}$$

$$\sum_{i\in V_k} \sum_{j=1}^{m} w_{ji} \leq 1, \ \forall k \in N, \tag{7c}$$

$$\sum_{j=1}^{m} f_{ij}^k - \sum_{j=1}^{m} f_{ji}^k = \left\{ \begin{array}{cl} 1, & i = \{0\}, \\ -y_i, & \forall i \in V_k, \\ 0, & \text{otherwise} \end{array} \right\}, \forall k \in N, \tag{7d}$$

$$0 \leq f_{ij}^k \leq w_{ij}, \ \forall (i,j) \in A, \forall k \in N, \tag{7e}$$

$$y_i, w_{ij} \in \{0,1\}, \ \forall i,j \in V. \tag{7f}$$

The first set of constraints (7b) guarantees that only one vertex is chosen from each cluster V_k. The set of constraints in (7c) allows for at most one incoming arc to each cluster. The set of constraints (7d) ensures flow conservation at all intermediate vertices $\{i\} \in V$. The final set of constraints in (7e) models the relationship between the design and the flow variables.

3.2 Constraint Generation Algorithm

We now describe a constraint generation algorithm to solve the (least) core problem. The motivation for using this technique is to resolve two technical issues: (a) the LP formulation for solving the least core has an exponentially large number of constraints and (b) the input of this LP requires solving 2^n GMSTPs. The main idea behind this algorithm is to solve problem (1) only with a limited subset of constraints $E_0 \subset F$ at the beginning rather than solving the whole problem with all $|F| = 2^n$ constraints included and later add only some necessary constraints $x(S) \leq c(S) + \epsilon$ into the system. At each step t, the algorithm initially solves the following restricted problem:

$$\begin{aligned} \mathbf{LP_t^r}: \quad & \min_{\epsilon^t, x_i} \ \epsilon^t \\ \text{s.t.} \quad & \epsilon^t \geq x(S) - c(S), \ \forall S \in E_t \\ & x(N) = c(N). \end{aligned} \tag{8}$$

After solving problem (8), we obtain a new solution $x^t = (x_1^t, \dots, x_n^t)$ and the value ϵ^t of $\mathbf{LP_t^r}$. In the separation step, we identify the most violated constraint in least core problem (1) by solving the following optimization problem with a fixed vector x^t:

$$\mathbf{P_2}: \quad \min_{S \subseteq N; S \neq \emptyset} \ c(S) - x^t(S). \tag{9}$$

A coalition S is described by the binary n-vector $s = \{s_k\}_{k \in N}$, where $s_k \in \{0, 1\}$ depends on whether the player $\{k\}$ is a member of the coalition S. Applying the multi-commodity flow formulation $\mathbf{P_1}$ of $c(S)$ to problem (9), the separation problem can be written in the form of a MILP with binary variables s_k, continuous variables f_{ij}^k and binary variables w_{ij}, y_i as shown in the following:

$$\begin{aligned} \mathbf{P_3}: \quad & \min_{f_{ij}^k, w_{ij}, y_i, s_k} \ \sum_{(i,j) \in A} c_{ij} w_{ij} - \sum_{k \in N} s_k x_k^t \\ \text{s.t.} \quad & \sum_{i \in V_k} y_i = s_k, \ \forall k \in N \cup \{0\}, \\ & \sum_{i \in V_k} \sum_{j=1}^{m} w_{ji} \leq s_k, \ \forall k \in N, \\ & \sum_{j=1}^{m} f_{ij}^k - \sum_{j=1}^{m} f_{ji}^k = \left\{ \begin{array}{ll} s_k, & i = \{0\}, \\ -y_i, & \forall i \in V_k, \\ 0, & \text{otherwise} \end{array} \right\}, \forall k \in N, \\ & 0 \leq f_{ij}^k \leq w_{ij}, \ \forall (i,j) \in A, \forall k \in N, \\ & y_i, w_{ij} \in \{0, 1\}, \ \forall i, j \in V, \\ & s_k \in \{0, 1\}, \ \forall k \in N. \end{aligned} \tag{10}$$

Compared to $\mathbf{P_1}$, formulation $\mathbf{P_3}$ has a new set of variables s which changes both the objective function and most of the constraints. Hence, the new formulation becomes more difficult to solve.

