

Juarez, Ruben; Wu, Michael

**Article**

## Routing-proofness in congestion-prone networks

Games

**Provided in Cooperation with:**

MDPI – Multidisciplinary Digital Publishing Institute, Basel

*Suggested Citation:* Juarez, Ruben; Wu, Michael (2019) : Routing-proofness in congestion-prone networks, Games, ISSN 2073-4336, MDPI, Basel, Vol. 10, Iss. 2, pp. 1-18, <https://doi.org/10.3390/g10020017>

This Version is available at:

<https://hdl.handle.net/10419/219240>

**Standard-Nutzungsbedingungen:**

Die Dokumente auf EconStor dürfen zu eigenen wissenschaftlichen Zwecken und zum Privatgebrauch gespeichert und kopiert werden.

Sie dürfen die Dokumente nicht für öffentliche oder kommerzielle Zwecke vervielfältigen, öffentlich ausstellen, öffentlich zugänglich machen, vertreiben oder anderweitig nutzen.

Sofern die Verfasser die Dokumente unter Open-Content-Lizenzen (insbesondere CC-Lizenzen) zur Verfügung gestellt haben sollten, gelten abweichend von diesen Nutzungsbedingungen die in der dort genannten Lizenz gewährten Nutzungsrechte.

**Terms of use:**

*Documents in EconStor may be saved and copied for your personal and scholarly purposes.*

*You are not to copy documents for public or commercial purposes, to exhibit the documents publicly, to make them publicly available on the internet, or to distribute or otherwise use the documents in public.*

*If the documents have been made available under an Open Content Licence (especially Creative Commons Licences), you may exercise further usage rights as specified in the indicated licence.*



<https://creativecommons.org/licenses/by/4.0/>

Article

# Routing-Proofness in Congestion-Prone Networks

Ruben Juarez \*  and Michael Wu

Department of Economics, University of Hawaii, 2424 Maile Way, Saunders Hall 542, Honolulu, HI 96822, USA; siukin@hawaii.edu

\* Correspondence: rubenj@hawaii.edu

Received: 31 December 2018; Accepted: 25 March 2019; Published: 3 April 2019



**Abstract:** We consider the problem of sharing the cost of connecting a large number of atomless agents in a network. The centralized agency elicits the target nodes that agents want to connect, and charges agents based on their demands. We look for a cost-sharing mechanism that satisfies three desirable properties: efficiency which charges agents based on the minimum total cost of connecting them in a network, stand-alone core stability which requires charging agents not more than the cost of connecting by themselves directly, and limit routing-proofness which prevents agents from profitable reporting as several agents connecting from A to C to B instead of A to B. We show that these three properties are not always compatible for any set of cost functions and demands. However, when these properties are compatible, a new egalitarian mechanism is shown to satisfy them. When the properties are not compatible, we find a rule that meets stand-alone core stability, limit routing-proofness and minimizes the budget deficit.

**Keywords:** cost sharing; core stability; routing proofness

## 1. Introduction

Congestion is a form of negative externality to the networks of any size. These networks can include but are not limited to road networks, internet networks, telecommunication networks as well as utility power grid networks and other networks [1–6]. The emergence of congestion can be attributed to the instance in which a large number of selfish agents, each wishing to route traffic via one's preferred links (or edges), without taking into account the actions of other agents in the network [7]. Such selfish routing behavior generates discrepancy between the socially optimal traffic pattern (minimizing overall delay or cost) and the selfish-routing behavioral outcome (also known as the non-cooperative Nash equilibrium). [8,9] suggest that the non-cooperative Nash equilibrium need not in general optimize the social welfare (minimizing overall delay or cost in this context). Indeed, the non-cooperative Nash equilibrium could generate a loss substantially higher than the social optimum. This implies that the discrepancy could cause some unwanted direct or indirect consequences such as network congestion, traffic paralysis as well as social welfare loss.

An earlier work by Roughgarden [8] investigates and quantifies the efficiency loss of the network performance due to the self-interested agents. The degradation is calculated by the worst-possible ratio between the Nash equilibrium and the social optimal traffic and is called the price of anarchy (POA) [8]. The closer the POA is to one, the less efficiency loss it generates between the Nash equilibrium and social optimum. Roughgarden shows that if the cost (latency) functions on the links in the network are linear, then the POA is equal to  $\frac{4}{3} \approx 1.333$ ; if the cost functions are cubic, then the POA is approximately 1.896; and if the cost functions are polynomials of degree  $\leq p$  where  $p > 0$ , then POA is asymptotically

equal to  $\theta \left( \frac{p}{\ln p} \right)$  as  $p \rightarrow \infty$ <sup>1</sup>. Finally, if the cost functions are unrestricted to any allowable class of functions (e.g., linear, quadratic, cubic, polynomials, etc.), then the POA is unbounded. These results indicate that the efficiency loss can be immensely large and the consequences generated by the selfish routing behavior can be extremely severe in the network.

In order to avoid such consequences, it is therefore important to manage the congestion of a network. In this paper, we consider a network with a set of nodes in which every node interconnects with at least one other node by a link. There is a variable congestion cost associated with each link that depends on the units of demands sent by agents. The cost may represent any maintenance cost or accessing fee of the links. Different links may have different congestion costs. There are a large number of self-interested agents who wish to route their traffic from an arbitrary node to another node using the links in the network. We assume the self-interested agents are interested in routing only via links that will incur the minimal cost or delay. In addition, we also include the possibility for a single agent performing a special type of strategic maneuver (or similarly, routing maneuver). The concept of strategic maneuver was first introduced by Dominique and Moulin [5], and followed by Moulin [10], who study mechanisms where coalition of agents in a large network can furtively create several alias and misreport themselves as several agents. Formally, that means an agent may report and route her traffic to the target node as two or more agents in hope of lowering her connection cost. To illustrate such scenario, we suppose there is an agent with initial node  $i$  and target node  $j$ . Then, she may report herself as two agents first connecting from node  $i \leftrightarrow k$ , then from node  $k \leftrightarrow j$  if she finds such strategic maneuver is less costly than connecting directly from node  $i \leftrightarrow j$ .

Unlike the decentralized networks such as road networks or internet networks, the networks being considered in this paper are operated in a centralized manner. We imagine there is a central authority that possesses the power to elicit the demand characteristics from all agents in the networks and is capable of computing and implementing the optimal traffic pattern at any time. Meanwhile, we imagine there is a centralized agency that will help satisfy the needs demanded by the agents at minimal total delay or cost. Since agents are self-interested, they need not have the incentive to reveal their true routing schemes. The centralized agency is therefore vulnerable to any misreports by individual units, or by a coalition of subgroup of units. Such strategic maneuver behavior will potentially distort the socially optimal traffic pattern (minimizing overall cost or delay) in the sense that links with relatively lower cost are more likely to be congested, which may in turns incurring higher total cost or delay.

Therefore, in order to prevent effectively such strategic maneuvers, thereby improving the performance of a network, there is a need to design cost-sharing mechanisms to charge on each agent in the network so that the action of strategic maneuver will make no difference in terms of lowering connection costs. Moulin calls any cost-sharing mechanisms or cost-sharing rules routing-proof if they are invulnerable to such strategic maneuvers [10]. Technically, it is not as difficult to find a routing-proof mechanism, if routing-proofness is the only constraint that we are aiming to satisfy. For instance, the simple average cost mechanism (AC) will do<sup>2</sup>. By splitting the total cost equally among all agents in the network, each agent is to pay a share of the average cost. As a result, any agent who intends to report as two or more agents will simply have to pay double or more of the average cost. Nevertheless, this mechanism may not be charging every agent or coalition fairly if the average cost exceeds the cost of a subnetwork meeting all connection needs of the group (stand-alone objections). This fairness-related constraint is known as stand-alone core stability and is the primary design constraint in the minimum cost spanning tree problem [10] and a variety of other problems.

<sup>1</sup> See [8] for more POAs of different classes of functions such as the quadratic, M/M/1 delay functions and M/G/1 delay functions.

<sup>2</sup> The average cost mechanism has been used successfully in a variety of network problems, e.g., [2,11].

Our work, therefore, has the objective to design or propose a cost-sharing mechanism satisfying the following three properties:

- Efficiency: the minimum total cost of connecting all agents to their target nodes;
- Stand-alone core stability: no groups of agents pay more than the cost of a subnetwork meeting all connection needs of the group;
- Routing-proofness: no agent can lower its cost by reporting as several agents along an alternative path connecting her target nodes.

### 1.1. Innovation of the Paper and Overview of the Results

While efficiency and stand-alone core stability are standard properties in the literature (see related literature section), the main innovation of this paper comes from studying the three properties together in a large economy with congestion. Previous work by Moulin [10] studied these properties in a model without congestion cost, where the cost associated with each link is fixed and the costs are independent of the units of traffic sent by agents. This assumption, while convenient, is not realistic in a variety of networks such as traffic, telecommunication and other large and complex networks (e.g., the internet). Our paper extends the model introduced by Moulin [10] to the case with variable congestion costs that depend on the units of traffic being connected in our network. The adoption of variable congestion costs substantially increases the difficulty of computing the efficiency of a general network. In particular, computing the social optimum (efficiency) of a general network is Non-deterministic Polynomial-time-hard (NP-hard) [12], unless the cost functions satisfy certain conditions (e.g., linearity or all cost functions are identical in the network [12]). Thus, our work is sharply different from [10] where the minimum cost (efficiency) can easily be computed by applying a simple greedy algorithm [13].

