



HAL
open science

An algorithm for improved proportional-fair utility for vehicular users: the multiple base-station case

Thi Thuy Nga Nguyen, Olivier Brun, Balakrishna Prabhu

► To cite this version:

Thi Thuy Nga Nguyen, Olivier Brun, Balakrishna Prabhu. An algorithm for improved proportional-fair utility for vehicular users: the multiple base-station case. 2020. hal-02892099v1

HAL Id: hal-02892099

<https://hal.science/hal-02892099v1>

Preprint submitted on 7 Jul 2020 (v1), last revised 24 May 2022 (v2)

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.

An algorithm for improved proportional-fair utility for vehicular users: the multiple base-station case*

T. T. Hang Nguyen^{1,2} Olivier Brun²
Balakrishna J. Prabhu²

¹Continental Digital Service in France, Toulouse, France
Thi.Thuy.Nga.Nguyen@continental-corporation.com

²LAAS-CNRS, Université de Toulouse, CNRS, Toulouse, France
{brun, balakrishna.prabhu}@laas.fr

Abstract

As the channel conditions experienced by vehicular users in cellular networks vary as they move, we investigate to which extent the quality of channel allocation could be improved by exploiting predictions on future data rates in non-stationary environments. Assuming mean future rates can be computed from Signal-to-Noise Ratio (SNR) maps, we propose an algorithm which predicts future throughputs over a short-term horizon at regular time intervals, and then uses this extra-knowledge for improved on-line channel allocation. The prediction of future throughputs is obtained by solving a relaxed version of the problem using a projected gradient algorithm. Using event-driven simulations, we compare the performance of the proposed algorithm against those of other channel allocation algorithms, including the Proportional Fair (PF) scheduler, which is known to be optimal in stationary environments, and the (PF)²S scheduler, which was devised for mobile nodes in non-stationary environments. The simulated scenarios include scenarios with multiple base stations and are based on realistic mobility traces generated using the road traffic simulator SUMO. Simulation results show that the proposed algorithm outperforms the other algorithms and that exploiting the knowledge of future radio conditions allows a significantly better channel allocation.

Index terms — proportional fairness, scheduling, mobility

1 Introduction

A central and challenging problem in cellular networks is channel allocation, that is, to decide which mobile user the base station (BS) should serve in each time

*A preliminary version of this paper appeared in [11].

slot. To this end, the BS gathers the channel state information (CSI) from users in order to know their radio conditions, which are mainly determined by their distances to the BS and by fading effects. As maximizing the overall throughput would lead to the starvation of distant users (those with the worst potential data rates), today cellular networks allocate the channel to the user with the highest potential rate proportionally to its time-average throughput¹. With this strategy, users with comparatively low allocated throughput are assigned a higher priority even when they are in worse channel conditions. This scheduling algorithm, which is known as the Proportional Fair (PF) scheduler, provides a fair and efficient sharing of bandwidth between users in the sense that it maximizes the aggregate logarithmic utility of obtained throughput in a fixed population of permanent users [1].

A number of studies have been devoted to the analysis of the performance of PF scheduling in wireless networks [2, 3, 4, 5, 6, 7], assuming either a static population of permanent users, or a dynamic setting in which random finite-size data transfers come and go over time. In both cases, it was shown that PF scheduling strikes a good balance between the overall network throughput and the degree of fairness among users. However, most of the literature is based on the assumption that users experience stationary channel conditions. This was partly motivated by the fact that a simple index-based allocation algorithm had been shown to be optimal for stationary channels [8]. Thus, even if they take into account the fast channel variations due to multi-path propagation, most studies ignore the variations of the channel conditions on slower time scale due to user mobility. Taking into account such slow fading effects is particularly important for vehicular users as the mean of the Signal-to-noise ratio (SNR) improves as a vehicle comes closer to a BS and then worsens as it moves away. Another usual assumption which is not realistic for vehicular users is the assumption of long sojourn times. Indeed, a vehicle typically stays in the coverage range of a BS for only a few minutes.

In this article, we investigate to which extent the quality of channel allocation could be improved by exploiting information on future radio conditions in non-stationary environments. Our main motivation comes from connected vehicles which will use cellular networks to exchange informations related to security and driving conditions with their environment. If the trajectory of a car is known or can be estimated from historical travel data and/or observations of the surrounding environment, then one can obtain good statistical predictions of the SNR that will be experienced by the car in the near future. In turn, these predictions could be used by the BS to achieve a channel allocation with a higher utility than that of the PF algorithm. In this paper, we propose a channel allocation policy exploiting this extra knowledge and evaluate the improvement in utility that it yields in non-stationary environments. Note that such an improvement in utility is not possible under the assumption of a stationary channel as knowing the car trajectory does not bring any new information on

¹The throughput is different from the data rate. While the latter is potential rate at which a user can be served, the former can be smaller since in some slots a user may not be served due to the presence of other users.

the future data rates.

The idea of using information on future radio conditions for channel allocation was already explored in [9]. It uses future information by looking at channel state of users in a few small time-slots. Different from their approach, we do not look at the predicted channel state in few time slots which may be different between users and difficult to predict correctly due to fast fading. Instead, we base our allocation on average rate the user will experience during the time interval this user stays inside the coverage range of the BS.

Another closely related work is [10] in which, using SNR maps obtained by measurements, the authors first show that PF scheduling may perform poorly in the presence of slow fading. They then propose a scheduling algorithm which is similar to PF in that the channel is allocated to the user with the highest potential rate proportionally to its total throughput. This new algorithm, which is called $(PF)^2S$ differs however from PF in that the total throughput includes an estimation of the future throughput whereas PF considers only the already allocated throughput. In order to estimate the future throughput, the authors proposed three methods: round-robin, blind estimation, and a local search heuristic. It was shown that even with this rough estimation of the future throughput, this new index leads to an improved utility compared to the PF algorithm in non-stationary environments. The channel allocation policy proposed in this paper is similar to the $(PF)^2S$ scheduling policy except that we use a different method for estimating future throughputs of vehicles. For the purposes of numerical comparisons, we shall assume in this paper that $(PS)^2S$ uses the round-robin policy. It was stated in [10] that, out of the three estimation methods, round-robin is the most robust to prediction errors.

1.1 Contributions

We present a heuristic algorithm for non stationary channels that improves the total utility of users compared to the PF and the $(PF)^2S$ algorithms. Our heuristic is similar to the $(PF)^2S$ algorithm, except that instead of computing an estimation of future throughput from a round-robin allocation, we compute it as the solution of a utility maximization problem over a short-term horizon assuming that the means of the future data rates are known over this short horizon.

