Optimal Utility-Based Multi-User Scheduling and Low-Complexity Alternatives

David Neumann, Andreas Dotzler and Wolfgang Utschick
Associated Institute for Signal Processing
Technische Universität München
{d.neumann, dotzler, utschick}@tum.de

Guido Dietl
DOCOMO Euro-Labs and
University of Applied Sciences Landshut
dietl@ieee.org

Abstract—**In this work the problem of utility-based multi-user scheduling is considered in a fading environment. The goal is to make use of the multi-user diversity while still guaranteeing short-term fairness. The performance of a scheduler is captured by a utility function. Based on the utility function and assuming statistical knowledge of the time-varying channel the optimization problem for the optimal scheduler can be formulated. Based on the optimal scheduler various simplifications are proposed which use estimations of future system states. Simulations show the performance gain compared to state-of-the-art methods.**

Index Terms—**opportunistic scheduling, predictive scheduling, multi-user diversity**

I. Introduction

Consider the downlink of a cellular multi-user system. The problem we discuss in this work is the optimization of the average service rates of each user, such that a utility function with respect to the average service rates is maximized. The optimization is with respect to the scheduling decisions at the base station. The system operates in discrete time, i.e., the time is divided in time slots. In each time slot, the scheduler has to make a decision concerning the service rates for the different users, where the set of possible decisions depends on the time-varying channel state, which is assumed to be constant within one time slot.

Previous work [1], [2] focuses on algorithms that are optimal if the service rates are averaged over a large number of time slots. For our work, we are interested in high performing solutions for short term averages, which are interesting for applications with tighter delay constraints. A first step in this direction, which is based on the modification of an established scheduling algorithm, can be found in [3].

To find a practical high performing scheduler we first discuss the optimal causal scheduler for our system model. The optimal scheduler uses information on previous channel states together with statistical information to estimate the system progression and scheduling decisions in future time slots. These estimates are then included in an optimization problem.

Since the optimal scheduler turns out to be too complex for application, we also present a less complex suboptimal approach. In order to reduce the complexity, we propose a scheduling approach with a simplified optimization problem, which still uses statistical information and information on previous channel states to improve the overall scheduling performance.

We propose further simplifications for larger systems and cases where no statistical information is available. We also suggest an optimization algorithm for the suboptimal scheduler, which is particularly efficient for complex physical layers, that offer a very large amount of possible scheduling decisions in each time slot. Finally, we present simulation results which show that our approach yields better short-term performance than state-of-the-art methods in the scenarios we considered.

II. System Model

The system model is partially adopted from [1]. We consider the downlink of a cellular system, where a base station serves N different users. In each time slot, the channel is in one of the possible states contained in a set $\mathcal{M}$. The sequence of channel states m_t $(t = 1, 2, \ldots)$ is assumed to be an ergodic stochastic process. Let $\mathcal{K}_m$ be the set of possible scheduling decisions for channel state m. Each scheduling decision $k \in \mathcal{K}_m$ has an associated vector of service rates $\boldsymbol{r}_m^k$ which contains the service rates for the different users when the channel is in state m and decision k is chosen.

We define the achievable rate region $\mathcal{R}_m$ of a channel state m as the set of all possible service rate allocations that can be obtained with a feasible scheduling decision

$$\mathcal{R}'_m = \left\{ \boldsymbol{r}_m^k, k \in \mathcal{K}_m \right\}. \tag{1}$$

For the practical implementation of our suboptimal algorithm, we include the possibility of time-sharing scheduling decisions within one time slot. We thus obtain a convex achievable rate region

$$\mathcal{R}_m = \mathsf{conv}\left[\mathcal{R}'_m\right], \tag{2}$$

where $\mathsf{conv}[\bullet]$ denotes the convex hull of a set.