The second innovation of the paper comes from studying the case where there are a large number of self-interested agents in the network such that each of them only controls a negligible fraction of the overall traffic (i.e., agents have zero mass). These types of assumptions are critical for the internet and other large and complex networks, where even if an agent reroutes in the network, then because the amount of traffic each agent controls is infinitesimally small relative to the overall traffic, the costs charged to agents should not be too sensitive and changing significantly. This assumption is also relevant for rerouting in airline networks, where agents wishing to fly from A to B, can instead buy two one-way tickets, from A to C and C to B, if proven cheaper than a direct ticket from A to B. One agent manipulating the airline system will have no effect on prices and a substantially large amount of population is needed for changes in prices to take effect. Furthermore, in practice, it is relatively hard to coordinate a large group of agents that change prices in a network. In order to address this feature, we introduce a new axiom, the limit routing-proofness, that adapts Moulin's concept of routing-proofness to the case where agents are extremely small relative to the overall traffic. We study the limitation of such an axiom in a congestion-prone network. In particular, we provide the necessary and sufficient conditions for the existence of rules that satisfy limit routing-proofness along with efficiency and stand-alone core stability (Theorem 1).

A third innovation of the paper comes from actually introducing a mechanism that satisfies these three properties when they are compatible. Our mechanism builds on perhaps the most studied mechanism in the cost-sharing literature, the uniform rule [14,15], and adapt it to our network setting with congestion in a way that Limit routing-proofness is satisfied. In order to compute the outcome of the mechanism, we actually provide an algorithm to compute the prices that agents pay, and show that such prices coincide with the cost-shares of our new mechanism (Proposition 1).

### 1.2. Related Literature

The stand-alone core stability and efficiency are two traditional properties in the network literature [4,10,16,17]; while routing-proofness is a relatively more recent idea [5,10,18]. The predecessor

of routing-proofness was first coined the axiom of no transit originated by Hanriet and Moulin in [5]. In [5], Hanriet and Moulin suggest two cost-allocation methods: private-cost and external cost methods that satisfy the axiom of additivity, sustainability and no transit. These methods work only for the case without congestion.

The closest paper to our work is Moulin [10], who studies a congestion-free network cost-sharing problem and where the no transit axiom is renamed as routing-proofness. The paper constructs two core stable and routing-proof cost-sharing rules: weighted Shapley value (WSH) and weighted spanning rule (WSP) together satisfying the stand-alone core stability and routing-proofness. The first family of the rules requires only that the connecting costs to be zero or one, whereas the second family allows for arbitrary costs but requires a spanning traffic. As such, neither of these rules work in our general congestion-prone setting.

Another related paper is a real-world application of routing-proofness by Dong, Guo and Wang [18]. In there, a highway toll pricing method is proposed to assign the toll charges to different types of agents in a linear network. The proposed method satisfies the stand-alone test for vehicles, cost recovery (for road sections with no cross-subsidizing) and a version of routing-proofness in the sense that no agents can reduce the toll charges by exiting and re-entering the highway consecutively [18]. In contrast with the papers above, the model presented in [18] is a congestion-prone network rather than congestion-free, but the proposed method only works in a linear network, whereas our networks would typically have cycles, and that is where the strength of our routing-proofness property comes from.

Related work also comes from the cost-sharing literature on the implementation of efficient costs. Juarez and Kumar [2] discuss the appropriateness on the implementation of various efficient cost-sharing mechanisms when agents choose their paths strategically as if they are in a routing game. A comparison of different classes of mechanisms such as the average cost mechanism, proportional to stand-alone mechanism, (weighted) egalitarian mechanism and the Shapley mechanism is done. It is important to note that the paper does not deal with routing-proofness issues, and as such, all the mechanisms studied there fail to meet our test.

Other implementation work include Hougaard and Tvede [19,20], who characterize cost-minimizing networks at the equilibrium of games by changing the announcement rule. The sequential sharing of costs and profits is studied axiomatically in Juarez et al. [11]. Kumar [21] also investigates secure implementation in cost-sharing models without congestion, while Hougaard et al. [22] study the sharing of profits in trees and Moulin et al. [23] study the division of the costs/profits of non-redundant items. In contrast with this literature, our work is the first to address from an axiomatic view the division of costs in congestion-prone networks by atomless agents.

## 2. The Model

A network cost-sharing problem (or simply a problem) is a triple  $P = \langle N, C, \theta \rangle$ , where  $N = (V, L)$  is a network with finite set of nodes  $V = \{n_1, \dots, n_r\}$  and a finite set of undirected links  $L = \{l_1, \dots, l_s\}$  connecting such nodes.  $C = (C_1(\cdot), \dots, C_s(\cdot))$  is a vector of cost functions associated to each link, where the cost function  $C_i : \mathbb{R}_+ \rightarrow \mathbb{R}_+$  associated to link  $l_i$  is strictly increasing and strictly convex. The vector  $\theta = (\theta_1, \dots, \theta_s) \in \mathbb{R}_+^s$  is the traffic demand associated to each link, where  $\theta_i$  representing the units of traffic demanded on link  $l_i$  (i.e.,  $\theta_i$  represents the mass of the agents interested in connected the two nodes of link  $l_i$ ). We denote by  $\mathcal{P}$  the set of all network cost-sharing problems.

For an arbitrary network cost-sharing problem  $P$ , the link  $l_i$  will be simply denoted by  $i$  if there is no confusion. Given the cost function  $C_i$  associated to link  $i$ , we denote by  $AC_i(x) = \frac{C_i(x)}{x}$  the average

cost associated to such link. Let  $\mathcal{P}_i = \{p_1, \dots, p_t\}$  be the set of all feasible paths without cycles that connect link  $i$  in the network.<sup>3</sup>

**Definition 1.** (a) Given a path  $k$  and  $\alpha > 0$ , we denote by  $\alpha \cdot k = (v_1, \dots, v_s) \in \mathbb{R}^s$  such that  $v_i = \alpha$  if  $i \in k$  and  $v_i = 0$  if  $i \notin k$ .

(b) Given any link  $i \in L$  and a demand  $\theta_i$  for that link, a connection vector  $(w_1^i, \dots, w_s^i)$  generates demand  $\theta_i$  if there exists non-negative weights  $\{\alpha^k\}_{k \in \mathcal{P}_i}$  such that  $\sum_{k \in \mathcal{P}_i} \alpha^k = \theta_i$  and  $(w_1^i, \dots, w_s^i) = \sum_{k \in \mathcal{P}_i} \alpha^k \cdot k$ .

(c) Given demands  $(\theta_1, \dots, \theta_s)$ , a connection vector  $(w_1, \dots, w_s)$  is feasible if for every link  $i \in \{1, \dots, s\}$  there exists a connection vector  $(w_1^i, \dots, w_s^i)$  that generates demand  $\theta_i$  such that  $(w_1, \dots, w_s) = \sum_{i=1}^s (w_1^i, \dots, w_s^i)$ .

While agents demanding link  $i$  are interested in connecting the nodes of such link, this connection can be achieved directly via link  $i$  or indirectly via alternative paths connecting link  $i$ . In particular, the demands of traffic  $\theta_i$  on link  $i$  can be split into multiple alternative paths where each particular path indirectly connects link  $i$  (part (b) in definition above). A connection vector  $(w_1, \dots, w_s)$  is said to be feasible if it is the sum of all these paths.

**Definition 2.** The total cost (or simply a cost) of a feasible connection vector  $(w_1, \dots, w_s)$  is  $C(w_1, \dots, w_s) = \sum_{i=1}^s C_i(w_i)$ .

With each element  $w_i$  represents all possible amounts of traffic on link  $i$ , the total cost of connecting all agents with connection vector  $(w_1, \dots, w_s)$  is the sum of each cost function  $C_i(\cdot)$  evaluated at  $w_i$  for all  $i = 1, \dots, s$ .

Recall that we are in a centralized setting where the planner has the ability to select the paths of some agents demanding a given link. A natural objective for the planner is to select a path that minimizes the cost. This is incorporated in the following definition.

**Definition 3.** Given a set of demands  $(\theta_1, \dots, \theta_s)$ , a feasible connection vector  $(w_1^*, \dots, w_s^*)$  is efficient if for any other feasible connection vector  $(\tilde{w}_1, \dots, \tilde{w}_s)$ , we have  $C(w_1^*, \dots, w_s^*) \leq C(\tilde{w}_1, \dots, \tilde{w}_s)$ .

We say that a feasible connection vector of a given set of demands is efficient if there exists a feasible connection vector  $(w_1^*, \dots, w_s^*)$  such that the total cost evaluated at  $(w_1^*, \dots, w_s^*)$  is always less than or equal to the total cost evaluated any other connection vectors  $(\tilde{w}_1, \dots, \tilde{w}_s)$ . Such connection vector will generate the least cost (also referred as efficient cost) of serving all the agents in the network.

**Definition 4** (Mechanism and efficient mechanism). (a) A cost-sharing mechanism (also called rule or solution) is a continuous function  $\xi : \mathcal{P} \rightarrow \mathbb{R}_+^s$  that allocates a set of cost shares for the use of every link at every network cost-sharing problem. (b) A cost-sharing mechanism is efficient at problem  $\mathbf{P}$  if it divides the efficient cost among agents:

$$\sum_{i \in L} \xi_i(\mathbf{P}) \cdot \theta_i = C^*(\mathbf{P})$$

where  $C^*(\mathbf{P})$  is the efficient cost at a problem  $\mathbf{P}$ .

We focus on a mechanism that allocates payments based on the demands of single links. That is, the planner cannot distinguish the names of agents who demand the same link. This is a standard assumption in the cost-sharing literature. An efficient cost-sharing mechanism is a centralized mechanism that meets the demands efficiently and redistribute this cost based only on the demand of

<sup>3</sup> A path is a sequence of consecutive nodes in the network. We say that a path has no cycles if the nodes do not repeat. We say that a path connects link  $i$  if the path starts in one of the nodes of link  $i$  and ends in the other. When there is no confusion, we also refer to a path by its links instead of its nodes.

every link. Thus, the sum of the charges assigned by the mechanism, when multiplied by the demands of each link, is equal to the efficient cost at a given problem.

