HAL
open science

Fair cost-sharing methods for the minimum spanning tree game

Eric Angel, Evripidis Bampis, Lélia Blin, Laurent Gourvès

To cite this version:

Eric Angel, Evripidis Bampis, Lélia Blin, Laurent Gourvès. Fair cost-sharing methods for the minimum spanning tree game. Information Processing Letters, 2006, 100, pp.29–35. 10.1016/j.ipl.2006.05.007 . hal-00341341

HAL Id: hal-00341341

https://hal.science/hal-00341341

Submitted on 19 Jul 2009

HAL is a multi-disciplinary open access archive for the deposit and dissemination of scientific research documents, whether they are published or not. The documents may come from teaching and research institutions in France or abroad, or from public or private research centers.

L'archive ouverte pluridisciplinaire **HAL**, est destinée au dépôt et à la diffusion de documents scientifiques de niveau recherche, publiés ou non, émanant des établissements d'enseignement et de recherche français ou étrangers, des laboratoires publics ou privés.

Fair Cost-Sharing Methods for the Minimum Spanning Tree Game

Eric Angel Evripidis Bampis Lélia Blin Laurent Gourvès

1. LaMI, CNRS UMR 8042, Université d'Évry, Tour Évry 2, 523 Place des terrasses de l'agora, 91000 Évry Cedex
{angel,bampis,lblin,lgourves}@lami.univ-evry.fr

Abstract

We study the problem of sharing in a *fair* manner the cost of a service provided to a set of players in the context of Cooperative Game Theory. We introduce a new fairness measure capturing the *dissatisfaction* (or *happiness*) of each player and we propose two cost sharing methods minimizing the maximum or average dissatisfaction of the clients for the classical *minimum spanning tree game*.

Keywords: cost sharing, fairness, minimum cost spanning tree game, Bird rule

1 Introduction

Cooperative Game Theory applies in situations where more than one decision makers are involved such as in the case where a group of decision makers decide to undertake a project together in order to increase (resp. decrease) their total revenue (resp. cost). They have then to solve two problems: i) how to execute their project in an optimal way and ii) how to allocate the revenue/costs among the participants. The second problem is the subject of Cooperative Game Theory which proposes profit/cost allocations taking into account the revenue/cost of all possible coalitions (subsets of the set of participants). Indeed, if one or more players conceive of a proposed allocation as being disadvantageous to them, they can decide to do not participate. Even when an allocation is individually rational, there may be a problem if a group of players figure out that they can do better without working with the others. This is not possible when the revenue cost allocation belongs to the *core* of the game. Thus, one important issue in Cooperative Game Theory consists in searching for a revenue/cost allocation belonging to the core of the game. But, if for some games the situation is problematic because of the *emptynness* of the core, for other games the situation becomes problematic because of the *huge* number of different allocations that belong to the core. In this later case, an individual player even if he has not incentive to do not participate, he may be unhappy with respect to the revenue/cost allocation when comparing with the best (for him) allocation that belongs to the core. Our goal, in this paper, is to introduce new tools allowing to take into account the *dissatisfaction* (or *happiness*) of the players. We thus introduce a new criterion for measuring the dissatisfaction of the players, the *dissatisfaction factor*, and study the problem of providing a solution in the core **and** minimizing the dissatisfaction factor. We illustrate our approach using the classical *minimum spanning tree game*.

1.1 Definitions and Notations

More formally, a *coalition game with transferable payoff* $\langle V, c\rangle$ consists of a finite set V of players, and a function c that associates with every nonempty subset S (a coalition) of V a real number

$c(S)$ which is the cost incurred by the coalition S. The quantity $c(S)$ is the amount that the players in the coalition S have to pay collectively in order to have access to a service. Let x_i, for $i \in V$, be the cost that the player i has to pay. The central question is how to fairly allocate the cost $c(V)$ among the set of players V?

A solution $x = (x_i)_{i \in V}$ belongs to the *core* if no coalition can obtain an outcome better for all its members than the current assignment $(x_i)_{i \in V}$ [9]. In other words, the *core* $\mathcal{C}$ of the coalition game with transferable payoff $\langle V, c \rangle$ is the set of cost vectors $(x_i)_{i \in V}$ such that $\sum_{i \in V} x_i = c(V)$ and $\forall S \subseteq V$ one has $\sum_{i \in S} x_i \leq c(S)$. An equivalent definition is to say that the core is the set of costs $(x_i)_{i \in V}$ for which there is no coalition S and cost vector $(y_i)_{i \in V}$ for which $\sum_{i \in S} y_i = c(S)$ and $y_i < x_i$ for all $i \in S$. Therefore, given a solution in the core, there is no incentive for an agent to leave the grand coalition V.

