HAL open science

# Lower Bounds for the Fair Resource Allocation Problem

Zaid Allybokus, Konstantin Avrachenkov, Jérémie Leguay, Lorenzo Maggi

## To cite this version:

Zaid Allybokus, Konstantin Avrachenkov, Jérémie Leguay, Lorenzo Maggi. Lower Bounds for the Fair Resource Allocation Problem. IFIP Performance 2017 - 35th International Symposium on Computer Performance, Modeling, Measurements and Evaluation, Nov 2017, New York, United States. pp.1-7. hal-01706142

**HAL Id: hal-01706142**

**https://inria.hal.science/hal-01706142v1**

Submitted on 10 Feb 2018

**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.

# Lower Bounds for the Fair Resource Allocation Problem

Zaid Allybokus*†, Konstantin Avrachenkov*, Jérémie Leguay† and Lorenzo Maggi†
* INRIA Sophia Antipolis
† Huawei Technologies, France Research Center
{zaid.allybokus, jeremie.leguay, lorenzo.maggi }@huawei.com
konstantin.avratchenkov@inria.fr

## ABSTRACT

The $\alpha$-fair resource allocation problem has received remarkable attention and has been studied in numerous application fields. Several algorithms have been proposed in the context of $\alpha$-fair resource sharing to distributively compute its value. However, little work has been done on its structural properties. In this work, we present a lower bound for the optimal solution of the weighted $\alpha$-fair resource allocation problem and compare it with existing propositions in the literature. Our derivations rely on a localization property verified by optimization problems with separable objective that permit one to better exploit their local structures. We give a local version of the well-known midpoint domination axiom used to axiomatically build the Nash Bargaining Solution (or proportionally fair resource allocation problem). Moreover, we show how our lower bound can improve the performances of a distributed algorithm based on the Alternating Directions Method of Multipliers (ADMM). The evaluation of the algorithm shows that our lower bound can considerably reduce its convergence time up to two orders of magnitude compared to when the bound is not used at all or is simply looser.

## Keywords

Weighted $\alpha$-fairness; Resource allocation; Network utility maximization; Proportional fairness; Max-min fairness; Alternating Directions Method of Multipliers.

## 1. INTRODUCTION

The $\alpha$-fair resource sharing model, first studied in [Mo and Walrand], has already been investigated in numerous application domains, as well as its weighted variants. The weighted $(\boldsymbol{w}, \alpha)$-fair resource allocation problem is to find a vector $\boldsymbol{x}^* \in \mathbf{R}_+^n$ such that 1) the utility

$$f^\alpha(\boldsymbol{w}, \boldsymbol{x}) = \begin{cases} \sum_{i=1}^n w_i \frac{x_i^{1-\alpha}}{1-\alpha}, & \alpha \neq 1, \\ \sum_{i=1}^n w_i \log(x_i), & \alpha = 1 \end{cases}$$

is maximized at $\boldsymbol{x} = \boldsymbol{x}^*$, and 2) $\boldsymbol{x}^*$ lies in a feasible set defined by linear constraints of the form $\mathbf{A}\boldsymbol{x} \leqslant \boldsymbol{c}$ where $\boldsymbol{c} \in \mathbf{R}_+^p$ is a capacity vector for a number $p$ of resources and $\mathbf{A}$ is the binary user-resource incidence $(p, n)$-constraint matrix, for a number $n$ of users, weighted by a positive vector $\boldsymbol{w} \in \mathbf{R}_+^n$. The family of $(\boldsymbol{w}, \alpha)$-fair metrics is general and includes popular fairness concepts such as *max-throughput* ($\alpha = 0$), *proportional fairness*, also called *Nash Bargaining Solution* ($\alpha = 1$), *min-delay* ($\alpha = 2$) and arbitrarily close approximations of *max-min fairness* ($\alpha \to \infty$).

In this paper, we study the general weighted $(\boldsymbol{w}, \alpha)$-fair resource allocation problem under linear constraints and we propose a novel lower bound on its optimal solution. A lower bound is a positive vector $\boldsymbol{d} \in \mathbf{R}_+^n$ respecting feasibility (that is, $\mathbf{A}\boldsymbol{d} \leqslant \boldsymbol{c}$) and such that $\boldsymbol{x}^* \geqslant \boldsymbol{d}$. Finding a lower bound in the context of fair resource sharing is of great interest – it permits one to automatically define a minimal share that is attributable to each resource user as initialization of any exact computation that could take time, and may be helpful in the phase of design of a system. We seek to derive *user-centric* formulas in the sense that their value for a specific user would depend only on the resources within a localized problem (and not on the global topology) and only on the users *that compete over the same resources.* We then evaluate the formulas under different instance regimes and compare them to the literature in order to appreciate the improvements they provide.

Remarkably, we also show how our lower bound can enhance the performance of a distributed algorithm based on the Alternating Directions Method of Multipliers (ADMM) (see [Boyd et al.]) that can be invoked to solve optimally the $\alpha$-fair resource allocation problem. The ADMM is well-known for its fast convergence properties to modest accuracy; however, its performance is highly conditioned by the initialization of its so-called *penalty parameter* that can, when badly tuned, induce an extremely poor convergence rate. Thus, tuning correctly the penalty parameter is a task that one should not neglect when using the ADMM. In light of recent studies (in particular, we exploit the results proven in [Deng and Yin]), we demonstrate how our lower bound permits one to accomplish this task for our particular problem.

A well known lower bound of the proportionally fair ($\alpha = 1$) resource allocation was brought in as a building block of the axiomatization of the Nash Bargaining Solution and is commonly referred to as the *midpoint domination* axiom [de Clippel]. It states that each user $i$ is given at least a fraction $\frac{w_i}{\sum_{j=1}^n w_j}$ of their *dictatorial allocation*, that is, the resource they would receive if the other users accept to receive 0. We refer to the bound given by the midpoint domination axiom as the *midpoint allocation.* One can imagine that the midpoint allocation becomes arbitrarily poor as the total number of users becomes large, and its utility as a first estimation of the optimum allocation, negligible. Indeed, the

*IFIP WG 7.3 Performance 2017.* Nov. 14-16, 2017, New York, NY USA
Copyright is held by author/owner(s).

formula includes the weights of the whole set of users and is independent of the problem's local structure. Similarly, the general lower bound found in [Marasevic et al.] may suffer from these dependencies.