**Definition 5** (Stand-alone core stability). *Consider a network cost-sharing problem  $\mathbf{P} = \langle N, C, \theta \rangle$ . A cost-sharing mechanism  $\xi$  satisfies the stand-alone core stability at  $\mathbf{P}$  if for any link  $i$ , we have that  $\xi_i(\mathbf{P})\theta_i \leq C_i(\theta_i)$ .*

Definition 5 ensures the absence of stand-alone objections from agents caused by overcharging from the implemented mechanism. This constraint plays an important role in terms of fairness to prevent agents from being charged more than the cost of the links they actually demanded. Moreover, it also helps to prevent agents who demand inexpensive links from subsidizing those who demand expensive links in a network.

**Definition 6** (Limit routing-proofness). *Consider a network cost-sharing problem  $\mathbf{P} = \langle N, C, \theta \rangle$ . A cost-sharing mechanism is limit routing-proof (LRP) at  $\mathbf{P}$  if for each link  $i$ , and each path  $p_k \in \mathcal{P}_i$  that connects link  $i$  we have that:*

$$\xi_i(\mathbf{P}) \leq \sum_{j \in p_k} \xi_j(\mathbf{P}). \quad (1)$$

Limit routing-proofness prevents agents from routing their demand along alternative paths that may also meet their demand. In particular, at a problem  $\mathbf{P}$ , by Equation (1), an agent demanding link  $i$  pays  $\xi_i(\mathbf{P})$ . If such an agent decides to demand an alternative path  $p_k$  that also connects  $i$  (by posing as separate agents along path  $p_k$ ) his charge will be  $\sum_{j \in p_k} \xi_j(\mathbf{P})$ , which is not smaller than  $\xi_i(\mathbf{P})$  under limit routing-proof (LRP).

### 3. Efficiency, Stand-Alone Core Stability and LRP

The trade-off between stand-alone core stability and incentive compatibility (here, interpreted as routing-proofness) for efficient mechanisms has already been identified in the literature. Indeed, in a related network model where the connection demands of agents need to be met, Juarez and Kumar [2] study two mechanisms: the average-cost mechanism (AC) that divides the cost of the efficient network equally among all the agents and the proportional to stand-alone mechanism (PR) that divides the cost of the efficient network in proportion to the cost of their demands.<sup>4</sup> In our setting, the AC mechanism meets LRP, as any agent pays the same cost-share regardless of the link demanded, thus an agent splitting their demand in two or more links will pay twice or more their cost-share. Unfortunately, it is easy to see that AC does not meet the stand-alone core stability, as it might happen that agents demanding a very inexpensive link cross-subsidize agents demanding very expensive links. PR is an exact opposite situation, meeting stand-alone stability but failing to meet LRP. Therefore, neither of them meet our three objectives. Other traditional mechanisms, like Aumann–Shapley, also fail to meet our demanding LRP. Therefore, in addressing the compatibility between stand-alone core stability, efficiency and limit routing-proofness, new mechanisms need to be developed.

One of the most studied rules in the cost-sharing literature is the uniform rule [14,15,24], which is a variation of AC that respect stand-alone core stability. This rule can be trivially adapted to our network setting by considering the cost of each link as the demand of the agents.<sup>5</sup> Both the uniform rule and AC divide the total cost of the network across all agents, however, their similarity begins to diverge when the charges reach the threshold of the stand-alone cost of a link (presumably the smallest stand-alone cost first). In that case, AC will disregard the stand-alone cost and continue increasing

<sup>4</sup> For brevity, we do not repeat the formal definition of these mechanisms already discussed in Juarez and Kumar [2].

<sup>5</sup> See [2,25] for the application of the Uniform rules to network problems.



charges to agents on each link  $i$  equally without bound. In contrast, the uniform rule will stop assigning charges to agents whose links have reached their stand-alone costs. The remaining cost will then be covered by the links in which the stand-alone costs are yet to be reached. Note that this characteristic will definitely preserve the property of stand-alone core stability. Unfortunately, the uniform rule does not meet LRP. The next section provides a variation of the uniform rule that meets LRP, stand-alone core stability and efficiency, whenever possible. We call it the egalitarian mechanism.

### 3.1. A New Egalitarian Mechanism

Given a network cost-sharing problem  $P = \langle N, C, \theta \rangle$ , our ultimate objective is to find a cost-sharing mechanism such that the set of cost shares allocated by the mechanism will be efficient, limit routing-proof and core stable. In order to achieve that, we formulate the network cost-sharing problem along with the constraints into a set of system of equations.

$$\sum_{i=1}^s \lambda_i \theta_i = EFF(\theta_1, \dots, \theta_s) \tag{2}$$

$$\lambda_i \theta_i \leq C_i(\theta_i) \quad \text{for all } i = 1, \dots, s \tag{3}$$

$$\lambda_i \leq \sum_{l \in p_k} \lambda_l \quad \text{for all } p_k \in \mathcal{P}_i \text{ and for all } i = 1, \dots, s \tag{4}$$

$$\lambda_i \geq 0 \quad \text{for all } i = 1, \dots, s. \tag{5}$$

We denote  $\lambda_i$  the payment charged to each agent on link  $i$ . Other variables are defined exactly the same way as the model in Section 2.  $EFF(\theta_1, \dots, \theta_s)$  represents the efficient cost of the network computed as a function of  $(\theta_1, \dots, \theta_s)$ . We express efficiency in this form because computing the efficiency (social optimality) of a network is generally a NP-hard problem.<sup>6</sup>

Equation (2) requires that the sum of the charges, when multiplied by the demands over all links  $i = 1, \dots, s$  is exactly equal to the efficiency of the network. In particular, this constraint represents the efficiency property.

Equation (3) describes the stand-alone core stability for all agents demanding link  $i$ . Essentially, this constraint says that the amount of assigned charges to each agent on link  $i$  should never exceed the cost of connecting the target pairs by themselves via link  $i$ .

Equation (4) represents the limit routing-proofness constraint for a general network. Recall that for any link  $i$ ,  $\mathcal{P}_i$  is the set of all feasible alternative paths  $\{p_1, \dots, p_t\}$  that connect link  $i$ . The limit routing-proofness constraint then implies that the charge  $\lambda_i$  on link  $i$  must be less than or equal to the sum of  $\lambda_l$  for  $l \in p_k$  on all possible paths  $p_k \in \{p_1, \dots, p_t\}$ <sup>7</sup>.

Equation (5) is an obvious one, which says that the cost-shares assigned to agents must be greater than or equal to zero. The systems (2)–(5) might have multiple solutions or they might not have any, it will be shown in the following sections.

**Definition 7** (Egalitarian solution at  $\lambda^*$ ). Consider a parameter  $\lambda^* > 0$ . The egalitarian solution at  $\lambda^*$ , denoted by  $(EG^{T\lambda^*})$ , assigns at the network cost-sharing problem  $\langle N, C, \theta \rangle$  the set of cost-shares  $(\lambda_1^*, \dots, \lambda_s^*)$  such that the cost share of link  $i = 1, \dots, s$  is determined by

$$\lambda_i^* = \min_{p_k \in \mathcal{P}_i} \left\{ \lambda^*, AC_i(\theta_i), \left[ \sum_{l \in p_k} \lambda_l^* \right] \right\}. \tag{6}$$

<sup>6</sup> Moulin [4] suggests that if no simple algorithm can characterize the cost minimization outcomes, then we will have no choice but to drop the cost minimization requirement and turn to the exogenously given suboptimality.

<sup>7</sup> Note that if a simple triangular network is analyzed here, then the routing-proofness constraint is just a set of triangle inequalities.



This solution achieves revenue equal to  $T_{\lambda^*} = \sum_{i=1}^s \lambda_i^* \theta_i$ .

Note that  $EG^{T_{\lambda^*}}$  is a well defined cost-sharing solution, as there exists a unique set of values that satisfy Equation (6). They can be easily computed by starting for small values, say at  $\lambda^* = 0$ , at which point the solution returns  $\lambda_i = \lambda^*$  for all  $i = 1, \dots, s$ . As we increase the value of  $\lambda^*$ , the stand-alone constraint ( $AC_i(\theta_i)$ ) or LRP constraints ( $\sum_{l \in p_k} \lambda_l^*$ ) bind. The formal computation of this rule is described in the next section.

By construction, egalitarian mechanism (EG),  $EG^{T_{\lambda^*}}$  meets stand-alone core stability and LRP, but it may not be efficient. Indeed, the revenue generated by  $EG^{T_{\lambda^*}}$  equals to  $T_{\lambda^*} = \sum_{i=1}^s \lambda_i^* \theta_i$  which may be smaller or larger than  $EFF(\theta_1, \dots, \theta_s)$ . Clearly,  $T_{\lambda^*}$  is non-decreasing in  $\lambda^*$ , and it may be that  $T_{\lambda^*}$  is actually constant for a large value of  $\lambda^*$ . Indeed, consider two values  $\underline{\lambda}^*$  and  $\bar{\lambda}^*$  such that  $\bar{\lambda}^* > \underline{\lambda}^* > AC_i(\theta_i)$  for all  $i$ . Under  $EG^{T_{\lambda^*}}$ , the payment  $\lambda_i^*$  to the agents along link  $i$  is not larger than  $AC_i(\theta_i)$ . Thus, any increases to  $\bar{\lambda}^* > \underline{\lambda}^*$  will also hit the same stand-alone constraints, and the cost-shares of the agents remain unchanged.

