Fair resource allocation over time

Evripidis Bampis, Bruno Escoffier, Sasa Mladenovic

Fair resource allocation over time

E. Bampis, B. Escoffier, S. Mladenovic.
LIP6, Sorbonne Université and CNRS.

November 19, 2018

Abstract

We consider the over-time version of the Max-Min Fair Allocation problem. Given a time horizon $t = 1, 2, \ldots, T$, with at each time t a set of demands and a set of available resources that may change over the time defining instance I_t, we seek a sequence of solutions $S_1, S_2, \ldots, S_T$ that (1) are near-optimal at each time t, and (2) as stable as possible (inducing small modification costs). We focus on the impact of the knowledge of the future on the quality and the stability of the returned solutions by distinguishing three settings: the off-line setting where the whole set of instances through the time horizon is known in advance, the on-line setting where no future instance is known, and the k-lookahead setting where at time t, the instances at times $t+1, \ldots, t+k$ are known. We first consider the case without restrictions where the set of resources and the set of agents are the same for all instances and where every resource can be allocated to any agent. For the off-line setting, we show that the over-time version of the problem is much harder than the static one, since it becomes $\mathcal{NP}$-hard even for families of instances for which the static problem is trivial. Then, we provide a $\frac{\rho}{\rho+1}$-approximation algorithm for the off-line setting using as subroutine a ρ-approximation algorithm for the static version. We also give a $\frac{\rho}{\rho+1}$-competitive algorithm for the online setting using also as subroutine a ρ-approximation algorithm for the static version. Furthermore, for the case with restrictions, we show that in the off-line setting it is possible to get a polynomial-time algorithm with the same approximation ratio as in the case without restrictions. For the online setting, we prove that it is not possible to find an online algorithm with bounded competitive ratio. For the 1-lookahead setting however, we give a $\frac{\rho}{2(2\rho+1)}$-approximation algorithm using as subroutine a ρ-approximation algorithm for the static version.

1 Introduction

Resource allocation problems have been intensively studied in the past [Lus12]. In such a problem, the aim is to find an allocation of a set of limited resources to a set of competing agents in a way to optimize a given objective function subject to a set of constraints (availability of resources, level of demands, compatibility between agents and resources). The study of a large variety of combinatorial optimization problems, including facility location, matching, scheduling problems etc..., has been motivated by applications in logistics, computer and telecommunication networks, air traffic management, energy management etc... A special focus on resource allocation problems has been given in multiagent settings, considering a fairness criterion/objective in the allocation [BLFL05, GP10, BCM16]. Typically, these problems are studied in the case where the instance is *static*, while in many applications the instance changes over time. Take for example an energy company producing electricity on a set of reactors. The electricity demands evolve over time, so the company has to decide a production plan, all along a period of time, in order (1) to satisfy the demand at every time, (2) to minimize the production cost at each time, and (3) to minimize the cost of turning on/off the reactors. The reader is refered to [LVB99, JMD05] for other applications in dynamic contexts. It is only recently that resource allocation problems start to be studied in such a *dynamic* context [EMS14, GTW14].

More formally, we are given a time horizon: $t = 1, 2, \ldots, T$ where at each time t we have a new instance I_t of the considered resource allocation problem. It is tempting to try to solve the problem for every new instance. However in most practical applications, there is a non-negligible transition cost for adopting modifications of the current allocation (solution). Hence, the goal is to determine a sequence of solutions $S_1, S_2, \ldots, S_T$ that both (1) are near-optimal (*quality*), and (2) induce small modification costs (*stability*). An important aspect in this dynamic setting is the impact of the knowledge of either the whole set of instances (*off-line case*), or a limited number of instances, say k, in the near future (*k-lookahead case*), or no future instance at all (*on-line case*) on the quality and the stability of the returned solutions.