The original utility maximization problem being computationally complex, we employ three techniques to obtain a lower complexity heuristic: *(i)* we relax the integer constraints of the original problem; *(ii)*, we shorten the time horizon over which the problem is solved; and *(iii)* we compute the solution over macroscopic time slots instead of microscopic ones that helps the algorithm run in real time. The relaxation turns the problem into a convex one and allows for its efficient resolution. Shortening of the time horizon and solving over macroscopic slots reduces the number of variables in the problem and decreases the computation time.

We compare the performance of the proposed algorithm against those of other channel allocation algorithms using event driven simulations. The simu-

lated scenarios include scenarios with multiple base stations and are based on realistic mobility traces generated using the open-source road traffic simulator SUMO with vehicles moving at either equal or different speeds. Simulation results show that the proposed algorithm outperform other algorithms and that exploiting the knowledge of future radio conditions allows a significantly better channel allocation.

A preliminary version of the paper limited to scheduling in a single base station setting and not including experiments with SUMO appeared in ASMTA2019 [11].

1.2 Organisation

In Section 2, we state the assumptions and define the objective function. In Section 3, we give some background on PF and (PF)²S algorithms. In Section 4, we present our heuristic for improving the utility based on estimations of future average data rate. Section 5 contains the numerical results for scenarios with homogeneous as well as heterogeneous vehicles. Finally, we end the paper in Section 6 with a few open problems.

2 Problem formulation

We consider a geographical region with a network of roads that is served by a set of M base stations $\{B_1, B_2, \dots, B_M\}$. The region is partitioned into M non-overlapping sub-regions each of which represents the coverage area of a base station. Users (vehicles, bicycles, pedestrians, etc) enter the network, move along different routes, and leave the network. Figure 1 shows an area within the city of Toulouse which will be later used in the numerical experiments. In the figure, the width of the box is approximately 1 km, and the height is around 0.65 km. The data for BS location can be found on the website² of the French Frequency Agency (ANFR), which manages all radio frequencies in France.

Every $\delta = 2$ ms each BS has to decide which user to serve in a decentralized fashion. We shall assume that the data rate received by a user depends on the distance between the BS and that user. The data rate depends upon the SNR which itself can vary along the road. In our numerical experiments, we assume that the data rate decays exponentially as in formula (1) below

$$r^m(x) = \begin{cases} 0 & \text{if } d(x, B_m) > d_m, \\ 1 + \kappa e^{-d(x, B_m)/\sigma} & \text{otherwise,} \end{cases} \quad (1)$$

where x is the position of the user, B_m is the position of BS m , $d(x, B_m)$ is the Euclidean distance between positions B_m and x , and κ and σ are adjustable parameters. The scheduling algorithm we propose does not however require this assumption to work.

²<https://data.anfr.fr/anfr/portail>

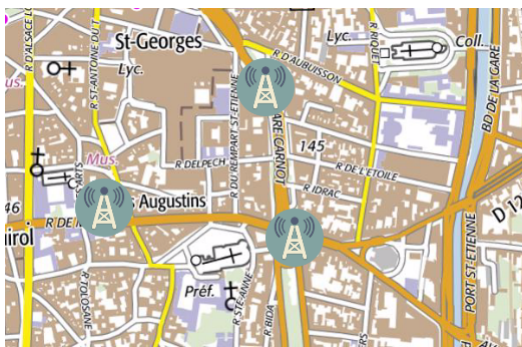


Figure 1: A selected area of Toulouse which is covered by three BSs (LTE1800) of the French mobile network operator Free Mobile.

Denote by T the time horizon over which the scheduling decisions are made, and let K be total number of users who pass through the considered region during that time. For simplicity, we assume that T is a multiple of δ . Our objective is to achieve the proportional-fairness between users, which is described by the following optimization problem (see, e.g., [2, 10, 12]):

$$\left\{ \begin{array}{l} \text{maximize} \quad O(\alpha) = \sum_{i=1}^K \log \left(\sum_{m=1}^M \sum_{j=1}^T \alpha_{ij}^m r_{ij}^m \right) \\ \text{subject to} \quad \sum_{i=1}^K \alpha_{ij}^m = 1, \quad j = 1, \dots, T, \quad m = 1, \dots, M, \\ \sum_{m=1}^M \alpha_{ij}^m \leq 1, \quad j = 1, \dots, T, \quad i = 1, \dots, K, \\ \alpha_{ij}^m \in \{0, 1\}, \quad j = 1, \dots, T, \quad i = 1, \dots, K, \quad m = 1, \dots, M. \end{array} \right. \quad (\text{I})$$

where:

- α_{ij}^m is a binary decision variable which is equal to 1 if the channel of BS m is allocated to user i at time j , and 0 otherwise.
- r_{ij}^m is the potential data rate of user i at time j if it served by BS m . This potential data rate is given by $r_{ij}^m = r^m(x_{ij})$, where x_{ij} is the position of the user at time j and the rate function $r^m(x)$ is defined in formula (1).

Constraints $\sum_{i=1}^K \alpha_{ij}^m = 1$ imply that each BS serves exactly one user at each time j . Constraints $\sum_{m=1}^M \alpha_{ij}^m \leq 1$ imply that each user i is served by at most one base station at each time j . Finally, the last constraints $\alpha_{ij}^m \in \{0, 1\}$ imply that a feasible solution is a binary vector α . To make the problem easier to solve, we will remove the constraints $\sum_{m=1}^M \alpha_{ij}^m \leq 1$ by assuming that a user can only be served by the closest BS.

3 Existing Algorithms

In this section, we present some of the existing heuristics for channel allocation. These heuristics will be later compared with the heuristics we propose in this paper.

3.1 Greedy allocation

In the greedy scheme, the channel is always allocated to the vehicle with the maximum rate, that is, at each time-slot j the channel of BS m is allocated to a vehicle $i_m^* \in \operatorname{argmax}_i(r_{i,j}^m)$.

3.2 Proportional Fair (PF) allocation

Remark that problem (I) is a discrete problem. Even though the number of options is finite, it is NP-hard to find the optimal solution (see, e.g., [10]). Nevertheless, a simple heuristic, called PF-EXP [8], is known to be optimal when the number of users is fixed and that the data rates $r_{i,j}^m$ are time stationary and ergodic, that is, there is no correlation between $r_{i,j}^m$ and $r_{i,j+1}^m$.

The PF algorithm chooses the user with the highest ratio of the current rate to the observed throughput, that is, it chooses the user i who maximizes the ratio $r_{i,j}^m/A_i(j-1)$, where

$$A_i(j) = A_i(0) + \sum_{t=1}^j \sum_{m=1}^M r_{i,t}^m \alpha_{i,t}^m,$$

is the total allocated rate to user i up to time j ($A_i(0)$ is the initial value for each user). In the long-run when T goes to ∞ , this algorithm was shown to be optimal for a stationary and ergodic channel and for a fixed number of users [8].