The achievable rate regions allow for a more convenient way to formulate the optimization problems in later sections. That is, we formulate the optimizations with respect to the rate allocations instead of the scheduling decisions. We assume that, given a feasible rate allocation $\boldsymbol{\mu} \in \mathcal{R}_m$, it is possible to reconstruct the necessary scheduling decisions.

III. Optimal Scheduling

The goal of the scheduler is, to maximize a utility function $U(\bullet)$ with respect to the average user rates in a window of fixed length T. The optimal rate allocation is the solution of the following problem

$$\{\boldsymbol{\mu}_t^\star\}_{t=1}^T = \underset{\substack{\boldsymbol{\mu}_t \in \mathcal{R}_{m_t},\\ t=1,\dots,T}}{\arg\max}\, U\left(\frac{1}{T}\sum_{t=1}^{T}\boldsymbol{\mu}_t\right), \tag{3}$$

where $\boldsymbol{\mu}_t$ is the rate allocation for time slot t.

Since the scheduling decision for time slot t has to be made without knowing the channel states of future time slots and, thus, the rate regions, we cannot use (3) to compute the schedule in a real system. However, we can use the results from (3) as an upper bound for the causal schedulers discussed in this work.

One approach to compute an optimal causal scheduling decision is to find the rate allocation for the current time slot which leads to the highest expected utility at the end of the time window. The expectation is with respect to the future channel states and calculated given the known channel states of the past and the current time slots collected in the vector $\boldsymbol{m}_t = [m_1, \dots, m_t]$. We formulate the optimal causal scheduler as

$$\boldsymbol{\mu}_{\mathrm{OC}}(t) = \underset{\boldsymbol{\mu}_t \in \mathcal{R}_{m_t}}{\arg\max}\, \mathsf{E}\left[U\left(\frac{1}{T}\left(\boldsymbol{c}(t) + \boldsymbol{\mu}_t + \sum_{\tau=t+1}^{T}\boldsymbol{\mu}_{\mathrm{OC}}(\tau)\right)\right)\middle|\, \boldsymbol{m}_t\right] \tag{4}$$

where $\boldsymbol{c}(t) = \sum_{\tau=1}^{t-1}\boldsymbol{\mu}_t$ denotes the cumulated rate allocations from previous time slots.

The recursive formulation in equation (4) can be transformed into

$$\boldsymbol{\mu}_{\mathrm{OC}}(t) = \underset{\boldsymbol{\mu}_t \in \mathcal{R}_{m_t}}{\arg\max}\, \mathsf{E}\left[\max_{\boldsymbol{\mu}_{t+1} \in \mathcal{R}_{m_t}} \mathsf{E}\left[\ \dots\ \mathsf{E}\left[\max_{\boldsymbol{\mu}_T \in \mathcal{R}_{m_T}} U\left(\frac{1}{T}\left(\boldsymbol{c}(t) + \sum_{\tau=t}^{T}\boldsymbol{\mu}_\tau\right)\right) \middle| \boldsymbol{m}(T-1)\right]\ \dots\ \middle| \boldsymbol{m}(t+1)\right] \middle| \boldsymbol{m}_t\right]. \tag{5}$$

For a finite number of possible channel states M, each expectation branches into M maximizations. The result is a number of optimization variables in the order of $(NM)^T$ for the maximization in the first time slot of a time window. If the conditional distribution of the channel states are known, and in case of a small number of states and a small time window, it is possible to calculate the optimal solution to the problem in (5). However, the huge increase in complexity with respect to all relevant system parameters makes the optimal scheduler impractical for real systems.