After the separation step, if the optimality of the problem (1) is not reached, the solution s of $\mathbf{P_3}$ is used to generate a new constraint. In particular, suppose $\mathbf{P_3}$ has an optimal solution s^t, then the coalition $S^t = \{s_j^t\}$ is the one that receives the least negative excess. A new constraint in the form $\epsilon \geq x(S^t) - c(S^t)$ is then added to the relaxed problem (8), which is resolved to produce a new set x^{t+1} of variables, and the algorithm continues in this manner.

A pseudo-code of the constraint generation algorithm for calculating the GM-STG least core is given in Algorithm 1.

Algorithm 1 **Constraint Generation Algorithm**

Initialize: $maxiter = 10^4$; $\delta = 10^{-6}$; iteration $t = 1$;
  Constraint set $E_1 = E_0 \subset F$;
repeat
  Compute solution x^t and value ϵ^t of $\mathbf{LP^r_t}$;
  Find S^t in F such that:

$$S^t = \text{argmin}_S\{c(S) - x^t(S)\};$$

  $\overline{\epsilon}(t) = x^t(S^t) - c(S^t)$;
  $\underline{\epsilon}(t) = \epsilon^t$;
  if $|\overline{\epsilon}(t) - \underline{\epsilon}(t)| \leq \delta$ **then**
    stop;
  end if
  $E_t = E_t \cup \{S^t\}$;
  $t := t + 1$;
until $t = maxiter$.

To solve problem (9), we use CPLEX 12.5.0 with Matlab interface to obtain the optimal set S^t. This approach is useful if the process of finding the coalition S^t that minimizes $(c(S) - x^t(S))$ over F is fast enough. In fact, this is one of the reason why the multi-commodity flow model $\mathbf{P_1}$ is used as a basis for formulating the separation problem.

We also have some inequalities in step t as following:

$$\underline{\epsilon}(t) := \epsilon^t \leq \text{val}(LP(1)) \leq \overline{\epsilon}(t) := x^t(S^t) - c(S^t),$$

where $\text{val}(LP(1))$ is the optimal value of the corresponding least core linear program. For a small enough δ, this procedure terminates when $|x^t(S^t) - c(S^t) - \epsilon^t| \leq \delta$.

3.3 Other approaches for cost sharing

In cooperative game theory, there are a few alternative mathematical formulations for calculating stable solutions. Sometimes, an optimal solution of the least core problem is such that some players are disadvantaged. This may happen, for example, when there are large differences between the shares of pairs of players. One way to avoid such a situation is called an equal profit method (Frisk et al. 2010), which is used to minimise the differences between the ratios of cost share over individual cost for every pair of players. Therefore we consider a model to find a stable cost allocation, such that the maximum difference in pairwise relative costs is minimised. This model will take as input an optimal solution ϵ^* of the least core problem and is presented below:

$$\begin{aligned}
\hat{\mathbf{P}}(\epsilon^*): \quad & \min_{x,f} \quad f && \text{(11a)}\\
\text{s.t.} \quad & f \geq \frac{x_i}{c(\{i\})} - \frac{x_j}{c(\{j\})}, \quad \forall i,j \in N, && \text{(11b)}\\
& \sum_{i\in S} x_i \leq c(S) + \epsilon^*, \quad \forall S \subset N, && \text{(11c)}\\
& \sum_{i\in N} x_i = c(N), && \text{(11d)}\\
& x_i \geq 0, \quad \forall i \in N. && \text{(11e)}
\end{aligned}$$

The new model $\hat{\mathbf{P}}(\epsilon^*)$ is an LP, and could be solved by adapting the CGA described earlier for a fixed ϵ^*. Because of the definition of the least core ϵ^*, there always exits a cost allocation x^* which satisfies constraints (11c)– (11e). Therefore, the feasible region of problem (11) is non-empty. The difference of the new CGA algorithm is in the relaxation step, where we add n^2 constraints (11b) into the LP formulation at each step t of $\widehat{\mathbf{LP}}^{\mathbf{r}}_{\mathbf{t}}(\epsilon^*)$, given below:

$$\begin{aligned}
\widehat{\mathbf{LP}}^{\mathbf{r}}_{\mathbf{t}}(\epsilon^*): \quad & \min_{x,f_t} \quad f_t \\
\text{s.t.} \quad & f_t \geq \frac{x_i}{c(\{i\})} - \frac{x_j}{c(\{j\})}, \quad \forall i,j \in N,\\
& \sum_{i\in S} x_i \leq c(S) + \epsilon^*, \ \forall S \in E_t \\
& \sum_{i\in N} x_i = c(N),\\
& x_i \geq 0, \quad \forall i \in N.
\end{aligned} \tag{12}$$

The outcome of the algorithm will generate new payoff vector $x^{*\prime}$.