As already mentioned, the stationarity assumption is not necessarily true for road traffic when all users always move instead of resting in the same place. As can be seen in Fig. 1, when users move on a given path, their rate can vary with the distance to the BSs. Thus, the rate process observed by vehicles need not be stationary, and the PF-EXP algorithm need not be optimal for vehicles moving in a network.

3.3 Predictive Finite-horizon PF Scheduling ((PF)²S)

In [10], a modified PF algorithm based on predicted future rate was proposed. This algorithm works as follows:

- It predicts future data rates $\hat{r}_{i,j}^m$ of cars in every future slot,
- it estimates future channel allocations $\hat{\alpha}^m$ based on the data rate predictions. As mentioned in Sec. 1, the estimations can be computed using either a round-robin policy, a blind estimation, or a local-search method. It is stated in [10] that, out of these three, round-robin is more robust

to prediction errors. Given this, we shall use *Round Robin Estimation* (RRE) as the estimation policy for (PS)²S in the numerical comparisons. As a reminder, RRE assumes that future time slots are allocated in a round-robin manner and each user receives an equal number of slots.

- for each time slot j , the BS m chooses the user who maximizes $M_{i,j}^m$, where

$$M_{i,j}^m = \frac{r_{i,j}^m}{\sum_{t=1}^{j-1} \alpha_{i,t}^m r_{i,t} + \hat{\alpha}_{i,j}^m r_{i,j} + \sum_{t=j+1}^T \hat{\alpha}_{i,t}^m \hat{r}_{i,t}^m}. \quad (2)$$

The index $M_{i,j}^m$ looks similar to that of the PF-EXP algorithm but includes the future allocation. It is related to the gradient of the utility function in (I). In the case of one BS (so that we can omit the index m), provided the future channel allocations $\hat{\alpha}$ can be predicted correctly, an optimal solution to problem (I) can be obtained, as stated in Proposition 1.

Proposition 1. *If there exist α^* satisfying $\alpha_{i^*,j}^* = 1$ and $\alpha_{i,j}^* = 0, \forall i \neq i^*$, where*

$$i_j^* \in \arg \max_{i \in \{1,2,\dots,K\}} \frac{r_{i,j}}{\sum_{t=1}^{j-1} \alpha_{i,t}^* r_{i,t} + \alpha_{i,j}^* r_{i,j} + \sum_{t=j+1} \alpha_{i,t}^* r_{i,t}}, \quad (3)$$

then α^* is the optimal solution of problem (I).

Proof. See Appendix B. □

Note that Condition (3) is a sufficient condition for α^* to be an optimal solution of problem (I), but not a necessary condition.

In the next section, we present our heuristic. The motivation for the heuristic comes from the observation that the formula of (PF)²S looks like one-step of the gradient descent with starting point chosen according to the round robin policy when the Round Robin Estimation is used. We may expect to get a better allocation if we do more iterations instead of only one, ensuring that in every iteration the allocation is in the feasible set. To do this we employ a projected gradient algorithm, as described in the next section.

4 Projected gradient approach

We shall assume that each BS allocates the channel independently, that is, in a decentralized manner and without coordination with the other BSs. The channel allocation is done by the BS by taking into account the future data rates of the users currently attached to this BS. Since each BS decides independently, we omit the index m of the BS for simplicity.

We propose two heuristic algorithms, Short Term Objective 1 (STO1) and Short Term Objective 2 (STO2), which are presented in the following. The two heuristics use a different method (to be explained below) for estimating the future throughput than the round-robin used in the (PF)²S algorithm. This

estimate is based on optimizing the objective with the future mean channel gains as an estimate for the actual realizations. This is similar in spirit to Stochastic Model Predictive Control [13]. The two heuristics differ in the time-scale on which updated future information is used as well as in the dimension of the optimization problem solved at each decision epoch.

Before describing the two heuristics, we explain the ideas common to them. The first step is to relax the integer constraints on the allocation variables in optimization problem (I), so as to obtain the following convex optimization problem

$$\left\{ \begin{array}{l} \text{maximize} \quad O(\alpha) = \sum_{i=1}^K \log \left(\sum_{j=1}^T \alpha_{ij} r_{ij} \right) \\ \text{subject to} \quad \sum_{i=1}^K \alpha_{ij} = 1, \quad j = 1, \dots, T, \\ \alpha_{ij} \in [0, 1], \quad j = 1, \dots, T, \quad i = 1, \dots, K, \end{array} \right. \quad (\text{II})$$

which is very similar to the original problem, except that α_{ij} can now be non-integer in $[0, 1]$. The relaxed problem (II) can be solved efficiently using the projected-gradient algorithm based on the formula for the projection on a simplex given in [14], as described below.

Denote by $D = \left\{ \alpha \in [0, 1]^{K \times T} : \sum_{i=1}^K \alpha_{ij} = 1, \quad j = 1, 2, \dots, T \right\}$ the feasible set of the relaxed problem. The set D is not a simplex, therefore we cannot apply directly the algorithm in [14]. However, for every j the feasible set of allocations is indeed a simplex. We can therefore obtain a projection on D by projecting independently on simplexes corresponding to each of the time-steps. The procedure for computing the projection Π_D on the set D is formalized in Appendix A.

The projected gradient algorithm then works as follows. Starting from an arbitrary initial solution $\alpha^0 \in D$, the algorithm computes at each iteration $n = 1, 2, \dots$ a new feasible solution using the formula

$$\alpha^{n+1} = \Pi_D(\alpha^n + \epsilon_n \nabla O(\alpha^n)), \quad (4)$$

where $\nabla O(\alpha^n)$ is the gradient of the objective function at iteration n and $\epsilon_n \in (0, 1)$ is the step size at that iteration. A new feasible solution is computed until convergence is reached. In our numerical examples, we have however limited the number of iterations to 20.

Denote by $\tilde{\nabla} O(\alpha) = \Pi_D(\alpha + \epsilon \nabla O(\alpha)) - \alpha$ with the step size $\epsilon \in (0, 1)$ small enough. Proposition 2 below says that if the iterations (4) converge, then the resulting allocation is optimal.

Proposition 2. *If $\alpha^* \in D$ and $\tilde{\nabla} O(\alpha^*) = 0$ then α^* is the optimal value of the relaxed problem (II).*

Proof. See Appendix B. □

Solving (II) using the projected gradient algorithm (4) requires the knowledge of all the future arrivals which may not be available. Further, the horizon T could be potentially large (tens of minutes giving roughly of the order of 300000 small slots). This means the BS will have to solve a very high dimensional problem every 2 ms.