IV. Rate Predictive Scheduling

In order to reduce the complexity of the optimal scheduler we reduce the number of optimization variables by solving an approximation of (5). Instead of the actual rate regions and the expected utility function values we take the expectations of the rate regions and a single deterministic utility function value. The result is the following scheduler,

$$\boldsymbol{\mu}_{\mathrm{RP}}(t) = \underset{\boldsymbol{\mu}_t \in \mathcal{R}_{m_t}}{\arg\max} \max_{\substack{\boldsymbol{\mu}_\tau \in \mathsf{E}[\mathcal{R}_{m_\tau}|\boldsymbol{m}_t],\\ \tau=t+1,\dots,T}} U\left(\frac{1}{T}\left(\boldsymbol{c}(t) + \sum_{\tau=t}^{T}\boldsymbol{\mu}_\tau\right)\right) \tag{6}$$

In other words, we predict the rate regions and solve the problem in (3) with the predicted rate regions for the future time slots. Note that the inner maximization in (6) gives an upper bound to the outer expectation in (5).

If we substitute $\tilde{\boldsymbol{\mu}} = \frac{1}{T-t}\sum_{\tau=t+1}^{T}\boldsymbol{\mu}_\tau$ we obtain the equivalent problem,

$$\boldsymbol{\mu}_{\mathrm{RP}}(t) = \underset{\boldsymbol{\mu}_t \in \mathcal{R}_{m_t}}{\arg\max} \max_{\tilde{\boldsymbol{\mu}} \in \tilde{\mathcal{R}}_t} U\left(\frac{1}{T}\left(\boldsymbol{c}(t) + \boldsymbol{\mu}_t + (T-t)\tilde{\boldsymbol{\mu}}\right)\right) \tag{7}$$

with

$$\tilde{\mathcal{R}}_t = \frac{1}{T-t}\sum_{\tau=t+1}^{T}\mathsf{E}[\mathcal{R}_{m_\tau}|\boldsymbol{m}_t]. \tag{8}$$

With this substitution we reduce the number of optimization variables, but shift the complexity into the constraint set $\tilde{\mathcal{R}}_t$. However, the structure of the constraint set does not change. For a finite number of states M, the expectation of an achievable rate region is a weighted vector sum of the achievable rate regions corresponding to the different states. This gives us the following result

$$\begin{aligned}\tilde{\mathcal{R}}_t &= \frac{1}{T-t}\sum_{\tau=t+1}^{T}\sum_{m\in\mathcal{M}} p_{m_\tau|\boldsymbol{m}_t}(m|\boldsymbol{m}_t)\mathcal{R}_m \\ &= \sum_{m\in\mathcal{M}} \frac{\sum_{\tau=t+1}^{t} p_{m_\tau|\boldsymbol{m}_t}(m|\boldsymbol{m}_t)}{T-t}\mathcal{R}_m \\ &= \sum_{m\in\mathcal{M}} p_{\tilde{m}|\boldsymbol{m}_t}(m|\boldsymbol{m}_t)\mathcal{R}_m.\end{aligned} \tag{9}$$

We can see that the reduction in complexity comes with the minimal cost of calculating the probability distribution $p_{\tilde{m}|\boldsymbol{m}_t}$.

If the channel states in different time slots are independent, the estimation in (8) simplifies to

$$\tilde{\mathcal{R}}^{\mathrm{AP}} = \mathsf{E}[\mathcal{R}_m], \tag{10}$$

which is independent of the time slot t. This can also be seen as an a-priori estimator which disregards available observations of the channel states.

A. Low complexity estimation

For systems with unkown statistics or systems with a large number of possible channel states it might not be possible to calculate the expectations used in the estimators in (8) and (10). In this case we propose to approximate the a-priori estimator in (10) based on a local average

$$\tilde{\mathcal{R}}_t^{\mathrm{LA}} = \frac{1}{W} \sum_{\tau=t-W+1}^{t} \mathcal{R}_{m_\tau} \approx \tilde{\mathcal{R}}^{\mathrm{AP}}. \tag{11}$$

In order to further reduce complexity, we propose to use an exponentially weighted average with a recursive update of the estimated region which is given by

$$\tilde{\mathcal{R}}_{t+1}^{\mathrm{WA}} = (1-\alpha)\tilde{\mathcal{R}}_t^{\mathrm{WA}} + \alpha \mathcal{R}_{m_t}, \tag{12}$$

with $\alpha = \frac{1}{W}$.