From the analysis, the maximum revenue generated under  $EG^{T_{\lambda^*}}$  occurs when  $\lambda^* = \max \{AC_1(\theta_1), \dots, AC_s(\theta_s)\}$ . We denote by  $T_{\lambda^*}^{max}$  such a maximum revenue. Note that if the revenue  $T_{\lambda^*}^{max} > EFF(\theta_1, \dots, \theta_s)$ , then by the continuity of  $T_{\lambda^*}(\lambda^*)$ , we can always find one  $\lambda^{**}$  such that  $T_{\lambda^{**}} = EFF(\theta_1, \dots, \theta_s)$ . This  $\lambda^{**}$  will generate an allocation  $EG^{T_{\lambda^{**}}}$ <sup>8</sup> and it will be called the egalitarian solution ( $EG^{T_{\lambda^*}}$ ). Note also that when this egalitarian solution exists, it will be efficient, core stable and limit routing-proof.

The computation of EG above is similar to the computation of the uniform rule, except for the additional LRP constraints,  $\left[ \sum_{l \in p_k} \lambda_l^* \right]$ , in Equation (6). Such a variation substantially changes how EG assigns cost shares to the agent in different links. In contrast with the uniform rule, the EG mechanism proposed in this paper might not exist for an arbitrary network cost-sharing problem  $P$ . The existence of EG depends on whether the last stand-alone cost being reached can cover the loss incurred by the replacement of the limit routing-proofness constraints. We see this in the next sections.

### 3.2. A Numerical Example of EG

In this section, we will provide a step-by-step numerical example illustrating how EG allocates charges to agents. Consider the network in Figure 1 and a demand  $\theta = (2, 6, 3)$ . That is, a total of 11 units of traffic were demanded: two units are demanded on link 1; six units were demanded on link 2; and three units were demanded on link 3.

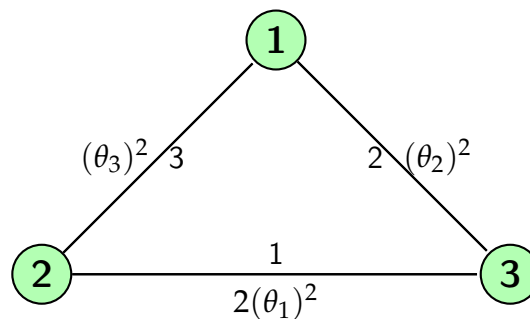


Figure 1. A simple network with three nodes.

<sup>8</sup> If it happens that  $T_{\lambda^{**}}$  is the maximum revenue that EG can achieve, then  $T_{\lambda^{**}} = T_{\lambda^{**}}^{max}$  and the allocation  $EG^{T_{\lambda^{**}}}$  coincides with  $EG^{T_{\lambda^{**}}^{max}}$ .

The computation of the efficient allocation for this network requires no re-routing of traffic, therefore providing an efficient cost equal to  $2(2)^2 + 6^2 + 3^2 = 53$ .

The stand-alone cost for each link is:

$$C_1(\theta_1) = 2(2)^2 = 8, C_2(\theta_2) = 6^2 = 36 \text{ and } C_3(\theta_3) = 3^2 = 9.$$

Now, we can formulate the problem and apply EG as in Equations (2)–(5):

$$2\lambda_1 + 6\lambda_2 + 3\lambda_3 = 53 \tag{7}$$

$$2\lambda_1 \leq 2(2)^2, 6\lambda_2 \leq (6)^2, 3\lambda_3 \leq (3)^2 \tag{8}$$

$$\lambda_1 + \lambda_2 \geq \lambda_3, \lambda_2 + \lambda_3 \geq \lambda_1, \lambda_1 + \lambda_3 \geq \lambda_2 \tag{9}$$

$$\lambda_1 \geq 0, \lambda_2 \geq 0, \lambda_3 \geq 0 \tag{10}$$

Under EG, we have:

$$\underbrace{2 \min \{ \lambda^*, 2(2), (\lambda_2^* + \lambda_3^*) \}}_{\lambda_1^* \theta_1} + \underbrace{6 \min \{ \lambda^*, (6), (\lambda_1^* + \lambda_3^*) \}}_{\lambda_2^* \theta_2} + \underbrace{3 \min \{ \lambda^*, (3), (\lambda_1^* + \lambda_2^*) \}}_{\lambda_3^* \theta_3} = 53$$

By Equation (6), we have

$$\lambda_1^* = \min \{ \lambda^*, 2(2), (\lambda_2^* + \lambda_3^*) \} \tag{11}$$

$$\lambda_2^* = \min \{ \lambda^*, (6), (\lambda_1^* + \lambda_3^*) \} \tag{12}$$

$$\lambda_3^* = \min \{ \lambda^*, (3), (\lambda_1^* + \lambda_2^*) \} \tag{13}$$

Solving the above system simultaneously will give us an egalitarian solution, if there exists one. We attacked this problem by gradually increasing  $\lambda^*$  until the first stand-alone constraint  $AC_i(\theta_i)$  for  $i = 1, 2, 3$  is reached. This happens when  $\lambda^* = 3$ .

Note the fact that the stand-alone constraints of  $\lambda_1^*$  and  $\lambda_2^*$  are still non-binding implies  $\lambda_1^* + \lambda_2^* > \lambda_3^*$ . So by (11), suppose  $\lambda^* = 3$  for all  $\lambda_i^*$ , we check if (11) is satisfied:

$$2\lambda^* + 6\lambda^* + 3\lambda^* = 2(3) + 6(3) + 3(3) = 33 < 53.$$

Obviously,  $\lambda^* = 3$  does not achieve efficiency. Hence, we must continue increasing  $\lambda^*$  on  $\lambda_1^*$  and  $\lambda_2^*$  while leaving  $\lambda_3^* = 3$  as its stand-alone constraint is reached.

Next, we applied the same logic and continue increasing  $\lambda^*$ . It is easy to see that the next stand-alone constraint being reached will be  $AC_1(\theta_1)$ . That is, when  $\lambda^* = 4$ .

We know that  $\lambda_3^* = 3$  and the stand-alone constraint of  $\lambda_2^*$  is still non-binding when  $\lambda^* = 4$ . This implies  $\lambda_2^* + \lambda_3^* = \lambda_2^* + 3 > 4 = \lambda_1^*$ . Thus, the minimum value in (11) will never be the limit routing-proofness constraint. We check (11),

$$2\lambda^* + 6\lambda^* + 3\lambda^* = 2(4) + 6(4) + 3(3) = 41 < 53$$

Again, efficiency is still not attained with  $\lambda^* = \lambda_1^* = \lambda_2^* = 4$  and  $\lambda_3^* = 3$ . Leaving  $\lambda_1^* = 4$  and  $\lambda_3^* = 3$ , we have by efficiency,

$$\lambda_2^* = \frac{EFF(\theta_1, \theta_2, \theta_3) - \lambda_1^* \theta_1 - \lambda_3^* \theta_3}{\theta_2} = \frac{53 - 2(4) - 3(3)}{6} = 6.$$

Recall that (12) implies

$$\lambda_2^* = \min \{ \lambda^*, 6, 6(4 + 3) \}$$

In this case, we need to be cautious about the limit routing-proofness constraint. If it happens that the minimum value is obtained from the limit routing-proofness constraint, then there will be no egalitarian solution satisfying Equation (11).

We proceed to check if  $(\lambda_1^*, \lambda_2^*, \lambda_3^*) = (4, 6, 3)$  satisfies all the constraints given by Equations (7)–(13).

$$\begin{aligned}
 \text{Equation (7)} &\Rightarrow 2(4) + 6(6) + 3(3) = 53 \\
 \text{Equation (8)} &\Rightarrow 2(4) = 2(2)^2, 6(6) = 36, 3(3) = 9 \\
 \text{Equation (9)} &\Rightarrow \lambda_1^* + \lambda_2^* = 4 + 6 = 10 > 3 = \lambda_3^* \\
 &\Rightarrow \lambda_2^* + \lambda_3^* = 6 + 3 = 9 > 4 = \lambda_1^* \\
 &\Rightarrow \lambda_1^* + \lambda_3^* = 4 + 3 = 7 > 6 = \lambda_2^* \\
 \text{Equation (10)} &\Rightarrow \lambda_1^* = 4 > 0, \lambda_2^* = 6 > 0, \lambda_3^* = 3 > 0.
 \end{aligned}$$

Hence, we have found an egalitarian solution  $EG_{\lambda^{**}}^{T^{max}} = (\lambda_1^*, \lambda_2^*, \lambda_3^*) = (4, 6, 3)$  such that efficiency, stand-alone core stability and limit routing-proofness are simultaneously satisfied. Interestingly, since the demands  $\theta_1, \theta_2$  and  $\theta_3$  are precisely the optimal allocations that generates the total minimum cost, the stand-alone core stability and efficiency force the egalitarian solution  $\lambda_i^* = AC_i(\theta_i)$  for all  $i = 1, 2, 3$ .

### 3.3. An Example for Problem Where EG does not Exist

Consider the network in Figure 2 and  $\theta = (1, 1, 4)$ . That is, where one unit of traffic demand link  $1 \leftrightarrow 3$ , one unit of traffic demand link  $1 \leftrightarrow 2$ ; and four units of traffic demand link  $1 \leftrightarrow 2$ .

