



HAL
open science

Configuring an heterogeneous smartgrid network: complexity and approximations for tree topologies

Dominique Barth, Thierry Mautor, Dimitri Watel, Marc-Antoine Weisser

► To cite this version:

Dominique Barth, Thierry Mautor, Dimitri Watel, Marc-Antoine Weisser. Configuring an heterogeneous smartgrid network: complexity and approximations for tree topologies. *Journal of Global Optimization*, 2023, 89, pp.223-257. 10.1007/s10898-023-01338-0 . hal-04289472

HAL Id: hal-04289472

<https://hal.science/hal-04289472v1>

Submitted on 17 Nov 2023

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.

Configuring an heterogeneous smartgrid network: complexity and approximations for tree topologies.*

Dominique Barth [†] Thierry Mauto [†] Dimitri Watel ^{‡§}
Marc-Antoine Weisser [¶]

Abstract

We address the problem of configuring a power distribution network with reliability and resilience objectives by satisfying the demands of the consumers and saturating each production source as little as possible. We consider power distribution networks containing source nodes producing electricity, nodes representing electricity consumers and switches between them. Configuring this network consists in deciding the orientation of the links between the nodes of the network. The electric flow is a direct consequence of the chosen configuration and can be computed in polynomial time. It is valid if it satisfies the demand of each consumer and capacity constraints on the network. In such a case, we study the problem of determining a feasible solution that balances the loads of the sources, that is their production rates. We use three metrics to measure the quality of a solution: minimizing the maximum load, maximizing the minimum load and minimizing the difference of the maximum and the minimum loads. This defines optimization problems called respectively min-M, max-m and min-R.

In the case where the graph of the network is a tree, it is known that the problem of building a valid configuration is polynomial. We show the three optimization variants have distinct properties regarding the theoretical complexity and the approximability. Particularly, we show that min-M is polynomial, that max-m is NP-Hard but belongs to the class FPTAS and that min-R is NP-Hard, cannot be approximated to within any exponential relative ratio but, for any $\varepsilon > 0$, there exists an algorithm for which the value of the returned solution equals the value of an optimal solution shifted by at most ε .

Complexity, Approximability, Electrical network flow, Tree topology

*This is a preprint version of <https://doi.org/10.1007/s10898-023-01338-0>

[†]dominique.barth@uvsq.fr, thierry.mauto@uvsq.fr, DAVID, University of Versailles-St Quentin, 45, avenue des États-Unis, Versailles, 78035, France

[‡]dimitri.watel@ensiie.fr, ENSIIE, 1, square de la Résistance, Evry, 91000, France

[§]SAMOVAR, 9 Rue Charles Fourier, Evry, 91000, France

[¶]marc-antoine.weisser@centralesupelec.fr, LISN, Centrale Supélec, Paris-Saclay University, Rue Noetzhlin, Gif-sur-Yvette, 91190, France

1 Introduction

Smart electrical distribution networks are becoming increasingly heterogeneous [10] with a multiplication of sources of production of different natures and capacities (wind, solar, biomass, etc.) with a diverse geographical distribution in particular at the scale of an urban territory. The configuration of these resilient networks must be dynamically optimized [6] by guaranteeing a reliability of service taking into account changes in the capacity of sources or in consumer demands but also breakdowns or degradations due for example to weather conditions [9, 11]. This dynamic configuration of the distribution network is essential for the current development of smartgrids or microgrids [20] implementing collective self-consumption grids interconnecting on a local territory a set of prosumers (i.e. consumers able to produce, store and therefore share energy) [2, 7, 10].

Configuring such a distribution network means deciding which components (lines, sources, switches) should be activated or not. Reliability can be defined as the capacity of the electrical system to supply electricity in the quantity and with the quality demanded by consumers by using available sources reliably and fairly. Thus, to guarantee reliability, the objective here is to find the configuration satisfying the demands of the consumers [6, 18, 22] by saturating as little as possible the production capacity of (heterogeneous) sources and the flow capacity of links and switches [3, 5].

Configuration problems with reliability and resilience objectives are often considered from a graph theory perspective [1, 19]. Thus, some solutions seek balanced configurations in terms of load power [21] and other ones configure the network in disjoint balanced subnetworks [12, 16, 20, 22], in particular in the form of spanning trees or sub-DAGS [17].

But, as the electric flow in a network is a direct consequence of the chosen configuration [21], the objective is more to determine if there exists a configuration satisfying the production and capacity constraints and the consumption demands than to compute an electric flow in a graph (such as in [8]). Moreover, the reliability of a network can be guaranteed by the existence of a configuration which does not use all the capacity of each link and each switch in order to contain the impact of the snowball effect during breakdowns. So given a desired maximum percentage s , the problem we solve allows us to determine if there is a network configuration using each switch and each link at most s percents of its capacity. This percentage is called the load of the node. In [3, 4] we prove that this existence problem is NP-complete for general network topologies but polynomial for trees.

This paper introduces an optimization perspective to the problem so that the objective is now to determine a feasible configuration of the network where the loads of the sources are as balanced as possible. Such a balance responds, on the one hand, to the need to avoid saturating sources that would not be able to adapt to failures or sudden changes in demand (to avoid snowball effects), and, on the other hand, to propose a fair use of the energy produced by each actor in the context of a collective self-consumption grid, for example [7, 10].

We therefore introduce three metrics that can answer this problematic. As done, in [3], we introduce the load reserve, which is the difference between the maximum and the minimum loads of the sources. This load reserve should be minimized. Another way to balance the network is also to maximize the minimum load or minimize the maximum load. We show in this paper that, when the network topology is a tree, the three problems have distinct properties regarding the theoretical complexity and the approximability. Minimizing the maximum load is polynomial, which is a direct consequence of the result of [4], while minimizing the load reserve and maximizing the minimum load are strongly NP-hard problems. The two latter problems differ in their approximability. There exists an FPTAS when the objective is to maximize the minimum load, but no polynomial approximation (with any exponential ratio) exists for the other problem. Finally, for the two problems, there exists an FPTAS with an absolute ratio, i.e., for any $\varepsilon > 0$, there exists an algorithm for which the value of the returned solution is equal to the value of an optimal solution shifted by at most ε .

The FPTAS algorithms are based on the fact that we can manipulate the flows with a small precision. An interesting property is that, if we consider the physical electrical network, we cannot (and we do not need) to do the calculations with a flow that has an infinite precision, and this limitation naturally leads to an approximation algorithm that can run in polynomial time.

The paper is organized as follows. First, we define our model and the related computational problems. We then prove the polynomial and NP-hardness results, and end with the approximability result.

2 Model

The electrical distribution network is represented by a graph $G = (V, E)$ in which there are three types of vertices S , W and P . S represents the set of *sources* of the network; each vertex $s \in S$ is characterized by a maximum supply capacity denoted $Prod(s) \in \mathbb{N}^*$. W corresponds to the set of *switches*; each vertex $w \in W$ has a maximum flow capacity $Cap(w) \in \mathbb{N}$. And P represents the set of *sinks* (or consumers); each vertex $p \in P$ is characterized by a demand $Pow(p) \in \mathbb{N}$. The edges are the connections between the vertices. In the following, we focus on graphs G that are trees.

Remark. *Note that we do not require the sources or the sinks to be leaves of the tree. Similarly, we do not require the switches to be internal nodes.*

An **orientation** \mathcal{O} of a graph G is a function associating each edge $[u, v] \in E$ with a couple (u, v) or (v, u) corresponding to the orientation of this edge. The directed graph obtained with \mathcal{O} must be a DAG to avoid circuits in the graph of the electrical flow. In this paper we only focus on the case where G is a tree, so the DAG constraint is necessarily satisfied.

Note that for a given orientation, a switch is *activated* only if it belongs to at least one induced path from a source to a sink. The activation of the switches

can thus be seen as a consequence of the orientation of the edges. Consequently, in the rest of the paper, only the orientation will be considered to determine the configuration of a network.

Note also that even if the use of expensive technologies based on diodes forbidding one direction of electric flow on a link is sometimes considered, they are not deployed in large electrical distribution networks. This orientation of each link of the graph is a consequence of the positions of sources and sinks on the network and their demand and production values, with the consequence that a brutal variation on production or demand can cause blackout situations (as the northeast one in 2023). In our model however, we consider that the orientation of the links is a decision variable (as for example in the case of restoration methods based on the search for the shortest paths considered as oriented from the source to the sinks [15, 19]). On the one hand, this avoids having to complicate the graph model by electrical considerations and on the other hand having to calculate a flow at each step of the optimization algorithms. In practice, with the considerations linked to the activation of the switches indicated above, the optimal solutions finally obtained turn out to be a realistic approximation of the real behavior of an electrical network in terms of orientation of the links, which is particularly true in the case of trees [13]. Thus, an orientation of the arcs can therefore be considered valid from the point of view of an electric flow if for each arc (u,v) , the total quantity of electricity in u is greater than the one in v , which will always be the case in the configurations discussed.

2.1 Flow in the oriented tree and feasible orientation

Given an orientation \mathcal{O} , the network G becomes a flow network with capacity only on the switches, with multiple sources and sinks. Such a flow network can be made compatible with the one used in [8], but the fundamental difference lies in the constraints satisfied by the flow. In addition to the classical conservation and capacity constraints, the (electric) flow entering each node in $W \cup P$ must be equitably distributed over all its incoming arcs (i.e., the arcs are considered equivalent to resistors with same resistance value). In the special case of a source, the same constraint applies except that the source production itself counts as an additional fictive arc. Consequently, the goal here is then not to determine a maximum flow since it is unique and determined by the orientation: since the directed graph we obtain with this orientation is a DAG, we can compute it by going up from P . Our objective is to know if there exists an orientation for which the implied unique flow satisfies the demands of the nodes of P and respects the supply capacities of the sources of S and the capacities of the switches of W . Figure 1 gives examples of an orientation and the flow induced by that orientation.

Definition 1. *Let \mathcal{O} be an orientation of the edges of the graph G . Let $\Gamma^+(v)$ and $d^-(v)$ be respectively the set of successors and the in-degree of v , the flow*

$F_{\mathcal{O}}(u, v)$ going through each arc (u, v) entering v is

$$F_{\mathcal{O}}(u, v) = \begin{matrix} \text{if} & v \in W, & v \in S, & v \in P \\ \frac{\sum_{w \in \Gamma^+(v)} F_{\mathcal{O}}(v, w)}{d^-(v)}, & \frac{\sum_{w \in \Gamma^+(v)} F_{\mathcal{O}}(v, w)}{d^-(v) + 1}, & \frac{\sum_{w \in \Gamma^+(v)} F_{\mathcal{O}}(v, w) + Pow(v)}{d^-(v)} \end{matrix}$$

Since u does not intervene in the formula, we also denote this value by $F_{\mathcal{O}}(v)$. We say the orientation \mathcal{O} is feasible if:

- it satisfies the demand constraints: for every switch $v \in W$ and every sink $v \in P$, we have $d^-(v) > 0$.
- it satisfies the capacity constraints: $F_{\mathcal{O}}(v) \cdot d^-(v) \leq Cap(v)$ for $v \in W$ and $F_{\mathcal{O}}(v) \leq Prod(v)$ for $v \in S$. Note that these capacity constraints include a supply capacity constraint.

Example 1. Assuming the input network is the one shown in Figure 1a. We give 5 examples of orientations, two of which satisfy the demand and the capacity constraints. Note that sources may not be leaves and may have entering arcs (as in Figure 1c).

Determining whether there is a feasible orientation for a given electrical network leads to the following decision problem, which is shown to be NP-complete for general graphs [3], but polynomial for trees [4].

Problem 1 (VALID). Given a graph $G = (V, E)$, three functions $Prod$, Cap and Pow , does there exist a feasible orientation \mathcal{O} of G .

Theorem 1. [4] If G is a tree, VALID can be solved in time $O(n^4 \log(n))$. In case of a positive answer, the algorithm builds a feasible orientation.

In the rest of the paper, we focus on this particular case where the input graph is a tree T .

2.2 Optimization version

To evaluate the quality of a solution \mathcal{O} , we consider the loads of the sources.

Definition 2. Given a feasible orientation \mathcal{O} , the load of a source $s \in S$ is defined as the ratio $load_{\mathcal{O}}(s) = F_{\mathcal{O}}(s)/Prod(s)$.

Maximizing the minimum load, minimizing the maximum load or minimizing the load reserve leads to a more balanced solicitation rate of the sources, knowing that the more balanced these rates are, the more resistant the network is to a reconfiguration in case of failures [12, 17]. Reliability and resilience of the network are thus related to the following three optimization problems.

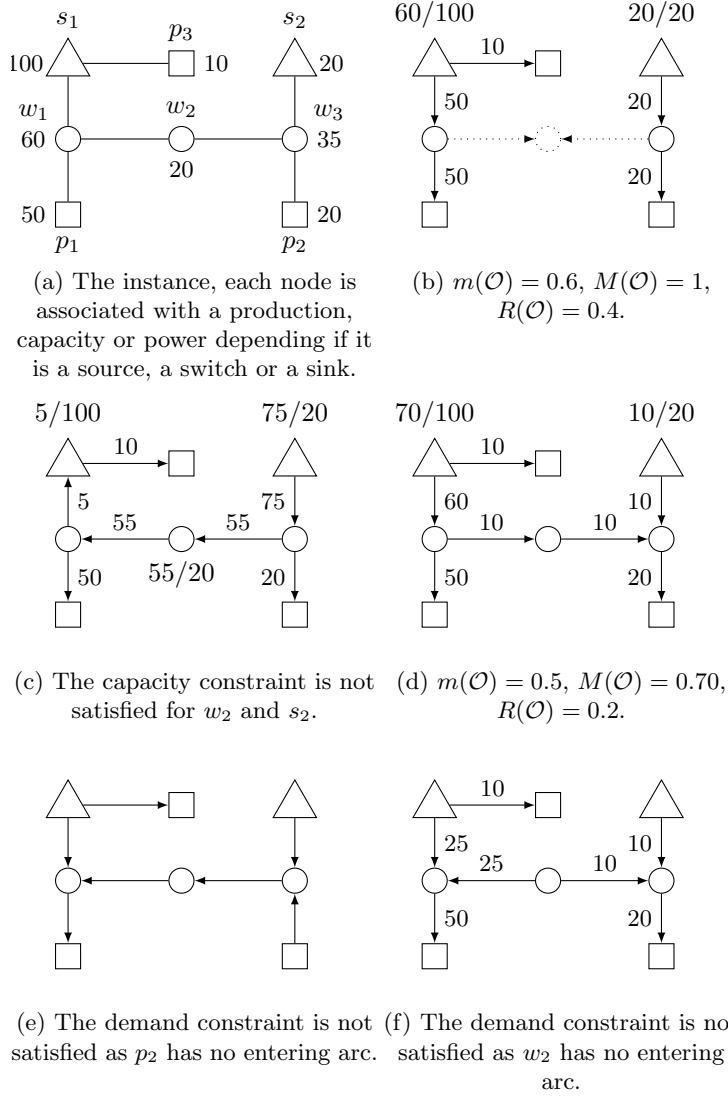


Figure 1: Example of an instance and orientations for that instance. Throughout the paper, we use triangles, circles and squares to represent sources, switches and sinks respectively. For each orientation, it is explained if it is feasible according to Definition 1. In this case, the metrics introduced in Definition 2 are given.

Problem 2 (max-m). Given a tree $T = (V, E)$, three functions *Prod*, *Cap* and *Pow*, compute an orientation \mathcal{O} of T satisfying the demand and the capacity constraints and maximizing the minimum load $m(\mathcal{O}) = \min_{s \in S} \text{load}_{\mathcal{O}}(s)$.

Problem 3 (min-M). Given a tree $T = (V, E)$, three functions $Prod$, Cap and Pow , compute an orientation \mathcal{O} of T satisfying the demand and the capacity constraints and minimizing the maximum load $M(\mathcal{O}) = \max_{s \in S} load_{\mathcal{O}}(s)$.

Problem 4 (min-R). Given a tree $T = (V, E)$, three functions $Prod$, Cap and Pow , compute an orientation \mathcal{O} of T satisfying the demand and the capacity constraints and minimizing the load reserve $R(\mathcal{O}) = M(\mathcal{O}) - m(\mathcal{O})$

Example 2. Figure 1 gives examples of values for $m(\mathcal{O})$, $M(\mathcal{O})$ and $R(\mathcal{O})$. Note that this example shows that maximizing $m(\mathcal{O})$ is not equivalent to minimize $R(\mathcal{O})$. Similarly, we could build an example where the solutions minimizing $M(\mathcal{O})$ are not the solutions minimizing $R(\mathcal{O})$.

3 Complexity of max-m, min-M, min-R

In this section, we explore the classical complexity of the three problems. We show that min-M is the easiest problem, since it is polynomial, while the other two are strongly NP-Hard.

Theorem 2. min-M is polynomial

Proof. The core idea to solve min-M is to use a dichotomy with the algorithm of Theorem 1. The dichotomy works as follows: at each iteration, we manipulate an interval $[a, b]$ containing the optimal maximum load (initialized with $a = 0$ and $b = 1$). We search for an orientation \mathcal{O} with $M(\mathcal{O}) \leq (a + b)/2$. If no such orientation exists, then a becomes $(a + b)/2$, otherwise, b becomes $M(\mathcal{O})$.

To do so, we solve the construction problem associated to min-M that, given a rational $M \in [0; 1]$, asks for a feasible orientation \mathcal{O} where $M(\mathcal{O}) \leq M$. This problem can be answered by determining the validity of the instance where, for every source $s \in S$, the production $Prod(s)$ is truncated to $M \cdot Prod(s)$. By the capacity constraint, the flow outgoing from s is less than $M \cdot Prod(s)$, consequently, any feasible orientation \mathcal{O} in that truncated instance is also feasible in the original instance and satisfies $M(\mathcal{O}) \leq M$. Note that each production should be an integer and that $M \cdot Prod(s)$ can be not in \mathbb{N} . We can get around this problem by noticing that multiplying each production, capacity and power by the same value does not change the instance. If $M = p/q$, then we can then multiply all the values by q to get an equivalent instance where each production, power and capacity is an integer.

The load reserve of any feasible orientation \mathcal{O} is a rational that can be encoded with a polynomial number of bits in the size of the instance. Then there exists a value ε such that $\lfloor \log(\varepsilon) \rfloor$ is polynomial and, for any two distinct feasible orientations \mathcal{O} and \mathcal{O}' , $|M(\mathcal{O}) - M(\mathcal{O}')| \geq \varepsilon$. So when $b - a < \varepsilon$, after $O(\log(\varepsilon))$ iterations, we stop and return the last found orientation \mathcal{O} and the corresponding value $M(\mathcal{O})$.