In this paper, we focus on the over-time version of different variants of a basic resource allocation problem, the Max-Min Fair Allocation problem where the aim is to determine an allocation of the resources to the agents in a *fair* way. An instance of the Max-Min Fair Allocation problem is characterized by a set $\mathcal{R}$ of n resources, a set $\mathcal{P}$ of m agents

and a set of nonnegative values $\ell(j,i)$, for every $j \in \mathcal{R}$ and every $i \in \mathcal{P}$. The value $\ell(j,i)$ corresponds to the valuation of the agent i for the resource j. An allocation is a partition $\mathcal{R}_1, \ldots, \mathcal{R}_m$ of the set of resources and the valuation of agent i is equal to $\ell(\mathcal{R}_i) = \sum_{j \in \mathcal{R}_i} \ell(j,i)$. The objective is to find an allocation with the best possible valuation for the worst-off agent, i.e. an allocation maximizing $\min_{i \in \mathcal{P}} \sum_{j \in \mathcal{R}_i} \ell(j,i)$. Another name used in the literature for this problem is the Santa Claus problem where the resources correspond to gifts and the agents to children. The objective is to find an allocation of gifts to the children so as to maximize the happiness of the least happy child. In scheduling terms, this problem corresponds to the problem of scheduling a set of jobs on a set of *unrelated machines* so as to maximize the load of the least loaded machine. Here the resources are the jobs and the agents are the machines. Important subproblems of the Max-Min Fair Allocation problem is the Restricted Max-Min Fair Allocation problem in which $\ell(j,i) \in \{0, \ell_j\}$ (each resource has a fixed value, but only some agents are interested in it), the Homogeneous Max-Min Fair Allocation problem in which $\ell(j,i) = \ell_j$ (each resource has a fixed value) and the Uniform Max-Min Fair Allocation problem in which $\ell(j,i) = \frac{\ell_j}{s_i}$ (where s_i is the speed of machine i in the scheduling context).

In the Over-Time Max-Min Fair Allocation problem, we are given a sequence $I_1, \ldots, I_T$ of instances of the Max-Min Fair Allocation problem, with $T \geq 2$. For every instance I_t, we have a set of m_t agents $M_t = \{1, \ldots, m_t\}$, a set of n_t resources $N_t = \{1, \ldots, n_t\}$ and a valuation of the agent i for the resource j, denoted by $\ell_t(j,i) \in \mathbb{R}$ for every $j \in N_t$ and $i \in M_t$. We are also given a reward equal to $K \in \mathbb{R}^+$ for every resource that is allocated to the same agent in two consecutive solutions (this *reward* for our maximization problem corresponds to the transition *cost* in [EMS14, GTW14] for minimization problems).

A solution sequence $S = (O_1, \ldots, O_T)$, where O_t is a solution of I_t, defines

- a *Santa Claus revenue*, $\ell(S) = \sum_{t=1}^{T} \ell_t(O_t)$ such that

$$\ell_t(O_t) = \min_{i \in \{1,\ldots,m_t\}} \sum_{j=1}^{n_t} l_t(j,i) \cdot x_{t,j,i}$$

  with $x_{t,j,i} = 1$ if the resource j is allocated to the agent i in O_t, and 0 otherwise. This corresponds to the sum of the valuations of the worst-off agents over all solutions O_t, for $t = 1, 2, \ldots T$,

- a *transition revenue*, $D(S) = \sum_{t=1}^{T-1} D_t(S)$, where

  $D_t(S) = K \times |\text{resources remaining on a same agent in } O_t \text{ and } O_{t+1}|$.

  Intuitively, the larger the transition revenue, the more stable the sequence of the solutions.

Overall, we seek for a solution sequence that yields a good trade-off between quality and stability, and in what follows our goal will be the maximization of the sum of $f(S) = \ell(S) + D(S)$.

Example 1. *Suppose that we have $n = 4$ resources and $m = 2$ agents, $K = 1$. We have 3 time steps. We are in the homogeneous case, and here are the valuation of resources:*

	$t = 1$	$t = 2$	$t = 3$
Resource 1	*3*	*6*	*2*
Resource 2	*2*	*4*	*1*
Resource 3	*1*	*8*	*2*
Resource 4	*2*	*1*	*3*

Let us consider this first solution sequence S_1 (for column $t = 1$ and line Agent 1, $(2, 4)$ are the resources allocated to Agent 1 at time 1):

	$t = 1$	$t = 2$	$t = 3$
Agent 1	*(2,4)*	*(1,2)*	*(2,4)*
Agent 2	*(1,3)*	*(3,4)*	*(1,3)*

We have $\ell(S_1) = 4 + 9 + 4 = 17$ (these are actually the best solutions at each time step). Since 2 resources remain allocated to the same agent between times 1 and 2, and 2 as well between times 2 and 3, we have $D(S_1) = 2K + 2K = 4$. Hence, $f(S_1) = 21$.