In the next section, we provide some computational results on the algorithm and formulation described above and present some comparative results. An illustrative example of the equal profit model will also be shown in Section 4.1.

4 Numerical Results

This section first presents results obtained with the constraint generation algorithm on a small-scale problem instance, and extends the results to a set of larger size instances. Our CGA algorithm was implemented in MATLAB and run on an Intel-core i5 PC with 2.6 GHz CPU and 4GB RAM.

4.1 An illustrative example

Example 2 **(Internet network)**

This example is motivated by an application in telecommunication networks, and concerns an internet provider located at vertex 0 wishing to establish its hubs (or gateways) at four different locations displayed as K, L, M and N, by setting up one hub in each city. Each city k will have some possible locations to set up its hub as shown in the network Fig 6. The internet provider can connect the potential hubs using optical fibres, where the costs between every pair of vertices are as shown in Table 2. For edges that do not exist in the graph, the corresponding cost in this table is shown as "—".

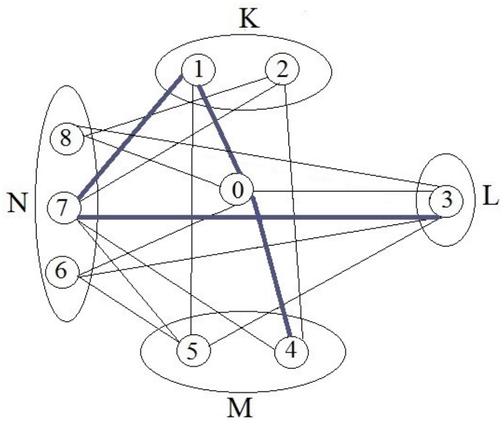

Fig. 6 An internet-cable network graph and the solution of GMSTP shown by bold lines

Vertices	0	1	2	3	4	5	6	7	8
0	0	89	—	514	114	—	385	—	315
1	89	0	0	—	—	296	—	40	—
2	—	0	0	—	47	—	—	358	347
3	514	—	—	0	—	326	457	80	44
4	114	—	47	—	0	0	—	306	—
5	—	296	—	326	0	0	252	331	—
6	385	—	—	457	—	252	0	0	0
7	—	40	358	80	306	331	0	0	0
8	315	—	347	44	—	—	0	0	0

Table 2 The cost matrix for the internet-cable network in Example 2

A coalition S of some players is the collaboration of a group of cities to minimize the network payments to the internet provider. The characteristic function of this game is defined such that $c(S)$ of a coalition S is the total cost of the minimum-cost spanning tree in this subgraph $S \cup \{0\}$. Solving the GMSTP for each coalition S, we obtain the characteristic function shown in Table 3:

Coalition S	$c(S)$	Coalition S	$c(S)$
$\emptyset$	0	$\{L, M\}$	628
$\{K\}$	89	$\{L, N\}$	359
$\{L\}$	514	$\{M, N\}$	420
$\{M\}$	114	$\{K, L, M\}$	675
$\{N\}$	315	$\{K, L, N\}$	209
$\{K, L\}$	603	$\{K, N, M\}$	243
$\{K, M\}$	161	$\{L, N, M\}$	473
$\{K, N\}$	129	$\{K, L, N, M\}$	323

Table 3 The characteristic function of an Internet cable network games with four players