For the heuristics, we circumvent these two issues as follows. First, we solve (II) for only cars that are actually present in the coverage range and ignore the future arrivals. Second, we reduce the computational complexity in two ways: (i) we solve the problem over a shorter horizon; and (ii) we compute the future allocations on a larger time-scale rather than the short time-scale of channel allocation slots δ , which is usually in the order of a few milliseconds. The distance travelled in δ ms by a vehicle is typically too small to observe large changes in the mean channel conditions. Therefore, we define the notion of a big-slot over which there is noticeable change in the mean channel conditions. For example, a big-slot can be $500 \times \delta$, giving a value of 1 second for the big-slot when $\delta = 2$ ms. The exact value of a big-slot is an adjustable parameter that can be set by the system designer.

Next, we describe the two heuristics.

4.1 Projected gradient short term objective algorithm (STO1)

Let Δ be the size of the big-slot in absolute time units and let $m = \Delta/\delta$ be the number of small slots in a big-slot. If \bar{r}_{ij} is the average rate in slot j for user i , then $\bar{\rho}_{i\tau} = \sum_{j=(\tau-1)m+1}^{\tau m} \bar{r}_{ij}$, is the total average data rate that user i will get in big-slot τ . Define $\bar{\alpha}_{i\tau}$ to be the allocation in future big slot τ . These allocations can be interpreted as the fraction of small slots that user i will be allocated in the big-slot τ .

In small time slot t , let $a_i(t) = \sum_{j=1}^t \alpha_{ij} r_{ij}$ be the cumulative allocated rate of user i until time slot t , and $K(t)$ be the number of users inside the coverage range.

The STO1 heuristic works in two steps. At each small slot t , it first solves the allocation for the current small slot and the future big-slots. In the second step, it allocates the channel to the user with the largest fractional allocation for the current slot. These steps are described below:

- **Step 1**– solve the following optimization problem over a short-term horizon of J big-slots using the projected gradient algorithm:

$$\left\{ \begin{array}{l} \text{maximize} \\ \text{subject to} \end{array} \right. \left\{ \begin{array}{l} \sum_{i=1}^{K(t)} U_i \\ \sum_{i=1}^{K(t)} \alpha_{it} = 1, \\ \sum_{i=1}^{K(t)} \bar{\alpha}_{i\tau} = 1, \quad \tau = 1, \dots, J \\ \alpha_{it}, \bar{\alpha}_{i\tau} \in [0, 1], \quad \tau = 1, \dots, J, \quad i = 1, \dots, K, \end{array} \right. \quad (\text{III})$$

where

$$U_i = \log \left(a_i(t-1) + \alpha_{it} r_{it} + \sum_{\tau=1}^J \bar{\alpha}_{i\tau} \bar{\rho}_{i\tau} \right).$$

The decision variables in Problem (III) are the allocations in the current small slot, α_{it} , and the allocations in the future big-slots, $\bar{\alpha}_{i\tau}$. Since the future allocations are only computed on the time-scale of big-slots, there is reduction of factor m in the number of variables in (III).

- **Step 2** – allocate the channel to the user who has the *largest allocation computed by* $(\alpha_{it})_{i=1, \overline{K(t)}}$ that is one user which is $\arg \max_i \alpha_{it}$.

The complexity of numerically optimal α computed in step 2 is equal to $20(J+1)\bar{K} \log(\bar{K})$ where 20 is the number of iteration steps of projected gradient in Step 1, \bar{K} is average number of users inside the coverage range, J is the number of big slots.

4.2 Projected gradient short term objective algorithm 2 (STO2)

In STO2, we further reduce the complexity by recomputing the future allocations only at the beginning of a big-slot. The future allocation thus computed is then used until the end of this big-slot. If one new user arrives to the system in the middle of big-slot, we just ignore it for this big-slot and wait until the beginning of next big-slot to update the state. Once the allocations for future big-slots are computed, then in every small slot of this big-slot, we apply an index-based policy as in (2).

The steps for STO2 are:

- **Step 1** – In each big slot τ , solve the following problem using projected gradient:

$$\left\{ \begin{array}{l} \text{maximize} \quad \sum_{i=1}^{K(\tau)} U_i \\ \text{subject to} \quad \sum_{i=1}^{K(\tau)} \bar{\alpha}_{i\tau} = 1, \quad \tau = 1, \dots, J, \\ \bar{\alpha}_{i\tau} \in [0, 1], \quad \tau = 1, \dots, J, \quad i = 1, \dots, K, \end{array} \right. \quad (\text{IV})$$

where

$$U_i = \log \left(a_i((\tau-1)m) + \sum_{\tau=1}^J \bar{\alpha}_{i\tau} \bar{\rho}_{i\tau} \right).$$

Here $a_i((\tau-1)m)$ is the total data rate received by user i up to big slot τ . The other quantities are the same as for algorithm STO1.

- **Step 2** – Inside a big-slot, in each small slot j , compute M_{ij} as in (2) where the future allocation \hat{a} is the solution \bar{a} of (IV).

Note that Step 1 in STO2 is computed only once every big-slot unlike m times in every big-slot as in STO1. By doing this, we further reduce the number of computations almost by a factor of m times since we calculate \bar{a} in each big-slot only.

5 Numerical results

We now compare the utility of the proposed heuristics with the PF-EXP, (PF)²S and a greedy algorithm. For the (PF)²S the future allocation was done using the round robin algorithm.

Denote by

$$O^A = \sum_{i=1}^K \log \left(\sum_{j=1}^T \alpha_{ij}^A r_{ij} \right),$$

the total reward of algorithm A and by $\bar{O}^A = \frac{1}{K} O^A$ its average reward over K users. Given two algorithms A and B , the ratio between A and B equals $\exp(\bar{O}^A - \bar{O}^B)$. The percentage of improvement of algorithm A over B is computed as $(\exp(\bar{O}^A - \bar{O}^B) - 1) \cdot 100\%$.

Due to the logarithm in the objective function, taking a different unit of measure for the rate will give a different percentage of improvement between algorithms. Although logarithm is an increasing function, we can know which algorithm is better than the other, but we will not get a consistent percentage of improvement across different units of measure. Therefore, by taking the difference as above we construct a consistent criterion for comparison.

5.1 One road network

In the first set of simulations, there is only one base station and one straight road. The road length is taken to be $L = 1000$ m with 0 at the leftmost edge. The closest point on the road to the BS is at $x = 500$ m. The data rate at position x along the road is given by:

$$r(x) = \eta \cdot (1 + \kappa \exp(|x - 500|/\sigma)), \quad (5)$$

where $\kappa \geq 0$ is a real number and η is uniform random variable whose range will be in $[0.7, 1.3]$ unless stated otherwise. A sample path of $r(x)$ is shown in Fig. 2,3. This function has the highest mean at the mid-point of the segment and the lowest mean at the two end points. We emphasize the algorithm itself is independent of the rate function. We chose the above rate function for convenience.