However, each agent may compare its current cost with the best cost (the smallest one) it could have payed in another solution in the core. Given a solution $(x_i)_{i \in V}$, we define the *dissatisfaction* of agent i as:

$$\Delta_i(x, \mathcal{C}) = \frac{x_i}{\min_{y \in \mathcal{C}} y_i}.$$

Two optimization problems naturally arise:

- MWD: minimize the worst dissatisfaction, i.e. find an allocation $x = (x_i)_{i \in V} \in \mathcal{C}$ which minimizes $\max_{i \in V} \Delta_i(x, \mathcal{C})$.
- MAD: minimize the average dissatisfaction, i.e. find an allocation $x = (x_i)_{i \in V} \in \mathcal{C}$ which minimizes $\sum_{i \in V} \Delta_i(x, \mathcal{C})$.

1.2 The spanning tree game and our contribution

To illustrate our approach we consider the problem of *broadcast routing*. Suppose that a service has to be provided to a set of clients V over a network $G(V^r, E)$. We consider that G is an undirected connected graph. Among all nodes of G, we distinguish the root (the provider), denoted by r. The set of nodes is denoted by V^r while V (the set of clients) denotes $V^r \backslash \{r\}$. Each edge $e \in E$ has a non-negative integral cost c_e. The service can be provided directly to a client or via others. Thus, the minimum substructure that can be used is a tree spanning all clients. This tree has a cost which must be shared by the clients. This is a classical *cooperative game* problem since cooperation may reduce aggregate costs. Formally, one has for any coalition $S \subseteq V$, $c(S) = \min\{\sum_{e \in T_S} c_e \,|\, T_S$ is a spanning tree of $G[S \cup \{r\}]\}$, where $G[S \cup \{r\}]$ denotes the subgraph of G induced by the set of vertices $S \cup \{r\}$.

Cost sharing for this problem has been first addressed by Claus and Kleitman [2], while Bird [1] treated this problem with game theoretic methods and proposed a cost allocation rule known as Bird's rule. It consists in assigning to each client the cost of the edge incident upon him on the unique path from him to the source/provider in a minimum cost spanning tree. Let T_G be the set of all minimum cost spanning trees of G and let C_{opt} be the cost of any tree in this set. Let T be a tree in T_G and v be a vertex in V. There is a unique path between r and v in T. This path uses exactly one edge $[x, v]$, where $x \in V^r \backslash \{v\}$. Let $\beta(T, v) = c_{[x,v]}$ be the cost of this edge. Bird's rule allocates to v the quantity $\beta(T, v)$, i.e. the fee of v is $x_v = \beta(T, v) = c_{[x,v]}$.

Bird's rule ensures that no coalition has incentive to be formed. In fact, the set of allocations $\mathcal{C}_{Bird}$ arising from this rule are always in the core $\mathcal{C}$ of the minimum spanning tree game (see Granot and Huberman [7], [3]). In addition, the set of Bird tree allocations is the unique non-empty solution for the minimum spanning tree game that satisfies three important properties *efficiency*,

leaf consistency and *converse leaf consistency*. (For more details on that see [5], [3].) However, since in general there are more than one minimum cost spanning trees for a given network, this way of dividing the costs does not necessarily lead to a unique cost allocation. Even worse, from an individual point of view, it may lead to a cost allocation that charges a very high fee to a client (or a subset of clients) compared to the one that he would pay with a different minimum cost spanning tree. As a consequence, the dissatisfaction of an agent grows as the price he is charged deviates from the best possible. We therefore define the *dissatisfaction* of v with respect to T and Bird's rule as follows:

$$\Delta_v(T, \mathcal{C}_{Bird}) = \frac{\beta(T, v)}{\min_{T' \in T_G}\{\beta(T', v)\}}.$$

The goal of this work is to design algorithms which provide cost allocations based on Bird's rule in order to ensure that they always belong to the core of the game, while taking into account the happiness of the users. To do so, as stated before, two directions can be typically followed: Minimize the *worst* dissatisfaction over all agents or minimize the *average* dissatisfaction. Therefore, we get two combinatorial optimization problems that we call:

SPANNING TREE-MWD: Find a minimum cost spanning tree that minimizes the *worst* dissatisfaction.

SPANNING TREE-MAD: Find a minimum cost spanning tree that minimizes the *average* dissatisfaction.

We provide polynomial time algorithms for these two problems.

Remark.
Notice that a different notion of fairness has been widely used in the context of bandwidth allocation in network routing, namely the notion of *max-min fairness* (see e.g. [8]). In our context it corresponds to a min-max fairness condition stating that in a fair allocation the maximum fee paid by a client should be as small as possible, then one should make sure that the next largest fee paid by a client should be as small as possible, and so on. In other words, an allocation x is fair if there is no way of decreasing a fee x_i without increasing some other fee x_j such that $x_j \geq x_i$. Notice that for the problem we consider, all allocations found using the rule of Bird have the same degree of fairness with respect to each other, and therefore no discrimination is possible with this notion of fairness. This is true since the multiset containing $\{x_1, \ldots, x_{|V|}\}$ is always the same for any allocation x found with the Bird's rule.

