Radboud University Nijmegen

Faculty of Science

Minimum Cost Village Spanning Tree

Thesis BSc Mathematics

Supervisor:
dr. Wieb Bosma
Radboud University

Author:
Bas Janssen

Second reader:
prof. dr. Peter Borm
dr. Ruud Hendrickx
Tilburg University

June 2022

Acknowledgements

I want to begin my thesis by thanking Peter Borm and Ruud Hendrickx from Tilburg University for guiding me throughout my thesis. They have given me great feedback and helped me out when I was struggling with a proof. It has truly been a masterclass in writing a thesis. Furthermore, I want to thank Wieb Bosma from Radboud University for being there to talk to about how I was doing and feeling about my thesis.
Laslty, I want to thank my dear friend Michelle Jager. She was the person to go to when I was feeling down. It helped me to stay motivated and sane during the writing of my thesis.

Contents

1 Introduction

Consider the following situation: there is a municipality which consists of several houses, where every house needs to have electricity to be livable. that is why a big power station is being build. Every house will be connected to the power station via cables. A single house can be connected either directly or via other houses. The construction of these cables have to be paid by the residents. Some connections will be more expensive than others for several reasons. Think about the different distances and accessibility of the houses. that is why the residents want the construction of the cables to be in the cheapest way possible. Furthermore, the municipality needs to worry about the allocation of these costs. They want to divide the minimal costs among the houses such that every house pays a fair amount. This problem is related to a well-known problem called: **Minimum Cost Spanning Tree problem** or **mcst problem** in short.
There are two problems that need to be solved. The first problem is connecting every house to the power station such that the costs are minimal. This problem belongs to the field of operations research, a discipline that looks at methods to improve decision-making. In this case, the construction company needs to decide which cables to place to have the minimal costs. The solution to this problem is called a **Minimum Cost Spanning Tree** or **mcst** in short. The second problem is deciding how to allocate the costs among the houses in a fair way. This belongs to the field of game theory. To be more specific, cooperative game theory, since the houses can cooperate to minimize their costs. In general, cooperative game theory talks about players that can form different coalitions instead of cooperating houses.
The first problem can be solved using algorithm. An algorithm needs to give back the cheapest set of cables such that every house is connected. Two well-known algorithm that are used to solve mcst problems were introduced by Prim [9] and Kruskal [8]. The second problem can be solved using cooperative game theory. The allocation problem can reformulated as a cooperative game which is called the **mcst game**. Fair allocations will be the core elements of the mcst game. Core elements are the allocations among the houses where the total cost of the allocation equals the cost of the mcst and the cost of each group of houses is higher than the sum of the allocated cost of each house in the group.
Finding these core elements can be difficult when the amount of houses increase. Using the algorithms for finding a mcst can be helpful for finding a fair allocation. Bird [1] introduced an allocation method that uses Prim's algorithm and is called the Bird rule. Granot and Huberman [6] proved that this allocation lies in the core of the mcst game. The other allocation method uses Kruskal's algorithm and is called the Equal Remaining Obligations rule [10].
This thesis will also dive in a variation of a mcst problem. Consider the following situation: there is a municipality which consists of several villages where every village consists of several houses. Every house needs to have electricity to be livable and that is why a big power station gets build. Every house will be connected to the power station via cables. To get an optimal electricity supply, the municipality chooses to connect every house in a village with each other. Every house still needs to be connected to the power station which can be done directly or via other houses. The construction of the cables has to be paid by the residents. That's why the construction of the cables has to be in the cheapest way possible such that every house will be supplied of electricity and the houses in every village are connected with each other. Furthermore, the municipality needs to worry about the allocation of these costs. They want to divide the minimal costs among the houses such that every house pays a fair amount while keeping the villages in mind. This problem is called the **Minimum Cost Village Spanning Tree problem** or **mcvst problem** in short.

Also here, there are two problems that need to be solved. The first problem is finding the cheapest way to connect all the houses to the power station while making sure that every village is connected when only using the cables within the village. The solution to a mcvst problem is called a **Minimum Cost Village Spanning Tree** or **mcvst** and can be found using algorithms. Prim's and Kruskal's algorithm are used as a basis and changed in such a way that the villages are kept in mind. The second problem is allocating the costs among the houses. This time there are villages involved which might impact the way we look at which allocation is fair. This allocation problem can again be solved by defining a **mcvst game**. The Bird rule and the Equal Remaining Obligation rule will be used to find core elements. These rules will be using the adapted algorithms such that the villages are taken into account.
Before diving into minimum cost village spanning trees, prior knowledge is needed. This thesis starts with preliminaries about graph and game theory in chapter 2. Then mcst problems and games are mathematically explained in chapter 3. The different algorithms and the allocation methods attached to it will be discussed. After that, mcvst problems and games will be discussed in chapter 4.

2 Preliminaries

This chapter talks about the basic notions of graph theory and game theory. The master thesis of Moor [4] has been used as inspiration for graph theory and the paper of Borm [2] for game theory.

2.1 Graph theory

A **graph** G is a pair (V, E), where V is the set of **vertices** and E the set of **edges**. Both sets are finite and each edge $e \in E$ connects two vertices $u, v \in V$ which is denoted as $\{u, v\}$. In this case, u and v are called the endpoints of e. A **subgraph** $G' = (V', E')$ of a graph $G = (V, E)$ is a graph such that $V' \subset V$ and $E' \subset E$.
A graph is called **complete** if for every pair of vertices $u, v \in V$ with $u \neq v$, there exists an edge $e \in E$ with u and v as its endpoints. A complete graph is denoted as (V, E_V) with $E_V = \{\{u, v\} | u, v \in V, u \neq v\}$.
Let $G = (V, E)$ be a graph, a **path** is a finite sequence of edges that joins a sequence of distinct vertices. In other words, a path $(e_1, e_2, ..., e_{n-1})$ is a sequence of edges in E for which there exists a sequence of distinct vertices $(v_1, v_2, ..., v_n)$ in V such that $e_i = \{v_i, v_{i+1}\}$. This can also be called a path between v_1 and v_n since it connects the two vertices through $v_2, ..., v_{n-1}$. A path where $v_1 = v_n$ is called a **cycle**. G is called **connected** if there exists a path between every pair of vertices in V.
A graph is called a **tree** if it's connected and does not contain a cycle. Consequently, a tree has $\#V - 1$ edges. Let $G = (V, E)$ be a graph, a **spanning tree** of G is a subgraph $G' = (V', E')$ with $V' = V, E' \subset E$ and G' a tree.

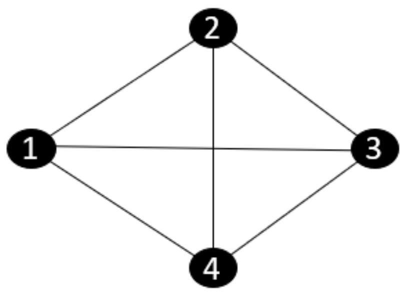

Figure 1: Complete graph of 4 vertices

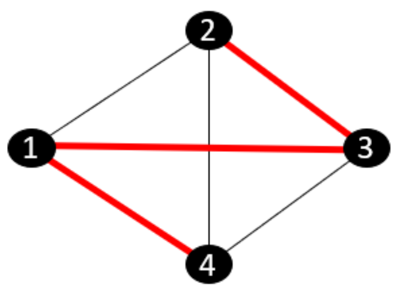

Figure 2: A path from vertex 2 to 4

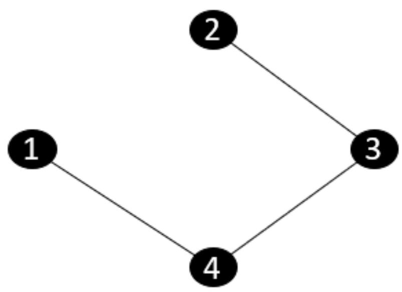

Figure 3: Spanning tree

2.2 Game theory

Let $N = \{1, 2, ..., n\}$ be a finite set of **players** and 2^N the set of all subsets of N. The elements of 2^N are called **coalitions** that the players can form. A **cost game** assigns a cost to each coalition $S \in 2^N$. Mathematically, a cost game is a pair (N, c) where $c : 2^N \to \mathbb{R}$ is called the **characteristic function** with $c(\emptyset) = 0$. The value $c(S)$ for every coalition $S \in 2^N$ is called the **cost** of coalition S. Consider a cost game (N, c) where the players want to allocate the total cost $c(N)$ in a fair way among each other. An allocation $x \in \mathbb{R}^N$ is a vector that assigns the cost x_i to player $i \in N$ for every player in N. To get a fair allocation, $x \in \mathbb{R}^N$ must satisfy two properties:

1. **Efficiency**: $\sum_{i \in N} x_i = c(N)$;
2. **Coalition rationality**: $\sum_{i \in S} x_i \leq c(S)$ for every coalition $S \in 2^N$.

The **core** of a cost game are all the allocations that satisfy these two properties. Formally, the core $C(c)$ of a cost game (N, c) is defined as

$$C(c) = \{x \in \mathbb{R}^N | \sum_{i \in N} x_i = c(N), \sum_{i \in S} x_i \leq c(S) \text{ for all } S \in 2^N\}.$$

Example 2.1. Let (N,c) be a cost game such that $N = \{1, 2, 3\}$, $c(\{1\}) = 4$, $c(\{2\}) = 8$, $c(\{3\}) = 6$, $c(\{1, 2\}) = 9$, $c(\{1, 3\}) = 8$, $c(\{2, 3\}) = 10$ and $c(\{1, 2, 3\}) = 11$. The fair allocations $x \in \mathbb{R}^3$ are the core elements of this cost game. This means that the following restrictions must hold for these allocations:

$$\begin{gathered} x_1 \leq 4,\ x_2 \leq 8,\ x_3 \leq 6 \\ x_1 + x_2 \leq 9 \implies x_3 \geq 2 \\ x_1 + x_3 \leq 8 \implies x_2 \geq 3 \\ ,\ x_2 + x_3 \leq 10 \implies x_1 \geq 1 \\ x_1 + x_2 + x_3 = 11 \end{gathered}$$

This results in $C(c) = \text{Conv}\{(4, 3, 4), (4, 5, 2), (1, 8, 2), (2, 3, 6), (1, 4, 6)\}$.

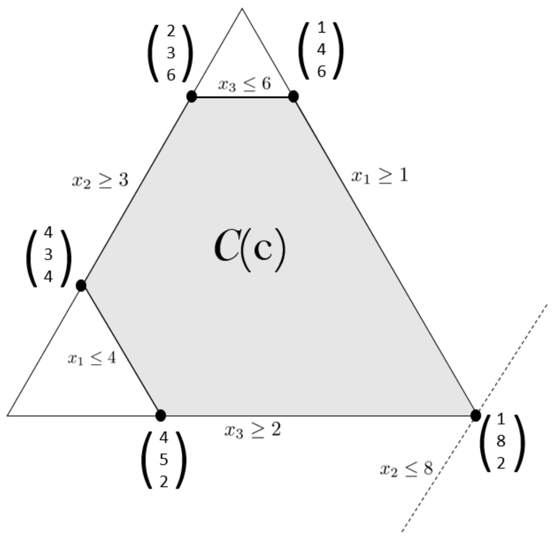

Figure 4

3 Minimum Cost Spanning Tree Problems and Games

This chapter discusses the mcst problems and games in more detail. It revolves around the algorithms to find the minimum cost such that every house is connected to the source and how we allocate the minimal cost in a fair way. Here, the master thesis of Moor [4], the paper of Borm [2] and the article from Borm, Hamers and Hendrickx [3] have been used as inspiration.

3.1 Mcst problem

The first part of the mcst problem is all about finding the minimum cost spanning tree. This means, finding the cheapest way to connect every house to the power station. From now on, this thesis talks about vertices instead of houses and the source instead of the power station.

Definition 3.1. A *mcst problem* is a triple $\mathcal{T} = (N, \star, t)$, where $N = \{1, 2, ..., n\}$, $\star$ is the source and $t : E_{N\cup\{\star\}} \to \mathbb{R}_+$ is the cost function that gives a non-negative cost to each edge in $E_{N\cup\{\star\}}$. For here on out, $N \cup \{\star\}$ is written as $N^\star$.

A solution to a mcst problem is called a mcst (minimum cost spanning tree) which must satisfy the following restrictions:

i. $(N^\star, R)$ is a tree;

ii. $t(R) = min\{t(S) : (N^\star, S) \text{ is a tree}\}$ where $t(S) = \sum_{s\in S} t(s)$

Remark. A mcst does not have to be unique since the costs of different edges can be the same. This means that a mcst problem might have more solutions with the same minimum cost.