Concerning proportional fairness ($\alpha = 1$), we give a more precise midpoint domination axiom, and provide a lower bound that we call *local midpoint*. Our lower bound on the proportionally fair allocation can be interpreted as a particular case of the midpoint domination axiom to locality – now, each user $i$ is proportionally fairly attributable at least a fraction $\frac{w_i}{\sum_{j\in S_i} w_j}$ of their dictatorial allocation, where $S_i$ is not the total set of users, but *the set of users in competition with the user i for some resource.* Few works attempted at providing lower bounds for the general $(\boldsymbol{w},\alpha)$-fair resource allocation. In fact, the most recent available bound is shown by [Marasevic et al.], and used by the authors for an initialization of their $\alpha$-fair heuristic. To the best of our knowledge, this is the best bound that could be found in the literature for the $\alpha$-fair resource allocation problem and we refer to it as the State-of-the-Art (SoA).

The remainder of the paper is organized as follows: Section 2 is dedicated to the model definition and problem statement. Our lower bound presentation is addressed in Section 3. In Section 4, we broadly remind the key features of the ADMM-based $\alpha$-fair distributed algorithm used for our illustration. The performance of the latter is shown in Section 5 and finally, Section 6 concludes the paper.

# 2. MODEL DEFINITION

Let us start by formalizing the weighted $\alpha$-fair resource allocation problem. In this work, we adopt the terminology of rate control in fixed communication networks. Thus, a resource will be referred to as a *link* and a user will be called a *connection request* (or shortly, *request*) from a source node to a destination node over a *route* formed of several links.

Let $J$ be the set of network links, each link $j$ having a capacity $c_j \in \mathbf{R}_+$. Let $R$ be the set of requests. Each request $r$ has a predefined route that identifies with a subset $J_r \subset J$ of links of the network. In turn, for each link $j \in J$, $R_j \coloneqq \{r \in R; j \in J_r\}$ is the set of all requests having a route that contains the link $j$. We define the link-route incidence $|J| \times |R|$-matrix $\mathbf{A}$ as:

$$\mathbf{A}_{jr} = \begin{cases} 1 & \text{if } j \in J_r \\ 0 & \text{otherwise} \end{cases}$$

For each request $r$, $x_r$ denotes the bandwidth allocated to $r$ along its route $J_r$. We say that an allocation $\boldsymbol{x} = (x_r)_{r\in R}$ belongs to the feasibility set $C$ (or is *feasible*) if it satisfies the capacity constraint (1) below:

$$\boldsymbol{x} \in C \Leftrightarrow \mathbf{A}\boldsymbol{x} \leqslant \boldsymbol{c},\ \boldsymbol{x} \geqslant 0 \tag{1}$$

where $\boldsymbol{c} = (c_j)_{j\in J}$. Each request $r$ is associated with a weight $w_r \in \mathbf{R}_+$. The weight vector $\boldsymbol{w} = (w_r)_{r\in R}$ accounts for a degree of relative importance of each request that can be defined at the discretion of the network. Weighted $\alpha$-fairness is formalized as in Definition 1 below.

Definition 1 ($(\boldsymbol{w},\alpha)$-fairness). *Let $C \subset \mathbf{R}_+^{|R|}$ be a feasibility set defined as in* (1)*, being a strict superset of* $\{0\}$*. Let $\boldsymbol{w} \in \mathbf{R}_+^{|R|}$ and $\boldsymbol{x}^* \in C$. We say that $\boldsymbol{x}^*$ is* $(\boldsymbol{w},\alpha)$-fair *(or simply $\alpha$-fair when there is no confusion on $\boldsymbol{w}$) if the following holds:*

$$\forall r \in R, \quad x_r^* > 0 \quad \textit{and} \quad \forall \boldsymbol{x} \in C, \quad \sum_{r\in R} w_r \frac{x_r - x_r^*}{x_r^{*\alpha}} \leqslant 0.$$

*Equivalently, $\boldsymbol{x}^*$ is $(\boldsymbol{w},\alpha)$-fair if, and only if $\boldsymbol{x}^*$ maximizes the $\alpha$-fair utility function $f^\alpha$ defined over $C - \{\mathbf{0}\}$:*

$$\max_{\boldsymbol{x}\in C} f^\alpha(\boldsymbol{w},\boldsymbol{x}) = \sum_{r\in R} f_r^\alpha(w_r, x_r), \tag{$P_\alpha$}$$

*where*

$$f_r^\alpha(w_r, x_r) = \begin{cases} w_r \frac{x_r^{1-\alpha}}{1-\alpha}, & \alpha \neq 1, \\ w_r \log(x_r), & \alpha = 1. \end{cases}$$

# 3. ALPHA-FAIRNESS – A LOWER BOUND

In this section, we derive an explicit lower bound on the general $(\boldsymbol{w},\alpha)$-fair resource allocation problem. Our lower bound only depends on the weight vector $\boldsymbol{w}$, the capacity vector $\boldsymbol{c}$ and the link-route incidence matrix $\mathbf{A}$. Moreover, the bound exploits the local structure of the problem, which prevents it from deteriorating systematically with the problem size. We compare it to the SoA bound that one can formulate as follows:

Proposition 1 ([Marasevic et al.]). *Let the vector $\boldsymbol{x}^*$ be the optimal solution to the $\alpha$-fair resource allocation problem. Then, for all $r \in R$:*

- *if* $0 < \alpha \leqslant 1$, $\quad x_r^* \geqslant m_r(\alpha) \coloneqq \left(\frac{w_r}{w_{\max} M} \min_{j\in J_r} \frac{c_j}{|R_j|}\right)^{1/\alpha} c_{\max}^{1-1/\alpha}$
- *if* $\alpha > 1$, $\quad x_r^* \geqslant m_r(\alpha) \coloneqq \left(\frac{w_r}{w_{\max} M}\right)^{1/\alpha} \min_{j\in J_r} \frac{c_j}{|R_j|} \left(\frac{c_{\min}}{c_{\max}}\right)^{1-1/\alpha}$

*where* $w_{\max} = \max w_r$, $M = \min\{|R|, |J|\}$, $c_{\min} = \min c_j$ *and* $c_{\max} = \max c_j$.

We seek to improve the above bound by removing the global dependencies on $w_{\max}$, $|R_j|$, and $M$, $c_{\min}$ and $c_{\max}$, those parameters being the major degradation factor when the size or congestion of the problems increase.