Let $\sigma = \max_{s \in S} Prod(s)$ and $\Pi = \sum_{p \in P} Pow(p)$. The load of a source is either 0 or belongs to $L = \{p/q | p \in [1; \Pi], q \in [1, n^n \cdot \sigma]\}$ Indeed, the flow cannot

be greater than the value asked by the consumers, it cannot be divided more than n times and is divided by at most n at each step, finally, in order to compute the load, we divide the flow by at most the maximum source production. The value ε is at least the minimum difference between two load reserves. Each load reserve is a difference between two values of L . Thus $\varepsilon \geq 1/(n^n \cdot \sigma)^4$. The number of iterations is then at most $4 \cdot n \log(n\sigma)$. By Theorem 1, the dichotomy is done in time at most $O(n^5 \log(n) \log(n\sigma))$. \square

Theorem 3. *max- m and min- R are strongly NP-hard. More precisely determining if there exists a feasible orientation \mathcal{O} with $R(\mathcal{O}) \leq \frac{2}{3}$ (respectively $m(\mathcal{O}) \geq \frac{1}{3}$) is NP-Complete.*

Proof. We describe below a reduction from the Subset Sum problem where, given n integers x_1, x_2, \dots, x_n and an integer B , the question is to decide whether there exists $I \subseteq \llbracket 1; n \rrbracket$ such that $\sum_{i \in I} x_i = B$. Let then $\mathcal{I} = (x_1, x_2, \dots, x_n, B)$ be such an instance.

This proof is done in three steps: first we describe a gadget to encode large integers, then we describe the construction of the reduction and some properties, and finally we prove the correctness of the reduction.

In the built instance, we will have $M(\mathcal{O}) = 1$ if \mathcal{O} is feasible, meaning that maximizing the minimum load and minimizing the load reserve are equivalent.

Encode large numbers As Subset Sum is weakly NP-Complete, we cannot use the values x_i or B in the powers, capacities and productions of the instance. First, we explain how it is possible to encode these integers with small productions and powers. Let $x \in \mathbb{N}$ be any integer such that a binary encoding of $x + 2$ is $\overline{b_{p-1} \dots b_1 b_0}$ (b_0 is the lowest bit and b_{p-1} is the highest). Let $m \geq p$ such that $m \geq 3$, we set $b_p = b_{p+1} = \dots = b_{m-1} = 0$. Then, we have $x + 2 = \overline{b_{m-1} \dots b_1 b_0} = \sum_{j=0}^{m-1} b_j 2^j$.

We then build a path of size $2m$ containing alternatively, for $j \in \llbracket 1; m \rrbracket$, a sink $p_j \in P$ with power $Pow(p_j) = 2 + b_{j-1}$ and a source $s_j \in S$ with production $Prod(s_j) = 2$ if $j = 1$ and 3 otherwise. Finally, we assume that the source s_m has another neighbor v (which can be of any type). Figure 2 illustrates this instance.

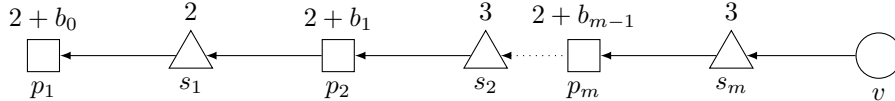


Figure 2: Gadget equivalent to a sink with power $2 + x \cdot 2^{-m}$ where $x + 2 = \overline{b_{m-1} \dots b_1 b_0}$. On the graph, each edge is directed toward the left as, otherwise, some demand constraints or some capacity constraints are not satisfied. Below each arc is given the flow going through the arc.

We now show an intermediate result showing that this path acts as a sink of value $2 + x \cdot 2^{-m}$.

Fact 1. Given a feasible orientation \mathcal{O} , then, for every $j \in \llbracket 1; m \rrbracket$, $\mathcal{O}([p_j, s_j]) = (s_j, p_j)$, for every $j \in \llbracket 1; m-1 \rrbracket$, $\mathcal{O}([p_{j+1}, s_j]) = (p_{j+1}, s_j)$ and $\mathcal{O}([v, s_m]) = (v, s_m)$. The load of each source s_j is at least 0.5 and $F_{\mathcal{O}}(v, s_m) = 2 + x \cdot 2^{-m}$.

Proof. We prove this property by induction on j , proving in addition that $F_{\mathcal{O}}(s_j, p_j) = \frac{2^{j+1}-2}{2^{j-1}} + \sum_{i=0}^{j-1} b_i 2^{i+1-j}$.

$[s_1, p_1]$ is directed toward p_1 otherwise the demand constraint is not satisfied for p_1 . In addition, $\mathcal{O}([s_1, p_2]) = (p_2, s_1)$ otherwise the flow outgoing from s_1 is at least 3 which is greater than $Prod(s_1)$. The flow entering p_1 is then $2 + b_0 = \frac{2^2-2}{2^0} + \sum_{i=0}^0 b_i 2^i$. Thus, the property is satisfied for $j = 1$.

Assuming the property is true for all $i \leq j$, we now show it is also true for $j + 1$. The edge $[s_{j+1}, p_{j+1}]$ is directed toward p_{j+1} otherwise the demand constraint is not satisfied for p_{j+1} . The flow $F_{\mathcal{O}}(s_{j+1}, p_{j+1})$ entering p_{j+1} is then

$$\begin{aligned} 2 + b_j + \frac{1}{2} \left(\frac{2^{j+1}-2}{2^{j-1}} + \sum_{i=0}^{j-1} b_i 2^{i+1-j} \right) &= \left(2 + \frac{2^{j+1}-2}{2^j} \right) + \left(\sum_{i=0}^{j-1} b_i 2^{i-j} + b_j \right) \\ &= \frac{2^{j+2}-2}{2^j} + \sum_{i=0}^j b_i 2^{i-j} \end{aligned}$$

If $j + 1 \neq m$ then $\mathcal{O}([s_{j+1}, p_{j+2}]) = (p_{j+2}, s_{j+1})$ otherwise s_{j+1} must power at least the flow $F_{\mathcal{O}}(s_{j+1}, p_{j+1})$ and half of $Pow(p_{j+2})$, which is strictly greater than $(2^{j+2}-2)/2^j \geq 3 = Prod(s_{j+1})$. If $j + 1 = m$ then s_m cannot power p_m alone as $j + 1 = m \geq 3$ and then $(2^{j+2}-2)/2^j > 3 = Prod(s_m)$. The property is then proved by induction.

Consequently, the flow $F_{\mathcal{O}}(v, s_m)$ entering s_m is the flow entering p_m divided by 2.

$$F_{\mathcal{O}}(v, s_m) = \frac{2^{m+1}-2}{2^m} + \sum_{i=0}^{m-1} b_i 2^{i-m} = 2 + \left(\sum_{i=0}^{m-1} b_i 2^i - 2 \right) \cdot 2^{-m} = 2 + x \cdot 2^{-m}$$

Note that $F_{\mathcal{O}}(s_j) = F_{\mathcal{O}}(s_j, p_j)/2$ is between 1.5 and 2.5 except when $j = 1$ where it is between 1 and 1.5. Then $load_{\mathcal{O}}(s_j)$ is greater than 0.5. \square

Thus we can now use sinks with called powers $2 - 2^{-m}x$ for any integer x .

Build the instance In this part, we will explain how we build an instance $\mathcal{J} = (T, Prod, Cap, Pow)$ of min-R and max-m from $\mathcal{I} = (x_1, x_2, \dots, x_n, B)$ using the previous gadget.

Without loss of generality, we assume that n is such that $0.1 > \frac{2}{3n+1}$ and that $B \geq x_i$ for all $i \in \llbracket 1; n \rrbracket$. We then choose m such that

$$1 > 2^{-m}B; \quad -1 < 2^{-m}(B - \sum_{i=1}^n x_i); \quad \forall i \in \llbracket 1; n \rrbracket \quad \frac{2}{3n+1} > 2^{-m}2x_i$$

Note that m is polynomial in n and the size of the encoding of x_i and B .

- We add $2n + 2$ switches $w_1, w_2, \dots, w_n, v_1, v_2, \dots, v_n, w$ and w_c with arbitrarily large capacities (for instance $\sum_{p \in P} Pow(p)$).
- We build n sinks p_i with power $4 - 2^{-m}2x_i$, for each $i \in \llbracket 1; n \rrbracket$, by building the gadget of Figure 2 with $x = 2^m - 2x_i$ and where the node v is replaced by a sink with power 1. We also build a sink p with power $2 + 2^{-m}B$ by using the gadget with $x = B$ where the node v is the switch w . Note that this procedure adds $(2m + 1) \cdot n + 2m$ nodes.
- We add n sinks q_1, q_2, \dots, q_n with power 4 and a sink p_c with power 10.
- We add $2n$ sources $t_1, t_2, \dots, t_n, r_1, r_2, \dots, r_n$ with production 4; n sources s_1, s_2, \dots, s_n with production 2; a source s with production 6 and a source s_c with production 10.
- For each $i \in \llbracket 1; n \rrbracket$, we add the 7 edges $[p_i, t_i]$, $[t_i, w_i]$, $[w_i, w]$, $[q_i, r_i]$, $[r_i, v_i]$, $[v_i, w]$ and $[s_i, w]$.
- We add the edges $[s_n, w_c]$, $[w_c, s_c]$, $[p_c, s_c]$ and $[s, w]$.

Figure 3 illustrates this instance.

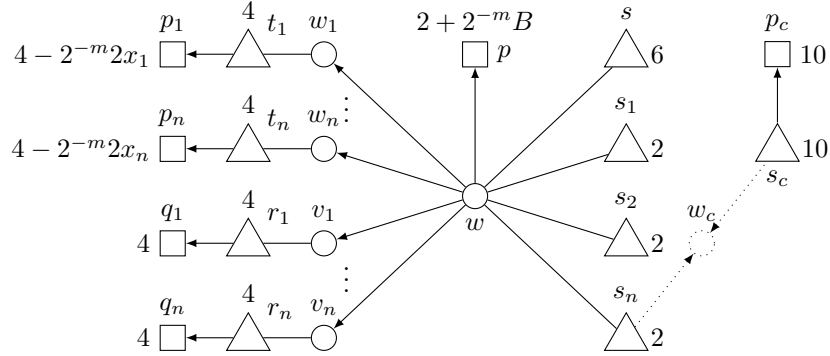


Figure 3: Instance \mathcal{J} of the reduction. Some edges are directed on the drawing, other orientations for these edges imply that the capacity or the demand constraints are not satisfied. The powers and productions are written next to each sink and source. Each switch can be considered to have an infinite capacity. Note that the switch w_c is not activated as no arc goes out of this switch.

This tree contains $N = 2m \cdot (n + 1) + 7n + 5$ nodes (recall that p and each sink p_i are gadgets creating $(2m + 1) \cdot n + 2m$ additional nodes), every power and production is between 2 and 10 and every capacity is at most $\sum_{p \in P} Pow(p) \leq 10 \cdot N$. This construction of \mathcal{J} is then done in polynomial time.

Properties of a feasible orientation of \mathcal{J} Let \mathcal{O} be any feasible orientation of \mathcal{J} . Due to the demand constraint, $\mathcal{O}([s_c, p_c]) = (s_c, p_c)$. Then the switch

w_c cannot be activated (the two incident edges of w_c are directed toward it), otherwise either s_c powers some sink through w_c which exceeds its production, or s_n powers at least a fraction $\frac{1}{4}$ of $Pow(p_c)$, that is 2.5, which exceeds its production too. Consequently, s_c powers p_c alone and $load_{\mathcal{O}}(c) = 1$.

Note that if \mathcal{O} directs one of the edges $[s, w]$ or $[s_i, w]$, for some $i \in \llbracket 1; n \rrbracket$, from w then the flow outgoing from the source is 0. In that case, the load reserve $R(\mathcal{O})$ is 1 and the minimum load $m(\mathcal{O})$ is 0. We then consider only feasible orientations where this case does not occur.

Again, due to the demand constraint, $\mathcal{O}([w, p]) = (w, p)$ and, for each $i \in \llbracket 1; n \rrbracket$, we have $\mathcal{O}([p_i, t_i]) = (t_i, p_i)$ and $\mathcal{O}([q_i, r_i]) = (r_i, q_i)$.

We now assume that $\mathcal{O}([w, w_i]) = (w_i, w)$ for some $i \in \llbracket 1; n \rrbracket$. The flow $F_{\mathcal{O}}(w_i, w)$ is at least $F_{\mathcal{O}}(w, p)/(3n+1) \geq 2/(3n+1)$. We can now examine the orientation of $[t_i, w_i]$. If $\mathcal{O}([t_i, w_i]) = (w_i, t_i)$, the demand constraint is not satisfied for w_i which has no entering arc. On the other hand, if $\mathcal{O}([t_i, w_i]) = (t_i, w_i)$, then $F_{\mathcal{O}}(t_i) = F_{\mathcal{O}}(t_i, p_i) + F_{\mathcal{O}}(t_i, w_i) \geq 4 - 2^{-m}2x_i + 2/(3n+1)$. As we chose m so that $2/(3n+1) > 2^{-m}2x_i$ and as the production of t_i is 4, the capacity constraint of t_i is not satisfied. Consequently, $\mathcal{O}([w, w_i]) = (w, w_i)$. Similarly, $\mathcal{O}([w, v_i]) = (w, v_i)$.

Hence, the only degree of freedom we have is to decide the orientation of the edges $[t_i, w_i]$ and $[r_i, v_i]$.

We now detail the load of each source:

- $load_{\mathcal{O}}(s_c) = 1$
- For each $i \in \llbracket 1; n \rrbracket$, $load_{\mathcal{O}}(r_i) = 1$ if $\mathcal{O}([r_i, v_i]) = (r_i, v_i)$ and 0.5 otherwise.
- For each $i \in \llbracket 1; n \rrbracket$, $load_{\mathcal{O}}(t_i) = 1 - 2^{-m}x_i/2$ if $\mathcal{O}([t_i, w_i]) = (t_i, w_i)$ and $0.5 - 2^{-m}x_i/4$ otherwise. Note that m has been chosen so that $2^{-m}2x_i \leq 0.1$. Consequently, the load of t_i is at least 0.4875.
- By Fact 1, the load of each hidden source in the gadgets is at least 0.5.
- Let α_v be the indexes $i \in \llbracket 1; n \rrbracket$ such that $\mathcal{O}([r_i, v_i]) = (v_i, r_i)$ and α_w be those for which $\mathcal{O}([t_i, w_i]) = (w_i, t_i)$.

$$\begin{aligned} load_{\mathcal{O}}(s) &= \frac{1}{Prod(s) \cdot (n+1)} \left(2 + 2^{-m}B + \sum_{j \in \alpha_w} (2 - 2^{-m}x_j) + \sum_{j \in \alpha_v} 2 \right) \\ &= \frac{1}{6 \cdot (n+1)} \left(2 \cdot (1 + |\alpha_w| + |\alpha_v|) + 2^{-m}(B - \sum_{j \in \alpha_w} x_j) \right) \end{aligned}$$

- For each $i \in \llbracket 1; n \rrbracket$, we have $load_{\mathcal{O}}(s_i) = 3 \cdot load_{\mathcal{O}}(s)$.

The load of s cannot be greater than $\frac{1}{3}$ as, otherwise, for each $i \in \llbracket 1; n \rrbracket$, we would have $load_{\mathcal{O}}(s_i) = 3 \cdot load_{\mathcal{O}}(s) > 1$, which means that \mathcal{O} is not feasible. Consequently, $m(\mathcal{O}) = load_{\mathcal{O}}(s)$.

Equivalence of the instances We end this proof by demonstrating that the minimum load reserve of \mathcal{J} is exactly $\frac{2}{3}$ if and only if \mathcal{I} is positive. Otherwise it is strictly greater.

If \mathcal{I} is positive, then there exists $I \subseteq \llbracket 1; n \rrbracket$ such that $\sum_{i \in I} x_i = B$. Let \mathcal{O} be the orientation where $\mathcal{O}([t_i, w_i]) = (w_i, t_i)$ and $\mathcal{O}([r_i, v_i]) = (r_i, v_i)$ if and only if $i \in I$. In that case $\alpha_w = I$, $\alpha_v = \llbracket 1; n \rrbracket \setminus I$ and $\sum_{j \in \alpha_w} x_j = B$. Thus $load_{\mathcal{O}}(s) = \frac{1}{6 \cdot (n+1)} (2 \cdot (1+n) + 0) = \frac{1}{3}$. The load of the sources s_i is 1. Thus \mathcal{O} is feasible, $R(\mathcal{O}) = \frac{2}{3}$ and $m(\mathcal{O}) = \frac{1}{3}$.

If \mathcal{I} is negative, then whatever the feasible orientation is, we have the relation $2^{-m}(B - \sum_{i: \alpha(w_i)=1} x_i) \neq 0$. In addition, this value is lower than $2^{-m}B < 1$ and greater than $2^{-m}(B - \sum_{i=1}^n x_i) > -1$.

Then $2 \cdot (1 + |\alpha_w| + |\alpha_v|) + 2^{-m}(B - \sum_{j \in \alpha_w} x_j)$ cannot be an integer, thus we have $load_{\mathcal{O}}(s) \neq \frac{1}{3}$. Consequently, $load_{\mathcal{O}}(s) < \frac{1}{3}$, $R(\mathcal{O}) > \frac{2}{3}$ and $m(\mathcal{O}) < \frac{1}{3}$.

This proves that min-R and max-m are strongly NP-Hard. \square

Remark. *Theorem 3 also proves that no interesting parameterized algorithm can be hoped for the decision versions of min-R and max-m (with respect to the natural parameters of those problems).*

Note also that we can arbitrarily increase the constant $\frac{2}{3}$ of Theorem 3 (respectively decrease the value $\frac{1}{3}$) by increasing the production of s . We can also decrease this constant (respectively increase), but in order to keep s to be the source minimizing the load, we need to use a trick used later in the proof of Theorem 4 in order to arbitrarily increase the load of all the sources.

4 Hardness of approximation for min-R

This section is devoted to extending the reduction of Theorem 3 in order to prove that there is no efficient approximation algorithm for min-R unless $P = NP$. Note that this proof does not apply to max-m. As we will see in the next section, there is indeed an approximation algorithm for this problem.

Theorem 4. *Unless $P = NP$, then, for any constant $c \in \mathbb{N}$, there is no polynomial approximation algorithm for min-R with ratio $2^{|T|^c}$ for a given tree T .*

Proof. We use a construction that is similar to the one given in the proof of Theorem 3 to build a GAP reduction from the Subset Sum problem. Let \mathcal{J} be the instance obtained in that proof, described in Figure 3. We then use the same notations in this proof (particularly n , the number of variables x_i in the Subset Sum instance).