Organization of the paper.
Section 2 study the dissatisfaction factor in the context of the Bird cost allocation. Sections 3 and 4 are respectively devoted to the SPANNING TREE-MWD and SPANNING TREE-MAD problems while Section 5 gives some concluding remarks.

2 The dissatisfaction factor with Bird's rule

Given a tree $T \in T_G$, the fee of a vertex in this tree using Bird's rule depends on the place of the root. This is why it is convenient to deal with arborescences instead of trees (we assume that arborescences are oriented from the root to the leaves). As a consequence, we build from $G = (V^r, E, c)$ a weighted digraph $H = (V^r, A, c)$ as follows: For each ordered pair of nodes $x, y \in V \times V^r$ such that $x \neq y$, put two arcs (x, y) and (y, x) in A if there is an edge $[x, y]$ in E and set $c_{(x,y)} = c_{(y,x)} = c_{[x,y]}$. Thus, for each minimum cost spanning tree of G, there exists a

corresponding minimum cost spanning arborescence rooted in r ($mcsa^r$ for short), B, in H. In the rest of the paper, we only work with arborescences on the digraph H, and we denote by T_H the set of $mcsa^r$'s of H.

In order to determine the dissatisfaction of a vertex $v \in V$ in an arborescence B, it is necessary to know what are the fees it can have to pay (this set is denoted by $F(v)$). It is clear by Bird's rule that $\beta(B, v)$ can only take values in $\{c_{(x,v)} \mid (x, v) \in A\}$. This condition is necessary but not sufficient. The following procedure, denoted by **FEE**, provides the set of fees a vertex could have to pay. It consists in removing some arcs incident upon the vertex and enforcing him to pay a particular price. We note $c_{|A'}$ the set of costs on the subset $A' \subseteq A$.

FEE

Input: A digraph $H = (V^r, A, c)$, a vertex $v \in V$
Step 1: Compute a $mcsa^r$ of H and let C_{opt} be its total cost
Step 2: $L := \{c_{(x,v)} \mid (x, v) \in A\}$
Step 3: $F(v) := \emptyset$
Step 4: *For* each l of L *do*
$A' := A \backslash \{(x, v) \mid c_{(x,v)} \neq l\}$
$H' = (V^r, A', c_{|A'})$
Compute a $mcsa^r$ B' in H'
If B' exists and its total cost is C_{opt}
Then $F(v) := F(v) \cup \{l\}$
End For
Output: $F(v)$

Then, we are able to determine, for each vertex v, the set $F(v)$ of fees it could pay in any arborescence $B \in T_H$. Among these fees, we distinguish the lowest denoted by $F_{min}(v)$ and the largest denoted by $F_{max}(v)$. Computing a $mcsa^r$ can take $\mathcal{O}(mn)$ time (where $m = |A|$ and $n = |V^r|$) [4]. Thus, **FEE** runs in time $\mathcal{O}(mn^2)$.

One can remark that if we run **FEE** on each vertex then one can generate the same arborescences again and again. By keeping track of the costs incurred by each vertex in all arborescences generated by the algorithm, we may save some computation time. However, in terms of worst case analysis, the time complexity of the algorithm remains unchanged.

In the next section, we study the SPANNING TREE-MWD problem for which we propose an optimal polynomial time algorithm.

3 The minimum worst dissatisfaction problem

We study the following problem: Among all spanning arborescences of minimum cost, find one that minimizes the worst dissatisfaction over all vertices. Formally, the SPANNING TREE-MWD problem can be described as follows:

$$\text{argmin}_{B \in T_H} \max_{v \in V} \{\Delta_v(B, \mathcal{C}_{Bird})\}$$

where

$$\Delta_v(B, \mathcal{C}_{Bird}) = \frac{\beta(B, v)}{\min_{B' \in T_H} \{\beta(B', v)\}}.$$

Even though the procedure **FEE** determines all the costs that Bird's rule can assign to the vertices, it is not itself sufficient to find a minimum spanning tree minimizing the worst dissatisfaction.