One may consider S_2 which is the same as S_1 but resource 4 is given to agent 1 in time step 2. Then $\ell(S_2) = \ell(S_1) - 1 = 16$, but $D(S_2) = 3K + 3K = 6$, so $f(S_2) = 22 > f(S_1)$.

Recall that a *ρ-approximation algorithm* for an optimization problem is a polynomial-time algorithm returning a solution whose value is within a factor of ρ of the value of an optimal solution for all the instances of the problem [WS11]. ρ is called the *approximation ratio* of the algorithm.

For a maximization problem like the Max-Min Fair Allocation problem studied in this paper, $0 \leq \rho \leq 1$ ($\rho = 1$ corresponds to an exact algorithm). The definition of a *c-competitive algorithm* is similar to the one of a *ρ-approximation algorithm* where the on-line algorithm is compared with respect to the optimal off-line algorithm that knows the entire sequence of the input data.

Variants. We distinguish between two cases: the case where there are restrictions on the set of agents on which a resource can be allocated (Section 3.2) and the case where there are no restrictions (Section 3.1).

2 Previous results

The (static) problem has been first studied in a game theoretic context [LMMS04]. Bezáková and Dani have shown that it is $\mathcal{NP}$-hard to approximate the Max-Min Fair Allocation problem to within a factor better than 1/2 and they provided the first non-trivial approximation algorithm [BD05]. Later, Bansal and Sviridenko [BS06] presented a configuration linear program for the problem with an integrality gap of $\Omega\left(\frac{1}{\sqrt{m}}\right)$. Asadpour and Saberi [AS10] proposed an $\Omega\left(\frac{1}{\sqrt{m}\log^3(m)}\right)$-approximation algorithm. This result has been improved by Saha and Srinivasan who gave an $\Omega\left(\frac{\log\log m}{\sqrt{m}\log m}\right)$-approximation algorithm [SS10]. The best result with respect to the number of resources is an $\Omega\left(\frac{1}{n^\epsilon}\right)$-approximation algorithm running in $O(n^{\frac{1}{\epsilon}})$, for any $\epsilon > 0$ [BCG09, CCK09].

A lot of progress has been done for the Restricted Max-Min Fair Allocation problem. While the same inapproximability result holds in this case, Bansal and Sviridenko [BS06] proposed an $\Omega\left(\frac{\log\log\log m}{\log\log m}\right)$-approximation algorithm. Haeupler et al. presented a constant factor approximation algorithm, but the constant was large and unspecified. Asadpour and Saberi [AS10] showed that the integrality gap of the configuration linear program is bounded by 1/4. Annamalai, Kalaitzis and Svensson proposed recently a combinatorial algorithm with approximation ratio $\frac{1}{6+2\sqrt{10}+\epsilon}$ which runs in $|I|^{\frac{1}{\epsilon^2}\log(\frac{1}{\epsilon})}$, for any $\epsilon > 0$ and where $|I|$ is the size of the instance [AKS17].

For the Homogeneous Max-Min Fair Allocation problem, Deuermeyer, Friesen and Langston [DFL82] showed that the classical LPT (*Longest*

Processing Time) algorithm is a $\frac{3}{4}$-approximation algorithm. Csirik, Kellerer and Woeginger [CKW92] improved the analysis showing that the approximation ratio of LPT is in fact $\frac{3m-1}{4m-2}$. Woeginger [Woe97] and also Ochel with Voecking [OV09] proposed a polynomial time approximation scheme for the problem: for any $\epsilon > 0$, there exists a (polynomial time) $(1-\epsilon)$-approximation algorithm. Epstein and Sgall proposed a polynomial time approximation scheme for the Uniform Max-Min Fair Allocation problem [ES04].

3 Our results

Our goal is to measure to what extend the over-time setting is harder to solve than the static one, with respect to the quality of solutions an algorithm can compute in polynomial time. In particular, for cases where $\ell(j,i) = \ell(j)$ (homogeneous case) or when $\ell(j,i) \in \{0, \ell(j)\}$, the static problem has constant ratio approximation algorithms as mentioned above; is it still the case in the over-time setting? We provide some answers which depend on the assumptions made on the knowledge of the future (off-line/on-line/k-lookahead), and on the possibility to have incompatibilities between resources and agents (case with restriction) or not.