In the worst case, this update leads to an exponential increase in rate points in the estimated region, as a result of the vector summation of the sets. As a consequence, we propose to track only a fixed amount of rate points on the boundary of the estimated region. These points can be defined as solutions to linear programs.

Specifically, we store a set of rate points $\tilde{\mathcal{R}}_t' = \{\tilde{\boldsymbol{r}}_t^l\}_{l=1}^L$ considering a corresponding set of weights $\mathcal{W} = \{\boldsymbol{w}^l\}_{l=1}^L$, where each rate point is a point on the boundary of the estimated region in (12) given by

$$\begin{aligned}
\tilde{\boldsymbol{r}}_t^l &= \operatorname*{arg\,max}_{\tilde{\boldsymbol{r}} \in \tilde{\mathcal{R}}_t^{\mathrm{WA}}} {\boldsymbol{w}^l}^{\mathsf{T}} \tilde{\boldsymbol{r}} \\
&= (1-\alpha) \operatorname*{arg\,max}_{\tilde{\boldsymbol{r}} \in \tilde{\mathcal{R}}_{t-1}^{\mathrm{WA}}} {\boldsymbol{w}^l}^{\mathsf{T}} \tilde{\boldsymbol{r}} + \alpha \operatorname*{arg\,max}_{\boldsymbol{r} \in \mathcal{R}_{m_{t-1}}} {\boldsymbol{w}^l}^{\mathsf{T}} \boldsymbol{r} \\
&= (1-\alpha)\tilde{\boldsymbol{r}}_{t-1}^l + \alpha \operatorname*{arg\,max}_{\boldsymbol{r} \in \mathcal{R}_{m_{t-1}}} {\boldsymbol{w}^l}^{\mathsf{T}} \boldsymbol{r}.
\end{aligned} \tag{13}$$

For this update we assume the weights to be fixed for all time slots. For example, they can be chosen as a fixed number of approximately uniformly distributed vectors on the positive part of the N-dimensional unit sphere.

With the optimization problem in (7) in mind, we see a possible advantage in adapting the weights over time, to sample areas of the estimated region more accurately, that are of higher importance to the optimization problem, i.e., areas that are more likely to be in the vicinity of the optimal solution to (7). This idea is based on the assumption that the optimal solution $\tilde{\boldsymbol{\mu}}^*$ of the inner maximization varies only slowly with time.

For time-varying weights $\mathcal{W}_t = \{\boldsymbol{w}_t^l\}_{l=1}^{L_t}$ the updated rate points are given by

$$\tilde{\boldsymbol{r}}_t^l = (1-\alpha) \operatorname*{arg\,max}_{\tilde{\boldsymbol{r}} \in \tilde{\mathcal{R}}_{t-1}'} {\boldsymbol{w}_t^l}^{\mathsf{T}} \tilde{\boldsymbol{r}} + \alpha \operatorname*{arg\,max}_{\boldsymbol{r} \in \mathcal{R}_{m_{t-1}}} {\boldsymbol{w}_t^l}^{\mathsf{T}} \boldsymbol{r}. \tag{14}$$

Note that with time-varying weights the rate points are no longer guaranteed to be on the boundary of the estimated region defined in (12).

The actual estimated rate region with this approach is given by the convex hull over the sampled rate points

$$\tilde{\mathcal{R}}_t^{\mathrm{LC}} = \mathsf{conv}[\tilde{\mathcal{R}}_t'] \subseteq \tilde{\mathcal{R}}_t^{\mathrm{WA}}. \tag{15}$$

V. Implementation Aspects

In this section we suggest an convex optimization algorithm to solve the problem in (7), which integrates well with the proposed estimation methods. It is very efficient when there are a lot of rate points in the constraint sets which is usually the case for the estimated region $\tilde{\mathcal{R}}_t$, but also for the achievable rate regions for some physical layers.