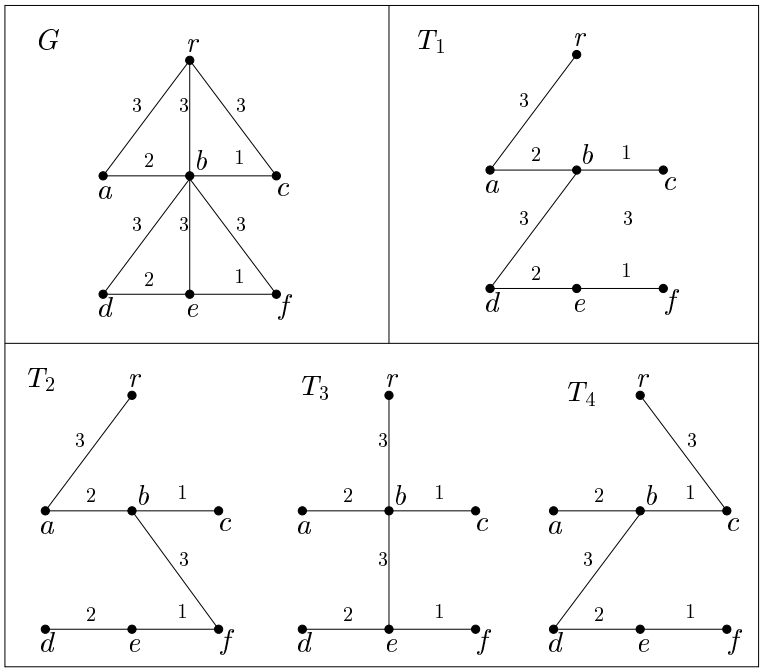

Figure 1: If we run **FEE** on G then only trees T_2, T_3 and T_4 can be generated while T_1, an optimum for the SPANNING TREE-MWD problem, is not built.

To see it, consider the instance given in Figure 1 for which the optimal tree does not belong to the set of arborescences generated by **FEE**.

The quantity $\max_{v \in V}\{F_{max}(v)/F_{min}(v)\}$ is an upper bound on the worst dissatisfaction. The idea of our algorithm is to decrease this upper bound until it reaches the optimal value. To do so, we propose an algorithm which iteratively deletes some arcs of H until any $mcsa^r$ of H is optimal with respect to the worst dissatisfaction.

ALGO 1

Input: A digraph $H = (V^r, A, c)$
Step 1: Compute a $mcsa^r$ B of H and let C_{opt} be its cost
Step 2: For each vertex $v \in V$, compute $F(v)$ with **FEE**
Step 3: $A' := A$
Step 4: Select $v' \in V$ such that $\frac{F_{max}(v')}{F_{min}(v')} = \max_{v \in V}\left\{\frac{F_{max}(v)}{F_{min}(v)}\right\}$
Step 5: $A'' := A'$
Step 6: $A' := A' \backslash \{(x, v') \mid c_{(x,v')} = F_{max}(v')\}$
Step 7: Compute a $mcsa^r$ B' on $H' = (V^r, A', c_{|A'})$
If B' does not exist or its cost is greater than C_{opt}
Then Goto **Step 8**
Else remove $F_{max}(v')$ from $F(v')$
Goto **Step 4**
Step 8: Compute a $mcsa^r$ B'' on $H'' = (V^r, A'', c_{|A''})$
Output: B''