The time horizon T was 4,000,000 small time slots which corresponds to 8000 seconds (slightly more than two hours). The big slot length Δ for our projected gradient short term objective algorithm was taken as 1 second or equivalently 500 small time slots.

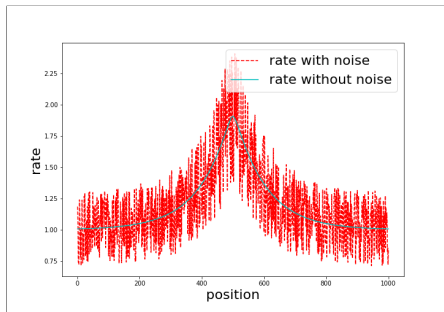


Figure 2: $\kappa = 1$.

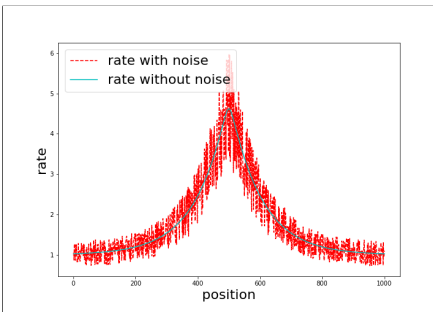


Figure 3: $\kappa = 4$.

Figure 4: Sample path of data rate at various positions along the road. $\sigma = 100$, and $\eta \in [0.7, 1.3]$.

5.1.1 Homogeneous vehicle velocities

First, we show the results when all vehicles move with the same velocity which is taken to be $v = 25$ m/s. That is, there are $N = 20,000$ spatial small slots in the coverage range and $J = 40$ seconds. A new car enters through the left edge in every second with probability p .

Figure 5 shows the average utility obtained by a vehicle for each of the four algorithms as a function of the arrival probability p . Figure 6 shows the percentage of improvement of the three other algorithms compared to PF-EXP. The proposed algorithm does better than PF-EXP and more importantly better than (PF)²S. Although, we have shown the greedy algorithm for comparison, we emphasize that greedy is not practically implemented because it can be very unfair to users that have heterogeneous rates. In the simulated scenario, all vehicles move along the same road and observe statistically identical but position-dependent radio conditions during their stay. These conditions are rather favorable for the greedy algorithm.

5.1.2 Comparison with the upper bound

Next, again for homogeneous velocities, we also include the solution of the relaxed problem (II) but for a smaller road length and shorter horizon because it is computationally expensive. The parameters for this setting are: $L = 100$ m, $J = 40$ s, $T = 500$ s, and the other parameters are the same as in the homogeneous case. We assume that the relaxed algorithm knows the future arrivals as well as the future data rates exactly whereas the other algorithms do not know this information. The solution to the relaxed problem gives an upper bound on the optimal solution of the original problem (I).

Figures 7 and 8 plot the average reward per car and percentage improvement for the five algorithms with respect to PF-EXP. It can be observed that the proposed algorithm is quite close to the upper bound in this scenario.

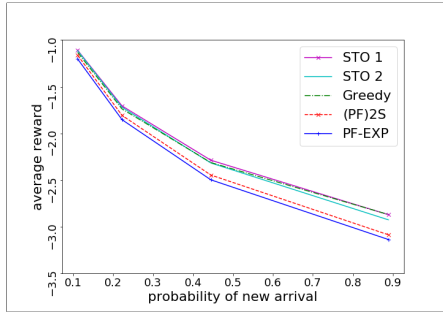


Figure 5: Average reward per car. Homogeneous velocities.

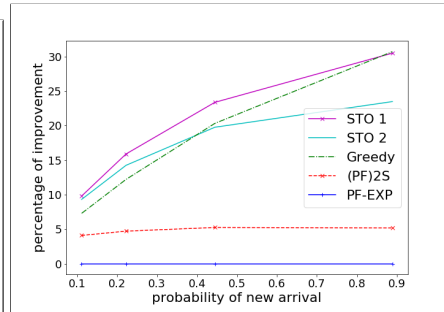


Figure 6: Percentage of improvement over PF-EXP. Homogeneous velocities.

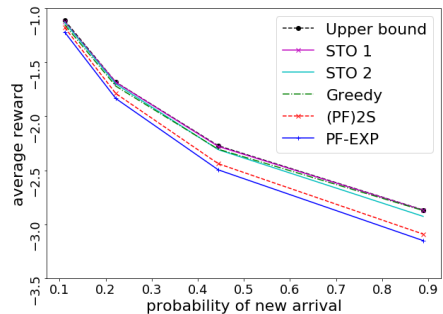


Figure 7: Average reward per car. Includes the upper bound from the solution of (II). Small setting of homogeneous velocities.

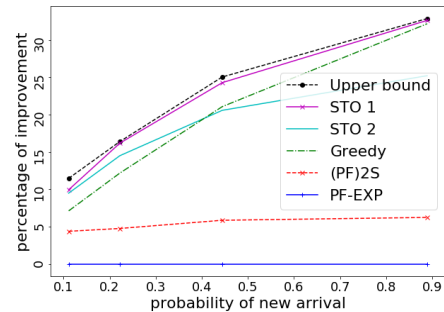


Figure 8: Percentage of improvement over PF-EXP. Small setting of homogeneous velocities.

In the following, unless otherwise specified, the parameters are chosen as follows: $\kappa = 4, \sigma = 100$, big slot $\Delta = 1s$ and the short-term horizon J is the maximal remaining staying time of the users that are currently inside the system. We calculate the allocation plan every one second. From now on, we do not compare STO 1 because STO 1 takes much longer to run and may not be computationally interesting on small time-scales. Also, we also do not show the performance of greedy here since some users may starve in a greedy allocation leading to a value of $-\infty$.

5.2 Network simulation with SUMO

Simulation of Urban MObility application (SUMO) [15] is an open source software designed for simulating mobility of vehicles in large traffic networks. One of the features of this simulator is that we can import maps of different cities and simulate realistic mobility traces. We use this application to simulate the

complex driving dynamic systems in a specific region of Toulouse city to have an objective comparison of our heuristics against existing algorithms in realistic scenarios.

The performance evaluation of heuristics is done in two steps: in the first step SUMO is used for generating the mobility traces of vehicles. These traces are then fed to a Python script which implements the different heuristics and computes the value of the objective function.

5.2.1 A simple network with 1 BS

Let us consider the network presented in in Figure 9. There are two classes of users: one that arrives from A then moves along the long road to B and D (the blue one), and another one that arrives from A then moves to B and then to C (the red one). If we apply the greedy heuristic in this situation, then many users of the second class are never allocated the channel. This is the reason we do not show performance of the greedy algorithm for this scenario. Figure 10 shows the numerical results for this case. In this scenario, it was observed that PF-EXP always gives priority to the new arrivals no matter what the initial value is. This leads to a higher sub-optimality of PF-EXP since the other heuristics focus on users that are closer to the base station and have a higher quality channel.