A. Complex Physical Layers

For complex physical layers with a large or even infinite amount of possible scheduling decisions, it is not feasible to calculate all of the corresponding service rate vectors. However, we only need a subset of the possible rate allocations to compute the optimal solution. To be able to solve the optimization problem with the suggested algorithm, we assume that it is possible to calculate single rate vectors of the achievable rate regions as the solution to a weighted sum rate (WSR) maximization problem

$$\boldsymbol{r}_m^i = \operatorname*{arg\,max}_{\boldsymbol{r} \in \mathcal{R}_m} {\boldsymbol{\lambda}^{i}}^{\mathsf{T}} \boldsymbol{r}. \tag{16}$$

The multiple-input-multiple-output (MIMO) broadcast channel (BC) capacity region would be one example.

B. Proposed Algorithm

Since the achievable rate regions are convex, the problem in (7) can be solved using convex optimization algorithms. One algorithm we found to be very efficient in solving the problem is the simplicial decomposition algorithm [4]. The algorithm solves the optimization problem by iteratively refining inner approximations of the current constraint sets $\mathcal{R}_{m_t}, \tilde{\mathcal{R}}_t$

$$\begin{aligned}
\mathcal{R}_{m_t}^i &= \mathsf{conv}\left[\{\boldsymbol{r}_{m_t}^j\}_{j=1}^i\right] \\
\tilde{\mathcal{R}}_t^i &= \mathsf{conv}\left[\{\tilde{\boldsymbol{r}}_t^j\}_{j=1}^i\right].
\end{aligned} \tag{17}$$

where the rate vectors $\boldsymbol{r}_{m_t}^j, \tilde{\boldsymbol{r}}_t^j$ are solutions to WSR maximizations over $\mathcal{R}_{m_t}, \tilde{\mathcal{R}}_t$ as in (16).

The intermediate solution vectors $\boldsymbol{\mu}_t^i$ and $\tilde{\boldsymbol{\mu}}_t^i$ are obtained by solving the optimization problem in (7) but replacing the constraint sets with the respective inner approximations

$$\{\boldsymbol{\mu}_t^i, \tilde{\boldsymbol{\mu}}_t^i\} = \operatorname*{arg\,max}_{\substack{\boldsymbol{\mu}_t \in \mathcal{R}_{m_t}^i, \\ \tilde{\boldsymbol{\mu}}_t \in \tilde{\mathcal{R}}_t^i}} U\left(\frac{1}{T}\left(\boldsymbol{c}(t) + \boldsymbol{\mu}_t + (T-t)\tilde{\boldsymbol{\mu}}_t\right)\right). \tag{18}$$

This optimization over a polytope can be solved with a projected gradient method.

The weights for the next WSR maximizations are chosen as the gradient at the intermediate solution of the last iteration

$$\boldsymbol{\lambda}^i = \nabla U \left(\frac{1}{T} \left(\boldsymbol{c}(t) + \boldsymbol{\mu}_t^{i-1} + (T-t)\tilde{\boldsymbol{\mu}}_t^{i-1} \right) \right). \tag{19}$$

With this choice, it can be shown that the solutions of (18) converge to the solution of the original problem in (7).

One important thing to note is that we can make use of the already calculated WSR maximizations for the update of the estimated region in (13) and (14) if we do not want to spend additional complexity. That is, we use the inner approximation $R_{m_t}^{I_t}$ to update the estimated region, where I_t is the number of optimization iterations preformed in time slot t.

If we use time varying weights $\mathcal{W}_t$ for the update of the estimated region can also use the weights $\{\boldsymbol{\lambda}^i\}_{i=1}^{I_t}$ that were used in the optimization to adapt the weights for the current time slot. One option would be to completely discard $\mathcal{W}_{t-1}$ and set $\mathcal{W}_t = \{\boldsymbol{\lambda}^i\}_{i=1}^{I_t}$.