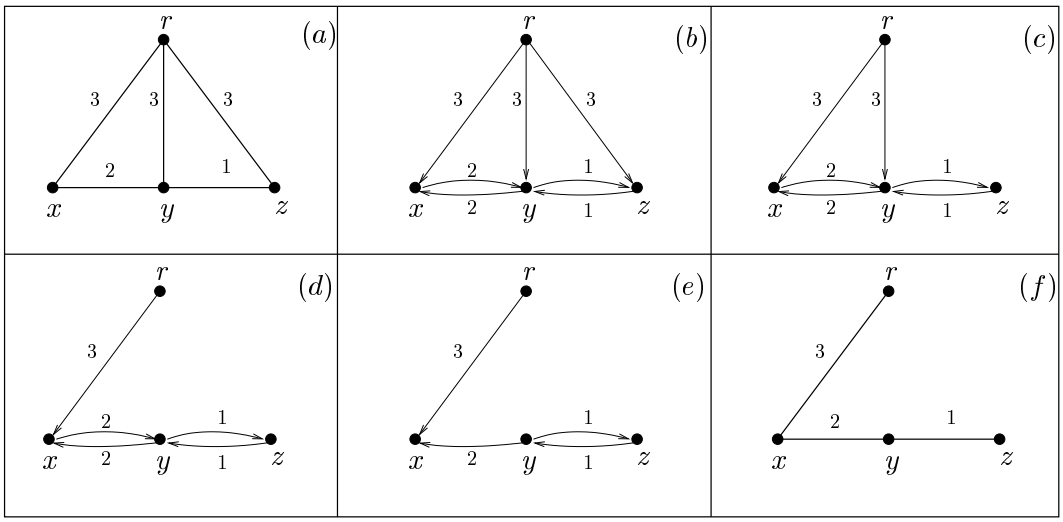

Figure 2: The instance G is given in (a). A minimum cost spanning tree in G has a cost $C_{opt} = 6$. The corresponding digraph H is given in (b). One has $F(x) = \{2, 3\}$, $F(y) = \{1, 2, 3\}$ and $F(z) = \{1, 3\}$, therefore $\frac{F_{max}(x)}{F_{min}(x)} = 3/2$, $\frac{F_{max}(y)}{F_{min}(y)} = 3$ and $\frac{F_{max}(z)}{F_{min}(z)} = 3$. The worst dissatisfaction occurs on nodes y and z. In (c), the algorithm deletes (r, z). With this new instance, it is still possible to compute a $mcsa^r$ with cost 6. One has $\frac{F_{max}(x)}{F_{min}(x)} = 3/2$, $\frac{F_{max}(y)}{F_{min}(y)} = 3$ and $\frac{F_{max}(z)}{F_{min}(z)} = 1$. In (d), the algorithm deletes (r, y). With this new instance, it is still possible to compute a $mcsa^r$ with cost 6. One has $\frac{F_{max}(x)}{F_{min}(x)} = 3/2$, $\frac{F_{max}(y)}{F_{min}(y)} = 2$ and $\frac{F_{max}(z)}{F_{min}(z)} = 1$. The worst dissatisfaction occurs on vertex y. In (e), the algorithm deletes (x, y) but there is no more a $mcsa^r$ with cost 6. Therefore, the algorithm computes on (d) a $mcsa^r$ and returns the corresponding tree (f). Finally, the worst dissatisfaction is 2.

Theorem 1 *The algorithm* **ALGO 1** *gives an optimal solution for the* SPANNING TREE-MWD *problem and runs in polynomial time.*

Proof. Suppose that the minimum worst dissatisfaction is equal to δ^*. Take the original digraph H and, for each vertex $v \neq r$, remove every arc (x, v) such that $c_{(x,v)} > \delta^* F_{min}(v)$. This processing produces a subgraph H^* for which, computing a $mcsa^r$ is possible and anyone of them is optimal for the SPANNING TREE-MWD problem. Suppose **ALGO 1** returns a $mcsa^r$ with worst dissatisfaction $\delta > \delta^*$. This means only arcs (x, v) with cost strictly larger than $\delta F_{min}(v)$ were removed and there is at least one arc (x', v') such that $c_{(x',v')} = \delta F_{min}(v')$. The algorithm stops if the removal of (x', v') leads to one of the following outcomes: No more $mcsa^r$ exists or any $mcsa^r$ has a total cost strictly greater than C_{opt}. However, such a digraph is a subgraph of H^*. We get a contradiction.

Step 2 runs in $\mathcal{O}(mn^3)$ time while the loop between Step 4 and 8 runs in $\mathcal{O}(m^2 n)$ time. □

A complete example is given in Figure 2.

Next section addresses the problem of minimizing the average dissatisfaction over all clients.

4 The minimum average dissatisfaction problem

We study the following problem: Among all spanning arborescences of minimum cost, find one that minimizes the average dissatisfaction. Notice that since the number of vertices is fixed, minimizing

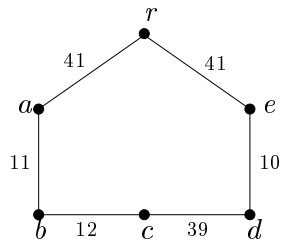

Figure 3: This instance has two possible minimum cost spanning trees. The first one, denoted by T, removes the edge $[r, e]$ and the second one, denoted by T', removes the edge $[r, a]$. For T, the worst dissatisfaction occurs on vertex d and is equal to 3.9 while the average dissatisfaction is $(41/11 + 1 + 1 + 39/10 + 1)/5 \approx 2.12$. For T', the worst dissatisfaction occurs on vertex e and is equal to 4.1 while the average dissatisfaction is $(41/10 + 1 + 39/12 + 12/11 + 1)/5 \approx 2.08$.

the sum or the average dissatisfaction is equivalent, therefore the SPANNING TREE-MAD problem can be described as follows:

$$\text{argmin}_{B \in T_H} \sum_{v \in V} \Delta_v(B, \mathcal{C}_{Bird}).$$

Even though for numerous instances, a solution which minimizes the average dissatisfaction also minimizes the worst dissatisfaction, this is not true in general (see Figure 3 for an example).

The idea of the second algorithm is to give, for each arc (x, y), a weight $w_{(x,y)}$ which is equal to the dissatisfaction of vertex y if this arc belongs to the chosen $mcsa^r$. Therefore we transform the initial graph $G = (V^r, E, c)$ into a digraph $H = (V^r, A, c)$ as follows: For each edge $[x, y]$ of G such that $x \in V^r$, $y \in V$ and $c_{(x,y)} \geq F_{min}(y)$, we add an arc (x, y) in A with cost $c_{(x,y)}$ and weight $w_{(x,y)} = c_{(x,y)}/F_{min}(y)$. It is clear that a spanning arborescence rooted in r which minimizes the weight also minimizes the sum of dissatisfactions. However, it is not necessarily optimal for the cost criterion (see Figure 4 for an example). For the SPANNING TREE-MAD problem, we seek among arborescences that are optimal for the cost, one of minimum weight. The idea of the algorithm is, for each arc (x, y), to combine $c_{(x,y)}$ and $w_{(x,y)}$ into a new composite cost $\tilde{c}_{(x,y)}$ such that any arborescence optimal for this new cost is also optimal for the SPANNING TREE-MAD problem:

$$\tilde{c}_{(x,y)} = \lambda c_{(x,y)} + (1 - \lambda) w_{(x,y)}$$

where $0 < \lambda < 1$. With a λ close enough to 1 but strictly inferior, one can get an optimal solution for the SPANNING TREE-MAD problem. In the following algorithm, λ depends on the instance and is given explicitly.