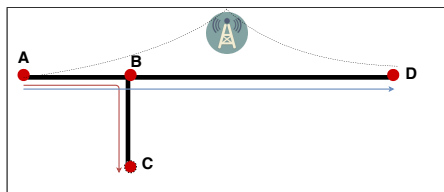


Figure 9: Utility Comparison: STO, (PS)²S and PF-EXP.

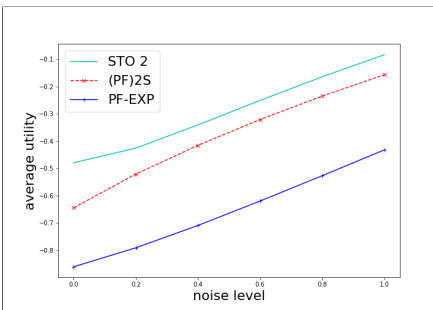


Figure 10: Utility Comparison: STO, (PS)²S and PF-EXP.

5.2.2 Place Wilson (Toulouse) scenario with 2 BSs

In this scenario, we evaluate the algorithms on users moving in the Place Wilson area of Toulouse with two BSs as shown in Fig. 11. The average utilities of the different heuristics are shown in Fig. 12. The various parameters for the rate function are the same as those indicated at the beginning of this section. It took 219 seconds, 229 seconds, 433 seconds and 833 seconds respectively to run greedy, PF-EXP, (PS)²S and STO2 for simulating 1.07 hours of traffic with 483 users (including cars, buses, and bicycles). The staying times of the users varied from 2s to 361s. We do not show greedy in the utility comparison since there were several starving users in this case. As expected, there is a trade-off

between the quality of the solution and the computation time. STO2 takes longer to solve but gives a better allocation.

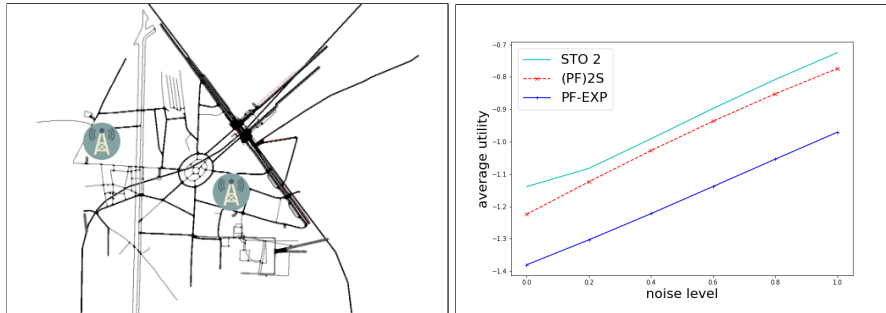


Figure 11: Place Wilson, Toulouse Figure 12: Utility Comparison: STO2, with 2 Free Mobile BSs 4G. (PS)²S and PF-EXP.

Now, we change some of the parameters to see how the performance of the heuristic is influenced by these parameters.

Figure 13a, 13b, 13c plot the average utilities for different values of κ with the same H and Δ . The gap between STO2 and (PS)²S become larger when κ increases.

Figure 14a, 14b, 14c and 14d illustrate the average utilities for different values of J with same κ, Δ . Remark that we assume (PS)²S and STO2 use the same information, so in (PS)²S the future information is estimated until J as well. It is seen that the more information we have, the better (PS)²S and STO2 perform.

Figure 15a, 15b, 15c illustrate for different values of big-slot Δ with the same values of J and κ . The performance of STO2 is almost the same for these different values of Δ but the running time is significant faster.

5.2.3 Jardin de Plantes (Toulouse) scenario with 4 BSs

In the final set of simulations, we take another area of Toulouse called Jardin des Plantes with four BSs as shown in Fig. 16. The average utilities for this scenario are plotted in Fig. 17. It took 976 seconds, 1009 seconds, 1502 seconds and 3403 seconds to run greedy, PF-EXP, (PS)²S and STO2 for a 1.05 hours of traffic with 740 users (including cars, buses, motorbikes, bicycles and pedestrian). Again, we do not show greedy in the utility comparison since there are several starving users in this case.

6 Conclusions and future work

We proposed two heuristics that use future mean channel gain information to improve the utility of users in cellular networks. In order to reduce the computational complexity, they solve the problem over a shorter time horizon as well

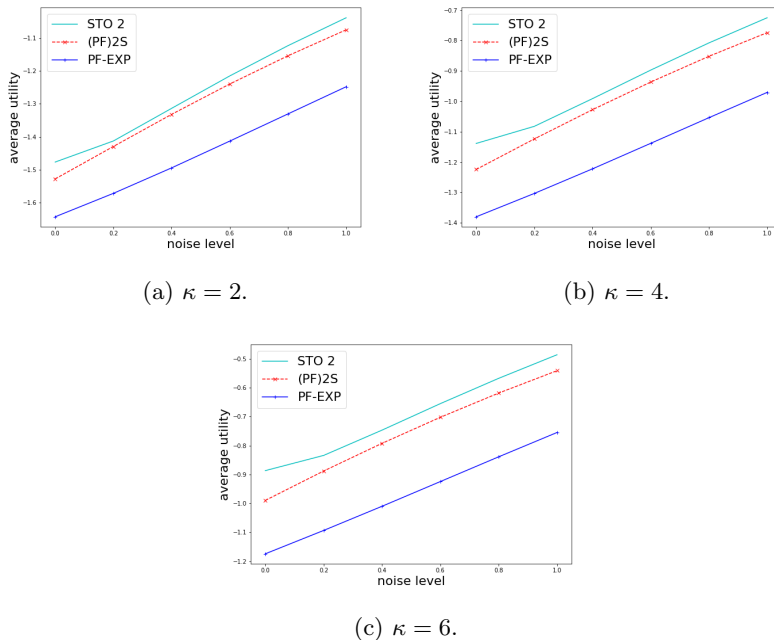


Figure 13: Place Wilson, utility comparison for different κ .

as on a larger time-scale. It was shown on numerical experiments carried out on traces generated from realistic mobility patterns that these heuristics give better utility compared to PF as well as (PS)²S algorithms.

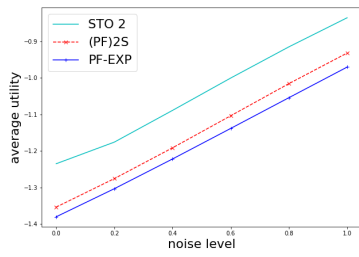
There are several directions in which this work can be taken. One avenue is to implement a centralized and coordinated version of these heuristics. Further, it will be interesting to design heuristics for networks in which there are a fractions of users whose future channel gain information is not known. These could be users who do not share their mobility information or users like pedestrians whose mobility can be random and hence not known exactly.