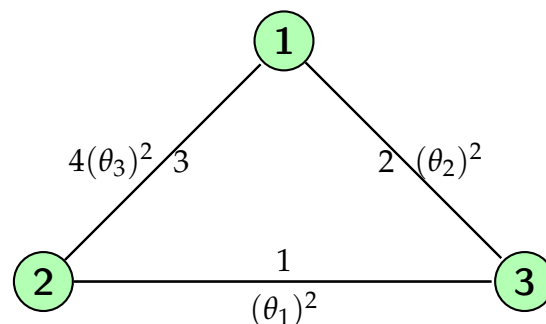


Figure 2. A simple network with three nodes and three links.

Efficiency requires  $\frac{7}{3}$  units of traffic demanding link  $1 \leftrightarrow 2$  to be routed along the alternative route on links  $1 \leftrightarrow 3$  and  $3 \leftrightarrow 2$ . This will generate an efficient cost equal to

$$EFF(\theta_1, \theta_2, \theta_3) = 4(4 - \frac{7}{3})^2 + (1 + \frac{7}{3})^2 + (1 + \frac{7}{3})^2 = \frac{100}{3}.$$

In order to compute the stand-alone cost note that

$$C_1(\theta_1) = (1)^2 = 1, \quad C_2(\theta_2) = (1)^2 = 1, \quad C_3(\theta_3) = 4(4)^2 = 64.$$

Thus, if the central authority decides to impose a personalized charge to every agent, then it can impose at most a total amount of \$1 to each agent on link 1 and 2; \$16 to each agent on link 3.

For EG to exist, it needs to recoup the efficient cost  $\frac{100}{3}$ . As we increase  $\lambda^*$ , the stand-alone constraints of link 1 and 2 are reached immediately followed by link 3. That is,  $\lambda_i^* = AC_i(\theta_i) = 1$  for  $i = 1, 2$ .

Hence, by efficiency, we have that

$$\begin{aligned}\lambda_3^* &= \frac{EFF(\theta_1, \theta_2, \theta_3) - \lambda_1^* \theta_1 - \lambda_2^* \theta_2}{\theta_3} \\ &= \frac{94}{12} = 7.8\bar{3} < \frac{64}{4} = 16.\end{aligned}$$

Thus, the charges assigned to each agent under EG are  $(\lambda_1^*, \lambda_2^*, \lambda_3^*) = (1, 1, 7.8\bar{3})$ . Note that this solution satisfies stand-alone core stability and efficiency but it fails to meet the limit routing-proofness constraint, as the agents demanding link 3 would prefer to pose as two agents demanding links 1 and 2.

#### 4. The RP Algorithm

We now turn our focus on investigating the necessary and sufficient conditions for the existence of an efficient, core stable and limit routing-proof egalitarian solution. To derive those conditions, we introduce an algorithm called the RP algorithm.

**Definition 8** (The RP algorithm). *Given a problem  $P$ , the RP algorithm assigns at any step a price to every link in the network. This price can be of two types: temporary or permanent. We define a semi-cycle to be a cycle of links such that all but one of the links have assigned permanent prices.*

- *Step 0.* We start from a network  $N$  and assign temporary prices  $T_i = AC_i(\theta_i)$  to every link  $i$  in the network. Continue to step 1.
- *Step 1.* The algorithm ends when all the assigned prices are permanent. For a given set of prices, replace the smallest temporary price with a permanent price of the same value. The rest of the prices are unchanged. If the new prices generate a semi-cycle, then we continue to step 2. Otherwise, we repeat step 1 with the new prices.
- *Step 2.* For every semi-cycle  $p_k$  with link  $i$  that has attached a temporary price  $T^i$ , we replace their temporary price with a new temporary price  $T'_i = \min \left\{ T^i, \sum_{l \in p_k} P_l \right\}$ . Moreover, if every path  $p_j \in \mathcal{P}_i$  is a semi-cycle, then  $T'_i$  becomes permanent. We continue to step 1 with the new set of prices.

The algorithm finishes in at most  $2s$  steps, where  $s$  is the number of links in the network, since every time that step 1 runs, a price moves from temporary to permanent.

By assigning temporary prices at the average cost in step 1, the RP algorithm ensures for stand-alone core stability. On the other hand, replacing a new temporary price either with the minimum of the average cost, or the sum of the prices on the alternative paths of a semi-cycle respects the LRP property.<sup>9</sup>

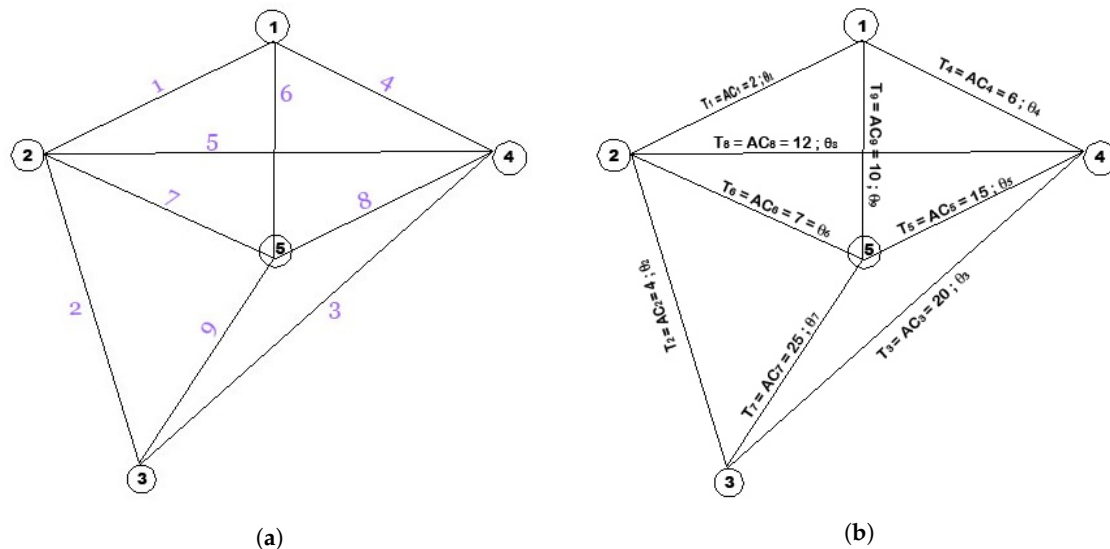
Note that the construction of the RP algorithm and EG are similar, but their objectives and operations are slightly different. To be precise, the RP algorithm always finds the feasible and maximal prices that it can charge the agents in order to meet the LRP constraint. This is in contrast with EG where it always allocates the minimal prices to agents subject to their stand-alone costs. Such discrepancy is due to the nature of the operation of EG and the RP algorithm. The RP algorithm always starts from the maximal prices possible (e.g., starts from the stand-alone prices) and solves the problem backwards, whereas EG attacks the problem from the lowest prices possible while gradually increasing it and stops when efficiency is reached.

<sup>9</sup> Similar to EG, the RP algorithm might not satisfy the efficiency property, but for any revenue (regardless of whether efficiency is achievable), it always generates prices that maximize the possible revenue.

Example of RP Algorithm

We provide a numerical example showing how prices are computed under the RP algorithm.

Consider an arbitrary network with the links numbered as in Figure 3a. Without loss of generality, suppose the average costs are given as in Figure 3b. By the RP algorithm, we start assigning a temporary price—average cost to each link until every link has a temporary price. For instance, we assign  $T_1 = AC_1 = 2$  to link 1 and  $T_2 = AC_2 = 4$  to link 2, etc. Starting from the smallest temporary price,  $T_1 = AC_1 = 2$ ,  $T_1$  will be replaced with a permanent price  $P_1 = 2$  followed by the second, third smallest temporary prices and so forth at every point in time until the first semi-cycle is formed.

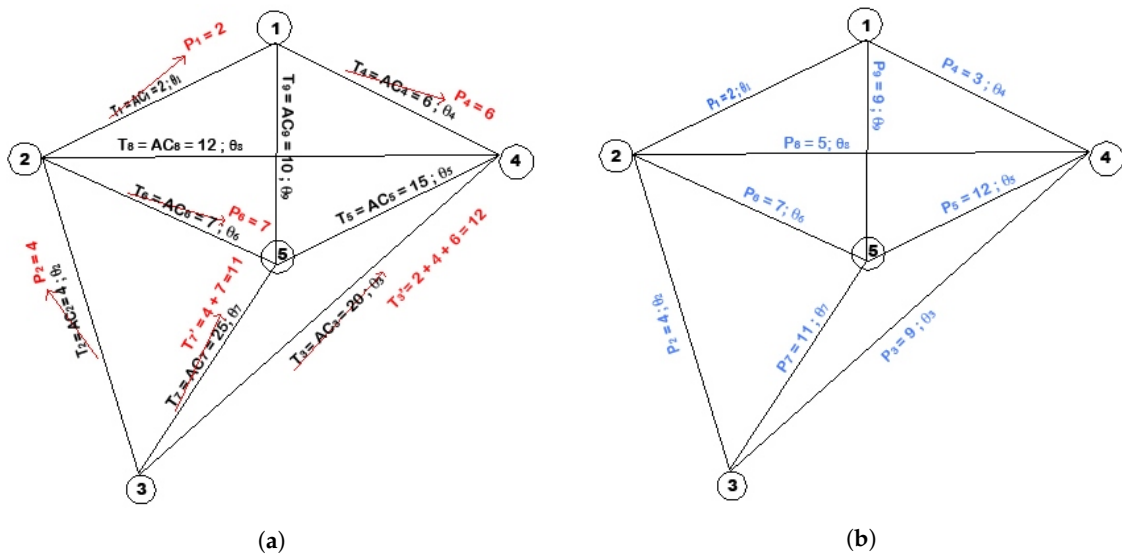


**Figure 3.** Step 1 of RP algorithm. (a) A general network with 5 nodes and 9 links numbered as shown; (b) A network with assigned temporary prices.