More precisely, in Section 3.1 we study the case without restrictions where the set of resources and the set of agents are the same for all instances and where every resource can be allocated to any agent. For the off-line setting, we first show that the over-time version of the problem is much harder than the static one with respect to the computational complexity. Then, we propose a $\frac{\rho}{\rho+1}$-approximation algorithm for the off-line setting using as subroutine a ρ-approximation algorithm for the static version, thus getting a constant ratio approximation algorithm for the over-time setting whenever the static problem has one. We improve this approximation ratio when the number of steps T or the number of agents m is bounded. Then, we show that a similar result can be obtained in the on-line setting, since we devise a $\frac{\rho}{\rho+1}$-competitive algorithm using also as subroutine a ρ-approximation algorithm for the static version.

In Section 3.2, we study the case with restrictions. We show that in the off-line setting it is possible to get a polynomial-time algorithm with the same approximation ratio as in the case without restrictions. For the on-line setting, we prove that it is *not* possible to find an on-line algorithm

with bounded competitive ratio. Interestingly, we show that knowing only one time step in advance (1-lookahead setting) is sufficient to again obtain a constant approximation algorithm whenever the static case has one; more precisely, we give a $\frac{\rho}{4\rho+2}$-approximation algorithm using as subroutine a ρ-approximation algorithm for the static version.

3.1 The case WITHOUT-RESTRICTIONS

We consider the case where the set of resources and the set of agents are the same for all the instances. Therefore, $n_1 = ... = n_T = n$ and $m_1 = ... = m_T = m$. In addition, every resource can be allocated to any agent.

As a first result, we illustrate the fact that the over-time problem is significantly harder than the static MAX-MIN FAIR ALLOCATION problem, since it remains $\mathcal{NP}$-hard even in a very restricted case.

Theorem 1. MAX-MIN FAIR ALLOCATION *is $\mathcal{NP}$-hard even in an off-line setting with no restriction, with only 2 agents and in the homogeneous case, with furthermore $\ell_t(j,i) = \ell_t(j) \in \{0,1\}$.*

Note that these instances are trivial in the static case.

Proof. We show this result using a reduction from the Maximum Cut problem. In this problem, known to be $\mathcal{NP}$-hard [GJ79], we are given an undirected graph $G = (V, E)$ and an integer s. The question is whether we can partition the set V of vertices into (V_1, V_2) in such a way that at least s edges have one endpoint in V_1 and one in V_2.

Given $G = (V, E)$ and s with $V = \{v_1, \ldots, v_{|V|}\}$ and $E = \{e_1, \ldots, e_{|E|}\}$, we build an instance of MAX-MIN FAIR ALLOCATION with 2 agents, $|V|$ resources (call them $\{v_1, \ldots, v_{|V|}\}$) and $T = |E|$ time steps. At time step i agents 1 and 2 give valuation 1 to the two endpoints of edge e_i, and 0 to all other resources. We set K sufficiently large, say $K = |E| + 1$. We claim that there is a cut with at least s edges in G if and only if there is a solution sequence of value at leat $K(T-1)|V| + s$ in the instance of MAX-MIN FAIR ALLOCATION.

If there is a cut (V_1, V_2) with at least s edges, then give vertices of V_1 to agent 1 and vertices of V_2 to agent 2, at any time (no modification of solution). Then the transition revenue is $K(T-1)|V|$. If an edge e_i is in the cut, then at time i one endpoint is given to agent 1, the other 1 to agent 2, and the Santa Claus revenue at time i is 1. Then we have a solution sequence of value at least $K(T-1)|V| + s$.

Conversely, note that the Santa Claus revenue is at most 1 at any time step, so at most $|E|$ in total. So, if a solution sequence makes at least one modification in the solutions, it will have value at most $K(T-1)|V| - K + |E| < K(T-1)|V|$. Hence, a solution sequence with value at least $K(T-1)|V| + s$ does not make any modification. Let V_1 be the set of resources given to agent 1, and V_2 the ones given to agent 2. The solution sequence has (at least) s time steps with Santa Claus revenue 1. This means that there are at least s edges with one endpoint in V_1 and one in V_2.

□

Now, we give some approximation algorithms for the problem, first in an off-line setting and then in an on-line setting.