Other directions of research include designing heuristics for different utility functions and QoS requirements such are latency and jitter. On the analytical side, obtaining heuristics with guaranteed sub-optimality bounds will be worth investigating.

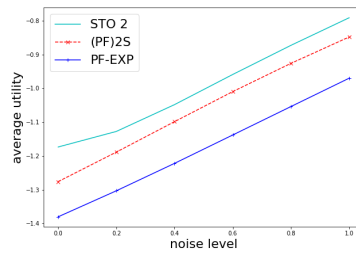
A Projection on feasible set D

The set D is a Cartesian product of J simplexes: $D = D_1 \times D_2 \cdots \times D_J$ where

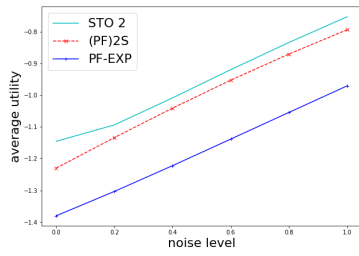
$$D_j = \{a_j = (\alpha_{ij})_{i=1, \overline{K}} \in [0, 1]^K, \sum_{i=1}^K \alpha_{ij} = 1\}$$



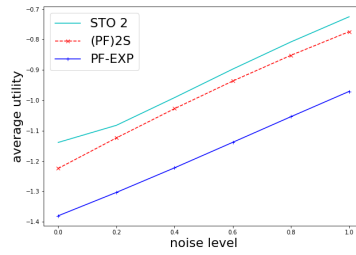
(a) $J = 20s$.



(b) $J = 60s$.

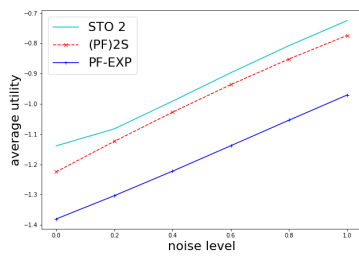


(c) $J = 120s$.

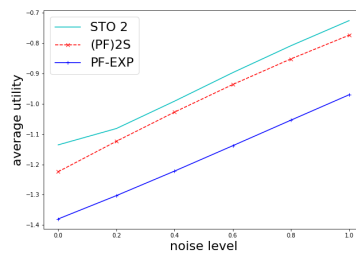


(d) $J =$ maximum remain time in term of big-slot of all users inside the system.

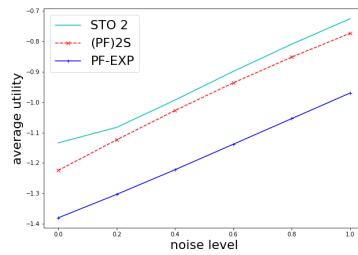
Figure 14: Place Wilson, utility comparison for different short time horizon J . Here we assume that (PS)²S and STO2 use the same information, so in (PS)²S the future information is estimated until J as well.



(a) $\Delta = 2s$ (833s to run STO2).



(b) $\Delta = 4s$ (568s to run STO2).



(c) $\Delta = 6s$ (438s to run STO2).

Figure 15: Place Wilson, utility comparison for different Δ . The performance are almost the same, but the running time is much faster when increasing big-slot.

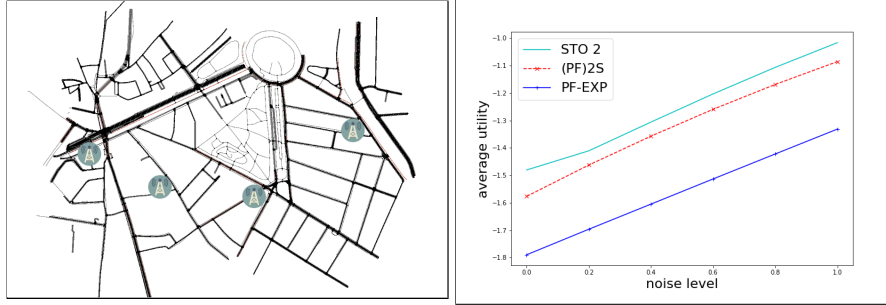


Figure 16: Jardin de Plantes, Toulouse Figure 17: Utility Comparison: STO2, with 4 BSs (Free and SFR) type 4G. (PS)²S and PF-EXP.

for all $j = 1, 2, \dots, J$.

Since $(D_j)_j$ are simplexes, we can compute the projection on D_j following [14]. The projection on D can thus be computed by the simple following lemma:

Lemma 1. *If $Y = (y_{ij})_{i=\overline{1,K},j=\overline{1,J}} \in \mathbb{R}^{K \times J}$, then*

$$\Pi_D(Y) = \Pi_{D_1}(Y_1) \times \Pi_{D_2}(Y_2) \times \dots \times \Pi_{D_J}(Y_J),$$

where $Y_j = (y_{ij})_{i=\overline{1,K}}$.

Proof. (of the lemma 1) Denote by $Z = \Pi_{D_1}(Y_1) \times \Pi_{D_2}(Y_2) \times \dots \times \Pi_{D_J}(Y_J)$. It is easy to check that for any $X \in D^{K \times J}$ then $\langle Y - Z, X - Z \rangle \leq 0$. \square

As described in [14], the complexity of finding Π_{D_j} is equal to $K \log(K)$ by observation in practice, and equal to $O(K^2)$ in the worst case. Therefore the complexity of finding projection on $D = D_1 \times D_2 \dots \times D_J$ is equal to $JK \log(K)$ in practice.

B Proofs

Proof. (proof of Proposition 2) The optimal is obtained by proving that for any $\alpha \in D$,

$$\nabla O(\alpha^*)(\alpha^* - \alpha) \leq 0.$$

From lemma 1, it follows that it is sufficient to prove the above property on D_1 . Assuming O is convex function on D_1 , we shall prove that if $\alpha^* = (\alpha_i^*)_{i=1,\dots,K} \in D_1$ satisfies

$$\Pi_{D_1}(\alpha^* + \epsilon \nabla O(\alpha^*)) = \alpha^* \tag{6}$$

where ϵ positive, then

$$\nabla O(\alpha^*)(\alpha^* - \alpha) \geq 0, \text{ for any } \alpha \in D_1$$

i.e, α^* is global optimal of O . Indeed, without loss of generality, we assume that

$$\alpha_1^* + \epsilon \frac{\partial O}{\partial \alpha_1^*} \geq \alpha_2^* + \epsilon \frac{\partial O}{\partial \alpha_2^*} \geq \dots \geq \alpha_M^* + \epsilon \frac{\partial O}{\partial \alpha_M^*} \geq \dots \geq \alpha_K^* + \epsilon \frac{\partial O}{\partial \alpha_K^*}$$

where M is the largest index such that

$$\frac{1}{M} \sum_{i=1}^M (\alpha_i^* + \epsilon \frac{\partial O}{\partial \alpha_i^*} - 1) \leq \alpha_M^* + \epsilon \frac{\partial O}{\partial \alpha_M^*}.$$