Example 3.2. Suppose that there are three houses that need electricity from the power station. The houses are connected to the power station via cables which can be done either directly or via other houses. The cost of every cable can be seen in figure 5. The vertices represent the houses and the star represents the power station. Figure 6 shows a mcst which has a total cost of 18. So the minimal cost to connect every house to the power station is 18.

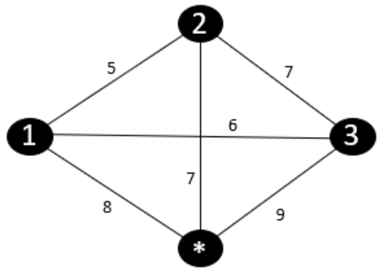

Figure 5: mcst problem

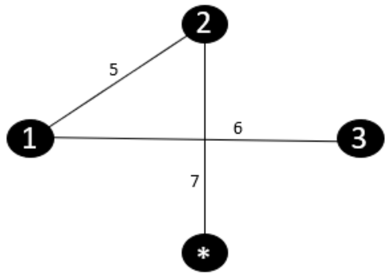

Figure 6: mcst

It was easy to find a mcst in this example, but it becomes harder when the amount of vertices increase. This is where algorithms come into play. The two well-known algorithms to solve mcst problems are Prim's and Kruskal's algorithm.

3.1.1 Prim's algorithm

Given a mcst problem $\mathcal{T} = (N, \star, t)$, Prim's algorithm starts by looking at all outgoing edges from the source and takes the edge with the minimal cost. Let I be the set of vertices that are connected to the source. The set I only contains $\star$ at the start of the algorithm. When this first edge is chosen, a vertex is added to set I. The algorithm continuous by looking at all edges $e = \{i, j\}$ with $i \in I$ and $j \notin I$. The edge with the minimum cost is chosen and the endpoint is added to I. It must be checked that every chosen edge does not introduce a cycle with the edges already chosen. The algorithm stops when all vertices are connected to the source.

Algorithm 3.1. Let $\mathcal{T} = (N, \star, t)$ be a mcst problem. Then the output is an edge set $R \subset E_{N^\star}$ of an mcst obtained as followed:

1. *Initialise* $R = \emptyset$ *and* $I = \{\star\}$;
2. *Find a minimal cost edge* $e_j = \{i, j\}$ *with* $i \in I$ *and* $j \notin I$ *in such a way that joining* e_j *to* R *does not introduce a cycle*;
3. *Join* e_j *to* R, j *to* I;
4. *If* $I \neq N^\star$, *go back to step 2.*

Example 3.3. Reconsider example 3.2 and use Prim's algorithm to solve it. The algorithm starts with the source and adds edge $\{\star, 2\}$ to R and 2 to I. Then it looks at all the outgoing edges from I and adds minimum cost edge $\{1, 2\}$. Lastly, it adds edge $\{1, 3\}$ since it is the cheapest edge that does not create a cycle. It connects the last remaining vertex to the source. This returns a mcst with edge set $\{\{\star, 2\}, \{1, 2\}, \{1, 3\}\}$. It becomes more clear when looking at figure 7.

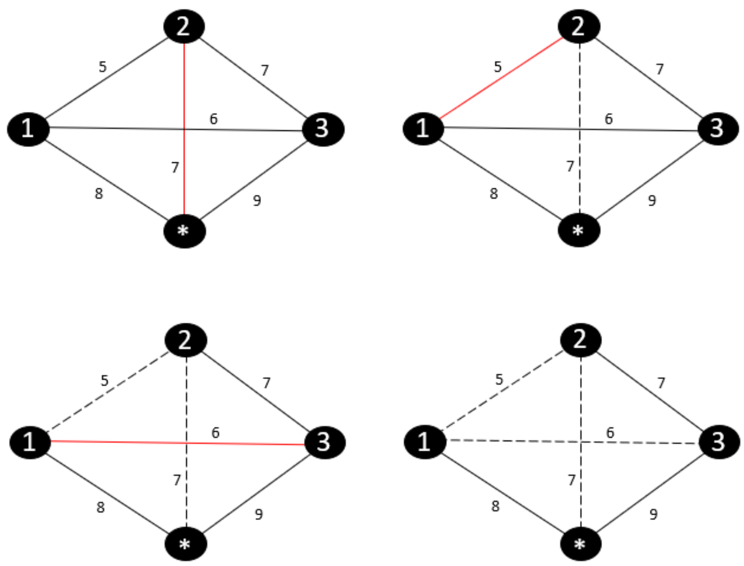

Figure 7

3.1.2 Kruskal's algorithm

Given a mcst problem $\mathcal{T} = (N, \star, t)$, Kruskal's algorithm keeps adding the minimum cost edge that has not been chosen. In every step, the cheapest edge is only chosen if it does not introduce a cycle with the edges already chosen. The algorithm stops when all vertices are connected to the source.

Algorithm 3.2. Let $\mathcal{T} = (N, \star, t)$ be a mcst problem. Then the output is an edge set $R \subset E_{N^\star}$ of an mcst obtained as followed:

1. *Initialise* $R = \emptyset$;
2. *Find a minimal cost edge* $e \in E_{N^\star} \backslash R$ *in such a way that joining* e *to* R *does not introduce a cycle*;
3. *Join* e *to* R;
4. *If* $(N^\star, R)$ *is not connected, go back to step 2.*

Example 3.4. Lets look at example 3.2 again and use Kruskal's algorithm to solve it. The algorithm just starts with adding the cheapest edge which is $\{1, 2\}$ to R. Then it looks at the next cheapest edges which is $\{1, 3\}$ and adds it. Lastly, edges $\{\star, 2\}$ and $\{2, 3\}$ will be the two next cheapest edges but $\{2, 3\}$ will create a cycle so $\{1, 2\}$ is added. This results in a mcst with edge set $\{\{\star, 2\}, \{1, 2\}, \{1, 3\}\}$. It becomes more clear when looking at figure 8.

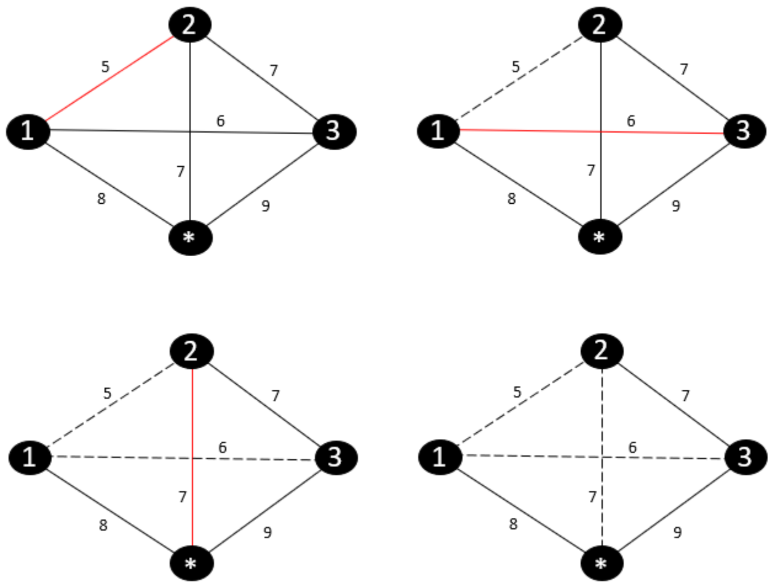

Figure 8

Theorem 3.5. *Let* $\mathcal{T} = (N, \star, t)$ *be a mcst problem. Then Prim's and Kruskal's algorithm give back an edge set* R *such that* $(N^\star, R)$ *is a mcst.*

The proof that Prim's and Kruskal's algorithm work, is given in Prim [9] and Kruskal [8] respectively.

3.2 Mcst game

The second part of the mcst problem is about allocating the minimal cost of a mcst. Here, the vertices are considered to be the players which can form coalitions. These coalitions will be given a certain cost given by a mcst game. Before the cost of each coalition is known, a mcst game has to be defined.

To every mcst problem $\mathcal{T} = (N, \star, t)$ is a mcst game $(N, c^{\mathcal{T}})$ associated where $c^{\mathcal{T}}(S)$ represents the value of the mcst of the graph $(S^\star, E_{S^\star})$:

$$c^{\mathcal{T}}(S) = min\{t(R) : R \subset E_{S^\star} \text{ and } (S^\star, R) \text{ is a tree}\}.$$

Here, $c^{\mathcal{T}}(S)$ is the minimal cost such that all players in coalition S are connected to the source. You can see this as the cost of the mcst of the mcst problem $(S, \star, t|_{E_{S^\star}})$.

Example 3.6. Reconsider example 3.2 and determine the mcst game associated to this mcst problem. The mcst game $(N, c^{\mathcal{T}})$ is:

S	$\{1\}$	$\{2\}$	$\{3\}$	$\{1,2\}$	$\{1,3\}$	$\{2,3\}$	$\{1,2,3\}$
$c^{\mathcal{T}}(S)$	8	7	9	12	14	14	18

Lets first start by looking at the coalition with only player 1. It can only be connected to the source by edge $\{\star, 1\}$ which costs 8. This goes the same way with player 2 and 3 which cost 7 and 9 respectively. When looking at a coalition of two players, it's important to understand that you cannot use all the edges. The coalition of $\{1,2\}$ can only use the edges $\{\star, 1\}$, $\{\star, 2\}$ and $\{1,2\}$. The cheapest way to connect player 1 and 2 to the source using these edge is by using $\{\star, 2\}$ and $\{1,2\}$. That is why coalition $\{1,2\}$ costs $5 + 7 = 12$. Doing this for the coalitions $\{1,3\}$ and $\{2,3\}$, it gives back the cost 14 for both. The cost of the coalition of all vertices $\{1,2,3\}$ $(= N)$ is the cost of the mcst in figure 6 which is 18.

3.2.1 Bird rule

Every time an edges is added in Prim's algorithm, a new player is connected to the source. This is used in the Bird rule to allocate the cost. When an edge causes a new player to be connected to the source, the new player gets the cost of that edge. It becomes more clear when looking at Prim's algorithm, where the Bird rule is added.

Algorithm 3.3. Let $\mathcal{T} = (N, \star, t)$ be a mcst problem. Then the output is an edge set $R \subset E_{N^\star}$ of an mcst and its corresponding Bird allocation $\beta^R(\mathcal{T})$ obtained as followed:

1. *Initialise* $R = \emptyset$ *and* $I = \{\star\}$;
2. *Find a minimal cost edge* $e_j = \{i, j\}$ *with* $i \in I$ *and* $j \notin I$ *in such a way that joining* e_j *to* R *does not introduce a cycle*;
3. *Join* e_j *to* R, j *to* I *and assign the cost* $\beta_j(\mathcal{T}) = t(e_j)$ *to* j ;
4. *If* $I \neq N^\star$, *go back to step 2.* .

Remark. When the algorithm is finished, we have an edge set R and an allocation $\beta(\mathcal{T})$. Since R does not have to be unique and $\beta(\mathcal{T})$ is dependent R, $\beta(\mathcal{T})$ does not have to be unique. When the algorithm is finished, the Bird allocation is written as $\beta^R(\mathcal{T})$ since then R is known.

Example 3.7. Reconsider 3.2 and use Prim's algorithm with Bird rule to find the mcst and allocate the minimum cost. The algorithm starts with adding edge $\{\star, 2\}$ to R which connects player 2 to the source. This means that $\beta_2(\mathcal{T}) = 7$. Then it adds edge $\{1, 2\}$ which connects player 1 to the source and $\beta_1(\mathcal{T}) = 5$. Lastly, edge $\{1, 3\}$ is added and the last vertex is connected to the source. This causes $\beta_3(\mathcal{T}) = 6$ and makes the Bird allocation vector complete with $\beta^R(\mathcal{T}) = (5, 7, 6)$. It might become more clear when looking at figure 9. The vertices that are connected to the source are made white.