ALGO 2

Input: A digraph $H = (V^r, A, c)$
Step 1: For each vertex $v \in V$, compute $F(v)$ with **FEE**
Step 2: Compute a $mcsa^r$ B of H and let $D = \sum_{v \in V} \Delta_v(B, \mathcal{C}_{Bird})$
Step 3: $\lambda := \frac{D}{D+1}$
Step 4: Compute a $mcsa^r$ B^* of $H^* = (V^r, A, \tilde{c})$
Output: B^*

Theorem 2 *The Algorithm* **ALGO 2** *gives an optimal solution for the* SPANNING TREE-MAD *problem and runs in polynomial time.*

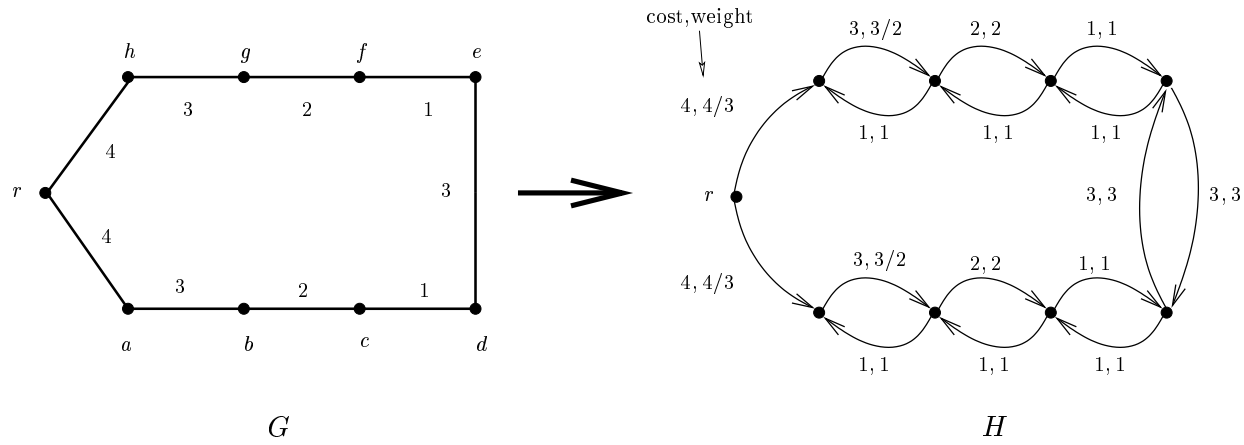

Figure 4: This instance only has two minimum cost spanning trees: T which removes the edge $[r, a]$ and T' which removes the edge $[r, h]$. They both have an average dissatisfaction of $(4/3+3/2+2+1+3+1+1+1)/8 \approx 1.47$. Nevertheless, the tree T'' which removes the edge $[e, d]$ has an average dissatisfaction of $(4/3+3/2+2+1+1+2+3/2+4/3)/8 \approx 1.45$ but T'' is not a minimum cost spanning tree.

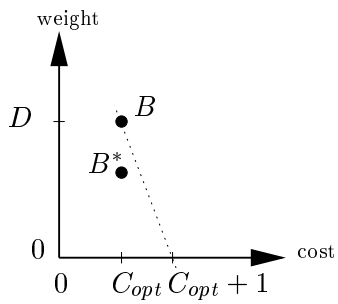

Figure 5: Illustration of the proof.

Proof. The proof is illustrated on Figure 5. Consider B^* the solution returned by **ALGO 2** and B computed at Step 2. We assume that B^* has a cost C^*_{opt} and average dissatisfaction $\frac{D^*}{|V|}$ while B has a cost C_{opt} and average dissatisfaction $\frac{D}{|V|}$. Since B^* is optimal for H^*, one has $\frac{D}{D+1}C^*_{opt} + \frac{1}{D+1}D^* \leq \frac{D}{D+1}C_{opt} + \frac{1}{D+1}D$ which means B^* is below the dotted line on Figure 5. Since costs are integral, any feasible solution below the dotted line also has a cost C_{opt}. Thus, B^* has a cost C_{opt} and minimizes the average dissatisfaction.

The running time of **ALGO 2** is $\mathcal{O}(mn^2)$ because of **FEE**. □

A complete example of **ALGO 2** on the instance of Figure 4 is given on Figure 6. One can remark that **ALGO 2** can also work with costs that are not integral. A *second best* solution s' is nearly optimal and its cost $C_{opt} + \epsilon$ is such that no feasible solution s'' with cost $C_{opt} + \delta$ verifies $0 < \delta < \epsilon$. For the minimal weight spanning tree problem, a second best tree with cost $C_{opt} + \epsilon$ can easily be computed [6]. Thus, by replacing $\lambda := \frac{D}{D+1}$ by $\lambda := \frac{D}{D+\epsilon}$ in **ALGO 2** we get the result.