For each request $r \in R$, let $b_r \coloneqq \min_{j\in J_r} c_j$. The so-called *utopia point* $\boldsymbol{b} \coloneqq (b_r)_{r\in R}$ is the (infeasible when the problem is non trivial) allocation representing the value each request would receive if they were alone in the network, that is, its dictatorial allocation. Our bound for the $(\boldsymbol{w},\alpha)$-fair allocation only depends on the utopia point (hence on the capacity vector $\boldsymbol{c}$), the matrix $\mathbf{A}$ and on the weight vector $\boldsymbol{w}$. For $r \in R$, let $R^r \coloneqq \{s \in R; J_r \cap J_s \neq \varnothing\}$, i.e., the set of requests sharing at least one resource with $r$ and $\overline{R^r} \coloneqq R - R^r$.

First of all, we use the separability of the objective function of Problem $(P_\alpha)$ to better estimate our lower bound on a restricted problem. Specifically, we prove a *restriction lemma* (see Lemma 1) that permits one to avoid unnecessary dependencies between requests that do not share resources together. Then, we prove our general lower bound on the corresponding restricted problems. Thanks to Lemma 1, the bound remains unchanged in the original problem.

## 3.1 A restriction lemma

In this paragraph, we show that instead of evaluating our bound on Problem $(P_\alpha)$, one can use a smaller request-centric problem. Specifically, let $\boldsymbol{x}^*$ denote the optimal solution of Problem $(P_\alpha)$ and let $r_0 \in R$ be *an arbitrary request.* We define the *restricted problem* at $r_0$, as the following:

$$\min \sum_{r \in R^{r_0}} -f_r^\alpha(w_r, x_r) \qquad (\widetilde{P_{r_0}})$$

$$\text{s.t.} \sum_{r \in R_j \cap R^{r_0}} x_r \leqslant \tilde{c}_j \coloneqq c_j \quad \forall j \in J_{r_0}$$

$$\text{and} \sum_{r \in R_j \cap R^{r_0}} x_r \leqslant \tilde{c}_j \coloneqq c_j - \sum_{r \in R_j \cap \overline{R^{r_0}}} x_r^* \quad \forall j \in J - J_{r_0}.$$

Intuitively, Problem $(\widetilde{P_{r_0}})$ arises when the allocations of all the requests that do not share any link with $r_0$ are fixed to their optimal $\alpha$-fair value (that is, following the vector $\boldsymbol{x}^*$), and one needs to compute the $\alpha$-fair allocation of the remaining requests, that is, the requests within $R^{r_0}$ that share at least one resource with $r_0$. The capacity constraints are thus updated taking into account the amounts of resources that are already allocated, as shows the second line of the constraints. Note in passing that all the links in $J - J_{r_0}$ that do not serve any of the requests within $R^{r_0}$ form trivial constraints in $(\widetilde{P_{r_0}})$ and can hence be removed without any loss.

We then have the following result:

Lemma 1. *The restriction to $(\widetilde{P_{r_0}})$ does not change the optimal allocation of the remaining requests: if $\boldsymbol{x}$ is the optimal solution of the Problem $(\widetilde{P_{r_0}})$, then, $x_s = x_s^*$, for $s \in R^{r_0}$.*

Proof. Consider the problem:

$$\min \sum_{r \in R} -f_r^\alpha(w_r, x_r) \qquad (2)$$

$$\text{s.t. } \mathbf{A}\boldsymbol{x} \leqslant \boldsymbol{c}$$

$$x_r \geqslant x_r^* \quad \forall r \in \overline{R^{r_0}}.$$

It suffices to show that the problems (2) and $(\widetilde{P_{r_0}})$ are equivalent. Then, the unicity of the solutions permits one to conclude.

We know that the problem (2) is feasible, as $\boldsymbol{x}^*$ is a feasible point. Denote its optimal solution by $\tilde{\boldsymbol{x}}$. We remark that both $\boldsymbol{x}^*$ and $\tilde{\boldsymbol{x}}$ are feasible for both problems $(P_\alpha)$ and (2). Hence, by optimality, we necessarily have $f^\alpha(\boldsymbol{w}, \tilde{\boldsymbol{x}}) = f^\alpha(\boldsymbol{w}, \boldsymbol{x}^*)$. Moreover, for instance, problem $(P_\alpha)$ has a unique optimal solution. Thus,

$$\boldsymbol{x}^* = \tilde{\boldsymbol{x}}.$$

Particularly for $r \in \overline{R^{r_0}}, x_r^* = \tilde{x}_r$. Thus, we can fix the values $x_r = x_r^*$ for $r \in \overline{R^{r_0}}$ without changing the optimum. Thus, Problem (2) is equivalent to the restricted problem $(\widetilde{P_{r_0}})$. □

Thanks to Lemma 1, we are now ready to present our lower bound on the $\alpha$-fair allocation based on the structure of the restricted problems.

## 3.2 Lower bound

We now show the main result of this paper. We define the *local midpoint* $\boldsymbol{p}$ as the following:

$$\forall r \in R \quad p_r \coloneqq \frac{w_r}{\sum_{s \in R^r} w_s} b_r.$$

Theorem 1. *Let $\boldsymbol{x}^*$ denote the optimal solution of problem $(P_\alpha)$. Let $r_0 \coloneqq \operatorname{argmin}_{s \in R} p_s$. Then, $\boldsymbol{x}^*$ can be lower bounded as follows:*

- *if $\alpha \geqslant 1$,* $\quad \forall r \in R \quad x_r^* \geqslant d_r(\alpha) \coloneqq p_{r_0}^{1-1/\alpha} p_r^{1/\alpha}$
- *if $0 < \alpha \leqslant 1$,* $\quad \forall r \in R \quad x_r^* \geqslant d_r(\alpha) \coloneqq \left( \dfrac{w_r b_r}{\sum_{s \in R^r} w_s b_s^{1-\alpha}} \right)^{1/\alpha}.$