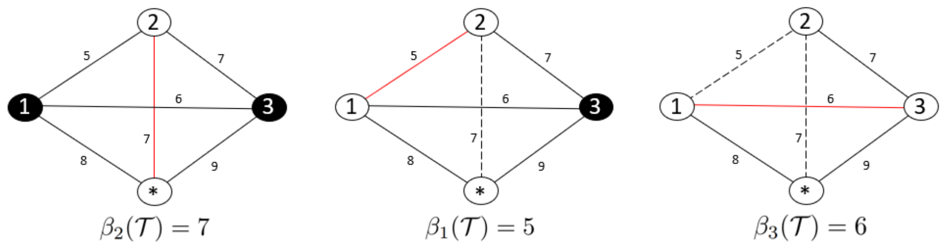

Figure 9

3.2.2 Equal Remaining Obligation rule

The equal remaining obligation rule means that every players has to pay a total of one unit of all chosen edges by Kruskal's algorithm. This could be in fractions where a player has to pay $\frac{1}{4}$ for one edge and $\frac{3}{4}$ for another. Determining what fraction of an edge each player pays will be as followed. Whenever a new edge is added by Kruskal's algorithm, a new player is added to a component of players. All players in the newly formed component have to pay an equal part of the new edge that has been added. A obligation vector obl^k is used where k stands for the k-th step. We talk about the k-th step when the k-th edge is added. In every step of Kruskal's algorithm, obl_i^k keeps track of how many players are connected to player i including itself. If player i is connected to the source, $obl_i^k = 0$. Else, obl_i^k will equal one over the amount of players that player i is connected with including itself. Furthermore, $O^k(\mathcal{T})$ is the cost contribution vector at step k of a mcst problem $\mathcal{T}$. This vector keeps track of what every player has to pay at step k. The cost contribution is notated as $O^R(\mathcal{T})$ when the algorithm is finished and an edge set R is returned.

Example 3.8. Reconsider 3.2 and use Kruskal's algorithm with the equal remaining obligation rule to find the mcst and allocate the minimum cost. It starts with $R = \emptyset$, $obl^0 = (1, 1, 1)$ and $O^0(\mathcal{T}) = (0, 0, 0)$. The algorithm begins with adding edge $\{1, 2\}$ to R which connects player 1 and 2 to each other. Since they are not yet connected to the source: $obl_1^1 = obl_2^1 = \frac{1}{2}$ and because player 3 is not connected to everything: $obl_3^1 = 1$. Since $obl^0 - obl^1 = (\frac{1}{2}, \frac{1}{2}, 0)$, players 1 and 2 have to pay half of $t(\{1, 2\}) = 5$ and player 3 has to play nothing. Furthermore, $O^1(\mathcal{T}) = O^0(\mathcal{T}) + (obl^0 - obl^1)t(\{1, 2\}) = (0, 0, 0) + (\frac{1}{2}, \frac{1}{2}, 0)5 = (2\frac{1}{2}, 2\frac{1}{2}, 0)$.

The next edge that will be added is $\{1, 3\}$ which connects all players to each other and nodboy is connected to the source. This means that $obl^2 = (\frac{1}{3}, \frac{1}{3}, \frac{1}{3})$ which leads to $obl^1 - obl^2 = (\frac{1}{2}, \frac{1}{2}, 1) - (\frac{1}{3}, \frac{1}{3}, \frac{1}{3}) = (\frac{1}{6}, \frac{1}{6}, \frac{2}{3})$. This means that player 1 and 2 have to pay $\frac{1}{6}$ of $t(\{1, 3\}) = 6$ and player 3 has to pay $\frac{2}{3}$ of 6. This gives us $O^2(\mathcal{T}) = (2\frac{1}{2}, 2\frac{1}{2}, 0) + (\frac{1}{6}, \frac{1}{6}, \frac{2}{3})6 = (3\frac{1}{2}, 3\frac{1}{2}, 4)$.

Lastly, edge $\{\star, 2\}$ will be added. All the players will be connected to the source, which causes $obl^3 = (0,0,0)$ such that $obl^2 - obl^3 = (\frac{1}{3}, \frac{1}{3}, \frac{1}{3})$. Every player has to pay equal part of $t(\{\star, 2\}) = 7$ which is $\frac{1}{3} \cdot 7 = 2\frac{1}{3}$. This leads to the final cost contribution vector $O^3(\mathcal{T}) = O^R(\mathcal{T}) = (3\frac{1}{2}, 3\frac{1}{2}, 4) + (\frac{1}{3}, \frac{1}{3}, \frac{1}{3})7 = (5\frac{5}{6}, 5\frac{5}{6}, 6\frac{1}{3})$.

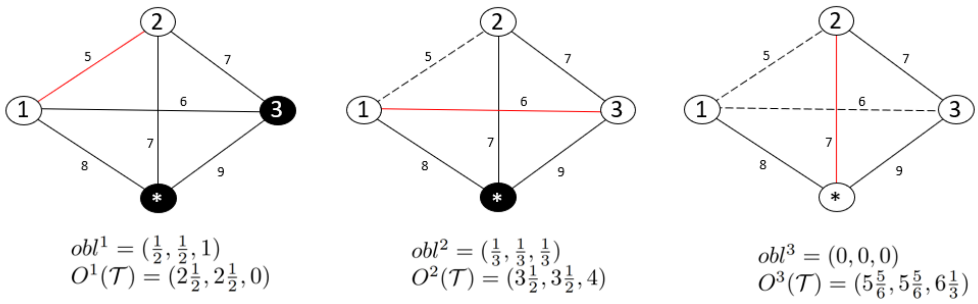

Figure 10

Theorem 3.9. *Let $\mathcal{T} = (N, \star, t)$ be a mcst problem and (N, c) the corresponding mcst game. Then the Bird allocation vector and the cost contribution vector are elements of the core $C(c)$.*

The proof that the Bird allocation vector and the cost contribution vector are core elements, is given in Granot and Huberman [6] and Tijs, Brânzi, Moretti and Norde [10] respectively.

4 Minimum Cost Village Spanning Tree

This chapter discusses the mcvst problems and games in more detail. It revolves around the algorithms to find the minimum cost such that every house is connected to the source while taking the villages into account. Furthermore, allocating the the minimal cost in a fair way with the restriction of the villages is analyzed.

4.1 Mcvst problem

The first part of the mcvst problem is about finding the minimum cost village spanning tree. This means finding the cheapest way to connect every vertex to the source while keeping villages into account.

Definition 4.1. A *mcvst problem* is a quadruple $\mathcal{T}_{\mathcal{V}} = (N, \star, t, \mathcal{V})$, where $N = \{1, 2, ..., n\}$, $\star$ is the source and $t : E_{N^\star} \to \mathbb{R}_+$ is the cost function that gives a non-negative cost to each edge in $E_{N^\star}$. The village set $\mathcal{V}$ is a partition of N which means that for $\mathcal{V}$ must hold that:

i. $V \neq \emptyset \quad \forall V \in \mathcal{V}$;

ii. $\displaystyle\bigcup_{V \in \mathcal{V}} V = N$;

iii. $V \cap W = \emptyset \quad$ for all $V \neq W$ in $\mathcal{V}$.

Furthermore, a solution for this problem has to be defined. These solutions are called minimum cost village spanning trees or mcvst in short. For a mcvst must hold that all the villages are connected. This means that a graph can only be valid to be a mcvst if it meets this requirement.

Definition 4.2. Let $\mathcal{T}_{\mathcal{V}} = (N, \star, t, \mathcal{V})$ be a mcvst problem, a graph $(N^\star, R)$ is called a *valid solution* if:

i. $(N^\star, R)$ is a tree;

ii. $(V, R \cap E_V)$ is connected $\forall V \in \mathcal{V}$.

Now the solution for a mcvst problem can be defined.

Definition 4.3. Let $\mathcal{T}_{\mathcal{V}} = (N, \star, t, \mathcal{V})$ be a mcvst problem, a graph $(N^\star, R)$ is a *mcvst* if:

i. $(N^\star, R)$ is a valid solution;

ii. $t(R) = min\{t(S) : (N^\star, S) \text{ is a valid solution}\}$ where $t(S) = \displaystyle\sum_{s \in S} t(s)$.

Remark. A mcvst problem $\mathcal{T}_{\mathcal{V}} = (N, \star, t, \mathcal{V})$ is the same as a mcst problem when the village set only consists of one village (i.e., $\mathcal{V} = \{N\}$) or when the village set consists of $\#N$ villages (i.e., $\mathcal{V} = \{\{i\} | i \in N\}$).

Example 4.4. Suppose that we have eight houses ($N = \{1, 2, ..., 8\}$), a source $\star$ and three villages with houses 1,3,4 in one village, 2,5,8 in another village and 6,7 in the remaining village or in short $\mathcal{V} = \{\{1, 3, 4\}, \{2, 5, 8\}, \{6, 7\}\}$. We define the cost function $t : E_{N^\star} \to \mathbb{R}_+$ as $t(e) = 0 \quad \forall e \in E_{N^\star}$. One can easily check that $\mathcal{V}$ satisfies the conditions in Definition 4.1. This leads to a mcvst problem $\mathcal{T}_\mathcal{V} = (N, \star, t, \mathcal{V})$ which looks as followed:

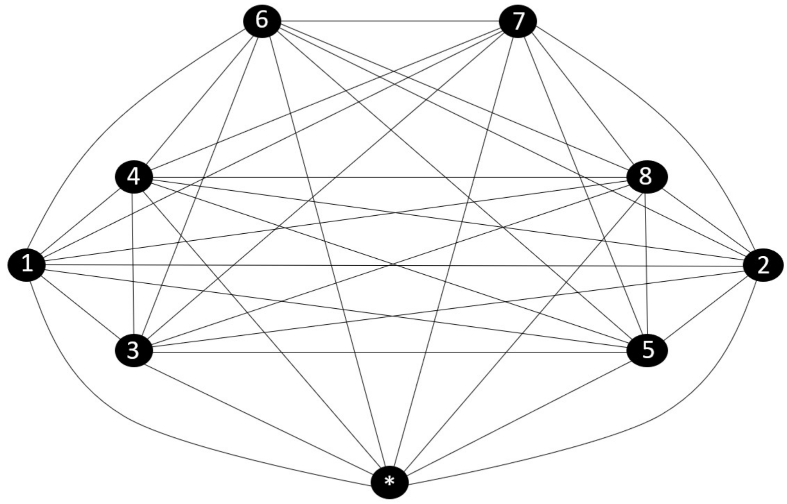

Figure 11

For simplicity, the vertices, which are the houses, have been rearranged in Figure 11 such that the once that belong in the same village are close together. Now we have to find a valid solution for this mcvst problem. It is easier to first make sure that the houses in each village are connected and then connect the villages with each other and the source. Keep in mind that you don't create any cycles as we are still trying to find a spanning tree. Looking at the following two graphs, it is easy to see which one is a valid solution.

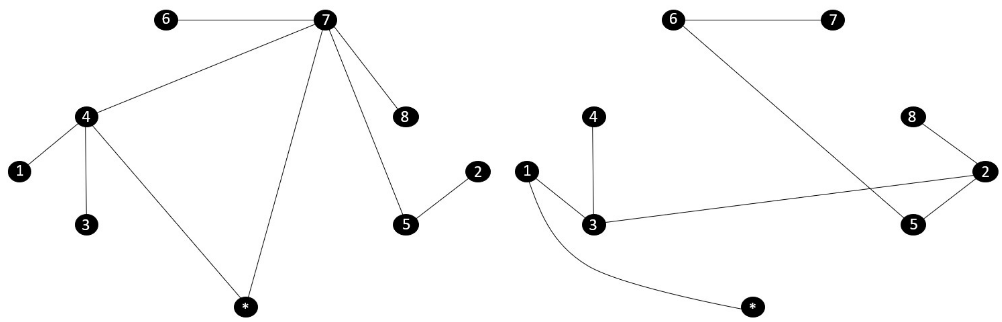

Figure 12

We can see in Figure 12 that the graph on the left is not a valid solution as it does not meet both criteria in Definition 4.2. First of all, there is a cycle with the vertices $\star$, 4 and 7. Secondly, the village {2,5,8} is not connected. The graph on the right however, is a valid solution as it is a spanning tree and the vertices in each village are connected.

4.1.1 Prim's algorithm

Given a mcvst problem $\mathcal{T}_{\mathcal{V}} = (N, \star, t, \mathcal{V})$, the first algorithm is inspired by Prim's algorithm. Also here, let I be the set of vertices that are connected to the source. When the first edge $\{\star, i\}$ is chosen, a vertex is added to the set I. This vertex must lie in a village. Define $V(i)$ as the village that contains vertex i. The algorithm continues by only looking at the cheapest edge with one endpoint in $V(i) \cap I$ and one endpoint in $V(i)\backslash I$. It keeps taking these edges until $V(i) \cap I = V(i)$. This means that the graph $(V(i), R \cap E_{V(i)})$ is connected. Then it repeats this process by first looking at the cheapest edge $\{j, \ell\}$ with $j \in I$ and $\ell \notin I$ and then taking the cheapest edges from $V(\ell)$ in the same way as above. The algorithm stops when $I = N^\star$.