Note that the first semi-cycle is formed by links 1, 2 and 4. Thus, the temporary price  $T_3$  on link 3 is replaced by a new temporary price  $T'_3 = \min \left\{ \underbrace{20}_{AC_3}, \underbrace{12}_{2+4+6} \right\} = 12$ . This new temporary price  $T'_3$  will become a permanent price only if all the ascending smaller prices are replaced with permanent prices. Once the price is replaced, we continue to replace the next smallest price, which is price of link 7 in this case. In there a second semi-cycle is formed by links 2 and 7 and we replace the temporary price  $T_9$  with  $T'_9 = \min \left\{ \underbrace{25}_{AC_9}, \underbrace{9}_{2+7} \right\} = 9$ .<sup>10</sup>

Continue replacing in this fashion and ensuring all the semi-cycles are extracted until each temporary price becomes permanent price as shown in Figure 4b.

<sup>10</sup> Depending on the configuration of the networks, a temporary price may be replaced more than once before it finally becomes a permanent price.



**Figure 4.** Step 2–4 of RP algorithm. (a) In the process of replacing temporary prices to permanent prices; (b) The RP algorithm ends when all links have permanent prices.

**5. Main Results**

We now turn to the main results of this paper. As mentioned in the previous section, our results will rely heavily on the application of RP algorithm.

**Proposition 1.** For any network cost-sharing problem the prices given by the RP algorithm coincide with  $EG^{\lambda^*}$ .

The proof is in Appendix A.

The RP algorithm takes the permanent price  $P_i = \min_{p_k \in \mathcal{P}_i} \{AC_i, \sum_{l \in p_k} P_l\}$  for all  $i$ . If the minimum temporary price is  $\sum_{l \in p_k} P_l$  for some  $p_k$ , then the RP algorithm will never replace any permanent prices less than that. Therefore, the revenue generated under these prices and demands is going to be maximal. That is, RP merely searches for the maximal prices that it can achieve while satisfying together the stand-alone core stability and limit routing-proofness.

On the other hand, EG also allocates charges to agents with respect to the stand-alone core stability and limit routing-proofness. It chooses, or controls the parameter  $\lambda^*$  where the revenue target (this might include efficiency) is just achieved and stops. Hence, if the revenue target is precisely the maximal revenue that is achievable by the RP algorithm, then the allocation  $EG^{\lambda^*} = (\lambda_1^*, \dots, \lambda_s^*)$  coincides with the price vector  $\bar{P} = (P_1, \dots, P_s)$  simply because EG and RP algorithm are constructed in an identical way. Indeed, since the RP algorithm always searches for the maximal revenue that it can achieve in a network, the total charge (resp. total revenue) allocated (resp. collected) by EG can do no better than that maximal revenue. Thus, the maximal revenue provided by the RP algorithm is an upper bound for which EG can achieve in a network.

**Theorem 1.** Consider an arbitrary network cost-sharing problem.

a. Let agent  $j^*$  be the last agent in the RP algorithm. A stand-alone core stable, efficient and limit routing-proof price vector  $\bar{P} = (P_1, \dots, P_s)$  exists if and only if one of following conditions is satisfied:

- i  $\frac{EFF(\theta_1, \dots, \theta_s) - \sum_{i \neq j^*} P_i \theta_i}{\theta_{j^*}} \leq \min_{p_k \in \mathcal{P}_{j^*}} \{AC_{j^*}, \sum_{l \in p_k} P_l\}$ , where  $\mathcal{P}_{j^*}$  is the set of all feasible alternative paths of  $j^*$ ;
- ii  $EFF(\theta_1, \dots, \theta_s) \leq \sum_{i=1}^s P_i \theta_i$

b. The egalitarian solution exists if and only if either condition (i) or (ii) is satisfied.

The proof is in Appendix A. Theorem 1a states the necessary and sufficient conditions for which the price vector  $\bar{P} = (P_1, \dots, P_s)$  is core stable, efficient and limit routing-proof. Note that condition (ii) is implied by condition (i), but added as a reference. On the other hand, part (b) tells us that the egalitarian solution is guaranteed to exist if either condition (i) or (ii) hold.

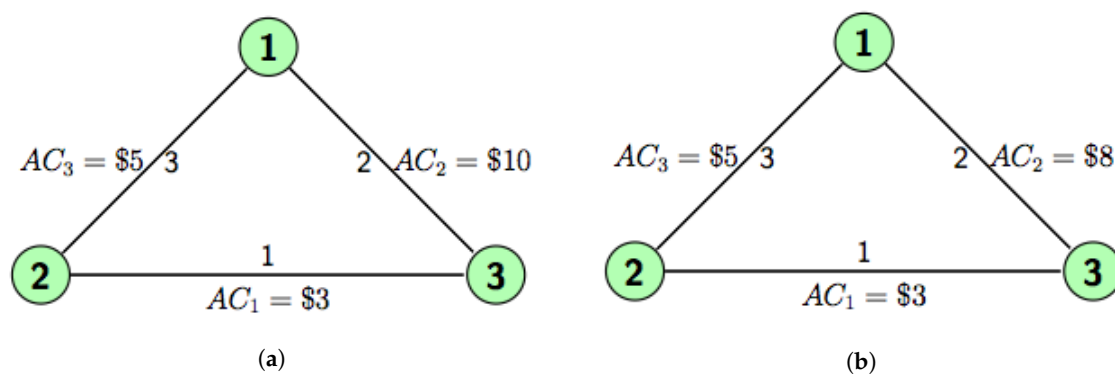
**Corollary 1.** *Given a general network, suppose that the average cost satisfies the (generalized)<sup>11</sup> triangle inequality, that is,*

$$AC_i(\theta_i) \leq \min_{p_k \in \mathcal{P}_i} \left\{ \sum_{l \in p_k} AC_l(\theta_l) \right\}$$

for any demand of link  $i$ , then the egalitarian solution exists.

The proof is in Appendix A. Note that this condition is sufficient but not necessary for the existence of egalitarian solution. To see this, consider the following network.

In Figure 5a, the average costs are given by \$3, \$10 and \$5 for link 1, 2, 3 respectively. Clearly, the triangle inequality is not satisfied since  $\$5 + \$3 = \$8 < \$10$ . Under the RP algorithm,  $AC_2$  is replaced with a new permanent price taken from the sum of  $\$5 + \$3 = \$8$ . We see that there exists an egalitarian solution whenever  $EFF(\theta_1, \theta_2, \theta_3) \leq 3\theta_1 + 5\theta_2 + 8\theta_3$ . In other words, no solution will be found if  $EFF(\theta_1, \theta_2, \theta_3) > 3\theta_1 + 5\theta_2 + 8\theta_3$ . Hence, (10) is a sufficient condition rather than a necessary condition.



**Figure 5.** An example illustrating the existence of egalitarian solution. (a) A network with assigned average costs. (b) The prices under routing-proof (RP) algorithm.

## 6. Conclusions and Open Questions

We have introduced the study of routing-proof mechanisms for network cost-sharing models in the presence of atomless agents for an arbitrary set of cost-functions. We have shown that two canonical properties in the cost-sharing literature, efficiency and stand-alone core stability, may be compatible with limit routing-proofness. In order to do so, we have introduced a new mechanisms, the egalitarian mechanism, as a natural selection when these three properties are compatible.

We find three critical open questions. The first pertaining to the study of mechanisms when agents are not atomless. Our work only provides a glimpse of the restrictions needed on the cost-functions to achieve efficiency, stand-alone core stability and a more general routing-proofness constraint where the payment allocated to agents may change depending on the mass rerouted.

The second question pertains to the demands of agents. Our work has focused on agents demanding single links between any pair of nodes. However, by extending the model into demanding multiple links should enhance the applicability of EG and related mechanisms.

<sup>11</sup> Note that this does not necessary have to be a triangle, it can be a “ $n$ -cycle” inequality where  $n$  is the number of links in the cycle.



Finally, our work has only shown the existence of a mechanism meeting efficiency, stand-alone core stability and limit routing-proofness, but has not provided a full characterization of a mechanism. More work is needed to develop a characterization using axioms in the spirit of limit routing-proofness.

**Author Contributions:** R.J. and M.W. conceptualized the paper, found the results and wrote them. They contributed equally to the study.

**Funding:** This research received no external funding.

**Conflicts of Interest:** The authors declare no conflict of interest.

## Appendix A. Proofs

### Appendix A.1. Proof of Proposition 1

**Proof.** Let  $(P_1, \dots, P_s)$  be the prices given by the RP algorithm and let  $(\lambda_1^*, \dots, \lambda_s^*)$  be the EG  $T_{\lambda^*}^{max}$  solution. Fix a problem  $P$  and let  $T_{\lambda^*}^{max}$  be the maximal revenue target achieved by EG and the RP algorithm, we will show that  $(P_1, \dots, P_s) = (\lambda_1^*, \dots, \lambda_s^*)$ . Without loss of generality, suppose that the average cost functions satisfy  $AC_1(\theta_1) < \dots < AC_s(\theta_s)$ .

Solving by RP algorithm: we assign temporary prices  $T_i = AC_i$  on each link  $i$ . At each point in time, the smallest temporary price is replaced by a permanent price, i.e.,  $T_i = AC_i(\theta_i) = P_i$  until any semi-cycle is formed. Let  $j$  be the last link that completes the semi-cycle. We replace  $T_j$  by  $T'_j = \min \{ AC_j, \underbrace{\sum_{l \in p_k} P_l}_{T_j} \}$ .

Case 1: if  $T'_j = \min \{ AC_j, \sum_{l \in p_k} P_l \} = AC_j$ , then we have:

Subcase 1: if  $T'_j = AC_j$  is the next smallest  $AC_j$ , then  $AC_j$  becomes  $P_j$ .