Proof. We first prove the proposition for $\alpha \geqslant 1$. Let us define the request $r_{\min}$ as the request with the least optimal allocation: $r_{\min} = \operatorname{argmin}_{s \in R} x_s^*$. By definition of $r_0$, we have:

$$p_{r_{\min}} \geqslant p_{r_0} \qquad (3)$$

Let $r \in R^r$. By Lemma 1, it suffices to show the inequality in the restricted problem $(\widetilde{P_r})$ associated to $r$. Let $C^r$ denote its feasible set. Thus, for all $(x_s)_{s \in R^r} \in C^r$ we have:

$$\sum_{s \in R^r} w_s \frac{x_s - x_s^*}{x_s^{*\alpha}} \leqslant 0,$$

This inequality holds for all feasible $(x_s)_{s \in R^r} \in C^r$. Thus, we evaluate it at the dictatorial allocation of $r$, that is, at the point $\boldsymbol{x}$ defined as $x_r = \tilde{b}_r$ and $x_s = 0$ for all $s \in R^r - \{r\}$. Let us note in passing that $\tilde{b}_r = \min_{j \in J_r} \tilde{c}_j = \min_{j \in J_r} c_j = b_r$. Thus,

$$w_r \tilde{b}_r = w_r b_r \leqslant x_r^{*\alpha} \sum_{s \in R^r} w_s x_s^{*1-\alpha} \leqslant \left( \sum_{s \in R^r} w_s \right) x_{r_{\min}}^{*1-\alpha} x_r^{*\alpha},$$

where we remind that $r_{\min} = \operatorname{argmin}_{s \in R} x_s^*$ and $1 - \alpha \leqslant 0$. Rearranging the terms, one gets:

$$\frac{w_r b_r}{\sum_{s \in R^r} w_s} x_{r_{\min}}^{*\alpha-1} \leqslant x_r^{*\alpha},$$

which yields:

$$p_r^{1/\alpha} x_{r_{\min}}^{*1-1/\alpha} \leqslant x_r^* \qquad (4)$$

In particular, applying equation (4) to $r = r_{\min}$, we get:

$$x_{r_{\min}}^* \geqslant p_{r_{\min}} \geqslant p_{r_0} \qquad (5)$$

Finally, we plug equation (5) in equation (4) to obtain the desired lower bound on $x_r^*$ (because $1 - 1/\alpha \geqslant 0$).

Next, we show the bound for $0 < \alpha < 1$. In the same fashion, we look at the restricted problem. Let $r \in R$ and consider its restricted problem. Then, one has:

$$\frac{w_r b_r}{x_r^{*\alpha}} \leqslant \sum_{s \in R^r} w_s x_s^{*1-\alpha} \leqslant \sum_{s \in R^r} w_s \tilde{b}_s^{1-\alpha} \leqslant \sum_{s \in R^r} w_s b_s^{1-\alpha}.$$

Rearranging the terms finally provides the desired bound. For any value of $\alpha$, one can remark that the bound $(d_r(\alpha))_{r \in R}$ only depends on the capacity vector $\boldsymbol{c}$, the weight vector $\boldsymbol{w}$, and the link-route incidence matrix $\mathbf{A}$. □

## 3.3 Illustration

To conclude this section, we illustrate a comparison of the two presented lower bounds $\boldsymbol{m}$ and $\boldsymbol{d}$ introduced in Proposition 1 and Theorem 1, respectively, under different regimes. Given the formulas, one can remark that the sensitivity of the bound to arbitrary problem sizes should be lessened as now more focused on local structures. For $\alpha \leqslant 1$, we obtain request-centric formulas. For general $\alpha > 1$, this elimination came with the dependency on the global minimum local

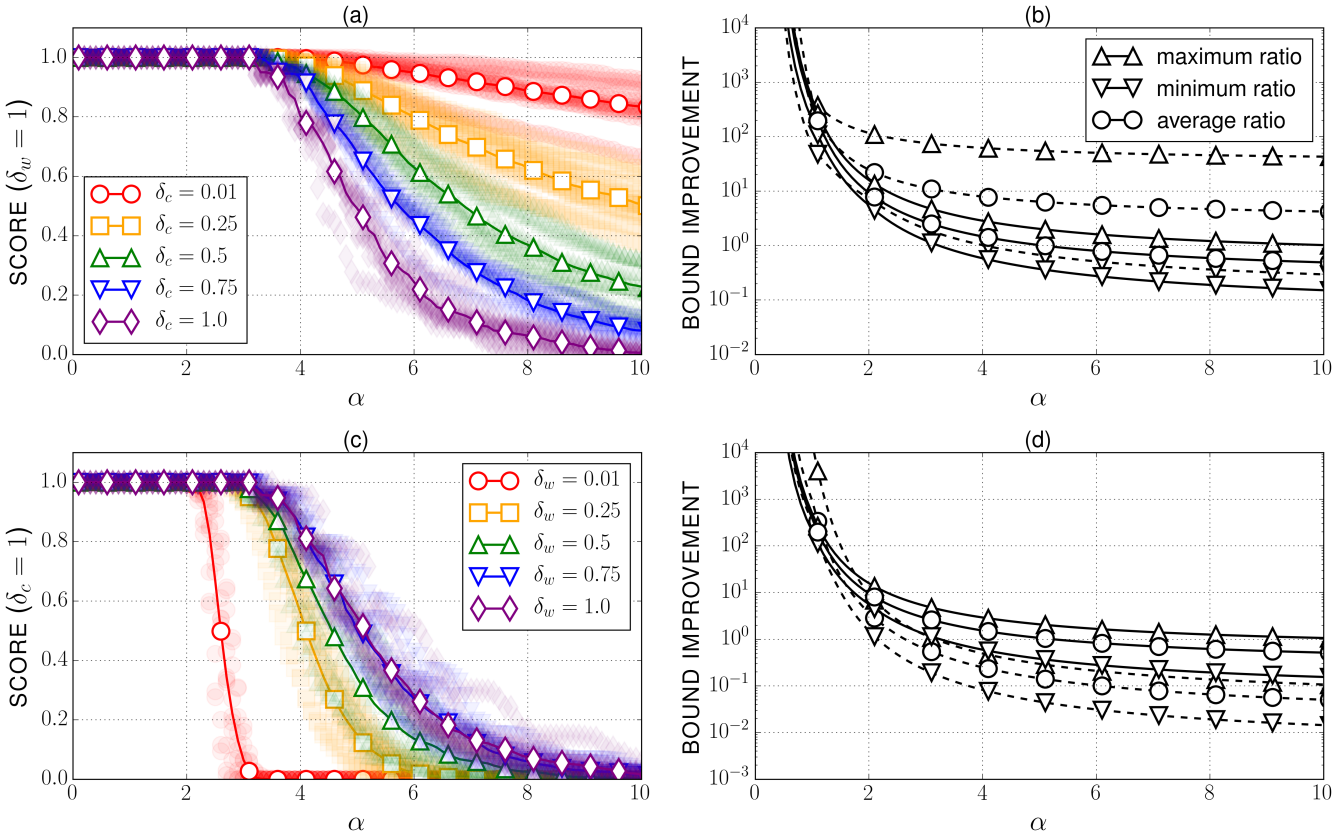


Figure 1: A comparison of the two bounds. The scores, and the minimum, average and maximum bound improvements are illustrated in the cases of (a)-(b) a constant $\delta_w$ for different values of $\delta_c$, and of (c)-(d) a constant $\delta_c$ for different values of $\delta_w$. Figures (b) and (d) show the bound improvements in the two extreme situations $\delta_c$ (resp. $\delta_w$) = 0.01 (resp. 1) in dashed lines (resp. solid lines).