Algorithm 4.1. Let a mcvst problem $(N, \star, t, \mathcal{V})$ be the input. Then the output is an edge set $R \subset E_{N^\star}$ of an mcvst obtained as followed:

1. *Initialise* $R = \emptyset$ *and* $I = \{\star\}$;
2. *Find a minimal cost edge* $e_j = \{i, j\}$ *with* $i \in I$ *and* $j \notin I$ *in such a way that joining* e_j *to* R *does not introduce a cycle*;
3. *Join* e_j *to* R, j *to* I;
4. *Find a minimal cost edge* $e_l = \{k, l\}$ *with* $k \in V(j) \cap I$ *and* $l \in V(j)\backslash I$ *in such a way that joining* e_l *to* R *does not introduce a cycle*;
5. *Join* e_l *to* R *and* l *to* I;
6. *If* $I \cap V(j) \neq V(j)$, *go back to step 4*;
7. *If* $I \neq N^\star$, *go back to step 2.*

Remark. To proof that this algorithm works, it's important to note that the edges are chosen in an order. Therefore, let $\pi : \{1, 2, ..., \#N\} \to N$ be a bijection where $\pi(a) = b$ means that b is the a-th vertex that was connected to the source. This results in $(e_{\pi(1)}, e_{\pi(2)}, ..., e_{\pi(n)})$ being the sequence of edges in the order that they have been chosen by algorithm 4.1. It also follows that vertex $\pi(i)$ is the i-th vertex that was added to I. Furthermore, the edge set R that the algorithm returns is not unique since there could be more than one edge with the same minimal cost in step 2 and 4. Therefore it is better to write π^R instead of π since π is dependent of R.

Example 4.5. Reconsider the mcvst problem in example 4.4 but with the cost function t higher than zero for every edge in $E_{N^\star}$. This leads to the mcvst problem $\mathcal{T}_{\mathcal{V}} = (N, \star, t, \mathcal{V})$ with $\mathcal{V} = \{\{1, 3, 4\}, \{2, 5, 8\}, \{6, 7\}\}$. For clarity, vertices within the same village have the same colour and the edges have the mixed colour of the colours of the two endpoints. The source $\star$ and its outgoing edges are black. This leads to the following graph:

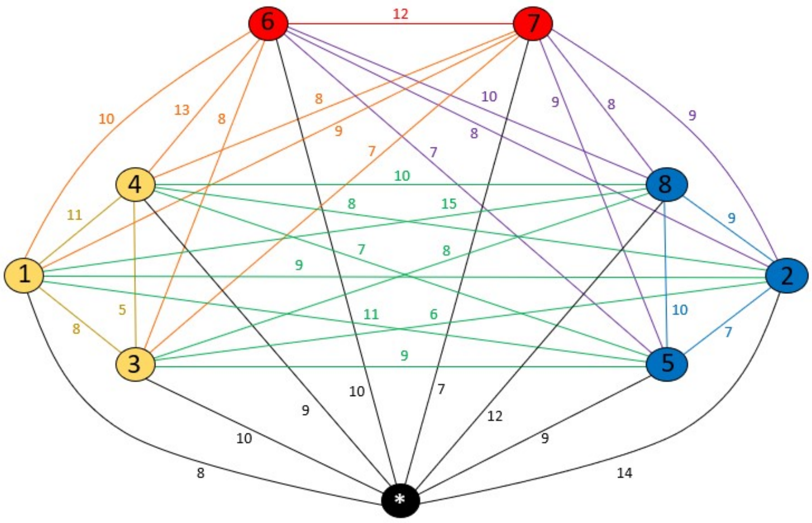

Figure 13

Step 1 of the algorithm is setting $R = \emptyset$ and $I = \{\star\}$. It continues with step 2 and 3 where for the first edge $e_j = \{i, j\}$ must hold that $i = \star$ and $j \notin I$ which means that $j \in N$. The cheapest edge that also meets these criteria is $\{\star, 7\}$. This edge will be added to R and vertex 7 to I since 7 is connected to the source.

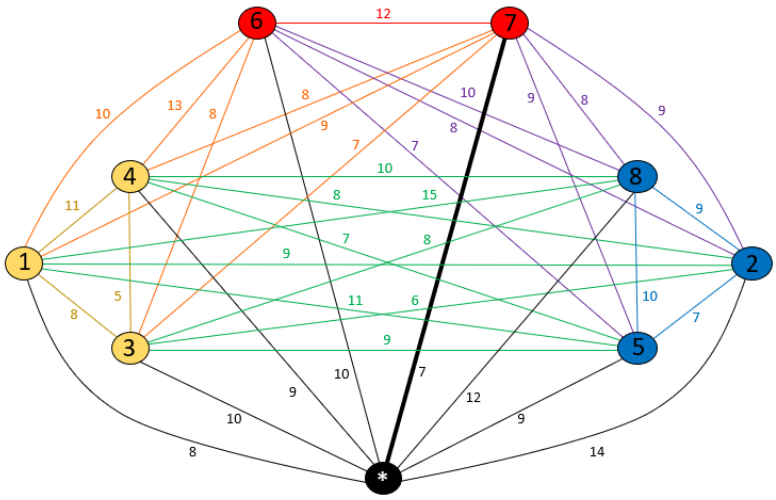

Figure 14

Step 4 looks at the village $V(7)$. The next edge that will be added to R is $\{6, 7\}$ since $7 \in V(7) \cap I$ and $6 \in V(7) \backslash I$. Vertex 6 is added to I which causes $I \cap V(7) = V(7)$. Since $I \neq N^\star$, the algorithm will return to step 2. The cheapest edges with only one endpoint in I are $\{3, 7\}$ and $\{5, 6\}$. Let's add edge $\{5, 6\}$ to R such that vertex 5 is added I and village $V(5)$ will be used in step 4.

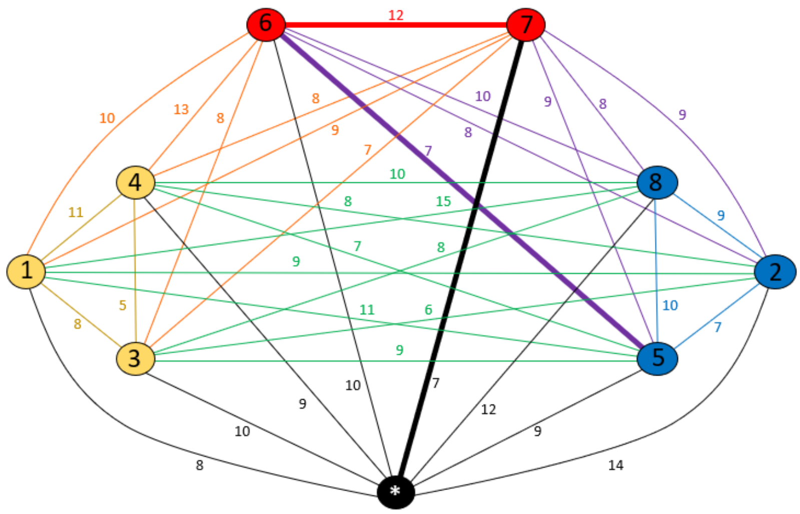

Figure 15

The algorithm continues with first adding edge $\{2, 5\}$ to R and then $\{2, 8\}$. Vertices 2 and 8 are added to I which causes $I \cap V(5) = V(5)$. The algorithm will go back to step 2 since $I \neq N^\star$. The edge that will be added to R is edge $\{2, 3\}$ and vertex 3 to I. Step 4 uses village $V(3)$ such that first edge $\{3, 4\}$ and then $\{1, 3\}$ are added to R. This means that vertices 4 and 1 are added to I such that $I = N^\star$. The algorithm returns the edge set $R = \{\{\star, 7\}, \{6, 7\}, \{5, 6\}, \{2, 5\}, \{2, 8\}, \{2, 3\}, \{3, 4\}, \{1, 3\}\}$.

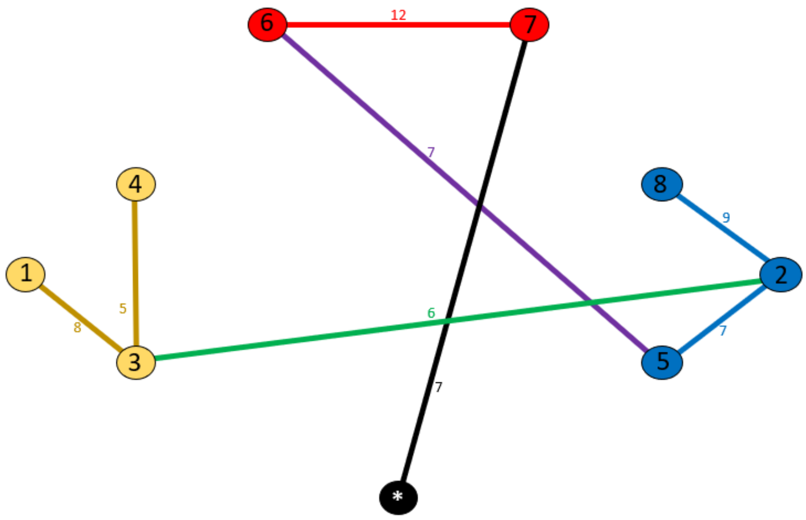

Figure 16

The cost of the mcvst $(N^\star, R)$ is $t(R) = 61$. This means that every mcvst of $\mathcal{T}_\mathcal{V}$ has the same minimal cost of 61. Note that the minimal cost is higher than when looking at the mcst of this problem by removing the villages. The cost of the mcst is 55 which can be seen in figure 17.

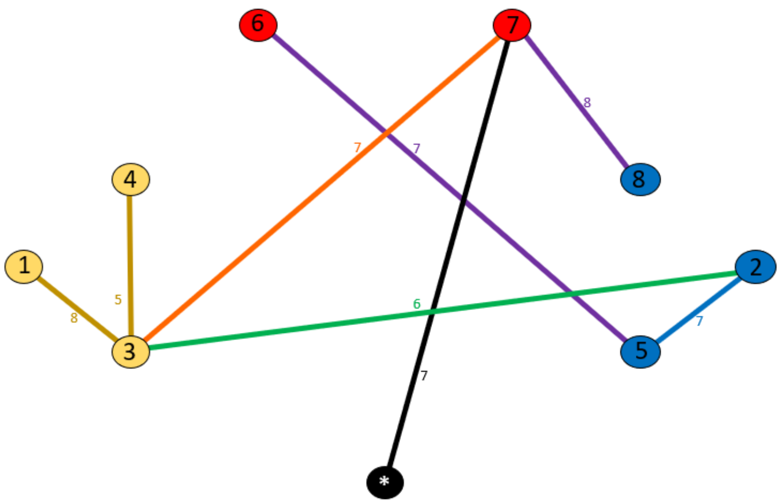

Figure 17

Theorem 4.6. *Let $\mathcal{T}_{\mathcal{V}} = (N, \star, t, \mathcal{V})$ be a mcvst problem. Then algorithm 4.1 returns an edge set R such that $(N^{\star}, R)$ is a mcvst.*

The proof of Prim's algorithm for mcst problems in Hein [7] has been used as inspiration for this proof.

Proof. That $(N^{\star}, R)$ is a valid solution is clear since algorithm 4.1 makes sure that there are not any cycles and it only stops when $(N^{\star}, R)$ and all villages are connected. The proof that $t(R) = min\{t(S) : (N^{\star}, S)$ is a valid solution$\}$ goes by contradiction.
Suppose that $t(R) \neq min\{t(S) : (N^{\star}, S)$ is a valid solution$\}$. Let $(e_{\pi(1)}, e_{\pi(2)}, ..., e_{\pi(n)})$ be the sequence of chosen edges in this order by algorithm 4.1. Let $\overline{R}$ be the edge set such that $(N^{\star}, \overline{R})$ is a mcvst and it contains $e_{\pi(1)}, e_{\pi(2)}, ..., e_{\pi(k)}$ where k is the largest possible integer. Let $I_k := \{\star, \pi(1), \pi(2), ..., \pi(k)\}$ be the vertex set I before $\pi(k+1)$ is connected to the source. Then $e_{\pi(k+1)} := \{i, \pi(k+1)\}$ with $i \in I$ will be the first edge that algorithm 4.1 adds to R that is not in $\overline{R}$. There are two cases for i and $\pi(k+1)$:

- **Case 1,**$V(i) = V(\pi(k+1)) \implies$ There is a path $i \rightsquigarrow \pi(k+1)$ in $\overline{R}$ within $V(i)$. Let $\{j, \ell\}$ be the first edge on this path with $j \in I_k$ and $\ell \notin I_k$;

Define edge set $S := (\overline{R} \cup \{\{i, \pi(k+1)\}\}) \backslash \{\{j, l\}\}$. S is the same as $\overline{R}$ except for one village. An edge is removed and a new edge is added to this village. The new edge makes sure that the village is still connected by how it is defined. This means that $(N^{\star}, S)$ is a valid solution.