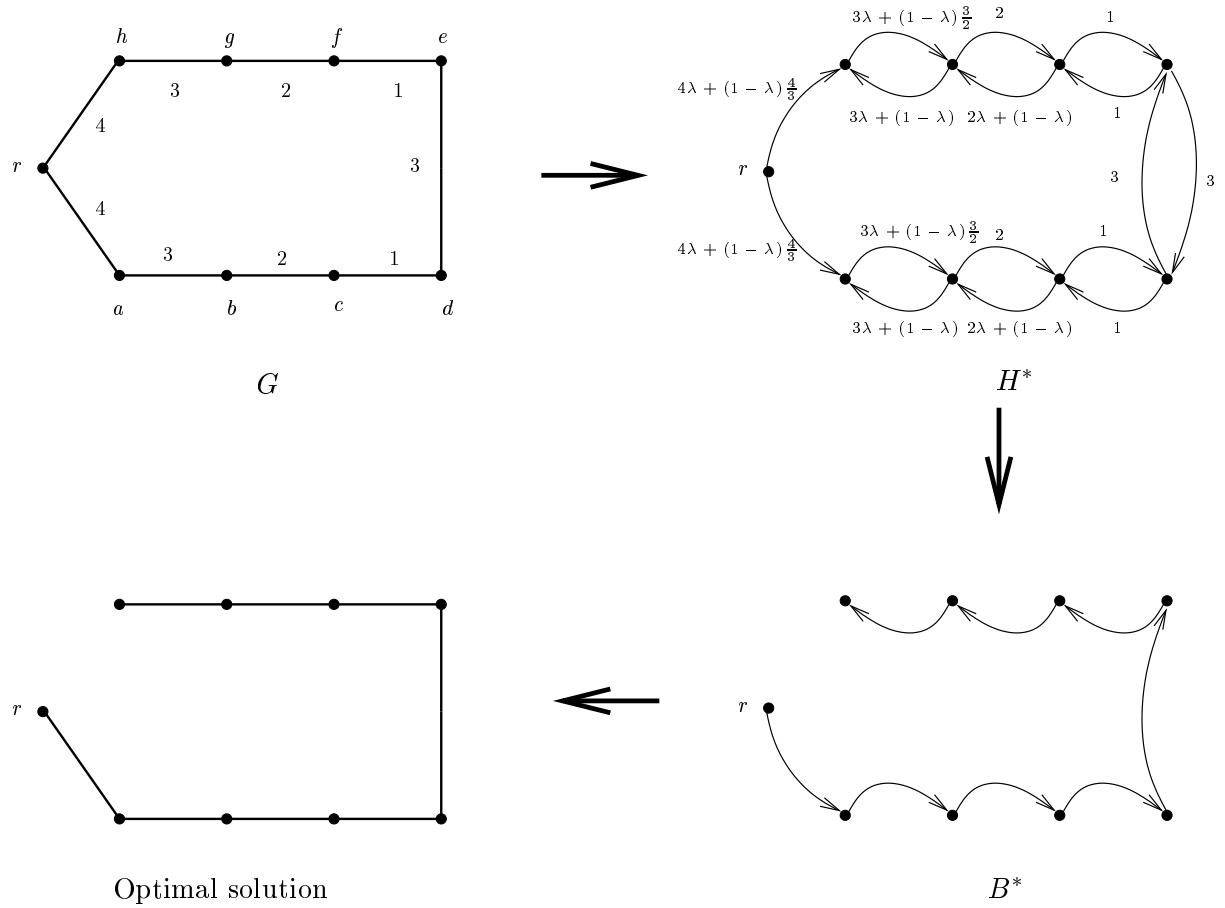

Figure 6: The list of possible fees is first computed: $F(a) = F(h) = \{3, 4\}$, $F(b) = F(g) = \{2, 3\}$, $F(c) = F(f) = \{1, 2\}$, $F(d) = F(e) = \{1, 3\}$. A $mcsa^r$ B is determined and its sum of dissatisfactions is $D = 4/3 + 3/2 + 1 + 3 + 1 + 1 + 1 = \frac{71}{6}$. One has $\lambda = D/(D+1) = 71/77$. H^* is then built and a $mcsa^r$ B^* is computed.

5 Concluding remarks

Fairness is subjective and especially in cost sharing problems, this notion, often conflicting in different situations, is not clearly established. In this work, we introduced a new measure taking into account the dissatisfaction/happiness of the agents and we applied this concept on a classical broadcast routing problem, which corresponds to the classical minimum spanning tree game. Without being restricted to our specific problem, we think that the dissatisfaction factor can be studied for a wide range of problems and particularly cost sharing ones.