midpoint value $p_{r_0}$. Intuitively, one can remark that the two bounds may react differently to a fluctuating asymmetry of the weight vector $\boldsymbol{w}$ or the capacity vector $\boldsymbol{c}$, namely, a variation of the two parameters $\delta_w \coloneqq \frac{\min w_r}{\max w_r}$ and $\delta_c \coloneqq \frac{c_{\min}}{c_{\max}}$. For a better vision, we illustrated this behavior in Figure 1.

The two bounds were compared on instances with 1000 requests over a same graph of type *barabasi(100,4)* (see [Albert and Barabási]). The routes were generated at random by taking the shortest path between pairs of sources and destinations drawn uniformly at random. The weights (resp. link capacities) were also drawn uniformly at random within intervals $I$ satisfying $\inf I / \sup I = \delta_w$ (resp. $\delta_c$). For each instance, and each $\alpha$, we define the *score* of $\boldsymbol{d}$ as the number $|\{r : d_r(\alpha) > m_r(\alpha)\}|/|R|$. The score represents the proportion of requests for which our bound $\boldsymbol{d}(\alpha)$ beats the SoA bound $\boldsymbol{m}(\alpha)$ for a particular $\alpha$. In Figure 1(a), the parameter $\delta_w$ was fixed to 1 (which namely means $\boldsymbol{w} = \mathbf{1}$) and we plotted the score of $\boldsymbol{d}$ versus $\alpha$ for different values of $\delta_c$. Figure 1(c) shows the score in the other extreme situation $\delta_c = 1$ (which means all the link capacities are equal) for different values of $\delta_w$.

In order to appreciate the quality of the bound improvement, if any, we plotted, in Figures 1(b) and 1(d), the corresponding bound improvements, measured with the values of the ratios $d_r(\alpha)/m_r(\alpha)$. To preserve readability of the plots, we represented only the extreme situations corresponding to the values $\delta_c = 0.01$ (dashed lines) and $\delta_c = 1$ (solid lines) for Figure 1(b) and to the values $\delta_w = 0.01$ (dashed lines) and $\delta_w = 1$ (solid lines) for Figure 1(d). Figures 1(b) and 1(d) show the best, worst, and average improvements encountered in the same problem instance. All the points represented in Figure 1 correspond to an average over 10 instances generated under identical conditions. In Figures 1(a) and 1(c), we also included the specific points as translucent scattered markers.

According to Figure 1, our bound is an absolute improvement for values of $\alpha$ in the interval [0,2] (thus including the max-throughput, proportional fairness, and min-delay popular concepts) in all situations. Particularly for proportional fairness, the simulations show that we improved the bound $\boldsymbol{m}$ by two orders of magnitude in all situations. For min-delay fairness, the bound is generally improved on average by a multiplicative factor between 1 and several tens. For greater values of $\alpha$, it is interesting to see that either $\boldsymbol{d}$ or $\boldsymbol{m}$ is more adapted to certain problem structures. For instance, $\boldsymbol{d}$ will be of greater interest when the network link capacities are more heterogeneous, $\delta_c \ll 1$ (which may correspond to situations where the network is asymmetrically congested), whereas $\boldsymbol{m}$ is more adapted to asymmetrically weighted problems, $\delta_w \ll 1$. One can thus conclude that the two available bounds complement each other for general $\alpha \geqslant 1$.

After presenting our bound, we now demonstrate how it permits one to boost the performance of an algorithm that

solves the $(\boldsymbol{w}, \alpha)$-fair resource allocation problem.

The next section is dedicated to the presentation of the algorithm, based on ADMM.

# 4. FAST AND DISTRIBUTED ADMM (FD-ADMM)

Several approaches may be used to tackle the $(\boldsymbol{w}, \alpha)$-fair resource allocation problem (e.g., see [Kelly et al.] and [Palomar and Chiang] for a tutorial). One of them is the Alternating Directions Method of Multipliers (ADMM) (see, e.g., [Boyd et al.]). The ADMM is well known for its distributivity properties that permit one to decouple constraints handled in parallel then plugged in together by means of *consensus constraints*. In [Allybokus et al.], these properties are used to design a fully distributed algorithm that solves optimally the problem in the context of traffic rate control in distributed Software-Defined Networks. For a description of the general ADMM framework, the reader may refer to [Boyd et al.], and for a more detailed construction of the presented algorithm, to [Allybokus et al.]. In this section, we briefly describe the design of the distributed algorithm used in the latter.

## 4.1 Algorithm overview

Assume the network links are split into a number $P \geqslant 1$ of domains. Each domain $p$ corresponds to a subset $J_p \subset J$ of links forming a covering of the whole set $J$:

$$\bigcup_{p=1}^{P} J_p = J.$$

For $p = 1 \dots P$, let $R_p = \{r \in R, J_p \cap J_r \neq \varnothing\}$ be the set of requests that traverse domain $p$, and $I_r = \{q \in [1, P]; r \in R_p\}$ the set of domains the request $r$ traverses. The problem $(P_\alpha)$ can thus be rewritten as: :

$$\begin{aligned}
&\min_{\boldsymbol{x} \in C} \sum_{r \in R} -f_r^\alpha(w_r, x_r) \\
&= \min_{\boldsymbol{x} \in C} \sum_{p=1}^{P} \sum_{r \in R_p} -\frac{1}{|I_r|} f_r^\alpha(w_r, x_r) \\
&= \min_{\boldsymbol{x}} \sum_{p=1}^{P} \left\{ \iota_p(\boldsymbol{x}) + \sum_{r \in R_p} -\frac{1}{|I_r|} f_r^\alpha(w_r, x_r) \right\} \\
&:= \min_{\boldsymbol{x}} \sum_{p=1}^{P} \iota_p(\boldsymbol{x}) + g_p(\boldsymbol{w}, \boldsymbol{x}), \qquad (6)
\end{aligned}$$

where $\iota_p$ is the indicator function of the capacity subset associated to domain $p$:

$$\iota_p(\boldsymbol{x}) = \sum_{j \in J_p} \iota_j(\boldsymbol{x}), \quad \iota_j(\boldsymbol{x}) = \begin{cases} 0 & \text{if } \sum_{r \in R_j} x_r \leqslant c_j \\ \infty & \text{otherwise} \end{cases}$$