- **Case 2,**$V(i) \neq V(\pi(k+1)) \implies$ There is a path $i \rightsquigarrow \pi(k+1)$ in $\overline{R}$. Let $\{j, \ell\}$ be the first edge on this path with $V(j) \neq V(l)$, $j \in I_k$ and $\ell \notin I_k$.

Define edge set $S := (\overline{R} \cup \{\{i, \pi(k+1)\}\}) \backslash \{\{j, l\}\}$. The villages in S remain unchanged such that they are all connected. The one thing that is different form $\overline{R}$ is an edge with endpoint in different villages. The new edge makes sure that the graph is still connected by how it is defined. This means that $(N^{\star}, S)$ is a valid solution.

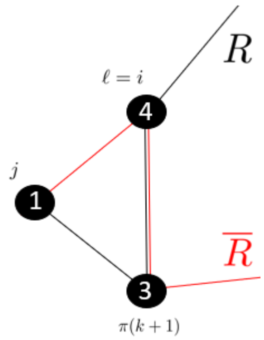

Figure 18: Case 1

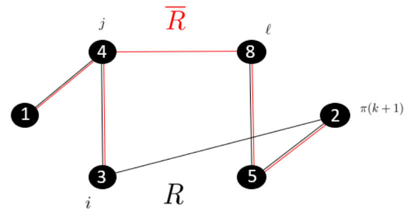

Figure 19: Case 2

In both cases, we have defined an edge set S such that $(N^\star, S)$ is a valid solution. This will be used when considering three possibilities for $t(\{i, \pi(k+1)\})$ and $t(\{j, \ell\})$:

1. $t(\{i, \pi(k+1)\}) < t(\{j, \ell\})$, then by constructing S we have replaced an edge in $\overline{R}$ with a cheaper edge. This means that $t(\overline{R}) > t(S)$, which cannot be possible since $\overline{R}$ is a mcvst;

2. $t(\{i, \pi(k+1)\}) = t(\{j, \ell\})$, then $t(\overline{R}) = t(S)$ which means that S is an mcvst. Since $e_{\pi(k+1)} = \{i, \pi(k+1)\}$ and $\{j, l\}$ cannot be any of the edges $e_{\pi(1)}, e_{\pi(2)}, ..., e_{\pi(k)}$, it follows that S contains $e_{\pi(1)}, e_{\pi(2)}, ..., e_{\pi(k+1)}$. This contradicts the definition of edge set $\overline{R}$ which contains $e_{\pi(1)}, e_{\pi(2)}, ..., e_{\pi(k)}$ for k the largest possible integer;

3. $t(\{i, \pi(k+1)\}) > t(\{j, \ell\})$, then the cost of edge $\{j, l\}$ is smaller. Algorithm 4.1 will select $\{j, \ell\}$ at step $k+1$, which contradict the definition of edge $\{i, \pi(k+1)\}$.

Since all possibilities lead to contradictions, our assumption must be false. This means that $t(R) = min\{t(S) : (N^\star, S) \text{ is a valid solution}\}$ which means that $(N^\star, R)$ is a mcvst. □

4.1.2 Kruskal's algorithm

Given a mcvst problem $\mathcal{T}_\mathcal{V} = (N, \star, t, \mathcal{V})$, the second algorithm is inspired by Kruskal's algorithm. The cheapest edge will again be taken from from $E_{N^\star}$, but it also takes villages into account. The algorithm starts by taking the minimal cost edge $\{i, j\}$ and looks at its vertices. If the vertices belong in the same village (i.e., $V(i) = V(j)$), the next edge that will be taken, is the cheapest edge within the village $V(i)$. The algorithm keeps taking the cheapest edges in this village until $(V(i), R \cap E_{V(i)})$ is connected. If the vertices belong in a different village (i.e., $V(i) \neq V(j)$), both villages will be connected separately in the same way as before. Then it continues with taking the cheapest edge and connects the corresponding villages.

Algorithm 4.2. Let $\mathcal{T} = (N, \star, t)$ be a mcst problem. Then the output is an edge set $R \subset E_{N^\star}$ of an mcst obtained as followed:

1. *Initialise* $R = \emptyset$;

2. *Find a minimal cost edge* $e_{ij} = \{i, j\} \in E_{N^\star} \backslash R$ *in such a way that joining* e_{ij} *to* R *does not introduce a cycle*;

3. *Join e_{ij} to R;*

4. *If $V(i) = V(j)$, do step 5 to 7 for $\boldsymbol{x} = i$. If $V(i) \neq V(j)$, do step 5 to 7 for $\boldsymbol{x} = i$ and for $\boldsymbol{x} = j$ where $(V(\boldsymbol{x}), E_{V(\boldsymbol{x})} \cap R)$ is not connected;*

5. *Find a minimal cost edge $e \in E_{V(\boldsymbol{x})} \backslash R$ in such a way that joining e to R does not introduce a cycle;*

6. *Join e to R;*

7. *If $(V(\boldsymbol{x}), E_{V(\boldsymbol{x})} \cap R)$ is not connected, go to step 5;*

8. *If $(N^\star, R)$ is not connected, go back to step 2.*

Example 4.7. Reconsider the mcvst problem in example 4.5 and use algorithm 4.2 to find the mcvst. Step 1 sets $R = \emptyset$ and continues to step 2 which looks at the cheapest edge of $E_{N^\star}$. The edge $\{3, 4\}$ is added to R and step 5 to 7 will be done with village $V(3) = V(4)$.

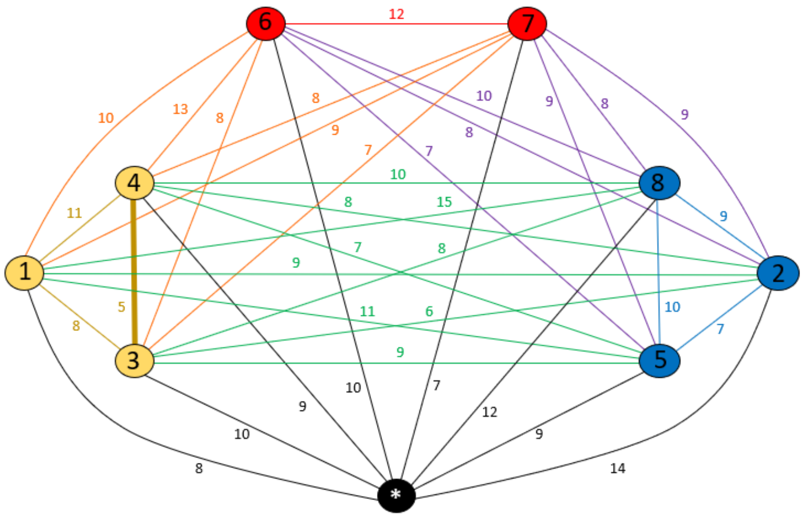

Figure 20

The next edge $e \in E_{N^\star}$ much hold that $e \in E_{V(3)} \backslash R$ which are $\{1, 3\}$ and $\{1, 4\}$. The cheapest of the two is edge $\{1, 3\}$ and will be added to R. The graph $(V(\mathbf{3}), E_{V(\mathbf{3})} \cap R)$ is connected and $(N^\star, R)$ is not connected. The algorithm will return to step 2 and adds the cheapest edge in $E_{N^\star} \backslash R$. The edge added to R is $\{2, 3\}$ which means that step 5 to 7 will be done with village $V(2)$.

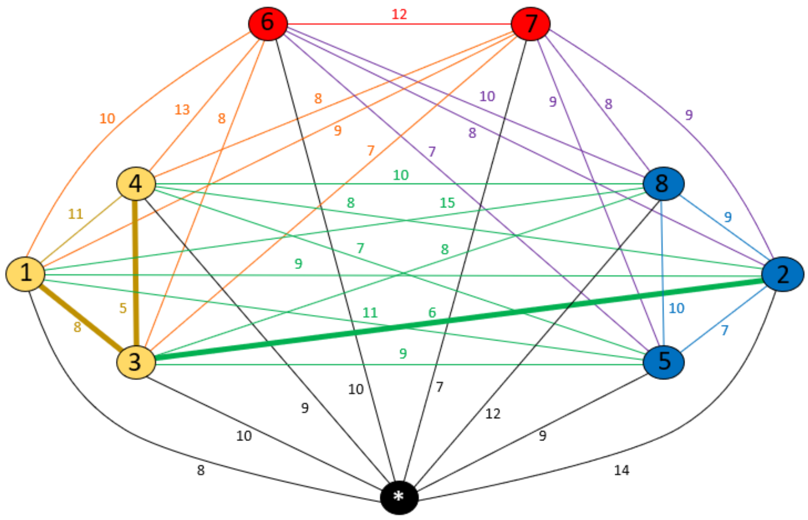

Figure 21

The algorithm continues with first adding edge $\{2, 5\}$ to R and then $\{2, 8\}$. The algorithm goes back to step 2 since $(N^\star R)$ is not connected. The two cheapest edge that can be added to R are $\{3, 7\}$ and $\{5, 6\}$. In this case, lets add edge $\{3, 7\}$ to R such that step 5 to 7 are done with village $V(7)$. Edge $\{6, 7\}$ will be added to R and the algorithm will go back to step 2 since $(N^\star R)$ still is not connected. The last edge that will be added to R is $\{\star, 7\}$ such that $(N^\star R)$ is connected.

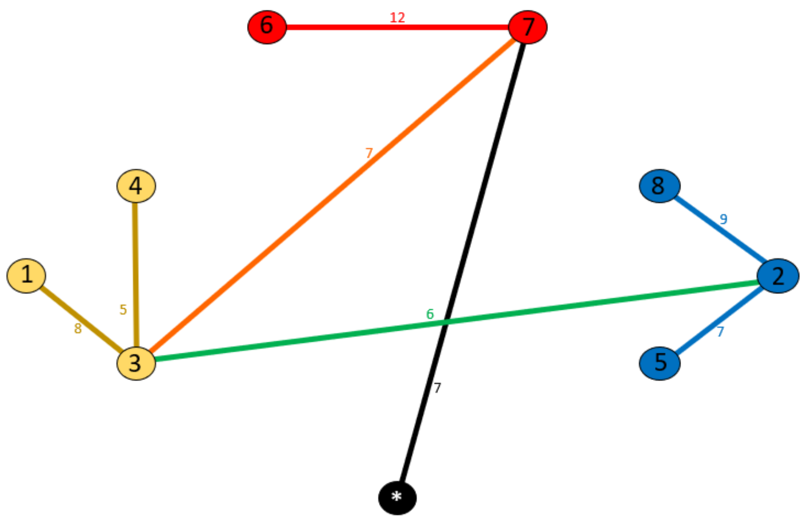

Figure 22

Theorem 4.8. *Let $\mathcal{T}_\mathcal{V} = (N, \star, t, \mathcal{V})$ be a mcvst problem. Then algorithm 4.2 returns an edge set R such that $(N^\star, R)$ is a mcvst.*

The proof of Kruskal's algorithm for mcst problems in Goodaire and Huberman [5] has been used as inspiration for this proof.

Proof. That $(N^\star, R)$ is a valid solution is clear since algorithm 4.2 makes sure that there

are not any cycles and it only stops when $(N^\star, R)$ and all villages are connected. The second thing that we need to check is that R has minimal cost.
Let $\overline{R}$ be an edge set such that $(N^\star, \overline{R})$ is a mcvst. If $R = \overline{R}$ then $(N^\star, R)$ is a mcvst. If $R \neq \overline{R}$, then there exists one or more edges in $\overline{R}$ that are different from the edges in R. Define the set $D = \{e \in \overline{R} | e \notin R\}$ and let $a = \{i, j\} \in D$ be the edge such that $t(a) = \min_{e \in D} t(e)$. There are two cases for edge a.

- **Case 1,** the vertices i and j lie within the same village ($V(i) = V(j)$).