We proposed two algorithms which allow us to minimize the worst or the average dissatisfaction and mentioned that an optimal solution for the first problem may differ from an optimal solution for the second problem. Therefore, a natural question arises: How far an optimal solution for the SPANNING TREE-MWD problem is from an optimal solution for the SPANNING TREE-MAD problem and vice et versa?

With a simple example, one can see that an optimal solution for the SPANNING TREE-MAD problem can have a worst dissatisfaction factor that is twice the best one (see Figure 7). Symmetrically, an optimal solution for the SPANNING TREE-MWD problem can have an average dissatisfaction factor that is twice the best one (see Figure 8). However, the question of whether in general these ratios are bounded by a constant remains open.

Finally, notice that the set of allocations found using Bird's rule, $\mathcal{C}_{Bird}$, is strictly included in the core $\mathcal{C}$ of the spanning tree game. Therefore the question of finding an allocation $x \in \mathcal{C}$ minimizing ($\max_i$ or $\sum_i$) $\Delta_i(x, \mathcal{C})$, instead of finding an $x \in \mathcal{C}_{Bird}$ minimizing ($\max_i$ or $\sum_i$) $\Delta_i(x, \mathcal{C}_{Bird})$, remains open.

References

[1] C. Bird. On cost allocation for spanning tree: a game theoretic approach. *Networks*, 6:335–350, 1976.

[2] A. Claus and D. Kleitman. Cost allocation for spanning tree. *Networks*, 3:289–304, 1973.

[3] I. Curiel. *Cooperative Game Theory and Applications.* Kluwer Academic Publishers, 1997.

[4] J. Edmonds. Optimum branchings. *Journal of Research of the National Bureau of Standards B 71*, pages 233–240, 1967.

[5] V. Feltkamp, S.H. Tijs, and S. Muto. Bird's tree allocation revisited. Technical Report CentER Discussion Paper 9435, Tilburg University, Tilburg, The Netherlands, 1994.

[6] H.N. Gabow. Two algorithms for generating weighted spanning trees in order. *SIAM Journal of Computing*, 6(1):139–150, 1977.

[7] D. Granot and G. Huberman. On minimum cost spanning tree games. *Mathematical Programming*, 21:1–18, 1981.

[8] J. Kleinberg, E. Tardos, and Y. Rabani. Fairness in routing and load balancing. In *Proceedings 40th Annual IEEE Symposium on Foundations of Computer Science*, pages 568–578, 1999.

[9] M.J. Osborne and A. Rubinstein. *A course in game theory.* The MIT Press, 1994.

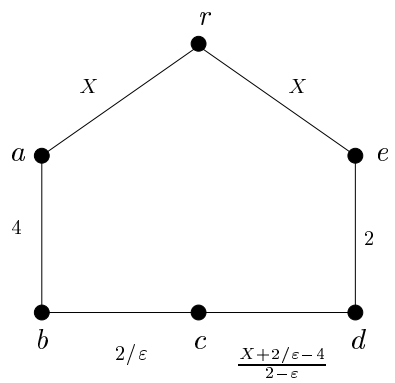

Figure 7: Let X be a very large value and ε be between 0 and 0.5. This instance shows that an optimal solution for the SPANNING TREE-MAD problem can have a worst dissatisfaction which is $2-\varepsilon$ times the best possible. Two minimum cost spanning trees exist: T which removes the edge $[r,e]$ and T' which removes the edge $[r,a]$. They have the same average dissatisfaction but T has a worst dissatisfaction $\max\{X/4, \frac{X+2/\varepsilon-4}{4-2\varepsilon}\}$ while T' has a worst dissatisfaction $X/2$. When X tends to infinity, the ratio between these two worst dissatisfactions is above $2-\varepsilon$.

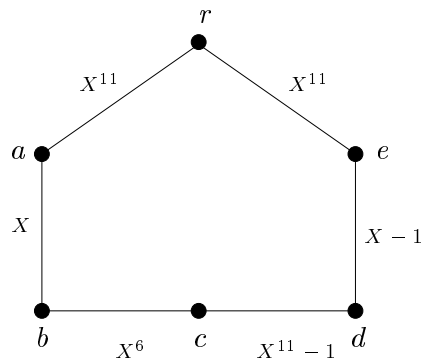

Figure 8: Let X be an integer ≥ 2. This instance shows that an optimal solution for the SPANNING TREE-MWD problem can have an average dissatisfaction which is 2 times the best possible. Two minimum cost spanning trees exist: T which removes the edge $[r,a]$ and T' which removes the edge $[r,e]$. The tree T' is optimal for the SPANNING TREE-MWD problem and has an average dissatisfaction $(X^{10}+1+1+(X^{11}-1)/(X-1)+1)/5$ while T has an average dissatisfaction $(X^{11}/(X-1)+1+(X^{11}-1)/X^6+X^5+1)/5$. When X grows to infinity, the ratio between these two average dissatisfactions tends to 2.