Subcase 2: if  $T'_j = AC_j$  is not the next smallest  $AC_j$ , then we continue replacing other temporary prices  $T_m, m \notin \{i \cup j\}$  with  $P_m, m \notin \{i \cup j\}$  until either  $T'_j$  becomes the smallest  $AC_j$  or another semi-cycle forms.

Note that if  $T'_j = AC_j$  is the smallest price, then  $T'_j = AC_j$  becomes  $P_j$ .

Case 2: if  $T'_j = \min \{ AC_j, \sum_{l \in p_k} P_l \} = \sum_{l \in p_k} P_l$ , then we have:

Subcase 1: if  $T'_j = \sum_{l \in p_k} P_l$  is the next smallest price, then  $T'_j = \sum_{l \in p_k} P_l$  becomes  $P_j$ .

Subcase 2: if  $T'_j = \sum_{l \in p_k} P_l$  is not the next smallest price, then we continue replacing other temporary prices  $T_m, m \notin \{i \cup j\}$  with  $P_m, m \notin \{i \cup j\}$  until  $T'_j = \sum_{l \in p_k} P_l$  becomes the smallest price or another semi-cycle forms.

Together, this implies that  $T_i = AC_i = P_i$  and  $T'_j = AC_j = P_j$  or  $T'_j = \sum_{l \in p_k} P_l$ .

(Remark: If another semi-cycle forms before  $T'_j$  becomes the smallest price at some point in time, then we can follow the same proof as described above.)

Now, we solve  $P$  by EG. From Equation (6), we know that  $\lambda_i^* = \min \{ \lambda^*, AC_i(\theta_i), [\sum_{l \in p_k} \lambda_l^*] \}$ . As  $\lambda^* \rightarrow \infty$ ,  $AC_1(\theta_1)$  is reached, followed by  $AC_2(\theta_2), \dots$  until a semi-cycle is formed. Then,  $\lambda_i^* = AC_i(\theta_i)$  for all  $i$  before a semi-cycle is reached. Suppose link  $j$  is the last link that completes the semi-cycle. Then we have,

Case 1: If  $\min \{ \lambda^*, AC_j(\theta_j), \sum_{l \in p_k} \lambda_l^* \} = AC_j(\theta_j)$ , then  $\lambda_j^* = AC_j(\theta_j)$ .

Case 2: If  $\min \{ \lambda^*, AC_j(\theta_j), \sum_{l \in p_k} \lambda_l^* \} = \sum_{l \in p_k} \lambda_l^*$ , then  $\lambda_j^* = \sum_{l \in p_k} \lambda_l^*$ .

Hence, we have that  $P_i = AC_i(\theta_i) = \lambda_i^*$  before any semi-cycle forms. Since EG is constructed as the same way as the RP algorithm, i.e.,  $T'_j = \min \{ AC_j, \sum_{l \in p_k} P_l \} \Leftrightarrow \min \{ \lambda^*, AC_j(\theta_j), \sum_{l \in p_k} \lambda_l^* \} = \lambda_j^*$ . Hence,  $T'_j = AC_j(\theta_j) = P_j \Leftrightarrow \lambda_j^* = AC_j(\theta_j)$ ; on the other hand,  $T'_j = \sum_{l \in p_k} P_l = P_j \Leftrightarrow \lambda_j^* = \sum_{l \in p_k} \lambda_l^*$ . Thus,  $T_{\lambda^*}^{max} = P_1\theta_1 + \dots + P_s\theta_s = \lambda_1^*\theta_1 + \dots + \lambda_s^*\theta_s$  since  $P_i = \lambda_i^*$  for all  $i = 1, \dots, s$ .  $\square$

Appendix A.2. Proof of Theorem 1

**Proof.** Part (a):  $\Rightarrow$  We first prove the sufficiency part, i.e., suppose a stand-alone core stable, efficient and limit routing-proof solution exists, then we have  $\frac{EFF(\theta_1, \dots, \theta_s) - \sum_{i \neq j^*} P_i \theta_i}{\theta_{j^*}} \leq \min_{p_k \in \mathcal{P}_{j^*}} \left\{ AC_{j^*, \sum_{l \in p_k} P_l} \right\}$ .

Step 1: to prove the sufficient part, we need to show the following first. Let  $(\lambda_1^*, \dots, \lambda_s^*)$  be some prices on a limit routing-proof and core stable allocation. Then for any  $i \neq j^*$ , we have  $\lambda_i^* \leq P_i$ .

Note that the temporary price  $T_i = AC_i \geq \lambda_i^*$  by the stand-alone property. We see that at any point in time, the RP algorithm will replace a permanent price on link  $i$ . First, note that prior to the formation of a semi-cycle,  $T_i = AC_i$  and then it becomes a permanent price  $\geq \lambda_i^*$ . Once a semi-cycle is formed, we replace  $T_i$  with a new temporary price  $T'_i = \min \left\{ \underbrace{AC_i}_{T_i}, \sum_{l \in p_k} P_l \right\}$  and  $P_l \geq \lambda_l^*$  for all  $l$  in

the path  $p_k$ . Now, we need to show that  $T'_i \geq \lambda_i^*$  for all  $i$ .

Note that  $\min \left\{ AC_i, \sum_{l \in p_k} P_l \right\} \geq \min \left\{ \lambda_i^*, \sum_{l \in p_k} \lambda_l^* \right\}$  since  $T_i = AC_i \geq \lambda_i^*$  and  $P_l \geq \lambda_l^*$ . Since  $T'_i = \min \left\{ AC_i, \sum_{l \in p_k} P_l \right\}$ , we have:

Case 1: If  $T'_i = \min \left\{ AC_i, \sum_{l \in p_k} P_l \right\} = AC_i \Rightarrow T'_i = AC_i \geq \lambda_i^*$ .

Case 2: If  $T'_i = \min \left\{ AC_i, \sum_{l \in p_k} P_l \right\} = \sum_{l \in p_k} P_l \Rightarrow T'_i = \sum_{l \in p_k} P_l \geq \sum_{l \in p_k} \lambda_l^* \geq \lambda_i^*$ . where the last inequality holds by limit routing-proofness. Together case 1 and case 2 imply  $T'_i \geq \lambda_i^*$ . Continue checking in this fashion for all  $i \neq j^*$ , we have that  $P_i \geq T'_i \geq \lambda_i^* \Rightarrow P_i \geq \lambda_i^*$ , as required.

Step 2: suppose a solution  $(\lambda_1^*, \dots, \lambda_s^*)$  satisfies limit routing-proofness, stand-alone core stability and efficiency, then by step 1, we have  $\sum_{i \neq j^*} \lambda_i^* \theta_i \leq \sum_{i \neq j^*} P_i \theta_i$ .

$$\Rightarrow \lambda_{j^*}^* = \frac{EFF(\theta_1, \dots, \theta_s) - \sum_{i \neq j^*} \lambda_i^* \theta_i}{\theta_{j^*}} \geq \frac{EFF(\theta_1, \dots, \theta_s) - \sum_{i \neq j^*} P_i \theta_i}{\theta_{j^*}}.$$

Since  $P_i \geq \lambda_i^*$  for all  $i \neq j^* \Rightarrow \min \left\{ \sum_{l \in p_k} P_l \right\} \geq \min \left\{ \sum_{l \in p_k} \lambda_l^* \right\} \geq \lambda_{j^*}^*$  where the last inequality holds by limit routing-proofness. By the stand-alone core stability,  $AC_{j^*} \geq \lambda_{j^*}^*$ . Together, this implies  $\min_{p_k \in \mathcal{P}_{j^*}} \left\{ AC_{j^*, \sum_{l \in p_k} P_l} \right\} \geq \frac{EFF(\theta_1, \dots, \theta_s) - \sum_{i \neq j^*} P_i \theta_i}{\theta_{j^*}} = \lambda_{j^*}^*$ , as required.

$\Leftarrow$ ) Now, we proceed to prove the necessity part. Note that by the construction of the RP algorithm, the allocation  $(P_1, \dots, P_s)$  is core stable and limit routing-proof. By Proposition 1, we know that the prices given by the RP algorithm will coincide with  $EG_{\lambda^*}^{T^{max}}$ , that is,  $\sum_{i=1}^s \lambda_i^*(\lambda^*) \theta_i = T_{\lambda^*}^{max} = \sum_{i=1}^s P_i \theta_i$ . Thus, the only thing left to show is the efficiency. If we can show that  $EFF(\theta_1, \dots, \theta_s) \leq T_{\lambda^*}^{max}$ , then egalitarian solution will be a solution that satisfies efficiency, stand-alone core stability and limit routing-proofness. By assumption,  $\frac{EFF(\theta_1, \dots, \theta_s) - \sum_{i \neq j^*} P_i \theta_i}{\theta_{j^*}} \leq \min_{p_k \in \mathcal{P}_{j^*}} \left\{ AC_{j^*, \sum_{l \in p_k} P_l} \right\}$ . By definition,  $T'_{j^*} = \min_{p_k \in \mathcal{P}_{j^*}} \left\{ \underbrace{AC_{j^*, \sum_{l \in p_k} P_l}}_{T_{j^*}} \right\} = P_{j^*}$ . Hence, we have,

$$\begin{aligned} \frac{EFF(\theta_1, \dots, \theta_s) - \sum_{i \neq j^*} P_i \theta_i}{\theta_{j^*}} &\leq \min_{p_k \in \mathcal{P}_{j^*}} \left\{ AC_{j^*, \sum_{l \in p_k} P_l} \right\} = P_{j^*} \\ \Leftrightarrow \quad EFF(\theta_1, \dots, \theta_s) - \sum_{i \neq j^*} P_i \theta_i &\leq P_{j^*} \theta_{j^*} \\ \Leftrightarrow \quad EFF(\theta_1, \dots, \theta_s) &\leq P_{j^*} \theta_{j^*} + \sum_{i \neq j^*} P_i \theta_i \\ &= \sum_{i=1}^s P_i \theta_i \\ &= T_{\lambda^*}^{max} \end{aligned}$$

as desired.