The set $R \cup a$ contains a cycle within village $V(i)$. It holds that every edge in this cycle has a cost lower or equal than $t(a)$ since the algorithm keeps taking the edge with minimal cost. Furthermore, there exists an edge $b = \{k, l\}$ in this cycle that is not contained in $\overline{R}$ since $(N^\star, \overline{R})$ is a mcvst (it does not contain a cycle). Furthermore, k and l lie within village $V(i)$ and one of the vertices may equal i or j. Consider the edge set $R_2 = (R \backslash \{b\}) \cup \{a\}$. This is a valid solution and $t(R_2) \geq t(R)$ since $t(a) \geq t(b)$.

- **Case 2,** the vertices i and j lie in different villages ($V(i) \neq V(j)$).

The set $R \cup a$ contains a cycle with two or more edge with endpoints in different villages. It holds that every edge in this cycle has a cost lower or equal than $t(a)$ since the algorithm keeps taking the edge with minimal cost. Furthermore, there exists an edge $b = \{k, l\}$ in this cycle that is not contained in $\overline{R}$ since $(N^\star, \overline{R})$ is a mcvst (it does not contain a cycle), where k and l lie in different villages. Consider the edge set $R_2 = (R \backslash \{b\}) \cup \{a\}$. This is a valid solution and $t(R_2) \geq t(R)$ since $t(a) \geq t(b)$.

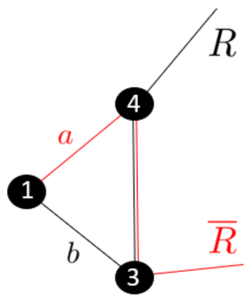

Figure 23: Case 1

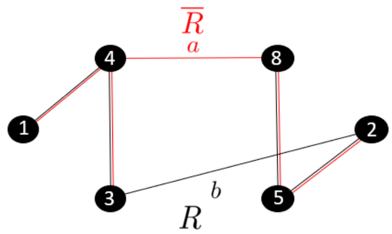

Figure 24: Case 2

Continuing this process with R_2 to find a valid solution R_3, the new edge set keeps getting more edges similar to the once in $\overline{R}$ (we keep removing an edge from R and add an edge from $\overline{R}$). This eventually ends up with the edge set $\overline{R}$ such that:

$$t(R) \leq t(R_2) \leq t(R_3) \leq ... \leq t(\overline{R})$$

Since $\overline{R}$ has a minimal cost because $(N^\star, \overline{R})$ is a mcvst, all the inequalities have to be equalities. This means that R has minimal cost such that $(N^\star, R)$ is a mcvst. $\square$

4.2 Mcvst game

The second part of the mcvst problem is about allocating the minimal cost for a mcvst problem. A mcvst game has to be defined to determine the cost of the coalitions. An intuitive way is to define it the same way as with mcst games.

Definition 4.9. Let $\mathcal{T}_\mathcal{V} = (N, \star, t, \mathcal{V})$ be a mcvst problem. For every coalition S , $\mathcal{V}_S = \{V \cap S : V \in \mathcal{V}\}$ be the village set of S. The associated *intuitive mcvst game* $(N, c^{\mathcal{T}_\mathcal{V}})$ is

$$c^{\mathcal{T}_\mathcal{V}}(S) = min\{t(R) : R \subset E_{S^\star} \text{ and } (S^\star, R) \text{ is a valid solution with village set } \mathcal{V}_S\}$$

$\forall S \subset N, S \neq \emptyset$ and $c^{\mathcal{T}_\mathcal{V}}(\emptyset) = 0$.

Remark. Note that every mcvst of $\mathcal{T}_\mathcal{V}$ has the same value as $c^{\mathcal{T}_\mathcal{V}}(N)$. This follows from the fact that $\mathcal{V}_N = \mathcal{V}$, which means that $c^{\mathcal{T}_\mathcal{V}}(N)$ looks for the minimal cost edge set R such that $(N^\star, R)$ is a valid solution with village set $\mathcal{V}$. The same is done when looking for a mcvst of $\mathcal{T}_\mathcal{V}$

This might seem like a well-defined mcvst game to find fair allocations, but there is a flaw with this definition.

Example 4.10. Consider the following mcvst problem $\mathcal{T}_\mathcal{V} = (N, \star, t, \mathcal{V})$ with $N = \{1, 2, 3, 4\}$ and $\mathcal{V} = \{\{1, 2\}, \{3, 4\}\}$:

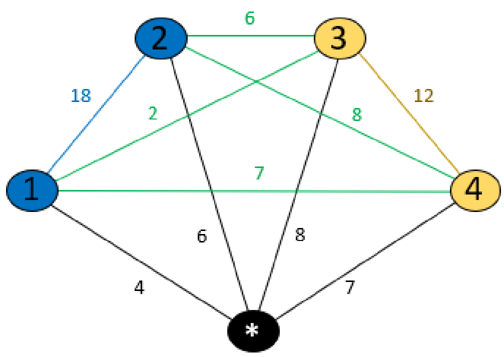

Figure 25

The mcvst game $(N, c^{\mathcal{T}_\mathcal{V}})$ is:

S	$\{1\}$	$\{2\}$	$\{3\}$	$\{4\}$	$\{1,2\}$	$\{1,3\}$	$\{1,4\}$	$\{2,3\}$	$\{2,4\}$	$\{3,4\}$
$c^{\mathcal{T}_\mathcal{V}}(S)$	4	6	8	7	22	6	11	12	13	19

S	$\{1,2,3\}$	$\{1,2,4\}$	$\{1,3,4\}$	$\{2,3,4\}$	$\{1,2,3,4\}$
$c^{\mathcal{T}_\mathcal{V}}(S)$	24	29	18	24	36

To find a fair allocation $x \in \mathbb{R}^4$ for this problem becomes impossible with the associated intuitive mcvst game. This becomes clear when looking at the core elements $C(c^{\mathcal{T}_\mathcal{V}})$. The fair allocations must satisfy several restrictions including

$$x_1 \leq 4,\ x_2 \leq 6,\ x_3 \leq 8,\ x_4 \leq 7$$
$$x_1 + x_2 + x_3 + x_4 = 36.$$

It can be easily seen that an allocation cannot satisfy all these restrictions which means that $C(c^{\mathcal{T}_\mathcal{V}}) = \emptyset$. This means that in this example, there does not exist a fair allocation when using the intuitive mcvst game. The problem with this game is that the villages are not taken into account. In this example, the edges with endpoints in the same village have much higher costs than the edges with endpoints in different villages. Let $(N^\star, R)$ be a mcvst of $\mathcal{T}_\mathcal{V}$, the edge $\{1,2\}$ must be added to the mcvst. If not, the graph $(V(1), R|_{E_{V(1)}})$ is not connected. This means that when looking at the singleton coalition $\{1\}$, the cost of edge $\{1,2\}$ must be taken into account. This is something that the intuitive mcvst game does not do. Therefore a new mcvst game must be defined.

4.2.1 Complete mcvst game

A possibility to avoid this problem is by looking at the complete village of every player in a coalition. This can be done by looking at all the villages that contain at least one player that is also in the coalition. New sets need to be defined to obtain these villages. For a given mcvst problem $\mathcal{T}_\mathcal{V} = (N, \star, t, \mathcal{V})$, all the villages that contain at least one player of a coalition $S \subset N$ is $C_S := \{V(i) \in \mathcal{V} : i \in S\}$. Furthermore, all the players contained in these villages are $T_S := \bigcup_{V \in \mathcal{C}_S} V$.

Definition 4.11. Let $\mathcal{T}_\mathcal{V} = (N, \star, t, \mathcal{V})$ be a mcvst problem. The associated *complete mcvst game* $(N, cc^{\mathcal{T}_\mathcal{V}})$ is

$$cc^{\mathcal{T}_\mathcal{V}}(S) = min\{t(R) : R \subset E_{T_S^\star} \text{ and } (T_S^\star, R) \text{ is a valid solution with village set } C_S\}$$

$\forall S \subset N, S \neq \emptyset$ and $cc^{\mathcal{T}_\mathcal{V}}(\emptyset) = 0$.

Remark. From this definition can be easily seen that for all coalitions $S \subset N$ with $S = T_S$ holds that $cc^{\mathcal{T}_\mathcal{V}}(S) = c^{\mathcal{T}_\mathcal{V}}(S)$ where $c^{\mathcal{T}_\mathcal{V}}$ is the associated intuitive mcvst game.

Example 4.12. Reconsider the mcvst problem used in example 4.10. The associated complete mcvst game $(N, cc^{\mathcal{T}_\mathcal{V}})$ gives:

S	$\{1\}$	$\{2\}$	$\{3\}$	$\{4\}$	$\{1,2\}$	$\{1,3\}$	$\{1,4\}$	$\{2,3\}$	$\{2,4\}$	$\{3,4\}$
$cc^{\mathcal{T}_\mathcal{V}}(S)$	22	22	19	19	22	36	36	36	36	19

S	$\{1,2,3\}$	$\{1,2,4\}$	$\{1,3,4\}$	$\{2,3,4\}$	$\{1,2,3,4\}$
$cc^{\mathcal{T}_\mathcal{V}}(S)$	36	36	36	36	36

It can be easily seen that $(9,9,9,9) \in C(cc^{\mathcal{T}_\mathcal{V}})$ which means that this is a fair allocation. One downside of the complete mcvst game is that it does not take the differences in coalitions into account. This can be seen when looking at the coalitions $\{1,3\}$ and $\{2,4\}$.

$C_{\{1,3\}} = \{V(1), V(3)\} = \{\{1,2\},\{3,4\}\}$ and $T_{\{1,3\}} = \{1,2,3,4\} \implies cc^{\mathcal{T}_\mathcal{V}}(\{1,3\}) = 36$
$C_{\{2,4\}} = \{V(2), V(4)\} = \{\{1,2\},\{3,4\}\}$ and $T_{\{2,4\}} = \{1,2,3,4\} \implies cc^{\mathcal{T}_\mathcal{V}}(\{2,4\}) = 36$

Although the coalitions are disjoint, the values are the same. In the table can be seen that eight collations have the same value as $cc^{\mathcal{T}_\mathcal{V}}(N)$. To avoid this problem, the intuitive and the complete mcvst games can be combined.

4.2.2 Combined mcvst game

A new mcvst game has to be defined that considers both the villages and the coalitions. Reconsider example 4.10 and again look at the coalitions $\{1,3\}$ and $\{2,4\}$. First use the associated intuitive mcvst game. For coalition $\{1,3\}$, the edges $\{\star,1\}$ and $\{1,3\}$ are used such that $c^{\mathcal{T}_\mathcal{V}}(\{1,3\}) = 6$. Keep in mind that the coalition $\{1,3\}$ is different from the edge $\{1,3\}$. For coalition $\{2,4\}$, it's $\{\star,2\}$ and $\{\star,4\}$ such that $c^{\mathcal{T}_\mathcal{V}}(\{2,4\}) = 13$. Define $R_0 := \{\{\star,1\},\{1,3\}\}$ and $\overline{R_0} := \{\{\star,2\},\{\star,4\}\}$.

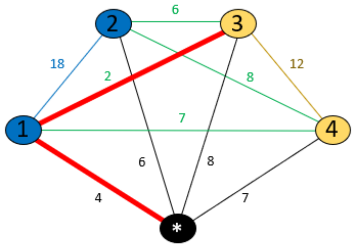

Figure 26: $S = \{1, 3\}$

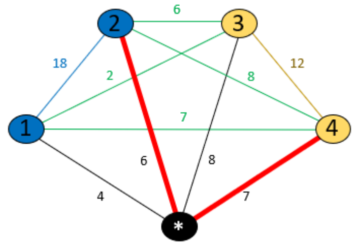

Figure 27: $S = \{2, 4\}$

Figure 28: Edge sets of $c^{\mathcal{T_V}}(S)$

Secondly, use the associated complete mcvst game. For both coalitions, the edge set $R = \{\{\star, 1\}, \{1, 2\}, \{1, 3\}, \{3, 4\}$ is used to get $cc^{\mathcal{T_V}}(\{1, 3\}) = cc^{\mathcal{T_V}}(\{2, 4\}) = 36$. However, this time the coalitions need to be taken into account. This is done by putting a restricting on the edge sets used for the complete mcvst game. Still, the edge set with minimal cost needs to be found but it must contain R_0 and $\overline{R_0}$ for coalitions $\{1, 3\}$ and $\{2, 4\}$ respectively. For coalition $\{1, 3\}$ can be seen that $R_0 \subset R$, which means that the value of this coalition is 36. For coalition $\{2, 4\}$ holds that $\overline{R_0} \not\subset R$. This means that we need to look at the minimal cost edge set $\overline{R}$ such that $\overline{R_0} \subset \overline{R} \subset E_{T^\star_{\{2,4\}}}$ and $(T^\star_{\{2,4\}}, R)$ is a valid solution with village set $C_{\{2,4\}}$. This leads to $\overline{R} = \{\{\star, 2\}, \{\star, 4\}, \{1, 2\}, \{3, 4\}\}$ such that the value of this coalition is 43.