The least core problem is presented as follows:

$$
\begin{aligned}
\mathbf{P_4} : \min_{x,\epsilon} \quad & \epsilon \\
\text{s.t.} \quad & x_K + x_L + x_M + x_N = 323, \\
& x_K \leq 89 + \epsilon, \quad x_L \leq 514 + \epsilon, \quad x_N \leq 315 + \epsilon, \quad x_M \leq 114 + \epsilon, \\
& x_K + x_L \leq 603 + \epsilon, \quad x_K + x_N \leq 129 + \epsilon, \quad x_K + x_M \leq 161 + \epsilon, \\
& x_L + x_N \leq 359 + \epsilon, \quad x_L + x_M \leq 628 + \epsilon, \quad x_M + x_N \leq 420 + \epsilon, \\
& x_K + x_N + x_M \leq 243 + \epsilon, \quad x_K + x_L + x_M \leq 675 + \epsilon, \\
& x_K + x_L + x_N \leq 209 + \epsilon, \quad x_L + x_M + x_N \leq 473 + \epsilon, \\
& x_K, x_L, x_M, x_N \geq 0.
\end{aligned}
\tag{13}
$$

In this example, the internet provider faces the GMSTP and GMSTG. Solving the GMSTP, the minimum-cost spanning tree is generated by vertices $(\{0\}, K : \{1\}, L : \{3\}, M : \{4\}, N : \{7\})$ as in Fig. 6. Using the previous CGA to find the core of the GMSTG, one has $\epsilon^* = 0$ (i.e., the core is non-empty) with the corresponding payoff vector $(0, 209, 114, 0)$. This means there is a stable cost allocation $x^* = (K : 0, L : 209, M : 114, N : 0)$ such that no city prefers to break the grand coalition. However, this cost allocation benefits players K and N as they have no share in the cost allocation. Using the equal profit method, we can calculate with another cost allocation $x^{*\prime} = (K : 20.26, L : 117.02, M : 114, N : 71.72)$ in the GMSTG core. This cost allocation is better with respect to relative costs of the players being proportionate to the individual cost.

Using the Bird rule, we can find a payoff vector $(K : 89, L : 80, M : 114, N : 40)$, which is not in the GMSTG core due to $x_K + x_M > 161$. However, as the OT(G) has 2 branches in Fig. 2, we could check that the payoff vector belongs to $\mathbf{Core}(\{M\}, c) \times \mathbf{Core}(\{K, N, L\}, c)$. Hence, we have the strict inclusion in the Proposition 4.

4.2 Results on larger instances

In this section, we present some computational results of the CGA to solve the GMSTG on randomly generated instances and GTSPLIB instances. The numbers of vertices for randomly generated instances range from 51 to 111 and the numbers of clusters range between 5 and 11. The inter-cluster edge costs are drawn from an uniform distribution of the range $[1, 1000]$ in a similar manner as in Golden et al. (2005).

Table 4 shows the results of CGA where column m is the total number of vertices in the graph and column n is the number of clusters. For each choice of (m, n) we generate 20 random instances, and each row of the table shows statistics averaged over 20 instances. In particular, *#Iter* is the number of iterations of the separation, *Ave.time per.Iter* is the average time (in seconds) per step and *CGA time* is the total time (in seconds) for solving the least core problem.

The total computation time for finding the least core is broken down into the time to solve the LP relaxation problems (column 5) and that to solve the separation step of CGA (column 6). Columns 5 and 6 show that the bottleneck is the separation step. The table also provides some comparisons between the CGA (*CGA time*) and a brute-force approach (*BF time*) for solving least core problem. For each problem configuration, column 9 shows the percentage of test instances where the least core value is greater than zero, i.e., when the core is empty. The last column of the table shows the average values of the normalized least core value $\bar{\epsilon}$ among all 20 random instances generated for each set of $\{m, n\}$. This value $\bar{\epsilon}$ is defined as the ratio of the least core value ϵ^* to the cost of the grand coalition.

Figure 7 shows the convergence of an experiment for a randomly generated instance with 101 vertices and 10 clusters for the least core problem using the constraint generation method, where the blue and red lines correspond to upper and lower bounds, respectively.