Let $c \in \mathbb{N}$ and $r = 2^{|T|^c}$. We set three values $H \in \mathbb{N}$, $\xi, L \in \mathbb{Q}$ such that:

$$L = 2 + \frac{2^{-m}}{r \cdot (n+1)}; \quad \frac{2}{L} < 1 - \xi; \quad \frac{H}{H+10} \geq 1 - \xi$$

Those values are polynomial with respect to the size of \mathcal{I} (recall that $\log(r)$ depends on $|T|$ which is polynomial in $|\mathcal{I}|$).

From \mathcal{J} , we build an instance \mathcal{K} by increasing the load of all the sources except s . For each such source s' , we attach to s' a sink with power H , and we increase $Prod(s')$ by H . Thus the load of s' is at least $H/(H+10) \geq 1 - \xi$. Note that this operation does not change the feasibility of any orientation. It also does not change the load of the source s_c which is 1. For the source s , we replace $Prod(s)$ by L . Note that this production is not an integer, but if we write $L = p/q$, we can simply multiply each power, capacity and production by q to get an equivalent instance with integer productions, powers and capacities.

Given a feasible orientation \mathcal{O} , the load of s is at most $\frac{2}{L} < 1 - \xi$ thus $\min(\mathcal{O}) = load_{\mathcal{O}}(s)$.

Let α_v be the indexes $i \in \llbracket 1; n \rrbracket$ such that $\mathcal{O}([r_i, v_i]) = (v_i, r_i)$ and α_w be those for which $\mathcal{O}([t_i, w_i]) = (w_i, t_i)$.

$$load_{\mathcal{O}}(s) = \frac{1}{L \cdot (n+1)} \left(2 \cdot (1 + |\alpha_w| + |\alpha_v|) + 2^{-m} (B - \sum_{j \in \alpha_w} x_j) \right)$$

If \mathcal{I} is positive, then there exists $I \subseteq \llbracket 1; n \rrbracket$ such that $\sum_{i \in I} x_i = B$. Let \mathcal{O} be the orientation where $\mathcal{O}([t_i, w_i]) = (w_i, t_i)$ and $\mathcal{O}([r_i, v_i]) = (r_i, v_i)$ if and only if $i \in I$. In that case $\alpha_w = I$, $\alpha_v = \llbracket 1; n \rrbracket \setminus I$ and $\sum_{j \in \alpha_w} x_j = B$. Thus the load of s is $load_{\mathcal{O}}(s) = \frac{1}{L \cdot (n+1)} (2 \cdot (1 + n) + 0) = 2/L$. The load of the sources s_i is 1. Thus \mathcal{O} is feasible, and $R(\mathcal{O}) = 1 - 2/L$.

If \mathcal{I} is negative, then whatever the feasible orientation is, we have the relation $2^{-m} (B - \sum_{i: \alpha(w_i)=1} x_i) \neq 0$. The difference $B - \sum_{i: \alpha(w_i)=1} x_i$ is necessarily negative (otherwise the capacity of s is exceeded) and is an integer. Thus

$$\begin{aligned} 2 \cdot (1 + |\alpha_w| + |\alpha_v|) + 2^{-m} (B - \sum_{j \in \alpha_w} x_j) &\leq 2 \cdot (n+1) - 2^{-m} \\ load_{\mathcal{O}}(s) &\leq \frac{2}{L} - \frac{2^{-m}}{L \cdot (n+1)} \\ R(\mathcal{O}) &\geq 1 - \frac{2}{L} + \frac{2^{-m}}{L \cdot (n+1)} \end{aligned}$$

$$\text{However } \frac{1 - \frac{2}{L} + \frac{2^{-m}}{L \cdot (n+1)}}{1 - \frac{2}{L}} = \frac{L - 2 + \frac{2^{-m}}{n+1}}{L - 2} = \frac{\frac{2^{-m}}{r \cdot (n+1)} + \frac{2^{-m}}{n+1}}{\frac{2^{-m}}{r \cdot (n+1)}} = 1 + r$$

Consequently, any polynomial approximation for min-R with ratio strictly less than $1 + r$ could decide in polynomial time whether \mathcal{I} is positive or not. Unless $P = NP$, there is no r -polynomial approximation for min-R. \square

5 Approximability of max-m and min-R

In this section, we prove that there exists an FPTAS¹ for max-m and an FPTAS with an absolute ratio² for max-m and min-R. Note that this last result does not contradict the inapproximability result of min-R as we do not use the same metric: there is no FPTAS (with a relative ratio) for min-R, but there exists an FPTAS with an absolute ratio.

The key idea to overcome the hardness of the reduction of Theorem 3 consists in noticing that, for many feasible solutions \mathcal{O} , the value $m(\mathcal{O})$ is close to $1/3$. In particular, it seems that the flow obtained in the arc (s, w) does not depend so much on the chosen orientation as this flow vary from $2 - \varepsilon$ to 2 (where ε is a very small value). The problem is NP-hard because it is hard to achieve the desired value 2 , but it is not hard to get a value really close to 2 . For example, one can set $\mathcal{O}([w_i, t_i]) = (w_i, t_i)$ and $\mathcal{O}([v_i, r_i]) = (r_i, v_i)$ for all i , to get $load_{\mathcal{O}}(s) = 1/3 + 2^{-m} \cdot (B - \sum_{j=1}^n x_j) / (6 \cdot (n + 1)) > 1/3 - 1/(6 \cdot (n + 1))$.

Similarly, Theorem 4 shows that it is hard to find a feasible solution with a small relative error for min-R. In the reduction, the load reserves of all the feasible orientations are close to 0 as all the loads are close to 1 . It is then not hard to find a feasible solution with a very small load reserve, in which case we achieve a very small absolute error. However, the relative error is very high as the optimal value is almost 0 .

In this context, one way to approximate the problem is to round the flow at each step of the calculation to get rid of the small variations. This defines, for each orientation, a new metric based on the rounded flow instead of the flow itself in which we focus on the rounded load of each source (that is the output rounded flow divided by the production of the source). Instead of searching for an optimal solution, we will focus on finding a feasible orientation that minimizes the rounded load reserve or maximizes the minimum rounded load.

Three main questions arise from this strategy:

- First, what makes the computation with the rounded flow easier? An important property of that flow is that there should be a polynomial number of possible values of a rounded flow. This allows to enumerate all these values and use them in a dynamic programming algorithm. Furthermore, this allows to enumerate all the possible values of rounded loads and thus all the possible objective values in polynomial time. To achieve such a result, it is necessary not to use a linear rounding, that is rounding to

¹A Fully Polynomial Time Approximation Scheme (or FPTAS) for a minimization problem (respectively maximization problem) is an algorithm that, given a value $\varepsilon > 0$, returns a feasible solution for which the objective value is equal to the optimal value multiplied by at most $1 + \varepsilon$ (respectively $1 - \varepsilon$). This algorithm is polynomial in the size of the instance and in $1/\varepsilon$.

²An FPTAS with absolute ratio for a minimization problem (respectively maximization problem) is an algorithm that, given a value $\varepsilon > 0$, returns a feasible solution for which the objective value is equal to the optimal value shifted by at most ε (respectively $-\varepsilon$). This algorithm is polynomial in the size of the instance and in $1/\varepsilon$.

the closest fraction of some value ε . Indeed, the real flow belongs to an interval that has an exponential size. To overcome this difficulty, we use a rounding that depends on the value of the flow: the larger the flow is, the larger the rounding error is. To do so, we cut each interval $[2^i; 2^{i+1}]$ into $\lceil \frac{1}{\varepsilon} \rceil$ pieces and we round the flow to the closest piece.

- Second, does this rounded flow provide a good approximation ratio? As we compute the rounding flow from the sinks to the sources, the rounding error propagates and increases. We show that with a polynomially small value ε this rounding error can be bounded by an arbitrarily small constant.
- Third, our technique is, in some way, adapted from the FPTAS algorithm for the Knapsack problem [14]. In this algorithm, the coefficients of the objective functions are rounded but not those of the constraints, which trivially makes the set of feasible solutions unchanged during the rounding process. In our problem, the objective function and the capacity constraints are based on the flow. Since the rounded flow is lower than the real flow, there may be some orientations that are feasible with respect to the rounded flow and not with the real flow. This is the tricky part of the algorithm. We want the rounded flow to be used only to define a new metric that is easier to optimize while not changing the set of feasible solutions. Thus, we must somehow keep the real flow in the computation to ensure that the capacity constraints are satisfied by any orientation that can be given by algorithm. To do this, we will use a technique that extends the one used in [4] to search for the existence of a feasible solution.

In this algorithm, for each possible orientation (u, v) of an edge, the algorithm computes two values: the maximum flow that can be produced by the sources and that can be sent to v through u , and the minimum flow that can be demanded by the sinks and that should come from u through v . If the former is greater than the latter, there exists a feasible orientation. This is done with a dynamic programming algorithm using the tree structure of the graph. In the optimization variant of the problem, we should search for the same two values (computed with the real flow also with an extension of the dynamic programming algorithm of [4]) with the additional constraint that the rounded loads of the sources lie in an interval $[\widetilde{m}; \widetilde{M}]$. If such values exist, we know there exists a feasible solution where the rounded load reserve is at most $\widetilde{M} - \widetilde{m}$. By enumerating all the possible couples of rounded loads, we get the desired feasible solution.

Note that this rounding method has a real physical meaning, since it is not possible to manipulate an electrical flow with an infinite precision. Thus a flow that is close to 2 will be measured as 2 and thus rounded. Consequently, the approximate solution can be seen as the best solution that could be obtained with regard to the physical constraints of the measurement.

This section is divided into three parts. The first part defines the rounded flow and proves some useful properties about it. The second part proves the

approximation ratio of the algorithms. The last part is dedicated to the description of a polynomial time algorithm that optimizes the rounded loads of the sources.

Definition 3. *Throughout this section, we denote with n the size of T ; with Π the sum $\sum_{p \in P} Pow(p)$; and with \mathcal{N} an arbitrary numbering of the nodes of T .*

5.1 Rounding the flow and computing the rounding error

The goal of rounding the flow is to manipulate a polynomial number of possible values in each arc.

Without rounding, the flow is a rational between $1/n^n$ (as the flow can be divided by at most n at most n times) and Π (which roughly leads to an exponential number $O(n^n \Pi)$ of possible values for the flow). A naive rounding method would consist in fixing a value $\varepsilon > 0$ and rounding to the closest multiple of ε/n . We can show that, as the rounding error propagates, it never exceeds ε which leads to an ε -approximation ratio of the loads. However, this decreases the number of values to $\varepsilon \Pi$ which is only pseudo-polynomial.

To overcome this obstacle, one should note that there is no need to round high flows to close values. Remember that we are interested in optimizing the loads of the sources, which means that the rounding error of a high rounded flow will be divided by the production of sources feeding that called power. And a high power should be necessarily powered by a source with a high production due to the capacity constraint. Consequently, the greater the flow value, the greater the rounding error can be.

We now fix two values $\varepsilon' \in]0; 1/2[$ and $\varepsilon = \varepsilon'/(n^2 + 1)^2$ and define hereinafter the rounded flow, the rounded minimum load and the rounded load reserve.

Definition 4 (Rounding a rational). *Let $f > 0$ be a rational. If $f \in [2^i; 2^{i+1}[$ then we set the error $e(f) = 2^i \varepsilon$. We define $a(f)$ as the closest multiple of $e(f)$ lower than f , that is $a(f) = \left\lfloor \frac{f}{e(f)} \right\rfloor \cdot e(f)$.*

Lemma 1. $a(f) \geq f \cdot (1 - \varepsilon)$.

Proof. Let i such that $f \in [2^i; 2^{i+1}[$, then $a(f) = \left\lfloor \frac{f}{e(f)} \right\rfloor \cdot e(f) \geq f - e(f) \geq f - \varepsilon 2^i \geq f - \varepsilon \cdot f$. \square

We now define the operator \oplus that is the rounding version of $\sum_{i=1}^p f_i/d$ and will be used in the formula of the rounded flow.

Definition 5. *Let $F = (f_1, f_2, \dots, f_p)$ be rationals and d be an positive integer. Then*

$$\oplus(F, d) = \begin{cases} 0 & \text{if } p = 0 \\ a(\oplus((f_1, f_2, \dots, f_{p-1}), d) + \frac{f_p}{d}) & \text{otherwise} \end{cases}$$

Remark. One may ask why we use this recursive formula leading to $a(a(\dots a(a(\frac{f_1}{d}) + \frac{f_2}{d}) + \dots) + \frac{f_p}{d})$ that accumulates many rounding errors where $a(\sum_{i=1}^p f_i/d)$ seems to do the same job with less errors. This has to do with the complexity of the approximation algorithm we will describe later. It is polynomial if we use the recursive formula and exponential if we use the latter formula.

Lemma 2. $\oplus(F, d) \geq \frac{1}{d} \sum_{i=1}^p f_i \cdot (1 - \varepsilon)^p$.

Proof. The proof is done by induction on p . If $p = 0$, the two values equal 0.

If we assume that $\oplus((f_1, f_2, \dots, f_{p-1}), d) \geq \frac{1}{d} \sum_{i=1}^{p-1} f_i \cdot (1 - \varepsilon)^{p-1}$.

Then

$$\oplus(F, d) = a(\oplus((f_1, f_2, \dots, f_{p-1}), d) + \frac{f_p}{d})$$

By Lemma 1,

$$\oplus(F, d) \geq \left(\oplus((f_1, f_2, \dots, f_{p-1}), d) + \frac{f_p}{d} \right) \cdot (1 - \varepsilon)$$

$$\oplus(F, d) \geq \left(\frac{1}{d} \sum_{i=1}^{p-1} f_i \cdot (1 - \varepsilon)^{p-1} + \frac{f_p}{d} \right) \cdot (1 - \varepsilon)$$

As $(1 - \varepsilon)^p \leq 1 - \varepsilon$

$$\oplus(F, d) \geq \left(\frac{1}{d} \sum_{i=1}^{p-1} f_i + \frac{f_p}{d} \right) \cdot (1 - \varepsilon)^p$$

□

We can now define the rounded flow.

Definition 6 (Rounded flow). Let \mathcal{O} be a feasible orientation.

Let $d^-(v)$ be the in-degree of v and (v_1, v_2, \dots, v_p) be the set of successors of v , ordered using the numbering \mathcal{N} . The rounded flow $\tilde{F}_{\mathcal{O}}(u, v)$ going through each arc (u, v) entering v is

- if $v \in W$, $\tilde{F}_{\mathcal{O}}(u, v) = \oplus((\tilde{F}_{\mathcal{O}}(v, v_i))_{i \in \llbracket 1; p \rrbracket}, d^-(v))$ if $d^-(v) \neq 0$ and $+\infty$ otherwise
- if $v \in S$, $\tilde{F}_{\mathcal{O}}(u, v) = \oplus((\tilde{F}_{\mathcal{O}}(v, v_i))_{i \in \llbracket 1; p \rrbracket}, d^-(v) + 1)$
- if $v \in P$, $\tilde{F}_{\mathcal{O}}(u, v) = \oplus((Pow(v), \tilde{F}_{\mathcal{O}}(v, v_i))_{i \in \llbracket 1; p \rrbracket}, d^-(v))$ if $d^-(v) \neq 0$ and $+\infty$ otherwise

We also denote by $\tilde{F}_{\mathcal{O}}(v)$ this value as u does not intervene in the formula.

Example 3. We reuse the network given in Figure 1. We show in Figure 4 the rounded flow of the orientations given in Figures 1b and 1d. We use the parameter $\varepsilon' = 0.1$. As $n = 8$, we have $\varepsilon = \varepsilon'/(n^2 + 1)^2 \simeq 2.36 \cdot 10^{-5}$.

We assume the numbering \mathcal{N} of the nodes of T gives the following order: $(s_1, s_2, w_1, w_2, w_3, p_1, p_2, p_3)$.

On the two figures, the rounded flow $\tilde{F}_{\mathcal{O}}(s_1, p_3)$ is $\oplus((\text{Pow}((p_3))), 1) = a(\text{Pow}((p_3))) = a(10)$. The value 10 is rounded to the highest multiple of $2^3 \cdot \varepsilon \simeq 1.89 \cdot 10^{-4}$ lower than 10, which is $2^3 \cdot \varepsilon \cdot 52812 \simeq 9.99991$. We can also compute $\tilde{F}_{\mathcal{O}}(w_3, p_2) = a(20)$, the highest multiple of $2^4 \cdot \varepsilon$ which is around 19.99981 and $\tilde{F}_{\mathcal{O}}(w_1, p_1) = a(50)$, the highest multiple of $2^5 \cdot \varepsilon$ which is around 49.99952.

On Figure 4a, $\tilde{F}_{\mathcal{O}}(s_1, w_1) = a(\tilde{F}_{\mathcal{O}}(w_1, p_1)) = a(a(50))$ is the highest multiple of $2^5 \cdot \varepsilon$ lower than $a(50)$ which is exactly $a(50) \simeq 49.99952$. The same occurs for $\tilde{F}_{\mathcal{O}}(s_2, p_3) = 19.99981$. The rounded flow in s_1 is

$$\begin{aligned} \tilde{F}_{\mathcal{O}}(s_1) &= \oplus((\tilde{F}_{\mathcal{O}}(s_1, w_1), \tilde{F}_{\mathcal{O}}(s_1, p_3)), 1) \\ &= a(a(\tilde{F}_{\mathcal{O}}(s_1, w_1)) + \tilde{F}_{\mathcal{O}}(s_1, p_3)) \\ &= a(a(a(a(50))) + a(10)) \\ &\simeq 59.99943 \end{aligned}$$

Similarly, $\tilde{F}_{\mathcal{O}}(s_2) = 19.99981$.

On Figure 4a, the same reasoning leads to $\tilde{F}_{\mathcal{O}}(s_2) \simeq 9.99991$ and

$$\begin{aligned} \tilde{F}_{\mathcal{O}}(s_1) &= \oplus((\tilde{F}_{\mathcal{O}}(s_1, w_1), \tilde{F}_{\mathcal{O}}(s_1, p_3)), 1) \\ &= a(a(\tilde{F}_{\mathcal{O}}(s_1, w_1)) + \tilde{F}_{\mathcal{O}}(s_1, p_3)) \\ &\simeq a(69.99934) \\ &\simeq 69.99858 \end{aligned}$$

Recall that we set a constant ε' and fixed $\varepsilon = \varepsilon'/(n^2 + 1)^2$. We now prove that the rounding error that is propagated through the arcs of the tree does not lead to a mistake greater than $(1 - \varepsilon')$.