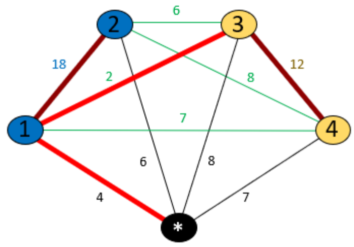

Figure 29: $S = \{1, 3\}$

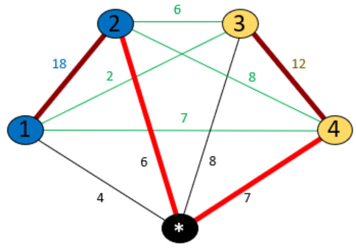

Figure 30: $S = \{2, 4\}$

Figure 31: Edge sets of $cc^{\mathcal{T_V}}(S)$ with restriction

Definition 4.13. Let $\mathcal{T_V} = (N, \star, t, \mathcal{V})$ be a mcvst problem. For every coalition $S \subset N$, let R_0 be the edge set such that $t(R_0) = c^{\mathcal{T_V}}(S) = min\{t(R) : R \subset E_{S^\star}$ and $(S^\star, R)$ is a valid solution with village set $\mathcal{V}_S\}$. The associated *combined mcvst game* $(N, sc^{\mathcal{T_V}})$ is

$$sc^{\mathcal{T_V}}(S) = min\{t(R) : R_0 \subset R \subset E_{T^\star_S} \text{ and } (T^\star_S, R) \text{ is a valid solution with village set } C_S\}$$

$\forall S \subset N, S \neq \emptyset$ and $sc^{\mathcal{T_V}}(\emptyset) = 0$

Remark. (1) From the definition can be concluded that for all coalitions $S \subset N$ with $S = T_S$ holds that $sc^{\mathcal{T_V}}(S) = c^{\mathcal{T_V}}(S)$ where $c^{\mathcal{T_V}}$ is the associated intuitive mcvst game. This follows from the fact that $(S^\star, R_0) = (T^\star_S, R_0)$ is a valid solution with village set C_S where $t(R_0) = c^{\mathcal{T_V}}(S) = c^{\mathcal{T_V}}(T_S)$.
(2) From the definition can also be concluded that $cc^{\mathcal{T_V}}(S) \leq sc^{\mathcal{T_V}}(S)$ for all $S \subset N$ where $cc^{\mathcal{T_V}}$ is the associated complete mcvst game. This follows from the fact that there is an extra restriction on the possible edge sets R that satisfy $t(R) = sc^{\mathcal{T_V}}(S)$.

Example 4.14. Reconsider the mcvst problem used in example 4.10. The associated combined mcvst game $(N, sc^{\mathcal{T}_\mathcal{V}})$ is:

S	$\{1\}$	$\{2\}$	$\{3\}$	$\{4\}$	$\{1,2\}$	$\{1,3\}$	$\{1,4\}$	$\{2,3\}$	$\{2,4\}$	$\{3,4\}$
$sc^{\mathcal{T}_\mathcal{V}}(S)$	22	24	20	19	22	36	41	42	43	19

S	$\{1,2,3\}$	$\{1,2,4\}$	$\{1,3,4\}$	$\{2,3,4\}$	$\{1,2,3,4\}$
$sc^{\mathcal{T}_\mathcal{V}}(S)$	36	41	36	42	36

It can be easily seen that just like with the associated complete mcvst game, $(9,9,9,9) \in C(sc^{\mathcal{T}_\mathcal{V}})$ which means that this is a fair allocation. The question can be asked: "How important is it to take the coalitions into account?" Let's compare the cores of example 4.12 and 4.14. For the allocations $x \in \mathbb{R}^4$ in $C(sc^{\mathcal{T}_\mathcal{V}})$ must hold that $x_2 \leq 24$. However, since for x must also hold that $x_1 + x_2 \leq 22$, follows that $x_2 \leq 22$. This is also the case with $x_3 + x_4 \leq 19 \implies x_3 \leq 19$ and $\sum_{i \in S} x_i \leq 36$ for all $sc^{\mathcal{T}_\mathcal{V}}(S) \geq 36$ since $\sum_{i=1}^{4} x_i \leq 36$. This means that in this case $C(sc^{\mathcal{T}_\mathcal{V}}) = C(cc^{\mathcal{T}_\mathcal{V}})$. It can be proven that this is the case for every mcvst problem $\mathcal{T}_\mathcal{V}$.

Lemma 4.15. *Let $\mathcal{T}_\mathcal{V} = (N, \star, t, \mathcal{V})$ be a mcvst problem and $cc^{\mathcal{T}_\mathcal{V}}$, $sc^{\mathcal{T}_\mathcal{V}}$ the associated complete and combined mcvst games respectively. Then $C(cc^{\mathcal{T}_\mathcal{V}}) = C(sc^{\mathcal{T}_\mathcal{V}})$.*

Proof. Suppose that $x \in C(cc^{\mathcal{T}_\mathcal{V}})$. It follows from $N = T_N$ that $cc^{\mathcal{T}_\mathcal{V}}(N) = sc^{\mathcal{T}_\mathcal{V}}(N)$. This means that $\sum_{i \in N} x_i = cc^{\mathcal{T}_\mathcal{V}}(N) = sc^{\mathcal{T}_\mathcal{V}}(N)$. Furthermore, for all $S \subset N$ holds that $cc^{\mathcal{T}_\mathcal{V}}(S) \leq sc^{\mathcal{T}_\mathcal{V}}(S)$, what leads to $x \in C(sc^{\mathcal{T}_\mathcal{V}})$.
Suppose that $x \in C(sc^{\mathcal{T}_\mathcal{V}})$. It follows from $cc^{\mathcal{T}_\mathcal{V}}(N) = sc^{\mathcal{T}_\mathcal{V}}(N)$ that $\sum_{i \in N} x_i = sc^{\mathcal{T}_\mathcal{V}}(N) = cc^{\mathcal{T}_\mathcal{V}}(N)$. Furthermore, for all $S \subset N$ with $S = T_S$ holds that $\sum_{i \in S} x_i \leq sc^{\mathcal{T}_\mathcal{V}}(S) = cc^{\mathcal{T}_\mathcal{V}}(S)$. From the definition of $sc^{\mathcal{T}_\mathcal{V}}$ follows that for all $\overline{S} \subset S$ with $T_{\overline{S}} = T_S$ holds that $sc^{\mathcal{T}_\mathcal{V}}(S) \leq sc^{\mathcal{T}_\mathcal{V}}(\overline{S})$. This implies that $\sum_{i \in \overline{S}} x_i \leq sc^{\mathcal{T}_\mathcal{V}}(S) = cc^{\mathcal{T}_\mathcal{V}}(S) \leq sc^{\mathcal{T}_\mathcal{V}}(\overline{S})$, what leads to $x \in C(cc^{\mathcal{T}_\mathcal{V}})$. □

From Lemma 4.15 can be concluded that it is not necessary to take coalitions into account. Therefore it becomes redundant to use the more complex associated combined mcvst game to find fair allocation when the associated complete mcvst game gives back the same set of fair allocations.

4.2.3 Bird rule

Every time an edges is added in algorithm 4.1, a new player is connected to the source. This is again used in Bird rule to allocate the cost for a mcvst problem.

Algorithm 4.3. Let $(N, \star, t, \mathcal{V})$ be a mcvst problem. Then the output is an edge set $R \subset E_{N^\star}$ of an mcvst and its corresponding Bird allocation $\beta^R(\mathcal{T}_\mathcal{V})$ obtained as followed:

1. *Initialise $R = \emptyset$ and $I = \{\star\}$;*
2. *Find a minimal cost edge $e_j = \{i, j\}$ with $i \in I$ and $j \notin I$ in such a way that joining e_j to R does not introduce a cycle;*
3. *Join e_j to R, j to I and assign the cost $\beta_j(\mathcal{T}_\mathcal{V}) = t(e_j)$ to j ;*
4. *Find a minimal cost edge $e_l = \{k, l\}$ with $k \in V(j) \cap I$ and $l \in V(j) \backslash I$ in such a way that joining e_l to R does not introduce a cycle;*
5. *Join e_l to R and l to I and assign the cost $\beta_l(\mathcal{T}_\mathcal{V}) = t(e_l)$ to l;*
6. *If $I \cap V(j) \neq V(j)$, go back to step 4;*

7. *If $I \neq N^{\star}$, go back to step 2.*

Remark. Again, we have an edge set R and an allocation $\beta(\mathcal{T}_{\mathcal{V}})$. Since R does not have to be unique and $\beta(\mathcal{T}_{\mathcal{V}})$ is dependent R, $\beta(\mathcal{T}_{\mathcal{V}})$ does not have to be unique. When the algorithm is finished, the Bird allocation is written as $\beta^R(\mathcal{T}_{\mathcal{V}})$ since then R is known.

Example 4.16. Reconsider example 4.10 and use algorithm 4.3 to find a fair allocation. The first edge that will be added to R is $\{\star, 1\}$ which connects player 1 to the source. This means that $\beta_1(\mathcal{T}_{\mathcal{V}}) = t(\{\star, 1\}) = 4$. The second edge that will be added is $\{1, 2\}$ which gives $\beta_2(\mathcal{T}_{\mathcal{V}}) = t(\{1, 2\}) = 18$. The third edge that will be added is $\{1, 3\}$ which gives $\beta_3(\mathcal{T}_{\mathcal{V}}) = t(\{1, 3\}) = 2$. Lastly, edge $\{3, 4\}$ is added and $\beta_4(\mathcal{T}_{\mathcal{V}}) = t(\{3, 4\}) = 12$. This gives the Bird allocation $\beta^R(\mathcal{T}_{\mathcal{V}}) = (4, 18, 2, 12)$.

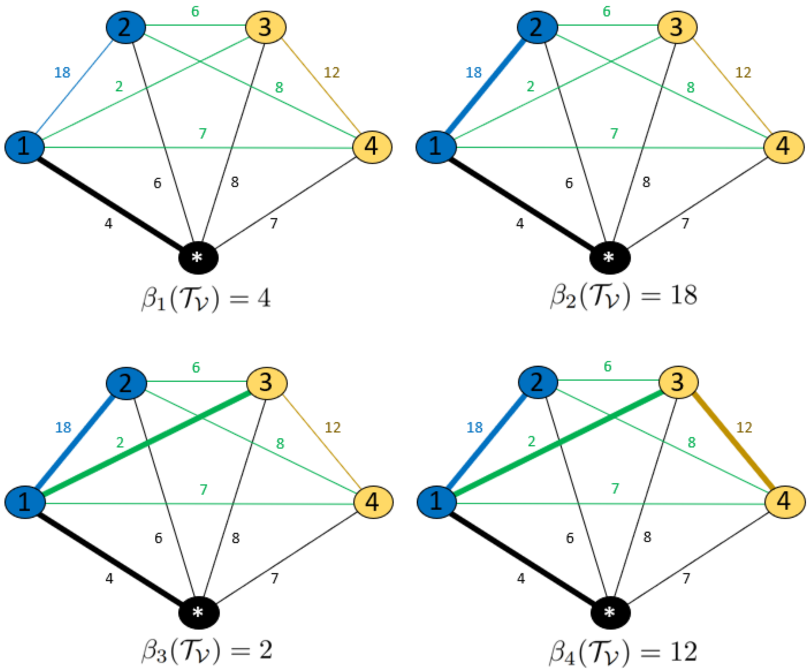

Figure 32

It can be easily checked that this is a core element of the associated complete mcvst game $cc^{\mathcal{T}_{\mathcal{V}}}$.

- $\sum_{i=1}^{4} \beta_i^R(\mathcal{T}_{\mathcal{V}}) = 36 = cc^{\mathcal{T}_{\mathcal{V}}}(N)$;
- $\beta_i^R(\mathcal{T}_{\mathcal{V}}) \leq cc^{\mathcal{T}_{\mathcal{V}}}(\{i\})$ for all $i \in N$;
- $\beta_1^R(\mathcal{T}_{\mathcal{V}}) + \beta_2^R(\mathcal{T}_{\mathcal{V}}) \leq cc^{\mathcal{T}_{\mathcal{V}}}(\{1, 2\})$;
- $\beta_3^R(\mathcal{T}_{\mathcal{V}}) + \beta_4^R(\mathcal{T}_{\mathcal{V}}) \leq cc^{\mathcal{T}_{\mathcal{V}}}(\{3, 4\})$.