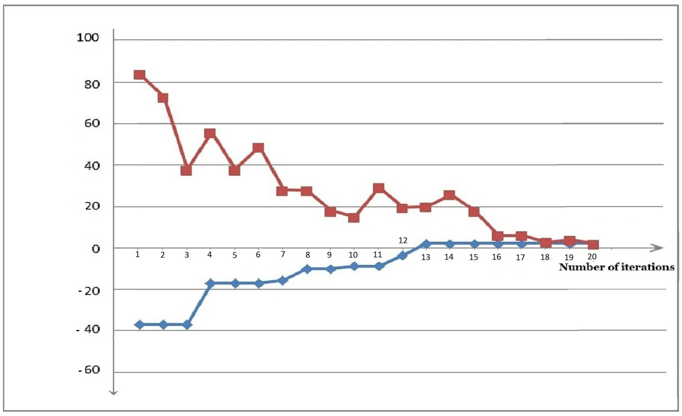

Fig. 7 Convergence of lower and upper bounds in the CGA for an instance with m=101 and n=10

From Table 4 and Fig. 7, the following observations can be made:

m	n	*#Iter*	*Ave.time per.Iter*	*LP time*	*Separation time*	*CGA time*	*BF time*	*%Empty core*	$\bar{\epsilon}$
51	5	6.25	3.97	0.03	21.70	24.83	54	10	0.045
	7	10.3	4.92	0.05	45.15	50.62	228	25	0.0088
71	5	5.07	15.27	0.03	72.24	77.43	101	20	0.022
	7	9.68	19.40	0.02	159.60	187.83	436	5	0.0038
91	5	5.53	35.72	0.05	163.97	197.55	217	0	0
	7	10.55	41.76	0.07	403.72	440.57	1616	15	0.0075
61	7	9.58	9.39	0.04	80.47	89.97	1116	15	0.0019
	9	15.25	13.09	0.12	163.83	199.63	6100	30	0.0079
81	7	10.24	20.95	0.05	181.32	214.51	1434	20	0.0061
	9	15.75	29.46	0.17	431.25	463.92	6178	15	0.0074
101	7	9.58	58.57	0.04	513.87	561.06	1998	10	0.0045
	9	13.91	125.03	0.22	1563.84	1739.19	10707	10	0.0045
71	9	14.4	28.82	0.21	419.52	415.06	5395	45	0.0225
	11	22.20	30.31	0.25	774.36	672.84	25840	35	0.0086
91	9	16.40	47.73	0.19	734.47	782.80	14255	10	0.006
	11	14.40	106.88	0.09	1385.38	1539.20	22047	30	0.0082
111	9	18.25	98.73	0.28	1651.14	1801.90	34736	20	0.0055
	11	24.60	197.34	0.29	4478.99	4854.60	49038	35	0.0138

Table 4 Computational time and the number of iterations involved in the Constraint Generation Algorithm

- As we increase the number of vertices but keep the number of clusters fixed, the average time for each iteration (with the main part of the separation step) increases as we would expect.
- When the number of clusters increases, but total number of vertices remains constant, the total number of iterations for the CGA increases.
- The number of iterations needed in CGA is significantly less than the total number 2^n of constraints required to solve problem (1).

In what follows, we present additional results to test the effect of the network density on the performance of the algorithm. The results are shown in Table 5 where the first two columns are the number of clusters (players) and number of vertices. For these experiments, we consider 10 instances for each tuple of (m, n, δ) where δ is the density of the network. Since we have four choices of δ, resulting in the generation of 40 random instances for each combination (m, n). Here, a density of $\delta = 0.25$ means the probability of having an inter-cluster edge between any two vertices of different clusters is 25% while a density of $\delta = 1$ means the network is fully connected. For intra-cluster edges which is not included in the graph, we set the corresponding variables in the formulation to zero.

The computational results indicate that when we fix the instance size but increase the density, the total time for CGA increases significantly. Moreover, the