Lemma 3. Let \mathcal{O} be a feasible orientation and (u, v) be an arc of the directed tree, then $\tilde{F}_{\mathcal{O}}(u, v) \in [F_{\mathcal{O}}(u, v) \cdot (1 - \varepsilon'); F_{\mathcal{O}}(u, v)]$.

Proof. The upper bound is trivial as we round down the flow. We prove the lower bound by induction on p , where p is the length of a longest path from v to a sink, that $\tilde{F}_{\mathcal{O}}(u, v) \geq F_{\mathcal{O}}(u, v) \cdot (1 - \varepsilon)^{np+1}$. Note that, if such a sink does not exist, then the flow and the rounded flow going out of v is 0.

If $p = 0$ then v is a sink and there is no path from v to another sink. In that case, $\tilde{F}_{\mathcal{O}}(u, v) = \oplus((\text{Pow}(v)), d^-(v)) = a(\text{Pow}(v)/d^-(v)) = a(F_{\mathcal{O}}(u, v)) \geq F_{\mathcal{O}}(u, v) \cdot (1 - \varepsilon)$ by Lemma 1.

We now assume that $p > 0$ and that the property is true for every node at distance at most $p - 1$ from a sink. Let v be a node for which the farthest sink

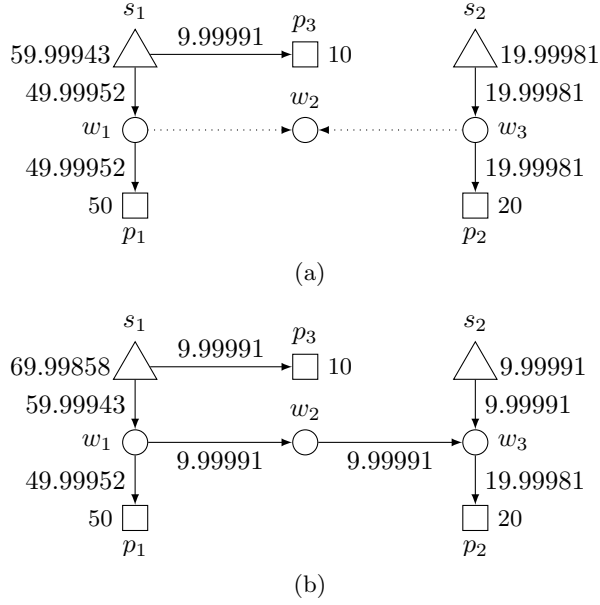


Figure 4: Example of rounded flow. Next to each arc and source is written the rounded flow going through it.

is at distance p and let (v_1, v_2, \dots, v_q) be the successors of v , ordered using the numbering \mathcal{N} . The longest path from v_i to a sink is $p_i \leq p - 1$. By induction, $\tilde{F}_{\mathcal{O}}(v, v_i) \geq F_{\mathcal{O}}(v, v_i) \cdot (1 - \varepsilon)^{np_i+1} \geq F_{\mathcal{O}}(v, v_i) \cdot (1 - \varepsilon)^{n \cdot (p-1)+1}$.

We also assume without loss of generality that $v \in W$. The calculations can be easily adapted to the cases where $v \in P$ or $v \in S$. Let

$$\tilde{F}_{\mathcal{O}}(u, v) = \oplus((\tilde{F}_{\mathcal{O}}(v, v_i))_{i \in \llbracket 1; p \rrbracket}, d^-(v))$$

By Lemma 2

$$\geq \frac{1}{d^-(v)} \sum_{i=1}^q \tilde{F}_{\mathcal{O}}(v, v_i) \cdot (1 - \varepsilon)^q$$

As $q \leq n$

$$\begin{aligned}
&\geq \frac{1}{d^-(v)} \sum_{i=1}^q \tilde{F}_{\mathcal{O}}(v, v_i) \cdot (1 - \varepsilon)^n \\
&\geq \frac{1}{d^-(v)} \sum_{i=1}^q F_{\mathcal{O}}(v, v_i) \cdot (1 - \varepsilon)^{n \cdot (p-1)+1} \cdot (1 - \varepsilon)^n \\
&\geq \frac{1}{d^-(v)} \sum_{i=1}^q F_{\mathcal{O}}(v, v_i) \cdot (1 - \varepsilon)^{np+1} \\
&\geq F_{\mathcal{O}}(u, v) \cdot (1 - \varepsilon)^{np+1}
\end{aligned}$$

In addition, as $p \leq n$

$$(1 - \varepsilon)^{pn+1} \geq (1 - \varepsilon)^{n^2+1}$$

$$\text{As } (1 - \varepsilon)^{n^2+1} = \sum_{i=0}^{n^2+1} \binom{n^2+1}{i} (-\varepsilon)^i$$

$$(1 - \varepsilon)^{pn+1} \geq \sum_{i=0}^{n^2+1} \binom{n^2+1}{i} (-\varepsilon)^i$$

$$(1 - \varepsilon)^{pn+1} \geq 1 - \sum_{i=1}^{n^2+1} \binom{n^2+1}{i} \varepsilon^i$$

As $\binom{a}{b} \leq a^b$

$$(1 - \varepsilon)^{pn+1} \geq 1 - \sum_{i=1}^{n^2+1} (\varepsilon \cdot (n^2 + 1))^i$$

$$\text{As } \varepsilon \cdot (n^2 + 1) = \frac{\varepsilon'}{n^2+1}$$

$$(1 - \varepsilon)^{pn+1} \geq 1 - \sum_{i=1}^{n^2+1} \left(\frac{\varepsilon'}{n^2+1} \right)^i$$

$$(1 - \varepsilon)^{pn+1} \geq 1 - (n^2 + 1) \frac{\varepsilon'}{n^2+1} = 1 - \varepsilon'$$

□

5.2 Approximation factor

With the rounded flow, we can now define the rounded loads, and thus a rounded version of $m(\mathcal{O})$, $M(\mathcal{O})$ and $R(\mathcal{O})$.

Definition 7 (Rounded objectives). *Given a feasible orientation \mathcal{O} , the rounded load of a source $s \in S$ is defined as the ratio $\widetilde{\text{load}}_{\mathcal{O}}(s) = \frac{\widetilde{F}_{\mathcal{O}}(s)}{\text{Prod}(s)}$.*

- The minimum rounded load $\widetilde{m}(\mathcal{O})$ is $\min_{s \in S} \widetilde{\text{load}}_{\mathcal{O}}(s)$.
- The maximum rounded load $\widetilde{M}(\mathcal{O})$ is $\max_{s \in S} \widetilde{\text{load}}_{\mathcal{O}}(s)$.
- The rounded load reserve $\widetilde{R}(\mathcal{O})$ is $\widetilde{M}(\mathcal{O}) - \widetilde{m}(\mathcal{O})$

In this part we prove that a feasible solution minimizing the rounded load reserve is a close approximation of the solutions minimizing the load reserve. Similarly, a solution that maximizes the minimum rounded load approximates the solutions that maximize the minimum load.

Note that rounding the flow may change the set of feasible orientations as such a flow is lower than the regular flow $F_{\mathcal{O}}$ and thus may be lower than some capacity exceeded by the flow. However, hereinafter, by *feasible solution*, we mean a solution satisfying the capacity constraints and the demand constraints with the regular flow as explained in Definition 1, not with the rounded flow $\widetilde{F}_{\mathcal{O}}$, which only changes the metric used to measure the quality of the feasible solutions. We show in the next subsection how to build solutions with good rounded metrics in polynomial time even though we use the regular flow to check the feasibility of the returned orientation.

Lemma 4. *Let \mathcal{O}^* be an orientation maximizing $m(\mathcal{O})$ and $\widetilde{\mathcal{O}}$ a feasible solution maximizing $\widetilde{m}(\mathcal{O})$ then $m(\widetilde{\mathcal{O}}) \geq m(\mathcal{O}^*) \cdot (1 - \varepsilon')$ and $m(\widetilde{\mathcal{O}}) \geq m(\mathcal{O}^*) - \varepsilon'$.*

Proof. Let s_m be the source with the lower (not rounded) load in $\widetilde{\mathcal{O}}$ and s_m^* be the source with the lower rounded load in \mathcal{O}^* , in other words: $m(\widetilde{\mathcal{O}}) = \frac{F_{\widetilde{\mathcal{O}}}(s_m)}{\text{Prod}(s_m)}$ and $\widetilde{m}(\mathcal{O}^*) = \frac{\widetilde{F}_{\mathcal{O}^*}(s_m^*)}{\text{Prod}(s_m^*)}$.

By Lemma 3 and definition of $\widetilde{m}(\widetilde{\mathcal{O}})$,

$$m(\widetilde{\mathcal{O}}) = \frac{F_{\widetilde{\mathcal{O}}}(s_m)}{\text{Prod}(s_m)} \geq \frac{\widetilde{F}_{\widetilde{\mathcal{O}}}(s_m)}{\text{Prod}(s_m)} \geq \widetilde{m}(\widetilde{\mathcal{O}})$$

As, by definition of $\widetilde{\mathcal{O}}$, it maximizes $\widetilde{m}(\widetilde{\mathcal{O}})$, then

$$\widetilde{m}(\widetilde{\mathcal{O}}) \geq \widetilde{m}(\mathcal{O}^*) = \frac{\widetilde{F}_{\mathcal{O}^*}(s_m^*)}{\text{Prod}(s_m^*)}$$

By Lemma 3 and definition of $m(\mathcal{O}^*)$,

$$\widetilde{m}(\mathcal{O}^*) \geq \frac{F_{\mathcal{O}^*}(s_m^*)}{\text{Prod}(s_m^*)} \cdot (1 - \varepsilon') \geq m(\mathcal{O}^*) \cdot (1 - \varepsilon')$$

Finally, as $m(\mathcal{O}^*) \leq 1$, we also have

$$m(\mathcal{O}^*) \cdot (1 - \varepsilon') \geq m(\mathcal{O}^*) - \varepsilon'$$

□

Lemma 5. *Let \mathcal{O}^* be an orientation minimizing $R(\mathcal{O})$ and $\tilde{\mathcal{O}}$ a feasible solution minimizing $\tilde{R}(\mathcal{O})$ then $R(\tilde{\mathcal{O}}) \leq R(\mathcal{O}^*) + 3\varepsilon'$.*

Proof. Let s_m and s_M be the sources with respectively the lower and higher (not rounded) loads in $\tilde{\mathcal{O}}$ and let s_m^* and s_M^* be the sources with respectively the lower and higher rounded loads in \mathcal{O}^* , in other words:

- $M(\tilde{\mathcal{O}}) = \frac{F_{\tilde{\mathcal{O}}}(s_M)}{Prod(s_M)}$ and $m(\tilde{\mathcal{O}}) = \frac{F_{\tilde{\mathcal{O}}}(s_m)}{Prod(s_m)}$.
- $\tilde{M}(\mathcal{O}^*) = \frac{\tilde{F}_{\mathcal{O}^*}(s_M^*)}{Prod(s_M^*)}$ and $\tilde{m}(\mathcal{O}^*) = \frac{\tilde{F}_{\mathcal{O}^*}(s_m^*)}{Prod(s_m^*)}$.

$$R(\tilde{\mathcal{O}}) = M(\tilde{\mathcal{O}}) - m(\tilde{\mathcal{O}}) = \frac{F_{\tilde{\mathcal{O}}}(s_M)}{Prod(s_M)} - \frac{F_{\tilde{\mathcal{O}}}(s_m)}{Prod(s_m)}$$

By Lemma 3,

$$R(\tilde{\mathcal{O}}) \leq \frac{\tilde{F}_{\tilde{\mathcal{O}}}(s_M)}{Prod(s_M)} \cdot \frac{1}{1 - \varepsilon'} - \frac{\tilde{F}_{\tilde{\mathcal{O}}}(s_m)}{Prod(s_m)}$$

By definition of $\tilde{M}(\tilde{\mathcal{O}})$ and $\tilde{m}(\tilde{\mathcal{O}})$

$$R(\tilde{\mathcal{O}}) \leq \tilde{M}(\tilde{\mathcal{O}}) \cdot \frac{1}{1 - \varepsilon'} - \tilde{m}(\tilde{\mathcal{O}})$$

As $\frac{1}{1 - \varepsilon'} = 1 + \frac{\varepsilon'}{1 - \varepsilon'}$ and $\tilde{M}(\tilde{\mathcal{O}}) \leq 1$

$$\tilde{M}(\tilde{\mathcal{O}}) \cdot \frac{1}{1 - \varepsilon'} - \tilde{m}(\tilde{\mathcal{O}}) \leq \tilde{M}(\tilde{\mathcal{O}}) - \tilde{m}(\tilde{\mathcal{O}}) + \frac{\varepsilon'}{1 - \varepsilon'} = \tilde{R}(\tilde{\mathcal{O}}) + \frac{\varepsilon'}{1 - \varepsilon'}$$

As, by definition of $\tilde{\mathcal{O}}$, it minimizes $\tilde{R}(\tilde{\mathcal{O}})$, then

$$\tilde{R}(\tilde{\mathcal{O}}) + \frac{\varepsilon'}{1 - \varepsilon'} \leq \tilde{R}(\mathcal{O}^*) + \frac{\varepsilon'}{1 - \varepsilon'} = \frac{\tilde{F}_{\mathcal{O}^*}(s_M^*)}{Prod(s_M^*)} - \frac{\tilde{F}_{\mathcal{O}^*}(s_m^*)}{Prod(s_m^*)} + \frac{\varepsilon'}{1 - \varepsilon'}$$

By Lemma 3

$$\tilde{R}(\mathcal{O}^*) + \frac{\varepsilon'}{1 - \varepsilon'} \leq \frac{F_{\mathcal{O}^*}(s_M^*)}{Prod(s_M^*)} - \frac{F_{\mathcal{O}^*}(s_m^*)}{Prod(s_m^*)} \cdot (1 - \varepsilon') + \frac{\varepsilon'}{1 - \varepsilon'}$$

By definition of $M(\mathcal{O}^*)$ and $m(\mathcal{O}^*)$

$$\tilde{R}(\mathcal{O}^*) + \frac{\varepsilon'}{1-\varepsilon'} \leq M(\mathcal{O}^*) - m(\mathcal{O}^*) \cdot (1-\varepsilon') + \frac{\varepsilon'}{1-\varepsilon'}$$

Finally, as $\varepsilon' \in]0; \frac{1}{2}[$, then $\frac{\varepsilon'}{1-\varepsilon'} \leq 2\varepsilon'$, and as $m(\mathcal{O}^*) \leq 1$

$$M(\mathcal{O}^*) - m(\mathcal{O}^*) \cdot (1-\varepsilon') + \frac{\varepsilon'}{1-\varepsilon'} \leq M(\mathcal{O}^*) - m(\mathcal{O}^*) + 3\varepsilon'$$

□

5.3 Compute a feasible orientation minimizing $\tilde{R}(\mathcal{O})$ or maximizing $\tilde{m}(\mathcal{O})$ in polynomial time

Recall that we denote by n the size of the tree T and by $\Pi = \sum_{p \in P} Pow(p)$. We prove it is possible to return in polynomial time the approximate solutions mentioned in Lemmas 4 and 5.

To do so, we adapt the algorithm of [4], which solved VALID in polynomial time, to those problems.

We first prove the following lemma that states the number of distinct values for a rounded flow is bounded by a polynomial.

Lemma 6. *Let*

$$\tilde{F} = \{\tilde{F}_{\mathcal{O}}(\mathcal{O}([u, v])), (u, v) \in T, \mathcal{O} \text{ is a feasible orientation}\}$$

Then, $|\tilde{F}| \leq (n \log(n) + \log(\Pi) + 1)(1 + \frac{1}{\varepsilon}) + 1 = (n \log(n) + \log(\Pi) + 1)(1 + \frac{(n+1)^2}{\varepsilon}) + 1 = O(\frac{1}{\varepsilon} \cdot (n^3 \log(n) + n^2 \log(\Pi)))$.

Proof. Let $f \in [2^i; 2^{i+1}[$ then $a(f)$ is the closest multiple of $2^i \varepsilon$ lower than f . Those multiples belong to the interval $[\lfloor \frac{2^i}{2^i \varepsilon} \rfloor \cdot 2^i \varepsilon; 2^{i+1}] \subset [2^i(1-\varepsilon); 2^{i+1}]$. Then there are at most $(2^{i+1} - 2^i \cdot (1-\varepsilon))/2^i \varepsilon = 1 + 1/\varepsilon$ such multiples. As a consequence, if the value i is between a and b then, there are at most $(1 + 1/\varepsilon) \cdot (b - a)$ possible values for the rounded flow.

Let \mathcal{O} be a feasible orientation, and (u, v) be an arc in that orientation, then $\tilde{F}_{\mathcal{O}}(u, v) \leq F_{\mathcal{O}}(u, v) \leq \Pi$. Thus $i \leq \log(\Pi)$.

The flow can also be split. This happens at most once per node in the tree and each node has at most n entering arcs, thus the flow $F_{\mathcal{O}}(u, v)$ cannot be less than $1/n^n$ unless if the flow is zero. By Lemma 3, $\tilde{F}_{\mathcal{O}}(u, v) \geq (1-\varepsilon')/n^n$. We recall that ε' was chosen between 0 and 1/2, thus $1-\varepsilon' \geq 1/2$ and, then, $(1-\varepsilon')/n^n \geq 1/(2 \cdot n^n) = 2^{-n \log(n)-1}$. Consequently, $i \geq -n \log(n) - 1$.

Thus, the number of non null rounded flows is $(1+1/\varepsilon) \cdot (\log(\Pi) - (-n \log(n) - 1))$. There is one additional possible rounded flow: the null flow. □

5.3.1 Feasible semi-orientations

Given an edge $[u, v]$, we consider the two trees T_u and T_v of the forest $T \setminus [u, v]$, respectively containing u and v .

We consider a feasible orientation \mathcal{O} of T , where $\mathcal{O}([u, v]) = (u, v)$. The flow $F_{\mathcal{O}}$ and the rounded flow $\tilde{F}_{\mathcal{O}}$ in T_v and (u, v) do not depend on the orientation of T_u . Similarly, the flow and the rounded flow in T_u depend only on the flow and the rounded flow in (u, v) and on the orientation of T_u , but not on the orientations of T_v . We express this idea with the notion of semi-orientations.

Definition 8. An semi-orientation \mathcal{HO} outgoing from (u, v) is an orientation of the edges of T_v and of $[u, v]$ such that $\mathcal{HO}([u, v]) = (u, v)$. The calculations of the flow $F_{\mathcal{HO}}$ and rounded flow $\tilde{F}_{\mathcal{HO}}$ in $T_v \cup \{[u, v]\}$ follow the Definitions 1 and 6. We say \mathcal{HO} is feasible if it satisfies the demand and the capacity constraints on the nodes of T_v .