VI. Simulation Results

In this section we compare the performance of different schedulers in numerical simulations. We analyse the schedulers based on the optimization problem in (7) for different estimators, namely the a-posteriori scheduler using the estimate in (8), the a-priori scheduler using the estimate in (10) and two schedulers using the recursively updated estimates in (13) and (14) respectively. The first one of those two uses fixed weights, which are approximately uniformly distributed on the unit sphere, and the other one uses a small number of dynamic weights based on the optimization weights. Additionally we include the gradient scheduler [1] and the upper bound in (3) as references.

As utility function we use the proportional fair utility given by

$$U(\boldsymbol{r}) = \sum_{n=1}^{N} \log r_n. \tag{20}$$

A. Orthogonal Scheduling

For the first set of simulations we consider sets of possible scheduling decisions $\mathcal{K}_m$, where each decision is associated with a single user. The rate vector $\boldsymbol{r}_m^k$ corresponding to the scheduling of user k has only one non-zero entry at component k denoting the achievable rate of the user given the channel state m. The resulting achievable rate regions are N-dimensional simplices, where N is the number of users.

The channel states are modeled as N-dimensional vectors. Each component denotes the channel state of a single user, which can be mapped directly to the corresponding achievable rate. The states of the different users are modeled as independent Markov chains.

The transition probabilities for the Markov processes are gathered from empirical simulations. To this end we generate channel coefficients for a large number of time slots according to Jakes' model [5] and calculate the corresponding Shannon rates. The rates are then quantized using C different rate configurations, each quantization interval representing one possible channel state for one user. Thus, the total number of possible states is given by $M = C^N$.

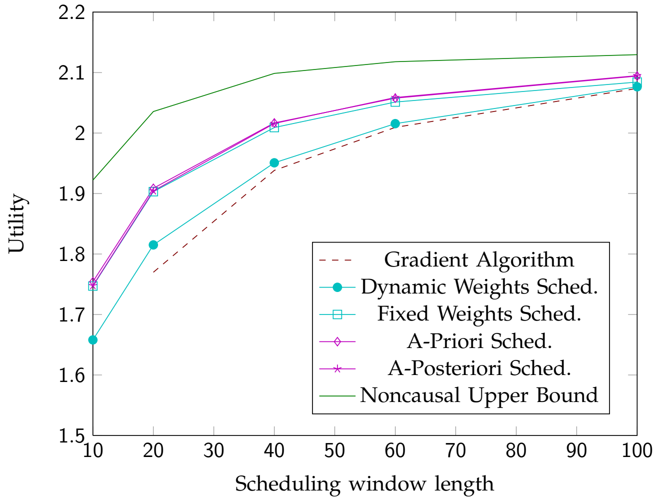

Fig. 1. Performance for orthogonal scheduling

The actual scheduling simulations are done for several realisations of the random channel state process over 5000 time slots. We choose $C = 4$ and $N = 4$ and varying scheduling window lengths T.

The performance of the different algorithms is shown in Figure 1. The two algorithms using statistical information show nearly the same performance indicating a small impact of current channel state information on the estimated rate regions. The scheduler using time-varying weights shows a slight performance gain over the gradient algorithm for all window lengths. Note that the gradient scheduler is equivalent to the proportional fair scheduler [6] under this system model.

The scheduler using fixed weights performs similar to the statistical schedulers for small window lengths but falls of for larger window lengths compared to all other algorithms. This can be explained by the fact that for small window lengths there is more variation in the average rates of the different windows, thus, there is also more variation in the results of the optimization. In this case the fixed weight scheduler benefits from the widely distributed sample rate points over the estimated region.