Further, we separate artificially the problem by creating a private variable $\boldsymbol{x}_p \in \mathbf{R}^{|R_p|}$ for each domain $p$, and by enforcing the agreement upon their values between domains with consensus constraints. Problem formulation (6) now reads:

$$\begin{aligned}
\min \quad & \sum_{p=1}^{P} g_p(\boldsymbol{w}, \boldsymbol{x}_p) + \iota_p(\boldsymbol{x}_p) \\
\text{s.t} \quad & x_{pr} = x_{qr} \quad \forall p, q \in I_r \quad \forall r \in R \qquad (7) \\
& \boldsymbol{x}_p \in \mathbf{R}^{|R_p|} \quad \forall p = 1 \dots P.
\end{aligned}$$

**Algorithm 1** Fast Distributed ADMM (FD-ADMM)

1: **procedure** of Domain $p$
2: Receive $\tilde{\boldsymbol{z}}_p = (\tilde{z}_r)_{r \in R_p}$
3: Receive updated reciprocal penalty $\lambda_0$ from master
4: **for** $j \in J_p$ **do**
5: $u_{jr} \leftarrow u_{jr} + y_{jr} - \tilde{z}_{pr} \quad \forall r \in R_j$
6: $y_j \leftarrow \mathrm{P}(j, \tilde{\boldsymbol{z}}_p - \boldsymbol{u}_j)$
7: **end for**
8: **for** $r \in R_p$ **do**
9: $v_{pr} \leftarrow v_{pr} + x_{pr} - \tilde{z}_{pr}$
10: $x_{pr} \leftarrow \operatorname{argmin}_x \left\{ -f_r^\alpha(w_r, x) + \frac{1}{2\lambda_0} \|x - (\tilde{z}_{pr} - v_{pr})\|^2 \right\}$
11: **end for**
12: **end procedure**
13: **procedure** of Master
14: Compute lower bound $\boldsymbol{d}$ and $\lambda_0$ using Eq. (10)
15: **while** termination condition not met **do**
16: $\lambda_0 \leftarrow \lambda$ updated using RB, by [He et al.]
17: **for** $r \in R$ **do**
18:
19: $\tilde{z}_r \leftarrow \frac{1}{|J_r| + |Ir|} \left( \sum_{j \in J_r} y_{jr} + \sum_{q \in I_r} x_{qr} \right)$
20: **end for**
21: **for** $p \in P$ **do**
22: Send $\tilde{\boldsymbol{z}}_p = (\tilde{z}_r)_{r \in R_p}$ to domain $p$
23: **end for**
24: **end while**
25: **end procedure**

Finally, we decompose the problem by separating the private objective of each domain. For each domain $p$, and each $j \in J_p$, the vector $\boldsymbol{y}_j$ defines a copy of the variable $\boldsymbol{x}_p$ for link $j$ and is reserved for the component function $\iota_j$. We can write Problem (7) in the following form:

$$\begin{aligned}
\min \quad & \sum_{p=1}^{P} g_p(\boldsymbol{w}, \boldsymbol{x}_p) + \sum_{j \in J_p} \iota_j(\boldsymbol{y}_j) := \sum_{p \in P} G_p(\boldsymbol{w}, \boldsymbol{x}_p, \boldsymbol{y}_p) \\
\text{s.t} \quad & x_{pr} = x_{qr} \quad \forall p, q \in I_r \quad \forall r \in R \qquad (8) \\
& x_{pr} = y_{jr} \quad \forall j \in J_p \quad \forall r \in R_j \\
& \boldsymbol{x}_p \in \mathbf{R}^{|R_p|} \quad \forall p = 1 \dots P \\
& \boldsymbol{y}_j \in \mathbf{R}^{|R_j|} \quad \forall j \in J \\
& \boldsymbol{y}_p = (\boldsymbol{y}_j)_{j \in J_p} \quad \forall p \in P
\end{aligned}$$

Let $\chi$ denote the indicator function of the feasible set (8). Then, the formulation takes the compact 2-block form:

$$\begin{aligned}
& \min \sum_{p \in P} G_p(\boldsymbol{w}, \boldsymbol{x}_p, \boldsymbol{y}_p) + \chi((\boldsymbol{x}'_p)_{p \in P}, (\boldsymbol{y}'_p)_{p \in P}) \qquad (9) \\
& \text{s.t. } (\boldsymbol{x}_p, \boldsymbol{y}_p) = (\boldsymbol{x}'_p, \boldsymbol{y}'_p)
\end{aligned}$$

Applied to the last formulation (9), the distributed ADMM is described in Algorithm 1. In lines 5 and 10, the variables $\boldsymbol{u}_j \in \mathbf{R}^{|R_j|}$ and $\boldsymbol{v}_p \in \mathbf{R}^{|R_p|}$ are dual variables associated with the constraints $\{\boldsymbol{y}_j = \boldsymbol{y}'_j\}$ and $\{\boldsymbol{x}_p = \boldsymbol{x}'_p\}$, respectively in (9). Also, $\mathrm{P}(j, \cdot)$ is the Euclidean projection onto the simplex $\{\boldsymbol{y}_j \in \mathbf{R}^{R_j}$ s.t. $y_{jr} \geqslant d_r$ and $\sum_{r \in R_j} y_{jr} \leqslant c_j\}$, $\lambda > 0$ is a scalar reciprocal penalty parameter, and $\boldsymbol{d} \in \mathbf{R}^R$ is a lower bound on the $(\boldsymbol{w}, \alpha)$-fair solution that will be computed with the input parameters.