Definition 9. An semi-orientation \mathcal{HO} entering (u, v) is an orientation of the edges of T_u and of $[u, v]$ such that $\mathcal{HO}([u, v]) = (u, v)$. Given $f \geq 0$ and $\tilde{f} \in \tilde{F}$, if we set $F_{\mathcal{HO}}(u, v) = f$ and $\tilde{F}_{\mathcal{HO}}(u, v) = \tilde{f}$ then we can compute the flow $F_{\mathcal{HO}}$ and rounded flow $\tilde{F}_{\mathcal{HO}}$ in $T_u \cup \{[u, v]\}$ with Definitions 1 and 6. We denote by semi-orientation \mathcal{HO} entering (u, v) with f and \tilde{f} the triplet $(\mathcal{HO}, f, \tilde{f})$ and we write $F_{(\mathcal{HO}, f, \tilde{f})}$ and $\tilde{F}_{(\mathcal{HO}, f, \tilde{f})}$ the associated flows. We say $(\mathcal{HO}, f, \tilde{f})$ is feasible if and only if $F_{(\mathcal{HO}, f, \tilde{f})}$ satisfies the demand and the capacity constraints on the nodes of T_u .

Remark. Considering a feasible semi-orientation \mathcal{HO} outgoing from (u, v) , there is no constraint on the flow $F_{\mathcal{HO}}(u, v)$: the demand and the capacity constraints should only be satisfied for the nodes of T_v .

Remark. Considering an semi-orientation entering (u, v) with f and \tilde{f} , then, even if v is a sink, there is no relation between f , \tilde{f} and $\text{Pow}(v)$.

Lemma 7. Given $0 \leq f' < f$ and $\tilde{f} \in \tilde{F}$. If $(\mathcal{HO}, f, \tilde{f})$ is feasible, then $(\mathcal{HO}, f', \tilde{f})$ is also feasible.

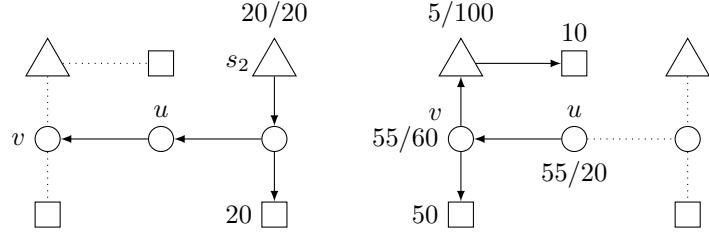
Proof. $F_{(\mathcal{HO}, f, \tilde{f})}$ satisfies the demand and the capacity constraint on the nodes of T_u . As $f' \leq f$, for every arc in T_u , $F_{(\mathcal{HO}, f', \tilde{f})}(u, v) \leq F_{(\mathcal{HO}, f, \tilde{f})}(u, v)$, thus $F_{(\mathcal{HO}, f', \tilde{f})}$ also satisfies the demand and capacity constraints. \square

Example 4. In Figure 5, we reuse the example of Figure 1 to show examples of feasible and not feasible semi-orientations.

5.3.2 Functions i and o

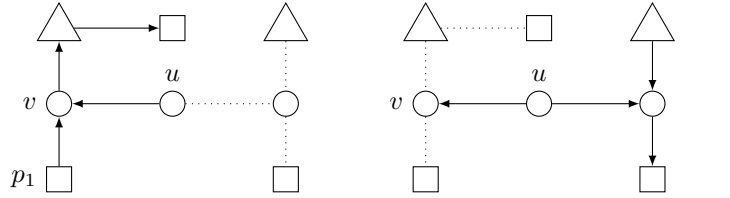
We adapt the notations $\tilde{M}(\mathcal{O})$ and $\tilde{m}(\mathcal{O})$ to the semi-orientations.

Given an semi-orientation \mathcal{HO} outgoing from (u, v) , $\tilde{M}(\mathcal{HO})$ and $\tilde{m}(\mathcal{HO})$ are the maximum and minimum rounded loads of the sources in T_v (not including u).



(a) Example of semi-orientation entering (u, v) . With $f = 0$ and any value \tilde{f} , this semi-orientation is feasible. However, if $f > 0$, it is not feasible anymore as the capacity constraint is not satisfied for s_2 .

(b) Example of semi-orientation outgoing from (u, v) . This semi-orientation is feasible as all the nodes of T_v satisfy the capacity and the demand constraints (even if u does not).



(c) This semi-orientation is not feasible as the demand constraint is not satisfied for p_1 .

(d) This semi-orientation is not feasible as the demand constraint is not satisfied for u .

Figure 5: Example of instance and orientations for that instance. For each orientation, it is explained if it is feasible according to Definitions 8 and 9.

Given an semi-orientation \mathcal{HO} entering (u, v) with f and \tilde{f} , $\tilde{M}(\mathcal{HO}, f, \tilde{f})$ and $\tilde{m}(\mathcal{HO}, f, \tilde{f})$ are the maximum and minimum rounded loads of the sources in T_u (not including v).

Remark. In case of an semi-orientation \mathcal{HO} outgoing from (u, v) where u is a source, as u is not part of T_v , the load of u does not intervene in the calculations of $\tilde{M}(\mathcal{HO})$ and $\tilde{m}(\mathcal{HO})$. This is necessary as, otherwise, only the flow going from u to v would count in the load of u and this would mistakenly reduce the value of $\tilde{M}(\mathcal{HO})$ and $\tilde{m}(\mathcal{HO})$.

Similarly, if v is a source and is considered in the definition of $\tilde{M}(\mathcal{HO}, f, \tilde{f})$ and $\tilde{m}(\mathcal{HO}, f, \tilde{f})$, where \mathcal{HO} is an semi-orientation entering (u, v) , we would have $\tilde{m}(\mathcal{HO}) = 0$ whatever the orientation of T_u is.

We write $\tilde{L} = \{\frac{\tilde{f}}{p_{rod}(s)} | \tilde{f} \in \tilde{F}, s \in S\}$. Note that $|\tilde{L}| = |\tilde{F}| \cdot |S|$ is polynomial

and that $\widetilde{M}(\mathcal{HO})$, $\widetilde{m}(\mathcal{HO})$, $\widetilde{M}(\mathcal{HO}, f, \widetilde{f})$ and $\widetilde{m}(\mathcal{HO}, f, \widetilde{f})$ are in \widetilde{L} .

Building a feasible orientation minimizing the rounded load reserve can be done using the following algorithm: given any two neighbors u and v and three values $f \in \widetilde{F}$, $\widetilde{M} \in \widetilde{L}$, $\widetilde{m} \in \widetilde{L}$, we search for a rational f such that there exists a feasible semi-orientation \mathcal{HO} outgoing from (u, v) satisfying $F_{\mathcal{HO}}(u, v) = f$ and $\widetilde{F}_{\mathcal{HO}}(u, v) = \widetilde{f}$ and a feasible semi-orientation \mathcal{HO}' entering (u, v) with f and \widetilde{f} . In addition, the loads should be such that $\widetilde{m}(\mathcal{HO}) \geq \widetilde{m}$, $\widetilde{m}(\mathcal{HO}, f, \widetilde{f}) \geq \widetilde{m}$, $\widetilde{M}(\mathcal{HO}) \leq \widetilde{M}$ and $\widetilde{M}(\mathcal{HO}, f, \widetilde{f}) \leq \widetilde{M}$.

If such semi-orientations exist, the union is a feasible orientation \mathcal{O} such that $\mathcal{O}([u, v]) = (u, v)$, $\widetilde{m}(\mathcal{O}) \geq \widetilde{m}$ and $\widetilde{M}(\mathcal{O}) \leq \widetilde{M}$. We start again for every possible values of \widetilde{f} , \widetilde{M} , \widetilde{m} . We then renew this operation by reversing the orientation of $[u, v]$. We get a feasible orientation minimizing the rounded load reserve by returning the one where $\widetilde{M} - \widetilde{m}$ is minimum, and a solution maximizing the minimum rounded load by returning the one where \widetilde{m} is maximum (in that case, enumerating all the values of \widetilde{M} is not necessary, we can set \widetilde{M} to 1).

It is possible to enumerate the triplets \widetilde{f} , \widetilde{m} and \widetilde{M} in polynomial time, but not all the possible values of f . This is why we use auxiliary functions i and o that will respectively give the lowest possible value f for \mathcal{HO} and the highest possible value f for \mathcal{HO}' .

Definition 10. Let $\widetilde{f} \in \widetilde{F}$, $\widetilde{M} \in \widetilde{L}$, $\widetilde{m} \in \widetilde{L}$ and $[u, v]$ be an edge of T . Let \mathcal{P} be the set of all values f such that there exists a feasible semi-orientation \mathcal{HO} outgoing from (u, v) such that $F_{\mathcal{HO}}(u, v) = f$, $\widetilde{F}_{\mathcal{HO}}(u, v) = \widetilde{f}$, $\widetilde{M}(\mathcal{HO}) \leq \widetilde{M}$ and $\widetilde{m}(\mathcal{HO}) \geq \widetilde{m}$. We then write $i(u, v, \widetilde{f}, \widetilde{M}, \widetilde{m}) = \min(\mathcal{P})$. If $\mathcal{P} = \emptyset$, then $i(u, v, \widetilde{f}, \widetilde{M}, \widetilde{m}) = +\infty$.

Definition 11. Let $\widetilde{f} \in \widetilde{F}$, $\widetilde{M} \in \widetilde{L}$, $\widetilde{m} \in \widetilde{L}$ and $[u, v]$ be an edge of T . Let \mathcal{P} be the set of all values f such that there exists a feasible semi-orientation \mathcal{HO}' entering in (u, v) with f and \widetilde{f} such that $\widetilde{M}(\mathcal{HO}', f, \widetilde{f}) \leq \widetilde{M}$ and $\widetilde{m}(\mathcal{HO}', f, \widetilde{f}) \geq \widetilde{m}$. We then write $o(u, v, \widetilde{f}, \widetilde{M}, \widetilde{m}) = \max(\mathcal{P})$. If $\mathcal{P} = \emptyset$, then $o(u, v, \widetilde{f}, \widetilde{M}, \widetilde{m}) = -\infty$.

$i(u, v, \widetilde{f}, \widetilde{M}, \widetilde{m})$ can be seen as the minimum power than should be called by the sinks of T_v and produced by the sources of T_u . On the contrary, $o(u, v, \widetilde{f}, \widetilde{M}, \widetilde{m})$ is the maximum power that can be produced by the sources of T_u and called by the sinks of T_v . This interpretation leads intuitively to Lemma 8.

Example 5. In the example of Figure 5, we set $\widetilde{m} = 0$, $\widetilde{M} = 1$ and u, v are the two nodes shown in the figure. There are two possible semi-orientations outgoing from (u, v) satisfying the demand constraint: the one of Figure 5b and the same orientation where we reverse the arc going from v to the source above. In the first case, the flow going through (u, v) is 55 and the rounded flow is around 54.991, and, in the second case, the flow going through (u, v) is 25 and the rounded flow is around 24.9976. Consequently, $i(u, v, 54.991, 1, 0) = 55$ and $i(u, v, 24.9976, 1, 0) = 25$.

There is no feasible orientation entering (u, v) with $f = 25$ and $\tilde{f} = 24.9976$ as the capacity constraint is, in that case, not satisfied for u . The maximum value f for which it is feasible is 0, thus, $o(u, v, 24.9976, 1, 0) = 0$. Similarly, $o(u, v, 54.991, 1, 0) = 0$.

As we can see, it is not possible to call a power greater than 0 to u whereas v needs at least 25. In other words, there is no feasible orientation where $\mathcal{O}([u, v]) = (u, v)$.

Lemma 8. Let $[u, v]$ be any edge of T , and $\tilde{M}, \tilde{m} \in \tilde{L}$, there exists a feasible orientation \mathcal{O} where $\tilde{M} \geq \tilde{M}(\mathcal{O})$ and $\tilde{m} \leq \tilde{m}(\mathcal{O})$ if and only if there exists $\tilde{f} \in \tilde{F}$ such that $i(u, v, \tilde{f}, \tilde{M}, \tilde{m}) \leq o(u, v, \tilde{f}, \tilde{M}, \tilde{m})$ or $i(v, u, \tilde{f}, \tilde{M}, \tilde{m}) \leq o(v, u, \tilde{f}, \tilde{M}, \tilde{m})$.

Proof. Assuming such a feasible orientation \mathcal{O} exists with $\mathcal{O}([u, v]) = (u, v)$, $f = F_{\mathcal{O}}(u, v)$ and $\tilde{f} = \tilde{F}_{\mathcal{O}}(u, v)$. Finally, let \mathcal{HO} be the orientation \mathcal{O} restricted to $T_v \cup [u, v]$, and \mathcal{HO}' be \mathcal{O} restricted to $T_u \cup [u, v]$.

Then, first, \mathcal{HO} is an semi-orientation outgoing from (u, v) . For every edge $e \in T_v$, we have $F_{\mathcal{O}}(e) = F_{\mathcal{HO}}(e)$ and $\tilde{F}_{\mathcal{O}}(e) = \tilde{F}_{\mathcal{HO}}(e)$. Consequently \mathcal{HO} is feasible and satisfies $F_{\mathcal{HO}}(u, v) = f$ and $\tilde{F}_{\mathcal{HO}}(u, v) = \tilde{f}$, $\tilde{M} \geq \tilde{M}(\mathcal{O})$ and $\tilde{m} \leq \tilde{m}(\mathcal{O})$. Thus $i(u, v, \tilde{f}, \tilde{M}, \tilde{m}) \leq f$.

Secondly \mathcal{HO}' is an semi-orientation entering (u, v) with f and \tilde{f} . For every edge $e \in T_u$, we have $F_{\mathcal{O}}(e) = F_{(\mathcal{HO}', f)}(e)$ and $\tilde{F}_{\mathcal{O}}(e) = \tilde{F}_{(\mathcal{HO}', f, \tilde{f})}(e)$. Consequently \mathcal{HO}' is feasible and satisfies $\tilde{M} \geq \tilde{M}(\mathcal{O})$ and $\tilde{m} \leq \tilde{m}(\mathcal{O})$. Thus $o(u, v, \tilde{f}, \tilde{M}, \tilde{m}) \geq f$. We then have $o(u, v, \tilde{f}, \tilde{M}, \tilde{m}) \geq i(u, v, \tilde{f}, \tilde{M}, \tilde{m})$.

We now assume that $o(u, v, \tilde{f}, \tilde{M}, \tilde{m}) \geq i(u, v, \tilde{f}, \tilde{M}, \tilde{m})$ for some value $\tilde{f} \in \tilde{F}$. We can first deduce that $i(u, v, \tilde{f}, \tilde{M}, \tilde{m}) \neq +\infty$ and $o(u, v, \tilde{f}, \tilde{M}, \tilde{m}) \neq -\infty$. By definition of $i(u, v, \tilde{f}, \tilde{M}, \tilde{m})$, there exists a feasible semi-orientation \mathcal{HO} outgoing from (u, v) such that $F_{\mathcal{HO}}(u, v) = i(u, v, \tilde{f}, \tilde{M}, \tilde{m})$, $\tilde{F}_{\mathcal{HO}}(u, v) = \tilde{f}$, $\tilde{M} \geq \tilde{M}(\mathcal{HO})$ and $\tilde{m} \leq \tilde{m}(\mathcal{HO})$. By definition of $o(u, v, \tilde{f}, \tilde{M}, \tilde{m})$, there exists a feasible semi-orientation \mathcal{HO}' entering (u, v) with $o(u, v, \tilde{f}, \tilde{M}, \tilde{m})$ and \tilde{f} such that $\tilde{M} \geq \tilde{M}(\mathcal{HO}')$ and $\tilde{m} \leq \tilde{m}(\mathcal{HO}')$. By Lemma 7, $(\mathcal{HO}', i(u, v, \tilde{f}, \tilde{M}, \tilde{m}), \tilde{f})$ is also feasible.

Let $\mathcal{O} = \mathcal{HO} \cup \mathcal{HO}'$. Note that, on the edges of $T_v \cup [u, v]$, the flow $F_{\mathcal{O}}$ and the rounded flow $\tilde{F}_{\mathcal{O}}$ coincide respectively with $F_{\mathcal{HO}}$ and $\tilde{F}_{\mathcal{HO}}$. In addition, on the edges of $T_u \cup [u, v]$, the flow $F_{\mathcal{O}}$ and the rounded flow $\tilde{F}_{\mathcal{O}}$ coincide respectively with $F_{(\mathcal{HO}', i(u, v, \tilde{f}, \tilde{M}, \tilde{m}), \tilde{f})}$ and $\tilde{F}_{(\mathcal{HO}', i(u, v, \tilde{f}, \tilde{M}, \tilde{m}), \tilde{f})}$. Then \mathcal{O} is feasible and $\tilde{M} \geq \tilde{M}(\mathcal{O})$ and $\tilde{m} \leq \tilde{m}(\mathcal{O})$. \square

The end of the section is devoted to explaining how we can compute the auxiliary functions in polynomial time. In order to prevent the technical details from obfuscating the key principles of this proof, the rest of the section is organized as follows.

- In Subsection 5.3.3, Lemma 9 gives the complexity of an algorithm computing the auxiliary functions o and i , given some assumption on the

existence of two functions h_o and h_i that can be computed in polynomial time,

- Those two functions are respectively described in Subsections 5.3.4 (in Lemmas 10 and 11) and 5.3.5 (in Lemmas 12 and 13). In order to prove their complexities, we make another assumption on the existence of another algorithm.
- This algorithm is then described in Subsection 5.3.6 and Lemma 14.

5.3.3 Complexity of the computation of the auxiliary functions

We assume $\widetilde{M} \in \widetilde{L}, \widetilde{m} \in \widetilde{L}, \widetilde{f} \in \widetilde{F}$ and $[u, v] \in T$ are given, and we want to compute $i(u, v, \widetilde{f}, \widetilde{M}, \widetilde{m})$ and $o(u, v, \widetilde{f}, \widetilde{M}, \widetilde{m})$. For readability, we denote hereinafter those values by $i(u, v, \widetilde{f})$ and $o(u, v, \widetilde{f})$ as, in the proofs, the values \widetilde{M} and \widetilde{m} never change.