3.1.1 Off-line case

In order to find a good solution sequence for the revenue, we use as a subroutine an algorithm A for the static Max-Min Fair Allocation problem. Then Algorithm 1 works as follows:

- Apply A on each instance I_t, denote by $\tilde{O}_t$ the obtained solution.
- Build a first solution sequence $S_0 = (\tilde{O}_1, \tilde{O}_2, \ldots, \tilde{O}_T)$.
- For any t consider the solution sequence $S_t = (\tilde{O}_t, \tilde{O}_t, \ldots, \tilde{O}_t)$ (no modification).
- Output the best of the $T+1$ previous solutions.

The idea is that the first solution performs well with respect to the Santa Claus revenue $\ell(S)$, while the others perform well with respect to the transition revenue $D(S)$. More precisely, we have the following result.

Theorem 2. *If A is a (polytime) ρ-approximation algorithm for the static case, then Algorithm 1 is a (polytime) $\frac{T\rho}{(T-1)\rho+T}$-approximation algorithm for the over-time problem.*

Note that this ratio decreases with T down to $\frac{\rho}{\rho+1}$ (T unbounded).

Proof. Let $S^* = (O_1^*, ..., O_T^*)$ be an optimal solution sequence, and denote by S the solution output by the algorithm. As A is a ρ-approximation algorithm for the static case, we get $\ell_t(\tilde{O}_t) \geq \rho \ell_t(O_t^*)$ for any t. Then:

$$f(S) \geq \ell(S_0) \geq \rho \sum_{i=1}^{T} \ell_t(O_t^*) = \rho \ell(S^*) \tag{1}$$

Now, since in S_i ($i \geq 1$) no modification is made, then clearly S_i has maximum transition revenue, so:

$$f(S) \geq f(S_i) \geq \rho \ell_i(O_i^*) + D(S^*) \tag{2}$$

Now, sum Equation (2) for $i = 1, \ldots, T$, multiply by ρ and add $(T-\rho)$ times Equation (1). This gives:

$$(T\rho + T - \rho) f(S) \geq \rho(\rho \ell(S^*) + TD(S^*)) + (T-\rho)\rho\ell(S^*)$$

We derive $(T\rho + T - \rho) f(S) \geq T\rho f(S^*)$, which gives the claimed ratio. □

Note that the analysis is tight when $\rho = 1$, for any T, even in the homogeneous case. Consider the following instance, with T agents and $n = T^2$ resources (i,t), $i = 1, \ldots, T$, $t = 1, \ldots, T$. The value of resource (i,t) is KT at time t, 0 at time $t' \neq t$. Then at time t exactly T resources ((i,t), $i = 1, \ldots, T$) have non zero value KT. So an optimal solution gives at time t (i,t) to agent i. The others have value 0 so they can be placed anywhere. Then consider that Algorithm 1 wrongly allocates these other resources, in such a way that no resource is given to the same agent between time t and time $t+1$. Then solution S_0 has value TKT (KT at each time step), while S_i has value $KT + (T-1)KT = TKT$.

Of course, an optimal solution sequence is to allocate (i,t) to i at any time, with value $TKT + (T-1)TK = 2TKT - KT$. The ratio is $\frac{1}{2-1/T}$, which is the claimed ratio for $\rho = 1$.

One might note that Algorithm 1 could easily perform better on the previous instance, by optimizing the number of resources that are not reallocated between the solutions computed at times t and $t+1$. As a matter of fact, we now show that the previous result can be slightly improved in the homogeneous case when the number of agents is bounded. Indeed, in this case a permutation on the agents does not modify the Santa Claus revenue at a given time step. Then we can improve solution S_0 of algorithm 1 by trying to find a permutation minimizing the number of re-allocations when moving from solution $\tilde{O}_t$ to $\tilde{O}_{t+1}$.

To do this, consider two allocations $(\mathcal{R}_1^t, \ldots, \mathcal{R}_m^t)$ and $(\mathcal{R}_1^{t+1}, \ldots, \mathcal{R}_m^{t+1})$ of resources to agents at times t and $t+1$. Build the complete bipartite graph with $2m$ vertices $a_1^t, \ldots, a_m^t$ and $a_1^{t+1}, \ldots, a_m^{t+1}$ where the edge $(a_j^t, a_{j'}^{t+1})$ has weight $|\mathcal{R}_j^t \cap \mathcal{R}_{j'}^{t+1}|$. Consider a perfect matching on this graph. The weight of this matching is the total number of resources that remain to the same agent if we permute the second allocation according to

the matching (if $(a_j^t, a_{j'}^{t+1})$ is in the matching the agent j receives resources $\mathcal{R}_j^t$ at time t and $\mathcal{R}_{j'}^{t+1}$ at time $t+1$). Then computing a perfect matching of maximum weight in this graph gives the permutation which maximizes the number of resources that remain allocated to the same agent. Consider now Algorithm $1'$ which is the same as Algorithm 1 up to the fact that, when computing S_0, we sequentially permute solutions $\tilde{O}_t$ in order to get the maximum number of 'non modified' resources from time $t-1$ to time t.