## 4.2 Performance

The convergence of ADMM is provably known since the 1990s (see [Eckstein and Bertsekas]), and its convergence rate has been widely studied. Today, the most general convergence rate of ADMM is known to be $O(1/T)$ ($T$ being the iteration count), and linear convergence rates are provably obtained for strongly convex problems. Nevertheless, the performance of the ADMM remains highly sensitive to the initialization and the update of the penalty parameter. In [Deng and Yin], the linear convergence rate of ADMM for strongly convex problems is quantified and optimized with regards to the penalty parameter, which yields an optimal tuning of it. Its value depends on the (global) strong convexity and the Lipschitz gradient moduli of the objective function, if those are finite. In [Allybokus et al.], this result is applied to a central strongly convex equivalent formulation of our problem to derive an approximate adaptive tuning of the distributed version of the algorithm. The adaptive penalty parameter is computed as the optimal parameter of the centralized formulation, $\lambda_0$, given according to the formula

$$\lambda_0 = \frac{1}{\sqrt{\sigma L}}, \tag{10}$$

where $\sigma$ is the strong convexity modulus of $f^\alpha(\boldsymbol{w}, \cdot)$ and $L$ is the Lipschitz modulus of its gradient. In fact, the fairness functions have singular values near 0, which make the Lipschitz modulus not globally defined, unless the feasible set is reduced from below by means of a positive lower bound $\boldsymbol{d}$ of the optimal solution. Thus, Equation (10) is applied to $L = L_{\boldsymbol{d}}$ where $L_{\boldsymbol{d}}$ is the Lipschitz modulus of the gradient of the objective over the set of feasible points $\boldsymbol{x}$ verifying $\boldsymbol{x} \geqslant \boldsymbol{d}$.

Adaptive penalty parameter schemes have been proposed to tackle this issue and provably bring consistent improvement of the convergence behavior of ADMM. One remarkable adaptive scheme can be found in [He et al.], in which the authors introduce the *residual balancing* (RB) principle which consists of shrinking or expanding the penalty parameter whenever the primal and dual residuals are unbalanced. For a definition of RB, we refer the interested reader to [He et al.]. Although this scheme helps making the ADMM less dependent from initialization, empirical behaviors of the algorithm however suggest that there is still room and interest for better initialization. To demonstrate this, we adopt residual balancing as a default adaptive scheme of our penalties in all the algorithms of the present paper.

In Section 3, we introduced the lower bound $\boldsymbol{d}$ (Theorem 1) on the $(\boldsymbol{w}, \alpha)$-fair optimal allocation. Next, we demonstrate how this bound permits one to enhance the performance of the ADMM for the $(\boldsymbol{w}, \alpha)$-fair resource allocation problem, and we compare it with the performance brought by the SoA bound $\boldsymbol{m}$ (Proposition 1). Although the lower bound permits one to adjust quickly a minimal individual resource allocation that would never be violated during the running time of the algorithm, the major feature of its introduction is in that it permits one to define an initialization of the penalty parameter that could enhance the algorithm performance. Indeed, the initialization can provide spectacular convergence acceleration, whereas reducing the feasible set at the projection line 6 of Algorithm 1 does not seem to matter, illustrating the fast convergence of FD-ADMM to modest accuracy. These observations are illustrated in the next section.

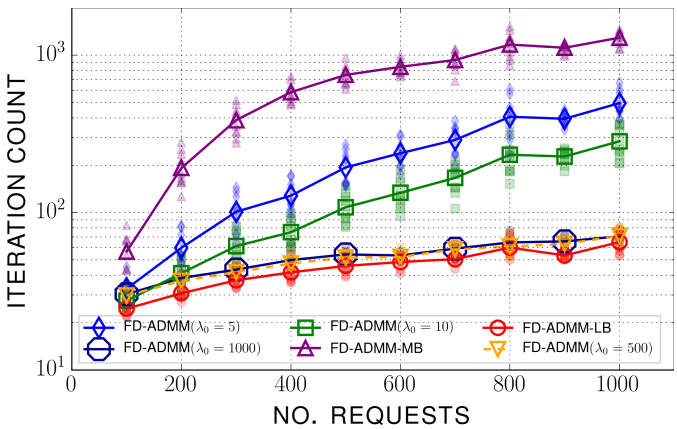


Figure 2: Iteration count versus the number of connection requests in situation 1. For FD-ADMM-LB and FD-ADMM-MB, the reciprocal penalty initialized value $\lambda_0$ lies in [110-150] and in [1, 6], respectively.

# 5. EXECUTION

In the present simulations, we dedicate our performance evaluation to the proportionally fair resource allocation problems ($\alpha = 1$). In this section, we demonstrate the gains achievable with only tuning the initial penalty parameter of the FD-ADMM by comparing several initialization schemes. Indeed, the only difference between the different algorithms that we compare is in that the initial penalty parameter $\lambda_0$ is chosen either arbitrarily – FD-ADMM($\lambda_0 = \lambda$), or according to Equation (10) applied to the bound $\boldsymbol{m}$ (FD-ADMM-MB) or $\boldsymbol{d}$ (FD-ADMM-LB).

The problem instances were generated under the same conditions as in Section 3.3. As it appears the parameter $\delta_w$ can deteriorate importantly the quality of our bound when small, we execute the simulations under two different situations 1) $w_r \in [.9, 1]$, and 2) $w_r \in [.1, 1]$.

## Performance results

In Figures 2 and 3, we plotted the iteration count of the algorithms under situations 1 and 2, respectively. The algorithms stop when the primal and dual residuals of the ADMM algorithm (see, e.g., [Boyd et al.]) fall below $10^{-2}$ (relatively modest accuracy). For each problem size in terms of number of different requests, we generated 10 instances of the corresponding size randomly and plotted the average performance. The specific points are also represented with scattered translucent markers to account for the exact performance of each algorithm. For each situation, we observed the performances of FD-ADMM-LB, in particular, its average initial reciprocal penalty parameter given by Equation (10) and chose several initialization values below and above this average to account for the effect of this initialization on the algorithm's performance.

**Situation 1 (Figure 2).** We observe a spectacular improvement of the FD-ADMM algorithm from the scheme FD-ADMM-MB to the scheme FD-ADMM-LB, corresponding to a reduction of the running iteration count of two orders of magnitude. When $\lambda_0$ is chosen larger than the one for FD-ADMM-LB, although the performances seem satisfactory, one can observe that FD-ADMM-LB still executes faster on average.