Lemma 9. *Given a node $u \in T$, we write $\Gamma(u)$ the neighbors of u . We assume that, for every edge $[u, v]$ in T and every \widetilde{f} ,*

- *there exists a function $h_o : \mathbb{N}^{2 \cdot (|\Gamma(u)|-1) \cdot |\widetilde{F}|} \rightarrow \mathbb{N}$ such that*

$$o(u, v, \widetilde{f}) = h_o \left(\left\{ o(w, u, \widetilde{f}'), i(u, w, \widetilde{f}') \mid w \in \Gamma(u) \setminus \{v\}, \widetilde{f}' \in \widetilde{F} \right\} \right)$$

and that can be computed in time $O(n^4 |\widetilde{F}|^4)$

- *there exists a function $h_i : \mathbb{N}^{2 \cdot (|\Gamma(v)|-1) \cdot |\widetilde{F}|} \rightarrow \mathbb{N}$ such that*

$$i(u, v, \widetilde{f}) = h_i \left(\left\{ o(w, v, \widetilde{f}'), i(v, w, \widetilde{f}') \mid w \in \Gamma(v) \setminus \{u\}, \widetilde{f}' \in \widetilde{F} \right\} \right)$$

and that can be computed in time $O(n^4 |\widetilde{F}|^3)$

then we can compute $i(u, v, \widetilde{f})$ and $o(u, v, \widetilde{f})$ for all $[u, v] \in T$ and $\widetilde{f} \in \widetilde{F}$ in time $O(n^5 |\widetilde{F}|^5)$.

Proof. If v is a leaf, the value $\Gamma(v) \setminus \{u\}$ is empty. In that case, $i(u, v, \widetilde{f}) = h_i(\emptyset)$ and can be computed in time $O(n^4 |\widetilde{F}|^3)$ for each value $\widetilde{f} \in \widetilde{F}$. Similarly, if u is a leaf, $o(u, v, \widetilde{f}) = h_o(\emptyset)$ can be computed, for each value $\widetilde{f} \in \widetilde{F}$, in time $O(n^4 |\widetilde{F}|^4)$ (in fact, those values can be computed in constant time, but this does not change the result of the lemma).

We can then compute $i(u, v, \widetilde{f})$ (respectively $o(u, v, \widetilde{f})$) for every edge $[u, v]$ such that the height of the tree T_v (respectively T_u) is 1. By induction on the height of the trees T_v and T_u , we can then compute step by step from the leaves all the values $i(u, v, \widetilde{f})$ and $o(u, v, \widetilde{f})$ for all $[u, v] \in T$ and $\widetilde{f} \in \widetilde{F}$. At each iteration, a new value $o(u, v, \widetilde{f})$ (respectively $i(u, v, \widetilde{f})$) is evaluated in time $O(n^4 |\widetilde{F}|^4)$ (respectively $O(n^4 |\widetilde{F}|^3)$). There are $O((n-1) \cdot |\widetilde{F}|)$ possible values for (u, v) and \widetilde{f} , thus the lemma follows. \square

5.3.4 Complexity of the computation of the function h_o of Lemma 9

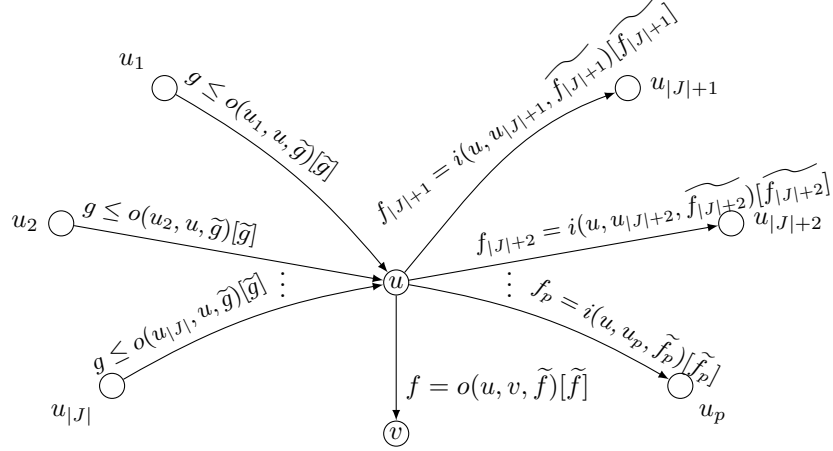


Figure 6: Illustration of Lemma 10, assuming $J = \{1, 2, \dots, |J|\}$. On each arc is written the flow followed by the rounded flow between square brackets.

Let u_1, u_2, \dots, u_p be the p neighbors of u in T_u (every neighbor distinct from v). In this part, we assume $u \in W$ and $Cap(u) = +\infty$. We will describe the formula of $o(u, v, \tilde{f})$ and explain how to adapt it to sources, sinks and switches with finite capacity. Without loss of generality, we also assume that v has the minimum number in \mathcal{N} (the numbering of the nodes of T) among all the neighbors of u .

The following lemma describes the function h_o mentioned in Lemma 9 that, given $o(w, u, \tilde{f}')$ and $i(u, w, \tilde{f}')$ for every neighbor w of u except v and every rounded flow \tilde{f}' , computes $o(u, v, \tilde{f})$.

Lemma 10. *Let $J \subseteq \llbracket 1; p \rrbracket$, with $J \neq \emptyset$, and $\tilde{f}_j \in \tilde{F}$ for $j \notin J$.*

We set $\tilde{g} = \oplus((\tilde{f}, \tilde{f}_j)_{j \notin J}, |J|)$ where $\tilde{f}_j = 0$ if $j \in J$, and

$$f(J, (\tilde{f}_j)_{j \notin J}) = \begin{cases} |J| \cdot \min_{j \in J} o(u_j, u, \tilde{g}) - \sum_{j \notin J} i(u, u_j, \tilde{f}_j) & \text{if this value is non-negative} \\ -\infty & \text{otherwise} \end{cases}$$

then

$$o(u, v, \tilde{f}) = \max_{\substack{J \subseteq \llbracket 1; p \rrbracket \\ J \neq \emptyset}} \max_{\substack{\tilde{f}_j \in \tilde{F} \\ j \notin J}} f(J, (\tilde{f}_j)_{j \notin J})$$

Example 6. *We use the instance of Figure 7 to illustrate the Lemma. We assume that $\varepsilon = 1$ to simplify the calculations of rounding values. We also assume that $\tilde{m} = 0.2$ and $\tilde{M} = 0.5$. We would like to compute $o(u, v, 4)$ that is the greatest power that can be sent from u to v if the rounded flow between*

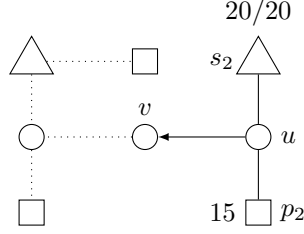


Figure 7: Illustration of Example 6.

them is 4. Intuitively, this value is 5 as the source s_2 can produce at most 20 and must send 15 to p_2 .

Now lets compute the formula of Lemma 10. We get:

$$o(u, v, 4) = \max \begin{cases} o(s_2, u, \tilde{g}) - i(u, p_2, \tilde{f}_2) & \text{where } \tilde{g} = \oplus((4, \tilde{f}_2), 1) \\ 2 \cdot \min(o(s_2, u, \tilde{g}), o(p_2, u, \tilde{g})) & \text{where } \tilde{g} = \oplus((4), 2) \\ o(p_2, u, \tilde{g}) - i(u, s_2, \tilde{f}_1) & \text{where } \tilde{g} = \oplus((4, \tilde{f}_1), 1) \end{cases}$$

The value $o(p_2, u, \tilde{g})$ is necessarily $-\infty$. Indeed, there is not feasible semi-orientation where the arc (p_2, u) is directed toward u as the demand constraint would not be satisfied for p_2 . This leave us with

$$o(u, v, 4) = o(s_2, u, \tilde{g}) - i(u, p_2, \tilde{f}_2) \text{ where } \tilde{g} = \oplus((4, \tilde{f}_2), 1)$$

The value $i(u, p_2, \tilde{f}_2)$ is $+\infty$ if \tilde{f}_2 is not the rounded value of $\text{Pow}(p_2)$, that is $a(15) = 8$ if $\varepsilon = 1$, and $i(u, p_2, 8)$ is $\text{Pow}(p_2) = 15$. The value \tilde{g} is then $\oplus((4, 8), 1) = a(8 + a(4)) = a(12) = 8$. We search for $o(s_2, u, 8)$. A rounded flow for s_2 of 8 is valid as its rounded load would be $8/20$, which is between \tilde{m} and \tilde{M} . As s_2 cannot produce more than 20, $o(s_2, u, 8)$ is 20.

Thus we get $o(u, v, 4) = 20 - 15 = 5$ as predicted.

Remark (Intuition). Figure 6 illustrates Lemma 10. Given a subset J and rounded flows $(\tilde{f}_j)_{j \notin J}$, the value $f(J, (\tilde{f}_j)_{j \notin J})$ corresponds to the maximum flow that can go through u to v if, for every $j \notin J$, the rounded flow in (u, u_j) is \tilde{f}_j and if, for every $j \in J$, all the edges (u_j, u) are directed toward u . In those arcs, assuming the rounded flow in (u, v) is f , the rounded flow is necessarily \tilde{g} .

$|J| \cdot \min_{j \in J} o(u_j, u, \tilde{g})$ is the maximum value that can be equitably distributed over all the input arcs of u . And $\sum_{j \notin J} i(u, u_j, \tilde{f}_j)$ is the value requested by the neighbors directed from u other than v , consequently, the maximum value that can go from u to v is the difference.

Note that J must be not empty in order to direct at least one arc toward u to satisfy the demand constraint.

Proof. Let $ma = \max_{\substack{J \subseteq [1;p] \\ J \neq \emptyset}} \max_{\substack{f_j \in \tilde{F} \\ j \notin J}} f(J, (f_j)_{j \notin J})$.

We assume that $o(u, v, \tilde{f}) \neq -\infty$. Then, by definition of $o(u, v, \tilde{f})$, there exists a feasible semi-orientation \mathcal{HO} entering (u, v) with $f = o(u, v, \tilde{f})$ and \tilde{f} such that $\tilde{M}((\mathcal{HO}, f, \tilde{f})) \leq \tilde{M}$ and $\tilde{m}((\mathcal{HO}, f, \tilde{f})) \geq \tilde{m}$. Let J be the set of indexes $j \in [1; p]$ such that $\mathcal{HO}([u, u_j]) = (u_j, u)$. The set J is not empty, otherwise the demand constraint is not satisfied for u . We write $g = F_{(\mathcal{HO}, f, \tilde{f})}(u_j, u)$, $\tilde{g} = \tilde{F}_{(\mathcal{HO}, f, \tilde{f})}(u_j, u)$, and $f_j = F_{(\mathcal{HO}, f, \tilde{f})}(u, u_j)$, $\tilde{f}_j = \tilde{F}_{(\mathcal{HO}, f, \tilde{f})}(u, u_j)$ for $j \notin J$. By definition of the rounded flow, and because v has the minimum number in \mathcal{N} among all the neighbors of u , $\oplus((\tilde{f}, \tilde{f}_j)_{j \notin J}, |J|) = \tilde{g}$. Finally, by definition of $F_{(\mathcal{HO}, f, \tilde{f})}$, $f = |J| \cdot g - \sum_{j \notin J} f_j$.

Let T_{u_j} be the subtree of T_u containing u_j in the forest $T_u \setminus [u, u_j]$, let $j \in J$, and let \mathcal{HO}_j be the semi-orientation entering (u_j, u) that equals \mathcal{HO} restricted to T_{u_j} then \mathcal{HO}_j is feasible with g and \tilde{g} . In addition, $\tilde{M}(\mathcal{HO}_j, g, \tilde{g}) \leq \tilde{M}(\mathcal{HO}, f, \tilde{f}) \leq \tilde{M}$ and $\tilde{m}(\mathcal{HO}_j, g, \tilde{g}) \geq \tilde{m}(\mathcal{HO}, f, \tilde{f}) \geq \tilde{m}$ thus $F_{(\mathcal{HO}, f, \tilde{f})}(u_j, u) \leq o(u_j, u, \tilde{g})$. As, for every edge $[u_j, u]$, $F_{(\mathcal{HO}, f, \tilde{f})}(u_j, u) = g$, we have, $g \leq \min_{j \in J} o(u_j, u, \tilde{g})$. We can similarly show that, for $j \notin J$, $f_j \geq i(u, u_j, \tilde{f}_j)$. Consequently, $f = |J| \cdot g - \sum_{j \notin J} f_j \leq |J| \cdot \min_{j \in J} o(u_j, u, \tilde{g}) - \sum_{j \notin J} i(u, u_j, \tilde{f}_j) \leq ma$. Thus

$o(u, v, \tilde{f}) \leq ma$. As $o(u, v, \tilde{f}) \neq -\infty$, we can also deduce that $ma \neq -\infty$.

We now assume that $ma \neq -\infty$. Let $J \neq \emptyset$ and \tilde{f}_j for $j \notin J$, such that $f(J, (\tilde{f}_j)_{j \notin J}) = ma = |J| \cdot \min_{j \in J} o(u_j, u, \tilde{g}) - \sum_{j \notin J} i(u, u_j, \tilde{f}_j)$ with the value \tilde{g}

being $\oplus((\tilde{f}, \tilde{f}_j)_{j \notin J}, |J|)$ (as $J \neq \emptyset$, this equality is defined). We build an semi-orientation \mathcal{HO} entering (u, v) satisfying $ma \leq o(u, v, \tilde{f})$. To do so, we set $\mathcal{HO}([u_j, u]) = (u_j, u)$ if $j \in J$ and (u, u_j) otherwise. As $ma \neq -\infty$, we know that, if $j \in J$, then $o(u_j, u, \tilde{g}) \neq -\infty$ and, if $j \notin J$, $i(u, u_j, \tilde{f}_j) \neq +\infty$. Thus, for every $j \in J$, there exists a feasible semi-orientation \mathcal{HO}_j entering (u_j, u) with $o(u_j, u, \tilde{g})$ and \tilde{g} such that $\tilde{M}(\mathcal{HO}_j, o(u_j, u, \tilde{g}), \tilde{g}) \leq \tilde{M}$ and $\tilde{m}(\mathcal{HO}_j, o(u_j, u, \tilde{g}), \tilde{g}) \geq \tilde{m}$. Similarly, for every $j \notin J$, there exists a feasible halt-orientation \mathcal{HO}_j outgoing from (u, u_j) such that $\tilde{M}(\mathcal{HO}_j) \leq \tilde{M}$, $\tilde{m}(\mathcal{HO}_j) \geq \tilde{m}$, $F_{\mathcal{HO}_j}(u, u_j) = i(u, u_j, \tilde{f}_j)$ and $\tilde{F}_{\mathcal{HO}_j}(u, u_j) = \tilde{f}_j$. Let then \mathcal{HO} be the union of all those semi-orientations plus $\mathcal{HO}([u, v]) = (u, v)$.

If $(\mathcal{HO}, ma, \tilde{f})$ is feasible, then, because we have $\tilde{M}(\mathcal{HO}, ma, \tilde{f}) \leq \tilde{M}$ and $\tilde{m}(\mathcal{HO}, ma, \tilde{f}) \geq \tilde{m}$, then $ma \leq o(u, v, \tilde{f})$ and $o(u, v, \tilde{f}) \neq -\infty$. First, note that, because $J \neq \emptyset$, the node u has an entering arc and the demand constraint is then satisfied for u . It is satisfied for every other node as all the semi-orientations \mathcal{HO}_j are feasible. Secondly, for all $j \notin J$, $F_{(\mathcal{HO}, ma, \tilde{f})}(u, u_j) = F_{\mathcal{HO}_j}(u, u_j) = i(u, u_j, \tilde{f}_j)$ and $\tilde{F}_{(\mathcal{HO}, ma, \tilde{f})}(u, u_j) = \tilde{f}_j$. For all $j \in J$, the flow $F_{(\mathcal{HO}, ma, \tilde{f})}(u_j, u) = \frac{1}{|J|} \cdot (ma + \sum_{j \notin J} F_{(\mathcal{HO}, ma, \tilde{f})}(u, u_j)) = \min_{j \in J} o(u_j, u, \tilde{g})$

and the rounded flow $\tilde{F}_{(\mathcal{H}\mathcal{O}, f, \tilde{f})}(u_j, u) = \oplus((\tilde{f}, \tilde{f}_j)_{j \notin J}, |J|) = \tilde{g}$ (again because v has the minimum number in \mathcal{N} among all the neighbors of u). Finally, for all $j \in J$, $(\mathcal{H}\mathcal{O}_j, o(u_j, u, \tilde{g}), \tilde{g})$ is feasible. According to Lemma 7, then $(\mathcal{H}\mathcal{O}_j, \min_{j \in J} o(u_j, u, \tilde{g}), \tilde{g})$ is also feasible. Consequently $(\mathcal{H}\mathcal{O}, ma, \tilde{f})$ is feasible. \square

The following lemma proves that we can compute the formula given in Lemma 10 in polynomial time. This lemma makes a hypothesis that is proven in Lemma 14.

Lemma 11.

- If we have access to $o(u_j, u, \tilde{f})$ and $i(u, u_j, \tilde{f})$ in $O(1)$ for every $j \in \llbracket 1; p \rrbracket$ and $\tilde{f} \in \tilde{F}$,
- and if, given $d \in \llbracket 1; p \rrbracket$, $k \in \llbracket 1; p - d + 1 \rrbracket$ and $\tilde{g} \in \tilde{F}$, we can compute

$$\min_{\substack{J \subseteq \llbracket 1; p \rrbracket \\ |J|=d \\ \min(J)=k}} \min_{\substack{\tilde{f}_j \in \tilde{F} \\ j \notin J}} \sum_{j \notin J} i(u, u_j, \tilde{f}_j)$$

in time $O(n^2 |\tilde{F}|^3)$,

then we can compute $o(u, v, \tilde{f})$ in $O(n^4 |\tilde{F}|^4)$.