Part (b):  $\Rightarrow$ ) Suppose the egalitarian solution  $EG^{T\lambda^{**}} = (\lambda_1^*, \dots, \lambda_s^*)$  exists, then it is efficient, core stable and limit routing-proof. By Proposition 1, we know that the maximal revenue given by the RP algorithm is always an upper bound for which EG can achieve. Moreover, the RP algorithm respects stand-alone core stability and limit routing-proofness. Thus, if the egalitarian exists, then there must exist a price vector  $\bar{P} = (P_1, \dots, P_s)$  that is also efficient, core stable and limit routing-proof. Hence, by Theorem 1a, condition (i) or condition (ii) is satisfied.

$\Leftarrow$ ) Firstly, note that condition (i) clearly implies condition (ii). Secondly, if condition (ii) is satisfied, then we have that  $EFF(\theta_1, \dots, \theta_s) \leq \sum_{i=1}^s P_i \theta_i = T_{\lambda^*}^{max}$  by Theorem 1. Hence, by proposition 1 and continuity, there exists  $\lambda^{**}$  such that  $T_{\lambda^{**}} = EFF(\theta_1, \dots, \theta_s)$  when we slowly decrease the revenue target from  $T_{\lambda^*}^{max}$  to  $EFF(\theta_1, \dots, \theta_s)$ . This  $\lambda^{**}$  will generate exactly the egalitarian solution  $EG^{T\lambda^{**}} = (\lambda_1^*, \dots, \lambda_s^*)$ .  $\square$

Appendix A.3. Proof of Corollary 1

**Proof.** Suppose without loss of generality that agents demand link  $i$ , EG starts charging  $\lambda^* > 0$  to all agents. At any point in time, the stand-alone constraint of every link will be reached. Without loss of generality, suppose the stand-alone constraint of link  $i$  is reached first, i.e.,  $\lambda^* = AC_i(\theta_i) = \lambda_i^*$ . By assumption,

$$\lambda_i^* = AC_i(\theta_i) \leq \min_{p_k \in \mathcal{P}_i} \left\{ \sum_{l \in p_k} AC_l(\theta_l) \right\} \Rightarrow \frac{C_i(\theta_i)}{\theta_i} \leq \sum_{l \in p_k} AC_l(\theta_l) \quad \forall p_k \in \mathcal{P}_i$$

$$\Rightarrow \lambda_i^* \leq \sum_{l \in p_1} AC_l(\theta_l), \dots, \lambda_i^* \leq \sum_{l \in p_i} AC_l(\theta_l)$$

This implies that agents on link  $i$  are charged at the lowest cost among all feasible alternative paths  $p_k$  that connect link  $i$ , so agents will not have an incentive to reroute. Limit routing-proofness is trivially satisfied by the assumption. Note that  $\lambda_i^*$  satisfies the stand-alone core stability since  $\lambda_i^* = AC_i(\theta_i)$ . It also satisfies efficiency since agents on link  $i$  are being charged at the lowest cost.

Now, it remains to check the cases  $\lambda_1^*, \dots, \lambda_{i-1}^*, \lambda_{i+1}^*, \dots, \lambda_{s-1}^*$ . Note that the assumption

$$AC_i(\theta_i) \leq \min_{p_k \in \mathcal{P}_i} \left\{ \sum_{l \in p_k} AC_l(\theta_l) \right\}$$

holds for all  $j = 1, \dots, s$ , hence by the same reasoning, we can deduce that  $\lambda_1^*, \dots, \lambda_{i-1}^*, \lambda_{i+1}^*, \dots, \lambda_{s-1}^*$  are also core stable, limit routing-proof and efficient. Now, without loss of generality, suppose the stand-alone cost of link  $s$  is reached last, then by efficiency,

$$\lambda_s^* = \frac{EFF(\theta_1, \dots, \theta_s) - \sum_{i \neq s} C_i(\theta_i)}{\theta_s}$$

The triangle inequality constraint assumed implies that,

$$\lambda_s^* = \frac{EFF(\theta_1, \dots, \theta_s) - \sum_{i \neq s} C_i(\theta_i)}{\theta_s} \leq AC_s(\theta_s) \leq \min_{p_k \in \mathcal{P}_i} \left\{ \sum_{l \in p_k} AC_l(\theta_l) \right\}$$

Thus, each agent who demands link  $s$  is paying at  $\lambda_s^*$ , which essentially is the lowest cost among all other feasible paths. Hence,  $\lambda_s^*$  is also routing-proof, core stable and efficient. Therefore, EG has a solution  $EG^{T\lambda^{**}} = (\lambda_1^*, \dots, \lambda_s^*)$  such that every  $\lambda_i^*$  for all  $i = 1, \dots, s$  is limit routing-proof, core stable and efficient.  $\square$

## References

1. Han, L.; Juarez, R. Free intermediation in resource transmission. *Games Econ. Behav.* **2018**, *111*, 75–84. [[CrossRef](#)]
2. Juarez, R.; Kumar, R. Implementing efficient graphs in connection networks. *Econ. Theory* **2013**, *54*, 359–403. [[CrossRef](#)]
3. Frederick, H.; Gerald, L. *Introduction to Operations Research*; McGraw-Hill: Columbus, OH, USA, 2009.
4. Moulin, H. Cost sharing in networks: Some open questions. *Int. Game Theory Rev.* **2013**, *15*, 1340001. [[CrossRef](#)]
5. Dominique, H.; Moulin, H. Traffic-based cost allocation in a network. *RAND J. Econ.* **1996**, *27*, 332–345.
6. Juarez, R. Group strategyproof cost sharing: The role of indifferences. *Games Econ. Behav.* **2013**, *82*, 218–239. [[CrossRef](#)]
7. Melo, E. Congestion pricing and learning in traffic network games. *J. Public Econ. Theory* **2011**, *13*, 351–367. [[CrossRef](#)]
8. Roughgarden, T. The Price of anarchy is independent of the network topology. *J. Comput. Syst. Sci.* **2003**, *67*, 341–364. [[CrossRef](#)]
9. Fotakis, D.; Spirakis, P. Cost-balancing tolls for atomic network congestion games. In Proceedings of the 3rd International Conference on Internet and Network Economics (WINE'07), San Diego, CA, USA, 12–14 December 2007; pp. 179–190.
10. Moulin, H. Pricing traffic in a spanning network. In Proceedings of the 10th ACM Conference on Electronic Commerce (EC09), Stanford, CA, USA, 6–10 July 2009.
11. Juarez, R.; Ko, C.Y.; Xue, J. Sharing sequential values in a network. *J. Econ. Theory* **2018**, *77*, 734–779. [[CrossRef](#)]
12. Chakrabarty, D.; Mehta, A.; Nagarajan, V. Fairness and optimality in congestion games. In Proceedings of the 6th ACM Conference on Electronic Commerce (EC '05), Vancouver, BC, Canada, 5–8 June 2005; pp. 52–57.
13. Joel, S. *Linear Programming Note X Integer Programming*; Mimeo University of California: San Diego, CA, USA, 2007.
14. Sprumont, Y. The division problem with single-peaked preferences: A characterization of the uniform allocation rule. *Econ. J. Econ. Soc.* **1991**, *59*, 509–519. [[CrossRef](#)]
15. Benassy, J.P. *The Economics of Market Disequilibrium (No. 330.1/B45e)*; Academic Press: Cambridge, MA, USA, 1982.
16. Bogomolnaia, A.; Holzman, R.; Moulin, H. Sharing the cost of a capacity network. *Math. Oper. Res.* **2010**, *35*, 173–192. [[CrossRef](#)]
17. Dutta, B.; Mishra, D. Minimum cost arborescences. *Games Econ. Behav.* **2011**, *74*, 120–143. [[CrossRef](#)]
18. Dong, B.; Guo, G.; Wang, Y. Highway toll pricing. *Eur. J. Oper. Res.* **2012**, *220*, 744–751. [[CrossRef](#)]
19. Hougaard, J.L.; Tvede, M. Truth-telling and nash equilibria in minimum cost spanning tree models. *Eur. J. Oper. Res.* **2012**, *222*, 566–570. [[CrossRef](#)]
20. Hougaard, J.L.; Tvede, M. Minimum cost connection networks: Truth-telling and implementation. *J. Econ. Theory* **2015**, *157*, 76–99. [[CrossRef](#)]
21. Kumar, R. Secure implementation in production economies. *Math. Soc. Sci.* **2013**, *66*, 372–378. [[CrossRef](#)]
22. Hougaard, J.L.; Moreno-Ternero, J.D.; Tvede, M.; Østerdal, L.P. Sharing the proceeds from a hierarchical venture. *Games Econ. Behav.* **2017**, *102*, 98–110. [[CrossRef](#)]
23. Moulin, H.; Laigret, F. Equal-need sharing of a network under connectivity constraints. *Games Econ. Behav.* **2011**, *72*, 314–320. [[CrossRef](#)]
24. Juarez, R.; You, J.S. Optimality of the uniform rule under single-peaked preferences. *Econ. Theory Bull.* **2018**, 1–10. [[CrossRef](#)]
25. Hougaard, J.L. *An Introduction to Allocation Rules*; Springer: Heidelberg/Berlin, Germany, 2009.