Note that for all $S \subset N$ with $cc^{\mathcal{T}_{\mathcal{V}}}(S) = 36$ holds that $\sum_{i \in S} \beta_i^R(\mathcal{T}_{\mathcal{V}}) \leq 36$ since $\sum_{i=1}^{4} \beta_i^R(\mathcal{T}_{\mathcal{V}}) = 36$. With these results follow that $(4, 18, 2, 12) \in C(cc^{\mathcal{T}_{\mathcal{V}}})$.

Theorem 4.17. *Let $\mathcal{T}_{\mathcal{V}} = (N, \star, t, \mathcal{V})$ be a mcvst problem and $(N^\star, R)$ be the mcvst obtained by Algorithm 4.3 with corresponding Bird allocation vector $\beta^R(\mathcal{T}_{\mathcal{V}})$. Let $(N, cc^{\mathcal{T}_{\mathcal{V}}})$ be the associated complete mcvst game. Then $\beta^R(\mathcal{T}_{\mathcal{V}})$ is an element of the core $C(cc^{\mathcal{T}_{\mathcal{V}}})$.*

Proof. Let $(N^\star, R)$ be a mcvst with $R = \{e_i\}_{i\in N}$ obtained by Algorithm 4.3 such that $\beta_i^R(\mathcal{T}_{\mathcal{V}}) = t(e_i), i \in N$. Here, edge e_i means that one of the endpoints equals i.
What is needed to prove:

1. $\sum_{i\in N} \beta_i^R(\mathcal{T}_{\mathcal{V}}) = cc(N)$;
2. $\sum_{i\in S} \beta_i^R(\mathcal{T}_{\mathcal{V}}) \leq cc(S)\ \forall S \subset N$.

The equation follows immediately from the definition of $cc^{\mathcal{T}_{\mathcal{V}}}(N)$:

$$\sum_{i\in N} \beta_i^R(\mathcal{T}_{\mathcal{V}}) = \sum_{i\in N} t(e_i) = c^{\mathcal{T}_{\mathcal{V}}}(N) = cc^{\mathcal{T}_{\mathcal{V}}}(N)$$

where $c^{\mathcal{T}_{\mathcal{V}}}$ is the associated intuitive mcvst game.

For the in inequality, let $S \in 2^N \backslash \{\emptyset\}$. Consider a valid solution $(T_S^\star, F)$ such that $F \in E_{T_S^\star}$ and $cc^{\mathcal{T}_{\mathcal{V}}}(S) = \sum_{e\in F} t(e)$. Define $G := \{e_i \in R | i \in N\backslash T_S\}$ and note that $F \cap G = \emptyset$ since every edge $e \in G$ has at least one endpoint that is not contained in $T_S^\star$. Furthermore $(N^\star, F \cup G)$ is a valid solution. To see this, look at the graph $(N\backslash T_S, G)$. This graph does not contain any cycles since G takes edges from R and $(N^\star, R)$ is a valid solution. From the definition of T_S follows that $N\backslash T_S$ is a finite union of villages. This means that all the villages in $(N\backslash T_S, G)$ are connected since G contains all the edges with both endpoints in $N\backslash T_S$. G also contains all the edges from R that lie between these villages. Since $\#G = \#N\backslash T_S$ and G does not contain any cycles, it follows that G must contain at least one edge with one endpoint in $T_S^\star$. This means that $(T_S^\star \cup N\backslash T_S, F \cup G) = (N^\star, F \cup G)$ is a valid solution. Hence,

$$cc^{\mathcal{T}_{\mathcal{V}}}(N) \leq \sum_{e\in F\cup G} t(e) = \sum_{e\in F} t(e) + \sum_{e\in G} t(e) = cc^{\mathcal{T}_{\mathcal{V}}}(S) + \sum_{i\in N\backslash T_S} \beta_i^R(\mathcal{T}_{\mathcal{V}})$$

and, consequently,

$$\sum_{i\in S} \beta_i^R(\mathcal{T}_{\mathcal{V}}) \leq \sum_{i\in T_S} \beta_i^R(\mathcal{T}_{\mathcal{V}}) = cc^{\mathcal{T}_{\mathcal{V}}}(N) - \sum_{i\in N\backslash T_S} \beta_i^R(\mathcal{T}_{\mathcal{V}}) \leq cc^{\mathcal{T}_{\mathcal{V}}}(S)$$

□

Corollary 4.17.1. *Let $\mathcal{T}_{\mathcal{V}} = (N, \star, t, \mathcal{V})$ be a mcvst problem and let (N, sc) be the corresponding combined mcvst game. Let $\beta^R(\mathcal{T}_{\mathcal{V}}) \in \mathbb{R}^N$ be a corresponding Bird allocation vector. Then $\beta^R(\mathcal{T}_{\mathcal{V}})$ is an element of the core $C(sc^{\mathcal{T}_{\mathcal{V}}})$.*

Proof. This follows from Lemma 4.15 and Theorem 4.17. □

4.2.4 Equal Remaining Obligation rule

The equal remaining obligation rule stays almost the same as stated in 3.2.2. This means that the definition of the obligation vector and the cost contribution vector remains unchanged. The only thing that is different is that in this case, algorithm 4.2 is used to determine the cost contribution vector. By doing this, the equal remaining obligation rule takes the villages into account.

Example 4.18. Reconsider example 4.10 and use algorithm 4.2 with the equal remaining obligation rule to find a fair allocation. It starts with $R = \emptyset$, $obl^0 = (1,1,1,1)$ and $O^0(\mathcal{T}_\mathcal{V}) = (0,0,0,0)$. The algorithm begins with adding edge $\{1,3\}$ to R which connects player 1 and 3 to each other. This means that $obl^1 = (\frac{1}{2}, 1, \frac{1}{2}, 1)$ such that $obl^0 - obl^1 = (\frac{1}{2}, 0, \frac{1}{2}, 0)$. Furthermore, $O^1(\mathcal{T}_\mathcal{V}) = (0,0,0,0) + (\frac{1}{2}, 0, \frac{1}{2}, 0)2 = (1,0,1,0)$.

The second edge that is being added is $\{1,2\}$. This means that $obl^2 = (\frac{1}{3}, \frac{1}{3}, \frac{1}{3}, 1)$ such that $obl^1 - obl^2 = (\frac{1}{6}, \frac{2}{3}, \frac{1}{6}, 0)$. Furthermore, $O^2(\mathcal{T}_\mathcal{V}) = (1,0,1,0) + (\frac{1}{6}, \frac{2}{3}, \frac{1}{6}, 0)18 = (4,12,4,0)$.

The third edge that is being added is $\{3,4\}$. This means that $obl^3 = (\frac{1}{4}, \frac{1}{4}, \frac{1}{4}, \frac{1}{4})$ such that $obl^2 - obl^3 = (\frac{1}{12}, \frac{1}{12}, \frac{1}{12}, \frac{3}{4})$. Furthermore, $O^3(\mathcal{T}_\mathcal{V}) = (4,12,4,0) + (\frac{1}{12}, \frac{1}{12}, \frac{1}{12}, \frac{3}{4})12 = (5,13,5,9)$.

The last edge that will be added is $\{\star, 1\}$ which connects all players to the source. This means that $obl^4 = (0,0,0,0)$ such that $obl^3 - obl^4 = (\frac{1}{4}, \frac{1}{4}, \frac{1}{4}, \frac{1}{4})$. Furthermore, $O^4(\mathcal{T}_\mathcal{V}) = (5,13,5,9) + (\frac{1}{4}, \frac{1}{4}, \frac{1}{4}, \frac{1}{4})4 = (6,14,6,10)$.

This leads to the final cost contribution: $O^R(\mathcal{T}_\mathcal{V}) = (6,14,6,10)$.

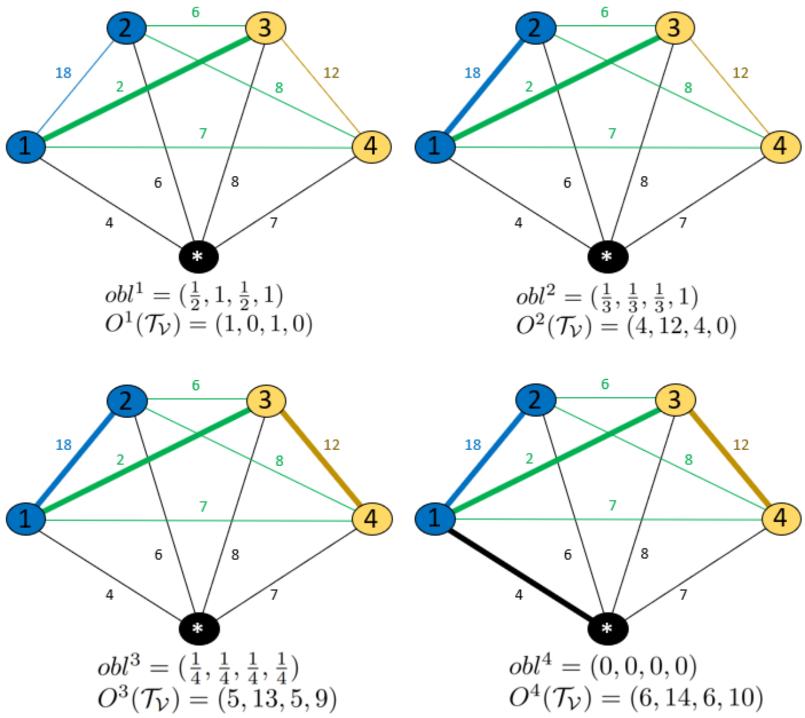

Figure 33

It can be easily checked that this is a core element of the associated complete mcvst game $cc^{\mathcal{T}_\mathcal{V}}$.

- $\sum_{i=1}^{4} O_i^R(\mathcal{T}_\mathcal{V}) = 36 = cc^{\mathcal{T}_\mathcal{V}}(N)$;

- $O_i^R(\mathcal{T}_\mathcal{V}) \leq cc^{\mathcal{T}_\mathcal{V}}(\{i\})$ for all $i \in N$;
- $O_1^R(\mathcal{T}_\mathcal{V}) + O_2^R(\mathcal{T}_\mathcal{V}) \leq cc^{\mathcal{T}_\mathcal{V}}(\{1,2\})$;
- $O_3^R(\mathcal{T}_\mathcal{V}) + O_4^R(\mathcal{T}_\mathcal{V}) \leq cc^{\mathcal{T}_\mathcal{V}}(\{3,4\})$.

Note that for all $S \subset N$ with $cc^{\mathcal{T}_\mathcal{V}}(S) = 36$ holds that $\sum_{i \in S} O_i^R(\mathcal{T}_\mathcal{V}) \leq 36$ since $\sum_{i=1}^{4} O_i^R(\mathcal{T}_\mathcal{V}) = 36$. With these results follow that $(6, 14, 6, 10) \in C(cc^{\mathcal{T}_\mathcal{V}})$.

In this thesis, it won't be proven that every cost contribution vector of a mcvst problem $\mathcal{T}_\mathcal{V}$ is a core element of the associated complete mcvst game.

References

[1] C. Bird. On cost allocation for a spanning tree: a game theoretic approach. *Networks*, 1976.

[2] P. Borm. Games, Cooperative Behavior and Economics. Tilburg, 2020.

[3] P. Borm, H. Hamers, and R. Hendrickx. Operations Research Games: A Survey. *TOP*, 2001.

[4] D. De Moore. Optimization and Allocation in Minimum Cost Spanning Tree Problems with Time. Master's thesis, Radboud University, 2019.

[5] E. Goodaire and M. Parmenter. *Discrete Mathematics with Graph Theory.* Pearson, 2005.

[6] D. Granot and G. Huberman. Minimum cost spanning tree games. *Mathematical Programming*, 1981.

[7] J. Hein. *Discrete Structures, Logic, and Computability.* Jones & Bartlett Learning, 2015.

[8] J. Kruskal. On the shortest spanning subtree of a graph and the travelling salesman problem. *Proceedings of the American Mathematical Society*, 1956.

[9] R. Prim. Shortest connection networks and some generalizations. *The Bell system technical journal*, 1957.

[10] S. Tijs, R. Brânzei, S. Moretti, and H. Norde. *Obligation Rules for Minimum Cost Spanning Tree Situations and their Monotonicity Properties.* CentER Discussion Paper, 2004.