Proof. By Lemma 10,

$$\begin{aligned} o(u, v, \tilde{f}) &= \max_{\substack{J \subseteq \llbracket 1; p \rrbracket \\ J \neq \emptyset}} \max_{\substack{\tilde{f}_j \in \tilde{F} \\ j \notin J}} f(J, (f_j)_{j \notin J}) \\ o(u, v, \tilde{f}) &= \max_{d=1}^p \max_{\substack{J \subseteq \llbracket 1; p \rrbracket \\ |J|=d}} \max_{\substack{\tilde{f}_j \in \tilde{F} \\ j \notin J}} f(J, (f_j)_{j \notin J}) \\ o(u, v, \tilde{f}) &= \max_{d=1}^p \max_{k=1}^{p-d+1} \max_{\substack{J \subseteq \llbracket 1; p \rrbracket \\ |J|=d \\ \min(J)=k}} \max_{\substack{\tilde{f}_j \in \tilde{F} \\ j \notin J}} f(J, (f_j)_{j \notin J}) \\ o(u, v, \tilde{f}) &= \max_{d=1}^p \max_{k=1}^{p-d+1} \max_{\tilde{g} \in \tilde{F}} \max_{\substack{J \subseteq \llbracket 1; p \rrbracket \\ |J|=d \\ \min(J)=k}} \max_{\substack{\tilde{f}_j \in \tilde{F} \\ j \notin J}} f(J, (f_j)_{j \notin J}) \end{aligned}$$

If, for every $\tilde{g} \in \tilde{F}$, we write $o_1(\tilde{g}), o_2(\tilde{g}), \dots, o_p(\tilde{g})$ the p values $o(u_j, u, \tilde{g})$ sorted in ascending order

$$o(u, v, \tilde{f}) = \max_{d=1}^p \max_{k=1}^{p-d+1} \max_{\tilde{g} \in \tilde{F}} \max_{\substack{J \subseteq \llbracket 1; p \rrbracket \\ |J|=d \\ \min(J)=k}} \max_{\substack{\tilde{f}_j \in \tilde{F} \\ j \notin J}} d \cdot o_k(\tilde{g}) - \sum_{j \notin J} i(u, u_j, \tilde{f}_j)$$

Given d, k and \tilde{g} , we can compute $\max_{\substack{J \subseteq \llbracket 1; p \rrbracket \\ |J|=d \\ \min(J)=k}} \max_{\substack{\tilde{f}_j \in \tilde{F} \\ j \notin J}} d \cdot o_k(\tilde{g}) - \sum_{j \notin J} i(u, u_j, \tilde{f}_j)$

to solve the problem, and this can be computed in time $O(n^2 |\tilde{F}|^3)$ by hypothesis. If the obtained value is negative, we return $-\infty$.

As a conclusion $o(u, v, f)$ can be computed in time $O(n^4 |\tilde{F}|^4)$ by enumerating all the possible values of d, k and \tilde{g} , and by returning the maximum result obtained with all the triplets. \square

Remark. We can adapt the formula of Lemma 10 to other cases.

- If the numbering of v in \mathcal{N} is not the smallest among the numbers of all the neighbors of u , the list $(\tilde{f}, \tilde{f}_j)_{j \notin J}$ should be sorted in the formula of \tilde{g} .
- If u is a switch with a finite capacity, we must replace $|J| \cdot \min_{j \in J} o(u_j, u, \tilde{g})$ by $\min(\text{Cap}(u), |J| \cdot \min_{j \in J} o(u_j, u, \tilde{g}))$ in the formula of the function $f(J, (\tilde{f}_j)_{j \notin J})$.
- If u is a source, then we can have $J = \emptyset$ and we must replace $|J| \cdot \min_{j \in J} o(u_j, u, \tilde{g})$ by $\min((|J| + 1) \cdot \text{Prod}(u), (|J| + 1) \cdot \min_{j \in J} o(u_j, u, \tilde{g}))$; in addition, $\oplus((\tilde{f}, \tilde{f}_j)_{j \notin J}, |J|)$ must be replaced by $\oplus((\tilde{f}, \tilde{f}_j)_{j \notin J}, |J| + 1)$ in the formula of \tilde{g} ; and finally, we have to replace $\max_{\tilde{g} \in \tilde{F}}$ by $\max_{\tilde{g} \in \tilde{F}, \tilde{m} \leq \frac{\tilde{g}}{\text{Prod}(u)} \leq \tilde{M}}$ in the formula of $o(u, v, \tilde{f})$.
- If u is a sink, we must replace the formula of \tilde{g} by $\oplus((\text{Pow}(u), \tilde{f}, \tilde{f}_j)_{j \notin J}, |J|)$ and replace $-\sum_{j \notin J} i(u, u_j, \tilde{f}_j)$ by $-\sum_{j \notin J} i(u, u_j, \tilde{f}_j) - \text{Pow}(u)$ in the formula of $f(J, (\tilde{f}_j)_{j \notin J})$.

5.3.5 Complexity of the computation of the function h_i of Lemma 9

Let v_1, v_2, \dots, v_p be the p neighbors of v in T_v (every neighbor distinct from u). In this part, we assume $v \in W$ and $\text{Cap}(v) = +\infty$. We will describe the formula of $i(u, v, \tilde{f})$ and explain how to adapt it to sources, sinks and switches with finite capacity.

The following lemma describes the function h_i mentioned in Lemma 9 that, given $o(w, v, \tilde{f}')$ and $i(v, w, \tilde{f}')$ for every neighbor w of v except u and every rounded flow \tilde{f}' , computes $i(u, v, \tilde{f})$.

Lemma 12. Let $J \subseteq \llbracket 1; p \rrbracket$, and $\tilde{f}_j \in \tilde{F}$ for $j \notin J$ such that $\tilde{f} = \oplus((\tilde{f}_j)_{j \notin J}, |J| + 1)$.

We set

$$f(J, (\tilde{f}_j)_{j \notin J}) = \begin{cases} \frac{1}{|J|+1} \sum_{j \notin J} i(v, v_j, \tilde{f}_j) & \text{if this value is lower than } \min_{j \in J} o(v_j, v, \tilde{f}) \\ +\infty & \text{otherwise} \end{cases}$$

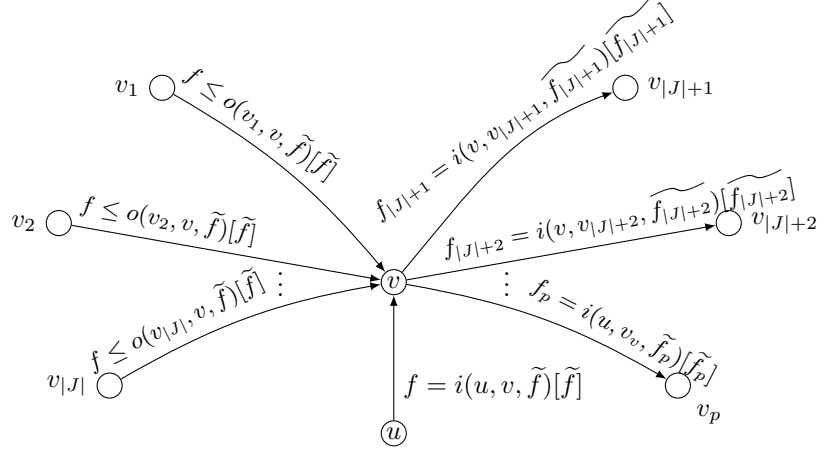


Figure 8: Illustration of Lemma 12. On each arc is written the flow followed by the rounded flow between square brackets.

If $J = \emptyset$, we set $\min_{j \in J} o(v_j, v, \tilde{f}) = +\infty$.

Then

$$i(u, v, \tilde{f}) = \min_{J \subseteq [1:p]} \min_{\substack{\tilde{f}_j \in \tilde{F} \\ j \notin J \\ \oplus((\tilde{f}_j)_{j \notin J, |J|+1}) = \tilde{f}}} f(J, (\tilde{f}_j)_{j \notin J})$$

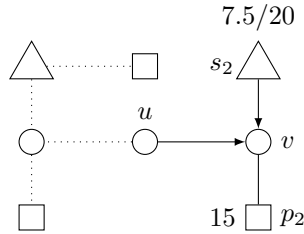


Figure 9: Illustration of Example 7.

Example 7. We use the instance of Figure 9 to illustrate the Lemma. We assume that $\varepsilon = 1$ to simplify the calculations of rounding values. We also assume that $\tilde{m} = 0.2$ and $\tilde{M} = 0.5$. We would like to compute $i(u, v, 4)$ that is the lower power that can be called by v to u if the rounded flow between them is 4. Intuitively, this value is 7.5 as the sink p_2 calls 15 to v , which can either equally calls 7.5 to each of u and s_2 or calls 15 to u alone.

Now lets compute the formula of Lemma 12. We get:

$$i(u, v, 4) = \min \begin{cases} \frac{1}{2} \cdot i(v, p_2, \tilde{f}_1) & \text{if it is lower than } o(s_2, v, 4) \text{ where } 4 = \oplus((\tilde{f}_1), 2) \\ \frac{1}{2} \cdot i(v, s_2, \tilde{f}_2) & \text{if it is lower than } o(p_2, v, 4) \text{ where } 4 = \oplus((\tilde{f}_2), 2) \\ i(v, p_2, \tilde{f}_1) + i(v, s_2, \tilde{f}_2) & \text{if it is lower than } +\infty \text{ where } 4 = \oplus((\tilde{f}_1, \tilde{f}_2), 1) \\ \frac{1}{3} \cdot 0 & \text{if it is lower than } \min(o(p_2, v, 4), o(s_2, v, 4)) \end{cases}$$

The value $o(p_2, v, 4)$ is necessarily $-\infty$. Indeed, there is not feasible semi-orientation where the arc (p_2, v) is directed toward v as the demand constraint would not be satisfied for p_2 . This leaves us with the first and third case of the formula.

The value $i(v, s_2, \tilde{f}_2)$ is $+\infty$ if $\tilde{f}_2 \neq 0$, indeed, no power is called by s_2 if (v, s_2) is directed toward s_2 . However, $i(v, s_2, 0)$ is also infinite as the source would have a rounded load of 0 and this value is not between \tilde{m} and \tilde{M} . Consequently,

$$i(u, v, 4) = \frac{1}{2} \cdot i(v, p_2, \tilde{f}_1) \text{ if it is lower than } o(s_2, v, 4) \text{ where } 4 = \oplus((\tilde{f}_1), 2)$$

The value $i(v, p_2, \tilde{f}_1)$ is $+\infty$ if \tilde{f}_1 is not the rounded value of $\text{Pow}(p_2)$, that is $a(15) = 8$ if $\varepsilon = 1$, and $i(v, p_2, 8) = \text{Pow}(p_2) = 15$. We have $\oplus((8), 2) = 4$. The rounded load of s_2 is $4/20$ if the rounded flow of s_2 is 4, which is between \tilde{m} and \tilde{M} . Thus $o(s_2, v, 4)$ is 20. Finally, $\frac{1}{2}i(v, p_2, 8) \leq o(s_2, v, 4)$.

Therefore, we get $i(u, v, 4) = \frac{1}{2} \cdot i(v, p_2, 8) = 7.5$ as predicted.

Remark (Intuition). Figure 8 illustrates Lemma 12. Given a subset J , and rounded flows $(f_j)_{j \notin J}$, the value $f(J, (f_j)_{j \notin J})$ is the minimum flow requested to v if, for every $j \notin J$, the rounded flow in (v, v_j) is f_j and if, for every $j \in J$, all the edges (v_j, v) are directed toward v . Assuming the rounded flow in (u, v) is \tilde{f} , as it is equal in all the input arcs of v , the rounded flow is then also \tilde{f} in (v_j, v) , and that value should equal $\oplus((f_j)_{j \notin J}, |J| + 1)$.

As the flow is also equal in all the input arcs, the minimum flow requested to v cannot be more than $\min_{j \in J} o(v_j, v, \tilde{f})$.

Note that, as v is a switch, it looks tempting to set $J = \llbracket 1; n \rrbracket$ so that the flow in (u, v) is 0 (and thus minimum). However, three problems may occur: first, it does not work if $\tilde{f} \neq 0$, secondly, this solution may not satisfy the choice of \tilde{m} and \tilde{M} , and thirdly, we may have $o(v_j, v, \tilde{f}) = -\infty$.

$$\text{Proof. Let } mi = \min_{J \subseteq \llbracket 1; p \rrbracket} \min_{\substack{\tilde{f}_j \in \tilde{F} \\ j \notin J \\ \oplus((\tilde{f}_j)_{j \notin J}, |J|+1) = \tilde{f}}} f(J, (\tilde{f}_j)_{j \notin J}).$$

If $i(u, v, \tilde{f}) \neq +\infty$, then there exists a feasible semi-orientation \mathcal{HO} outgoing from (u, v) such that $\tilde{M}(\mathcal{HO}) \leq \tilde{M}$, $\tilde{m}(\mathcal{HO}) \geq \tilde{m}$, $F_{\mathcal{HO}}(u, v) = i(u, v, \tilde{f})$ and $\tilde{F}_{\mathcal{HO}}(u, v) = \tilde{f}$. We hereinafter denote by f the value $i(u, v, \tilde{f})$. Let J be the set of indexes $j \in \llbracket 1; p \rrbracket$ such that $\mathcal{HO}([v, v_j]) = (v_j, v)$. We write $\tilde{f}_j =$

$\tilde{F}_{\mathcal{H}\mathcal{O}}(v, v_j)$ for $j \notin J$. By definition of the rounded flow, $\oplus((\tilde{f}_j)_{j \notin J}, |J| + 1) = \tilde{f} = \tilde{F}_{\mathcal{H}\mathcal{O}}(v_j, v)$ for $j \in J$. Finally, by definition of the flow, for $j \in J$, $\tilde{f} = \tilde{F}_{\mathcal{H}\mathcal{O}}(v_j, v) = 1/(|J| + 1) \sum_{j \notin J} f_j$.

Let T_{v_j} be the subtree of T_v containing v_j in the forest $T_v \setminus [v, v_j]$, let $j \in J$, and let $\mathcal{H}\mathcal{O}_j$ be the semi-orientation entering (v_j, v) that equals $\mathcal{H}\mathcal{O}$ restricted to T_{v_j} then $\mathcal{H}\mathcal{O}_j$ with f and \tilde{f} is feasible. In addition, $\tilde{M}(\mathcal{H}\mathcal{O}_j, f, \tilde{f}) \leq \tilde{M}(\mathcal{H}\mathcal{O}) \leq \tilde{M}$ and $\tilde{m}(\mathcal{H}\mathcal{O}_j, f, \tilde{f}) \geq \tilde{m}(\mathcal{H}\mathcal{O}) \geq \tilde{m}$ thus $f \leq o(v_j, v, \tilde{f})$, and consequently $f \leq \min_{j \in J} o(v_j, v, \tilde{g})$. We can similarly show that, for $j \notin J$, $f_j \geq i(v, v_j, \tilde{f}_j)$. Finally $f = 1/(|J| + 1) \sum_{j \notin J} f_j$, consequently, $1/(|J| + 1) \sum_{j \notin J} i(v, v_j, \tilde{f}_j) \leq \min_{j \in J} o(u_j, u, \tilde{f})$. Therefore $f(J, (\tilde{f}_j)_{j \notin J})$ is not infinite and equals f ; and then $mi \leq f(J, (\tilde{f}_j)_{j \notin J}) \leq f$. As $f = i(u, v, \tilde{f})$, we conclude that $mi \leq i(u, v, \tilde{f})$. We can also deduce that $mi \neq +\infty$.

We now assume $mi \neq +\infty$. Let J and \tilde{f}_j , for $j \notin J$, such that $\oplus((\tilde{f}_j)_{j \notin J}, |J| + 1) = \tilde{f}$ and $f(J, (\tilde{f}_j)_{j \notin J}) = mi = 1/(|J| + 1) \sum_{j \notin J} i(v, v_j, \tilde{f}_j)$. We build an semi-orientation $\mathcal{H}\mathcal{O}$ outgoing from (u, v) satisfying $mi \geq i(u, v, \tilde{f})$. To do so, we set $\mathcal{H}\mathcal{O}([v_j, v]) = (v_j, v)$ if $j \in J$ and (v, v_j) otherwise. As $mi \neq +\infty$ then $f(J, (\tilde{f}_j)_{j \notin J}) \leq \min_{j \in J} o(v_j, v, \tilde{f})$. Thus if $j \in J$, $o(v_j, v, \tilde{f}) \neq -\infty$ and,

if $j \notin J$, $i(v, v_j, \tilde{f}_j) \neq +\infty$. Consequently, there exists, for every $j \in J$, a feasible semi-orientation $\mathcal{H}\mathcal{O}_j$ entering (v_j, v) with $o(v_j, v, \tilde{f})$ and \tilde{f} such that $\tilde{M}(\mathcal{H}\mathcal{O}_j, o(v_j, v, \tilde{f}), \tilde{f}) \leq \tilde{M}$ and $\tilde{m}(\mathcal{H}\mathcal{O}_j, o(v_j, v, \tilde{f}), \tilde{f}) \geq \tilde{m}$. Similarly, for every $j \notin J$, there exists an semi-orientation $\mathcal{H}\mathcal{O}_j$ outgoing from (v, v_j) such that $\tilde{M}(\mathcal{H}\mathcal{O}_j) \leq \tilde{M}$, $\tilde{m}(\mathcal{H}\mathcal{O}_j) \geq \tilde{m}$, $F_{\mathcal{H}\mathcal{O}_j}(v, v_j) = i(v, v_j, \tilde{f}_j)$ and $\tilde{F}_{\mathcal{H}\mathcal{O}_j}(v, v_j) = \tilde{f}_j$. Let $\mathcal{H}\mathcal{O}$ be the union of all those semi-orientations plus $\mathcal{H}\mathcal{O}([u, v]) = (u, v)$.

If $\mathcal{H}\mathcal{O}$ is feasible, $F_{\mathcal{H}\mathcal{O}}(u, v) = mi$ and $\tilde{F}_{\mathcal{H}\mathcal{O}}(u, v) = \tilde{f}$, then, because we have $\tilde{M}(\mathcal{H}\mathcal{O}, f, \tilde{f}) \leq \tilde{M}$ and $\tilde{m}(\mathcal{H}\mathcal{O}, f, \tilde{f}) \geq \tilde{m}$, then $mi \geq i(u, v, \tilde{f})$ and then $i(u, v, \tilde{f}) \neq +\infty$. Note firstly that the demand constraint is satisfied for u as it has an input arc and it is satisfied for all the other nodes as all the semi-orientations are feasible. For all $j \notin J$, $F_{\mathcal{H}\mathcal{O}}(v, v_j) = F_{\mathcal{H}\mathcal{O}_j}(v, v_j) = i(v, v_j, \tilde{f}_j)$ and $\tilde{F}_{\mathcal{H}\mathcal{O}}(v, v_j) = \tilde{F}_{\mathcal{H}\mathcal{O}_j}(v, v_j) = \tilde{f}_j$. For all $j \in J$, $F_{\mathcal{H}\mathcal{O}}(u, v) = F_{\mathcal{H}\mathcal{O}}(v_j, v) = 1/(|J| + 1) \cdot \sum_{j \notin J} F_{\mathcal{H}\mathcal{O}}(v, v_j) = mi$. In addition, we made the assumption that $\oplus((\tilde{f}_j)_{j \notin J}, |J| + 1) = \tilde{f}$, hence $\tilde{F}_{\mathcal{H}\mathcal{O}}(u, v) = \tilde{F}_{\mathcal{H}\mathcal{O}}(v_j, v) = \tilde{f}$. Finally, for every $j \in J$, $(\mathcal{H}\mathcal{O}_j, o(v_j, v, \tilde{f}), \tilde{f})$ is feasible. As $mi \leq \min_{j \in J} o(v_j, v, \tilde{f})$ then, by Lemma 7, the triplet $(\mathcal{H}\mathcal{O}_j, F_{\mathcal{H}\mathcal{O}}(v_j, v), \tilde{f})$ is also feasible. As a consequence $\mathcal{H}\mathcal{O}$ is feasible. \square

The following lemma proves we can compute the formula given in Lemma 12 in polynomial time. This lemma makes a hypothesis that is proven in Lemma 14.