For larger time windows, however, the average rates vary only slightly, which leads to optimization results which are focused on a small area of the estimated region. In this case the fixed weight scheduler has a lot of useless information, i.e., unnecessary sample points, and not enough sample points in the important area of the estimated region resulting in a relative performance

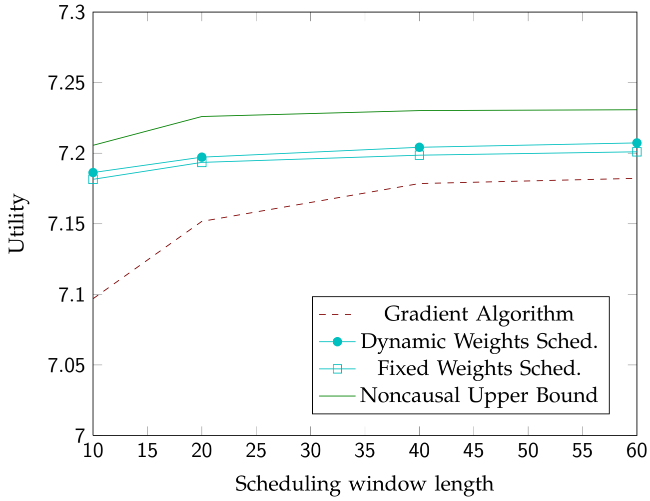

Fig. 2. Performance for the MIMO-BC capacity region

degradation for large window sizes, where accuracy in the estimate is important to further increase the utility.

Note that for large window sizes, the a-priori and a-posteriori schedulers outperform the gradient scheduler only slightly. Thus, in most cases the gradient scheduler is the preferable solution for long term fairness due to the low complexity.

B. Scheduling for the MIMO-BC Channel

A further set of simulations analyses the application of our algorithm to the MIMO-BC channel, where we assume that the achievable rate regions correspond to the capacity region, which is achievable with dirty paper coding [7]. We assume two antennas at the transmitter and at each receiver. The complex flat fading channel coefficients are generated independently for each user using Jakes' model. The total channel state is then given as the combination of all coefficients. The number of users is set to $N = 6$.

In this scenario it is no longer reasonable to use the a-priori and a-posteriori estimators due to prohibitively high complexity.

In Figure 2 we show the results for the fixed and dynamic weights schedulers compared to the upper bound and the gradient scheduler. We notice that the dynamic weights scheduler yields a better relative performance. A possible explanation is that for the MIMO-BC with dirty paper coding, the average user rates are more stable than for the orthogonal scheduling case.

VII. Conclusion

We proposed effective methods for multi-user scheduling based on estimates in the rate space and could show significant performance gains for tighter delay requirements.

References

[1] Alexander L. Stolyar. On the asymptotic optimality of the gradient scheduling algorithm for multiuser throughput allocation. *Operations Research*, 53(1):12–25, January 2005.

[2] Leonidas Georgiadis, Michael J. Neely, and Leandros Tassiulas. Resource allocation and cross-layer control in wireless networks. *Foundations and Trends in Networking*, 1(1):1–144, 2005.

[3] H. J Bang, T. Ekman, and D. Gesbert. Channel predictive proportional fair scheduling. *IEEE Transactions on Wireless Communications*, 7(2):482–487, February 2008.

[4] Balder Von Hohenbalken. Simplicial decomposition in nonlinear programming algorithms. *Mathematical Programming*, 13:49–68, 1977.

[5] W. Jakes. *Microwave Mobile Communications*. New York: Wiley, 1974.

[6] F.P. Kelly, A.K. Mauloo, and D.K.H. Tan. Rate control for communication networks: shadow prices, proportional fairness and stability. *Journal of the Operational Research Society*, 19(3):237– 252, March 1998.

[7] H. Weingarten, Y. Steinberg, and S. Shamai. The capacity region of the gaussian Multiple-Input Multiple-Output broadcast channel. *IEEE Transactions on Information Theory*, 52(9):3936–3964, September 2006.