m	n	δ	$\#Iter$	$Ave.time$ $per.Iter$	$Separation$ $time$	CGA $time$	BF $time$	$\%Empty$ $core$	$\bar{\epsilon}$
		0.25	1.1	9.44	2.73	10.38	62.17	0	0
51	5	0.5	1.8	7.46	5.48	13.44	42.32	0	0
		0.75	2.4	7.78	10.78	18.68	81.75	0	0
		1	4.2	5.08	19.15	21.34	58.27	10	0.0023
		0.25	1.7	7.36	3.72	12.32	125.79.3	0	0
51	7	0.5	2.5	6.11	7.16	15.28	131.65	0	0
		0.75	4.3	5.96	17.95	25.64	178.30	0	0
		1	8.7	6.68	51.17	58.15	341.62	30	0.0146
		0.25	1.0	26.29	5.24	26.29	87.53	0	0
71	5	0.5	1.7	19.0	10.61	32.3	121.65	0	0
		0.75	3.3	15.76	29.81	52.01	88.16	0	0
		1	5.4	20.86	88.32	112.67	171.49	30	0.0342
		0.25	1.2	19.95	5.21	23.84	453.12	0	0
71	7	0.5	3.9	11.57	25.39	45.14	491.63	0	0
		0.75	5.0	15.6	53.16	78.03	478.1	10	0.0031
		1	8.3	17.80	126.61	147.76	369.20	10	0.0076
		0.25	1.2	33.05	8.38	39.67	119.82	0	0
91	5	0.5	2.0	26.5	17.69	53.02	81.93	0	0
		0.75	3.1	25.16	45.95	78.01	103.19	0	0
		1	4.4	36.11	103.38	158.9	184.73	0	0
		0.25	1.2	60.65	12.07	72.78	513.63	0	0
91	7	0.5	2.4	38.02	36.82	91.24	561.16	0	0
		0.75	7.6	35.69	210.34	271.27	631.35	10	0.0022
		1	7.9	50.37	309.89	397.92	1826.79	20	0.0149

Table 5 Computation time of the least core with instances of different density.

core is likely to be non-empty when the edge cost matrix is less dense. The reason might be that there is fewer valid constraints in the core formulation of problem (1). For problem instances with the same number of players and the density, if the network graph has more vertices then the calculation time for separation problem increases, although the number of iterations in CGA does not change considerably.

For finding the least core using CGA, columns 4–6 show the computational statistics such as the average number of iterations, average time for each iteration and total time for the CGA. Column 7 presents the time required by the brute-force method to solve the least core problem. Comparing columns 6 and 7, one can see the computational advantages of CGA.

To see the algorithm's performance on more realistic instances, we performed tests on some GTSPLIB instances (Zverovich 2002) with the number of vertices $14 \leq m \leq 70$ and the number of clusters $3 \leq n \leq 14$. These instances are derived from TSPLIB instances where the vertex set has already been partitioned. However, they contain no source vertex, which is needed to model the GMSTG.

For this reason, we choose the last vertex of the last non-singleton cluster to be the source vertex so that the total number of clusters in the corresponding GMSTP instance is retained. For these instances, we also calculate the total cost of the grand coalition $c(N)$ to compute the normalized least core values. The results are presented in Table 6.

Problem name	m	n	*#Iter*	*Separation time*	*CGA time*	*BF time*	$c(N)$	$\bar{\epsilon}$
3burma14	14	3	1	0.06	0.31	0.45	9.2	0
4ulysses16	16	4	3	0.20	0.45	0.75	25.49	0
4gr17	17	4	2	0.21	0.51	0.84	736	0
5gr21	21	5	6	0.68	0.87	1.25	1160	0
5ulysses22	22	5	4	0.81	1.09	2.69	33.76	0
5gr24	24	5	4	1.20	1.61	2.38	259	0
6fri26	26	6	10	4.46	5.3	6.29	388	0
10att48	48	10	63	316.64	325.42	554.43	12725	0
11eil51	51	11	89	750.44	761.24	1240.7	134.9	0
11berlin52	52	11	80	717.96	730.21	1891.4	2938.4	0.00045
14st70	70	14	136	6210.23	6245.9	36713	243.1	0

Table 6 Computation time of the least core with instances of different density.

The results presented in Table 6 indicate that when the number of vertices is less than 71, the computational time of CGA speeds up as the number of players increases to 14. Indeed, the performance of the algorithm on GMSTG least core problem depends on both number of vertices and number of clusters. Combining all numerical results, it can be clearly seen that the CGA outperforms the brute-force method in terms of the computational time.

The only case of "11berlin52", where the core is empty, constitutes less than 10% of all the instances in Table 6. This result parallels those shown in Table 4 and 5, in which the %*Empty core* column shows that the core is empty in a relatively small percentage of all cases. Another general observation is that in case of empty-core instances, the least core value is less than 5% of the cost of the grand coalition. The computational results presented in Table $4--6$ collectively suggest a high probability of the existence of the core in GMSTGs, which is related to the theoretical property of the MSTG in which the core is always non-empty.