Lemma 13.

- If we have access to $o(v_j, v, \tilde{f})$ and $i(v, v_j, \tilde{f})$ in $O(1)$ for every $j \in [1; p]$ and $\tilde{f} \in \tilde{F}$,

- and if, given $d \in \llbracket 0; p \rrbracket$ and $k \in \llbracket 1; p-d+1 \rrbracket$, we can compute

$$\min_{\substack{J \subseteq \llbracket 1; p \rrbracket \\ |J|=d \\ \min(J)=k}} \min_{\substack{\tilde{f}_j \in \tilde{F} \\ j \notin J}} \sum_{j \notin J} i(v, v_j, \tilde{f}_j)$$

in $O(n^2|\tilde{F}|^3)$ (where $\min(\emptyset)$ is arbitrarily set to $p+1$ so that the formula is defined for $d=0$)

then we can compute $i(u, v, \tilde{f})$ in $O(n^4|\tilde{F}|^3)$.

Proof. By Lemma 12

$$i(u, v, \tilde{f}) = \min_{J \subseteq \llbracket 1; p \rrbracket} \min_{\substack{\tilde{f}_j \in \tilde{F} \\ j \notin J \\ \oplus((\tilde{f}_j)_{j \notin J, |J|+1}) = \tilde{f}}} f(J, (\tilde{f}_j)_{j \notin J})$$

$$i(u, v, \tilde{f}) = \min_{d=0}^p \min_{\substack{J \subseteq \llbracket 1; p \rrbracket \\ |J|=d}} \min_{\substack{\tilde{f}_j \in \tilde{F} \\ j \notin J \\ \oplus((\tilde{f}_j)_{j \notin J, d+1}) = \tilde{f}}} f(J, (\tilde{f}_j)_{j \notin J})$$

$$i(u, v, \tilde{f}) = \min_{d=0}^p \min_{k=1}^{p-d+1} \min_{\substack{J \subseteq \llbracket 1; p \rrbracket \\ |J|=d \\ \min(J)=k}} \min_{\substack{\tilde{f}_j \in \tilde{F} \\ j \notin J \\ \oplus((\tilde{f}_j)_{j \notin J, d+1}) = \tilde{f}}} f(J, (\tilde{f}_j)_{j \notin J})$$

If we write $o_1(\tilde{f}), o_2(\tilde{f}), \dots, o_p(\tilde{f})$ the p values $o(v_j, v, \tilde{f})$ sorted in ascending order

$$i(u, v, \tilde{f}) = \min_{d=0}^p \min_{k=1}^{p-d+1} \min_{\substack{J \subseteq \llbracket 1; p \rrbracket \\ |J|=d \\ \min(J)=k}} \min_{\substack{\tilde{f}_j \in \tilde{F} \\ j \notin J \\ \oplus((\tilde{f}_j)_{j \notin J, d+1}) = \tilde{f}}} \begin{cases} \frac{1}{d+1} \sum_{j \notin J} i(v, v_j, \tilde{f}_j) \\ \text{if this value is lower than } o_k(\tilde{f}) \\ +\infty \text{ otherwise} \end{cases}$$

Given d and k , we can compute $\min_{\substack{J \subseteq \llbracket 1; p \rrbracket \\ |J|=d \\ \min(J)=k}} \min_{\substack{\tilde{f}_j \in \tilde{F} \\ j \notin J \\ \oplus((\tilde{f}_j)_{j \notin J, d+1}) = \tilde{f}}} \sum_{j \notin J} i(v, v_j, \tilde{f}_j)$ to

solve the problem, and this can be computed in time $O(n^2|\tilde{F}|^3)$ by hypothesis. We then compare the returned value to $o_k(\tilde{f})$, if it is lower, we return it, otherwise we return $+\infty$.

As a conclusion $i(u, v, \tilde{f})$ can be computed in $O(n^4|\tilde{F}|^3)$ by enumerating all the possible values of d and k , and by returning the minimum result obtained with all the couples. \square

Remark. We can adapt the formula of Lemma 12 to other cases:

- if v is a switch with a finite capacity, we must replace $\min_{j \in J} o(v_j, v, \tilde{g})$ by $\min(\text{Cap}(v), \min_{j \in J} o(v_j, v, \tilde{g}))$ in the formula of the function $f(J, (\tilde{f}_j)_{j \notin J})$.
- if v is a source, we must replace $\frac{1}{|J|+1} \sum_{j \notin J} i(v, v_j, \tilde{f}_j)$ if this value is lower than $\min_{j \in J} o(v_j, v, \tilde{f})$ by $\frac{1}{|J|+2} \sum_{j \notin J} i(v, v_j, \tilde{f}_j)$ if this value is lower than $\min(\text{Prod}(v), \min_{j \in J} o(v_j, v, \tilde{f}))$; in addition, $\oplus((\tilde{f}_j)_{j \notin J}, |J| + 1) = \tilde{f}$ must be replaced by $\oplus((\tilde{f}_j)_{j \notin J}, |J| + 2) = \tilde{f}$ and, finally, we must set $i(u, v, \tilde{f})$ to $+\infty$ if we do not check the inequalities $\tilde{m} \leq \tilde{f} / \text{Prod}(v) \leq \tilde{M}$
- if v is a sink, we must replace the formula $\oplus((\tilde{f}_j)_{j \notin J}, |J| + 1) = \tilde{f}$ by $\oplus((\text{Pow}(v), \tilde{f}_j)_{j \notin J}, |J| + 1) = \tilde{f}$; and replace $\sum_{j \notin J} i(v, v_j, \tilde{f}_j)$ by $\sum_{j \notin J} i(v, v_j, \tilde{f}_j) + \text{Pow}(v)$ in the formula of $f(J, (\tilde{f}_j)_{j \notin J})$.

5.3.6 Hypothesis of Lemmas 11 and 13

This final part proves the hypothesis made in the two lemmas. Let $\tilde{f}, \tilde{g} \in \tilde{F}$, u be a node of T and u_1, u_2, \dots, u_p be a subset of the neighbors of u in T . Let $d \in \llbracket 0; p \rrbracket$, $d' \in \llbracket d; d+1 \rrbracket$ and $k \in \llbracket 1; p-d+1 \rrbracket$ if $d \neq 0$ and $k = p+1$ otherwise. We arbitrarily set $\min(\emptyset) = p+1$. We compute the following value:

$$\Xi = \min_{\substack{J \subseteq \llbracket 1; p \rrbracket \\ |J|=d \\ \min(J)=k}} \min_{\substack{\tilde{f}_j \in \tilde{F} \\ j \notin J}} \sum_{j \notin J} i(u, u_j, \tilde{f}_j) \\ \oplus((\tilde{f}, \tilde{f}_j)_{j \notin J}, d') = \tilde{g}$$

Remark. Recall the remark on page 17 interrogating why the formula operator \oplus introduces many rounding errors. This hypothesis is where this choice intervenes as it makes Lemma 14 true.

Lemma 14. Assuming we can access to $i(u, u_j, \tilde{f})$ in $O(1)$ for every $j \in \llbracket 1; p \rrbracket$ and $\tilde{f} \in \tilde{F}$, then we can compute Ξ in $O(n^2 |\tilde{F}|^3)$.

Proof. Ξ can be equivalently seen as the search of p couples $(\tilde{f}_j, b_j)_{j \in \llbracket 1; p \rrbracket}$ where b_j is a boolean determining whether $j \in J$ or not and where:

1. b_k is true and there exist $d-1$ other true b_j with $j > k$; every other boolean is false;
2. if b_j is true then $\tilde{f}_j = 0$;
3. $\tilde{g} = \oplus((\tilde{f}, \tilde{f}_1, \tilde{f}_2, \dots, \tilde{f}_p), d')$;
4. $\sum_{\substack{j=1 \\ b_j \text{ is false}}}^p i(u, u_j, \tilde{f}_j)$ is minimum.

Note that if $d = 0$ and $k = p + 1$, then all the booleans are false.

We use a dynamic programming algorithm to compute the following recursive function H . We set $H(i, \tilde{f}', s)$ as the function that equals the minimum value

$\sum_{\substack{j=1 \\ b_j \text{ is false}}}^i i(u, u_j, \tilde{f}_j)$ that can be obtained with i couples $(\tilde{f}_1, b_1), (\tilde{f}_2, b_2), \dots, (\tilde{f}_i, b_i)$ satisfying

1. if $i \geq k$ then b_k is true and there exist $s - 1$ other true b_j with $j > k$; every other boolean is false; if $i < k$ then $s = 0$ and all the booleans are false.
2. if b_j is true then $\tilde{f}_j = 0$
3. $\tilde{f}' = \oplus((\tilde{f}, \tilde{f}_1, \tilde{f}_2, \dots, \tilde{f}_i), d')$

If no such pairs exists then $H(i, \tilde{f}', s) = +\infty$. By computing $H(p, \tilde{g}, d)$, we obtain the desired result.

We then have the following recurrence formula.

$$H(0, \tilde{f}', s) = 0 \text{ if } s = 0 \text{ and } \tilde{f}' = a\left(\frac{\tilde{f}}{d'}\right), \text{ and } +\infty \text{ otherwise}$$

Indeed, if $i = 0$, then the sum we want to minimize is null, no boolean is true and $\tilde{f}' = \oplus((\tilde{f}), d')$.

$$H(i, \tilde{f}', s) = \min \begin{cases} H(i - 1, \tilde{f}'', s) - i(u, u_i, \tilde{f}_i) \text{ if } i \neq k \text{ and } a(\tilde{f}'' + \frac{\tilde{f}_i}{d'}) = \tilde{f}' \\ H(i - 1, \tilde{f}', s - 1) \text{ if } i \geq k \\ +\infty \end{cases}$$

Note that, if $d = 0$ and $k = p + 1$, then the second case never occurs, meaning that s should equal 0 to reach the terminal state $H(0, \tilde{f}', 0)$.

The first case sets b_i to false and the second one to true. With a dynamic programming algorithm, this formula can be computed in time $O(n^2 |\tilde{F}|^3)$ as there are n possible values for i and s , and $|\tilde{F}|$ for \tilde{f}' ; and as the first part of the formula is computed in time $O(|\tilde{F}|^2)$ by enumerating all the couples $(\tilde{f}'', \tilde{f}_i)$ and the second one in $O(1)$. \square

5.3.7 Main result

Theorem 5. *There exists an FPTAS for max- m and an FPTAS with absolute ratio for max- m and min- R running in time $O(n^7 |\tilde{F}|^7)$ where $|\tilde{F}| = O(\frac{1}{\epsilon} \cdot (n^3 \log(n) + n^2 \log(\Pi)))$.*

Proof. By Lemma 8, 9, 11, 13 and 14, given $\tilde{M}, \tilde{m} \in \tilde{L}$, we can decide whether there exists a feasible orientation \mathcal{O} where $\tilde{M} \geq \tilde{M}(\mathcal{O})$ and $\tilde{m} \leq \tilde{m}(\mathcal{O})$ or not in $O(n^5 |\tilde{F}|^5)$. By computing this for every value of \tilde{M} and \tilde{m} , in $O(n^5 |\tilde{F}|^5 |\tilde{L}|^2)$, we can get a feasible orientation \mathcal{O}_r minimizing $\tilde{R}(\mathcal{O})$ or a feasible orientation

\mathcal{O}_m maximizing $\tilde{m}(\mathcal{O})$. By Lemmas 4 and 5, \mathcal{O}_r is an approximate solution with absolute ratio ε' , and \mathcal{O}_m is an approximate solution with absolute ratio ε' and relative ratio $(1 - \varepsilon')$.

Consequently, this algorithm is an FPTAS for max-m and an FPTAS with absolute ratio for max-m and min-R. The complexity follows from the fact that $|\tilde{L}| = O(n|\tilde{F}|)$ and from Lemma 6. \square

6 Conclusion

In this paper, we have studied the theoretical complexity and approximability of three new optimization flow problems where a constraint of equality is added at the entrance of each node, and whose objective is to return the orientation that balance as much as possible the load of the sources of the network. A reasonable continuation of this work is to implement and evaluate the performance of the polynomial algorithm for min-M and the FPTAS algorithms for max-m and min-R on real electrical distribution networks, to determine their runtime efficiency and the resilience of the proposed configurations.

With this in mind, further work is needed first to improve the complexity of the algorithms. For example, it may not be necessary to enumerate all the values of \tilde{M} , \tilde{m} when searching for a solution optimizing the rounded load reserve with Lemma 8. The computation of the functions i and o should also be improved.

Another continuation of the work would be to adapt the algorithms to networks that are *almost* trees, such as graphs with small treewidth or with small cyclomatic number.

Declarations

- This work is partially supported by the ANR project Labcom HYPHES 21-LCV1-0002.
- The authors have no relevant financial or non-financial interests to disclose.
- No extra material was used to produce this paper

References

- [1] ATKINS, K., CHEN, J., KUMAR, V. A., AND MARATHE, A. The structure of electrical networks: a graph theory based analysis. *International Journal of Critical Infrastructures* 5, 3 (2009), 265.
- [2] BARTH, D., COHEN-BOULAKIA, B., AND EHOUNOU, W. Distributed reinforcement learning for the management of a smart grid interconnecting independent prosumers. *Energies* 15, 4 (2022).

- [3] BARTH, D., MAUTOR, T., DE MOISSAC, A., WATEL, D., AND WEISSER, M.-A. Optimisation of electrical network configuration: Complexity and algorithms for ring topologies. *Theoretical Computer Science* 859 (2021), 162–173.
- [4] BARTH, D., MAUTOR, T., WATEL, D., AND WEISSER, M. A. A polynomial algorithm for deciding the validity of an electrical distribution tree. *Information Processing Letters* 176 (2022).
- [5] BLASI, T. M., DE AQUINO, C. C., PINTO, R. S., DE LARA FILHO, M. O., FERNANDES, T. S. P., AND TABARRO, F. H. Active Distribution Networks with Microgrid and Distributed Energy Resources Optimization Using Hierarchical Model. *Energies*, 15 (2022).
- [6] CALHAU, F. G., AND MARTINS, J. S. B. A electric network reconfiguration strategy with case-based reasoning for the smart grid. In *8th Brazilian Conference on Intelligent Systems* (2019).
- [7] CEJKA, S., FRIEDEN, D., AND KITZMÜLLER, K. Implementation of self-consumption and energy communities in Austria’s and EU member states’ national law: A perspective on system integration and grid tariffs. In *CIREN 2021 Conference* (2021).
- [8] CHRISTIANO, P., KELNER, J. A., MADRY, A., SPIELMAN, D. A., AND TENG, S.-H. Electrical flows, laplacian systems, and faster approximation of maximum flow in undirected graphs. In *Proceedings of the 43rd annual ACM symposium on Theory of computing* (New York, New York, USA, 2011), ACM Press, p. 273.
- [9] DEVAUX, O. The advance management of distribution grid. <https://www.edf.fr/sites/default/files/Lot%203/CHERCHEURS/Publications/advanced-management-distribution-grids.pdf>, 2014. Online report published in EDF Technology unveiled, accessed 08-November-2023.
- [10] FRIEDEN, D., TUERK, A., NEUMANN, C., D’HERBEMONT, S., AND ROBERTS, J. Collective self-consumption and energy communities: Trends and challenges in the transposition of the eu framework. *Theoretical Computer Science* (2020).
- [11] GHIANI, E., MOCCI, S., AND PILO, F. Optimal reconfiguration of distribution network according to the microgrid paradigm. In *Proceedings of 2005 International Conference Futur Power System* (2005).
- [12] GUO, J., HUG, G., AND TONGUZ, O. K. Intelligent Partitioning in Distributed Optimization of Electric Power Systems. *IEEE Transactions on Smart Grid* 7, 3 (may 2016), 1249–1258.

- [13] HONG, H., HU, Z., GUO, R., MA, J., AND TIAN, J. Directed graph-based distribution network reconfiguration for operation mode adjustment and service restoration considering distributed generation. *Journal of Modern Power Systems and Clean Energy* 5, 1 (2017), 142–149.
- [14] IBARRA, O. H., AND KIM, C. E. Fast Approximation Algorithms for the Knapsack and Sum of Subset Problems. *Journal of the ACM (JACM)* 22, 4 (1975), 463–468.
- [15] LABRINI, HAYTHAM. Graph-based model for distribution systems: Application to planning problem. Master’s thesis, 2015.
- [16] LI, J. *Reconfiguration of power networks based on graph-theoretic algorithms*. PhD thesis, Iowa State University, Digital Repository, Ames, 2010.
- [17] LI, J., MA, X.-Y., LIU, C.-C., AND SCHNEIDER, K. P. Distribution System Restoration With Microgrids Using Spanning Tree Search. *IEEE Transactions on Power Systems* 29, 6 (nov 2014), 3021–3029.
- [18] MOSLEHI, K., AND KUMAR, R. A Reliability Perspective of the Smart Grid. *IEEE Transactions on Smart Grid* 1, 1 (jun 2010), 57–64.
- [19] QUIRÓS-TORTÓS, J., AND TERZIJA, V. A graph theory based new approach for power system restoration. In *2013 IEEE Grenoble Conference* (2013), pp. 1–6.
- [20] SAHUA, A., UTKARSH, K., AND DING, F. A Fast and Scalable Genetic Algorithm- Based Approach for Planning of Microgrids in Distribution Networks. In *IEEE Power and Energy Society General Meeting Denver* (2022).
- [21] SHEN, T., LI, Y., XIANG, J., SHEN, T., LI, Y., AND XIANG, J. A Graph-Based Power Flow Method for Balanced Distribution Systems. *Energies* 11, 3 (feb 2018), 511.
- [22] TANG, L., YANG, F., AND MA, J. A survey on distribution system feeder reconfiguration: Objectives and solutions. In *2014 IEEE Innovative Smart Grid Technologies - Asia (ISGT ASIA)* (2014), pp. 62–67.