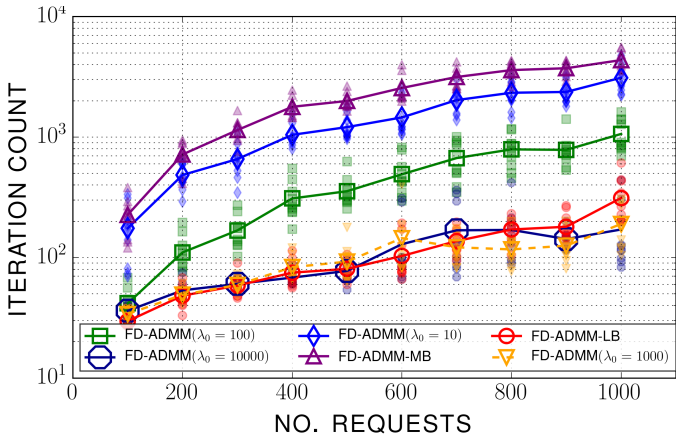


Figure 3: Iteration count versus the number of connection requests in situation 2. For FD-ADMM-LB and FD-ADMM-MB, the reciprocal penalty initialized value $\lambda_0$ lies in [500,1000] and in [0.1, 2.5], respectively.

**Situation 2 (Figure 3).** The same improvement of the performances, related to the introduction of our lower bound, is observed. It seems that for lower initialization value of $\lambda_0$, the algorithms demonstrate poorer performances. Nevertheless, one can observe that higher values of $\lambda_0$ can provide algorithms with, although not consistently, better performances than FD-ADMM-LB on average. Although this phenomenon can seem surprising after a look at situation 1, one can explain it with the fact that when the vector $\boldsymbol{w}$ is highly unbalanced (as it is the case when its values are uniformly drawn at random within $[.1, 1]$) the objective function $f^\alpha(\boldsymbol{w}, \cdot)$ obtains highly asymmetric structure. Indeed, the computation of the strong convexity modulus of $f^\alpha$ in [Allybokus et al.] in order to obtain a desirable initialization $\lambda_0$, shows that the factor $\sigma$ in Equation (10) in fact corresponds to the smaller strong convexity modulus of the functions $f_r^\alpha(w_r, \cdot)$, which is proportional to $w_r$. Not surprisingly then, this evaluation becomes poorer when the vector $\boldsymbol{w}$ becomes unbalanced. Thus, it is worth considering that an accurate penalty parameter tuning becomes more difficult when the weighted fairness function symmetry is poor. Nevertheless, our simulations suggest that initializing the reciprocal penalty parameter according to Equation (10) applied to our lower bound permits one to obtain a satisfactory performance of the FD-ADMM. We believe this scheme can be improved in order to tackle a potential performance issue under highly asymmetric realizations of the $(\boldsymbol{w}, \alpha)$-fair resource allocation problem characterized by a very low value of $\delta_w$.

## 6. CONCLUSION

We studied the structure of weighted $(\boldsymbol{w}, \alpha)$-fair allocation problems and proposed a lower bound that permits one to better understand the problem's features. The $(\boldsymbol{w}, \alpha)$-fair allocation can be lower bounded individually and locally (that is, each user, or request, has a minimal guarantied allocation that depends on its individual weight and that of a locally reduced subset of users). We compared experimentally our bound with the best bound available in the literature, and showed that we can provide consistent improvement in the case of high asymmetry of the capacity vector $\boldsymbol{c}$ (which may describe congested networks situations) or in the case of a suitable symmetry of the fairness measures (which may cover situations where the requests have balanced relative priorities). We believe that the bound derived in the present paper for general fairness concepts ($\alpha > 1$) can be further improved, and intend to soften its dependencies on the global minimum local midpoint value $p_{r_0}$. Our intuition suggests this would improve considerably the quality of our general bound. To demonstrate the utility of our derivation, we showed as an illustration how the introduction of this lower bound can remarkably improve the performances of an iterative distributed algorithm, the FD-ADMM, that solves the problem optimally, by a simple initialization of a penalty parameter. We also observed that the initialization scheme allows a remarkably satisfactory tuning of the FD-ADMM, and that this accuracy may impoverish as the asymmetry of the weighted problem strengthens. In the future, we envision to study this situation and strengthen our bound in order to possibly empower the initialization scheme, providing more robustness to the technique with respect to asymmetry.

## References


Réka Albert and Albert-László Barabási. 2002. Statistical mechanics of complex networks. *Rev. Mod. Phys.* 74 (Jan 2002), 47–97. Issue 1.

Zaid Allybokus, Konstantin Avrachenkov, Jérémie Leguay, and Lorenzo Maggi. 2017. Real-Time Fair Resource Allocation in Distributed Software Defined Networks. In *29th International Teletraffic Congress (ITC)*. 19–27.

Stephen Boyd, Neal Parikh, Eric Chu, Borja Peleato, and Jonathan Eckstein. 2011. Distributed optimization and statistical learning via the alternating direction method of multipliers. *Foundations and Trends in Machine Learning* 3, 1 (2011), 1–122.

Geoffroy de Clippel. 2007. An axiomatization of the Nash bargaining solution. *Social Choice and Welfare* 29, 2 (01 Sep 2007), 201–210.

Wei Deng and Wotao Yin. 2016. On the global and linear convergence of the generalized alternating direction method of multipliers. *Journal of Scientific Computing* 66, 3 (2016), 889–916.

Jonathan Eckstein and Dimitri P. Bertsekas. 1992. On the Douglas—Rachford splitting method and the proximal point algorithm for maximal monotone operators. *Mathematical Programming* 55, 1 (01 Apr 1992), 293–318.

BS He, Hai Yang, and SL Wang. 2000. Alternating direction method with self-adaptive penalty parameters for monotone variational inequalities. *Journal of Optimization Theory and applications* 106, 2 (2000), 337–356.

Frank P Kelly, Aman K Maulloo, and David KH Tan. 1998. Rate control for communication networks: shadow prices, proportional fairness and stability. *Journal of the Operational Research society* 49, 3 (1998), 237–252.

Jelena Marasevic, Clifford Stein, and Gil Zussman. 2016. A Fast Distributed Stateless Algorithm for alpha-Fair Packing Problems. In *43rd International Colloquium on Automata, Languages, and Programming (ICALP 2016) (Leibniz International Proceedings in Informatics (LIPIcs))*, Vol. 55. 54:1–54:15.

Jeonghoon Mo and Jean Walrand. 2000. Fair end-to-end window-based congestion control. *IEEE/ACM Transactions on Networking (ToN)* 8, 5 (2000), 556–567.

Daniel Pérez Palomar and Mung Chiang. 2006. A tutorial on decomposition methods for network utility maximization. *IEEE Journal on Selected Areas in Communications* 24, 8 (2006), 1439–1451.