Denote by $\tau = \frac{1}{M} \sum_{i=1}^M (\alpha_i^* + \epsilon \frac{\partial O}{\partial \alpha_i^*} - 1)$, by proposition 10 in [14] we have:

$$\Pi_{D_1}(\alpha^* + \epsilon \cdot \nabla(\alpha^*)) = (\alpha_1^* + \epsilon \frac{\partial O}{\partial \alpha_1^*} - \tau, \alpha_2^* + \epsilon \frac{\partial O}{\partial \alpha_2^*} - \tau, \dots, \alpha_M^* + \epsilon \frac{\partial O}{\partial \alpha_M^*} - \tau, 0, \dots, 0).$$

Using (6) to compare term by term we get:

1. $\alpha_{M+1}^* = \dots = \alpha_K^* = 0$,
2. $\alpha_{M+1}^* + \epsilon \frac{\partial O}{\partial \alpha_{M+1}^*} \leq \tau, \dots, \alpha_K^* + \epsilon \frac{\partial O}{\partial \alpha_K^*} \leq \tau$. Now, from the first item we have $\alpha_{M+1}^* = \dots = \alpha_K^* = 0$. It implies $\epsilon \frac{\partial O}{\partial \alpha_{M+1}^*} \leq \tau, \dots, \epsilon \frac{\partial O}{\partial \alpha_K^*} \leq \tau$,
3. $\epsilon \frac{\partial O}{\partial \alpha_1^*} = \dots = \epsilon \frac{\partial O}{\partial \alpha_M^*} = \tau$.

Thus,

$$\begin{aligned} \epsilon \nabla O(\alpha^*)(\alpha^* - \alpha) &= \sum_{i=1}^K \epsilon \frac{\partial O}{\partial \alpha_i^*} (\alpha_i^* - \alpha_i) \\ &= \sum_{i=1}^M \epsilon \frac{\partial O}{\partial \alpha_i^*} (\alpha_i^* - \alpha_i) + \sum_{i=M+1}^K \epsilon \frac{\partial O}{\partial \alpha_i^*} (\alpha_i^* - \alpha_i), \\ &= \sum_{i=1}^M \tau (\alpha_i^* - \alpha_i) + \sum_{i=M+1}^K \epsilon \frac{\partial O}{\partial \alpha_i^*} (\alpha_i^* - \alpha_i), \\ &= \sum_{i=1}^K \tau \alpha_i^* - \sum_{i=1}^K \tau \alpha_i + \sum_{i=M+1}^K (\epsilon \frac{\partial O}{\partial \alpha_i^*} - \tau) (\alpha_i^* - \alpha_i), \\ &= \tau - \tau + \sum_{i=M+1}^K (\epsilon \frac{\partial O}{\partial \alpha_i^*} - \tau) (0 - \alpha_i) \\ &\geq 0. \end{aligned}$$

The last sum is less than 0 since all its terms are greater than or equal to 0. \square

Proof. (proof of Proposition 1) In fact the condition (3) implies that $\tilde{\nabla} O(\alpha^*) = 0$ and from Proposition 2 we can conclude. \square

References

- [1] Frank Kelly. Charging and rate control for elastic traffic. *European Transactions on Telecommunications*, 8(1):33–37, 1997.
- [2] S. Borst. User-level performance of channel-aware scheduling algorithms in wireless data networks. *IEEE/ACM Transactions on Networking*, 13(3):636–647, June 2005.
- [3] L. Tan, Z. Zhu, F. Ge, and N. Xiong. Utility maximization resource allocation in wireless networks: Methods and algorithms. *IEEE Transactions on Systems, Man, and Cybernetics: Systems*, 45(7):1018–1034, July 2015.
- [4] S. Borst, N. Hegde, and A. Proutiere. Mobility-driven scheduling in wireless networks. In *IEEE INFOCOM 2009*, pages 1260–1268, April 2009.
- [5] Yung Yi and Mung Chiang. Stochastic network utility maximisation: a tribute to Kelly’s paper published in this journal a decade ago. *European Transactions on Telecommunications*, 19(4):421–442, 2008.
- [6] H. Zhou, P. Fan, and J. Li. Global proportional fair scheduling for networks with multiple base stations. *IEEE Transactions on Vehicular Technology*, 60(4):1867–1879, May 2011.
- [7] P. Chandur, R. M. Karthik, and K. M. Sivalingam. Performance evaluation of scheduling algorithms for mobile wimax networks. In *2012 IEEE International Conference on Pervasive Computing and Communications Workshops*, pages 764–769, March 2012.
- [8] H. J. Kushner and P. A. Whiting. Convergence of proportional-fair sharing algorithms under general conditions. *IEEE Transactions on Wireless Communications*, 3(4):1250–1259, July 2004.
- [9] H. J. Bang, T. Ekman, and D. Gesbert. Channel predictive proportional fair scheduling. *IEEE Transactions on Wireless Communications*, 7(2):482–487, February 2008.
- [10] Robert Margolies, Ashwin Sridharan, Vaneet Aggarwal, Rittwik Jana, N. K. Shankaranarayanan, Vinay A. Vaishampayan, and Gil Zussman. Exploiting mobility in proportional fair cellular scheduling: Measurements and algorithms. *IEEE/ACM Trans. Netw.*, 24(1):355–367, February 2016.
- [11] N. Nguyen, O. Brun, and B. Prabhu. An algorithm for improved proportional-fair utility for vehicular users. *The 25th International Conference on Analytical and Stochastic Modelling Techniques and Applications ASMTA-2019*, May 2019.
- [12] H. Abou-zeid, H. S. Hassanein, and N. Zorba. Long-term fairness in multi-cell networks using rate predictions. In *2013 7th IEEE GCC Conference and Exhibition (GCC)*, pages 131–135, Nov 2013.

- [13] A. Mesbah. Stochastic model predictive control: An overview and perspectives for future research. *IEEE Control Systems Magazine*, 36(6):30–44, Dec 2016.
- [14] Laurent Condat. Fast projection onto the simplex and the l1 ball. *Math. Program.*, 158:575–585, 2016.
- [15] Pablo Alvarez Lopez, Michael Behrisch, Laura Bieker-Walz, Jakob Erdmann, Yun-Pang Flötteröd, Robert Hilbrich, Leonhard Lücken, Johannes Rummel, Peter Wagner, and Evamarie Wießner. Microscopic traffic simulation using sumo. In *The 21st IEEE International Conference on Intelligent Transportation Systems*. IEEE, 2018.