5 Conclusion and Future Work

In this paper, we have introduced a new class of cooperative games, namely the Generalized Minimum Spanning Tree Game, and proposed a constraint generation method to solve the least core problem. We have provided an example showing an empty core for the GMSTG. Numerical results have shown that randomly generated instances defined on complete graphs with up to 111 vertices and 11 clusters can be solved to optimality using the proposed method. We also tested some GTSPLIB instances with up to 70 vertices and with up to 14 clusters and observed the clear advantage of using the constraint generation algorithm. By introducing the new generalized minimum spanning tree game, its applications and an algorithm for calculating a stable cost share, we hope the new game will help to promote further practical usage of cooperative game theory in cost allocation problems.

Acknowledgement

Thanks are due to two anonymous reviewers for their valuable comments. The first author gratefully acknowledges the Faculty Scholarship provided by Southampton Business School.

References

Bergantiños G, Gómez-Rúa M (2015) An axiomatic approach in minimum cost spanning tree problems with groups. Annals of Operations Research 225(1):45–63

Bird CG (1976) On cost allocation for a spanning tree: A game theory approach. Networks 6:335–350.

Bogomolnaia A, Moulin H (2010) Sharing a minimal cost spanning tree: Beyond the Folk solution. Games and Economic Behaviour 69:238–248

Chalkiadakis G, Elkind E Wooldridge M (2011). Computational aspects of cooperative game theory. Morgan Claypool Publishers

Claus A, Kleitman DJ (1973) Cost allocation for a spanning tree. Networks 3:289–304

Dror M, Haouari M, Chaouachi J (2000) Generalized spanning trees. European Journal of Operational Research 120:583–592

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

Faigle U, Kern W (1992) The Shapley value for cooperative games under precedence constraints. International Journal of Game Theory 21:249–266

Faigle U, Kern W (1997) On the complexity of testing membership in the core of min-cost spanning tree games. International Journal of Game Theory 26:361–366

Feremans C, Labbé M , Laporte G (2002) A comparative analysis of several formulations for the generalized minimum spanning tree problem. Networks 39:29–34

Fiestras-Janeiro MG, Garcia-Jurado I, Meca A, Mosquera MA (2011) Cooperative game theory and inventory management. European Journal of Operational Research 210:459-466

Frisk M, Göthe-Lundgren M, Jörnsten K, Rönnqvist M (2010) Cost allocation in collaborative forest transportation. European Journal of Operational Research 205:448–458

Granot D, Huberman G (1981) Minimum cost spanning tree games. Mathematical Programming 21:1–18

Gerla M, Frata L (1988) Tree structured fiber optics MAN's. IEEE Journal. Selected Areas in Communications 6:934–943

Golden B, Raghavan S, Stanojević D (2005) Heuristic search for the generalized minimum spanning tree problem. INFORMS Journal on Computing, 17:290–304

Grabisch M (2013) The core of games on ordered structures and graphs. Annals of Operations Research 204:33–64

Horowitz E, Sahni S (1978) Fundamentals of Computer Algorithms. Computer Science Press, Maryland

Ihler E, Reich G, Widmayer P (1999) Class Steiner trees and VLSI-design. Discrete Applied Mathematics 90:173–194

Kruskal JB (1956) On the shortest spanning tree of a graph and the travelling salesman problem. Proceedings of the American Mathematical Society 7:48–50

Maschler M, Solan E, Zamir S (2013) Game Theory. Cambridge University Press, Cambridge

Myerson RB (1977) Graphs and cooperation in games. Mathematics of Operations Research 2:225–229

Myung YS, Lee CH, Tcha DW (1995) On the generalized minimum spanning tree problem. Networks 26:231–241

Pop PC(2009) A survey of different integer programming formulations of the generalized minimum spanning tree problem. Carpathian Journal of Mathematics, 25:104–118

Prim RC (1957) Shortest connection networks and some generalizations. Bell System Technical Journal 36:1389–1401

Prisco JJ (1986) Fiber optic regional area networks in New York and Dallas. IEEE Journal in Selected Areas in Communications 4:750–757

Tijs S, Branzei R, Moretti S, Norde H (2006) Obligation rules for minimum cost spanning tree situations and their monotonicity properties. European Journal Operational Research 175:121–134

Zverovich A (2002) GTSP instances. www.cs.rhul.ac.uk/home/zvero/GTSPLIB/