Proposition 1. *If A is a ρ-approximation algorithm, then in the homogeneous case Algorithm $1'$ is a $\frac{T\rho-\rho/m}{(T-1)\rho+T-T/m}$-approximation algorithm.*

Proof. Given two allocations at times t and $t+1$, if we apply a random permutation (all permutations being chosen with the same probability) on the agents at time $t+1$, then a resource remains given to the same agent between time t and $t+1$ with probability $1/m$. So, in average (among all permutations) a fraction $1/m$ of resources are not re-allocated. A maximum weight perfect matching performs at least as well as average (since it is optimal), so at least a fraction $1/m$ of resources are not re-allocated in the computed solution. Then Equation (1) becomes:

$$f(S) \geq \rho\ell(S^*) + \frac{1}{m}D(S^*) \tag{3}$$

Now, summing Equation (2) for $i = 1, \ldots, T$, multiplying by $m\rho - 1$ and adding $(T-\rho)m$ times Equation (3) leads to the claimed ratio. □

As pointed out in introduction, Max-Min Fair Allocation has an approximation scheme in the homogeneous case, and a $r = \frac{1}{6+2\sqrt{10}+\epsilon}$-approximation algorithm in the case where $\ell(j,i) \in \{\ell_j, 0\}$. We get the following:

Corollary 1. *For any $\epsilon > 0$,* Over-Time Max-Min Fair Allocation *is approximable within ratio:*

- $\frac{T-1m}{2T-1-T/m} - \epsilon \xrightarrow{T\to\infty} \frac{1}{2-1/m} - \epsilon$ *in the homogenenous case;*
- $\frac{Tr}{(T-1)r+T} - \epsilon \xrightarrow{T\to\infty} \frac{1}{7+2\sqrt{10}} - \epsilon$ *in the case where $\ell_t(j,i) \in \{\ell_t(j), 0\}$.*

So in the homogeneous case the ratio is $(1/2-\epsilon)$ if the number of agents is unbounded. For 2 agents, the improvement with the matching technique allows to increase the ratio up to $(3/4-\epsilon)$.

Note that for the case where $\ell_t(j,i) \in \{\ell_t(j), 0\}$, the ratio is only slightly decreasing from the static case to the over-time setting, from (nearly) 1/12.3 to 1/13.3.

3.1.2 On-line case

In the on-line setting, we only know the instance I_t at time t. We have to build the solution at time t without knowledge on what will be the instances at times $t' > t$.

Algorithm 1 is of course no longer possible in the on-line setting. We cannot build the solution sequences, neither compare them at the beginning of the time period. We now devise an on-line algorithm which gives the same approximation ratio as the off-line one when the number of steps is unbounded. Algorithm 2 also uses as subroutine an algorithm A for the static case, and works as follows:

- Apply A on instance I_1, denote by $\tilde{O}_1$ the obtained solution. Set $O_1 = \tilde{O}_1$
- For t from 2 to T:
  - Apply A on instance I_t.
  - If the Santa Claus value of this solution is greater than Kn then set $O_t = \tilde{O}_t$.
  - Otherwise set $O_t = O_{t-1}$.
- Output $S = (O_1, \ldots, O_T)$.

Theorem 3. *If A is a ρ-approximation algorithm, then Algorithm 2 is a $\frac{\rho}{\rho+1}$-approximation algorithm.*

Proof. Let $S^* = (O_1^*, ..., O_T^*)$ be an optimal solution sequence. By construction we have:

- $\ell_1(S) \geq \rho \ell_1(O_1^*)$, and for any $t = 2, \ldots, T$, $D_{t-1}(S) + \ell_t(S)$ is at least $\rho \ell_t(O_t^*)$; so $D(S) + \ell(S) \geq \rho \ell(S^*)$.
- For any $t = 2, \ldots, T$, $D_{t-1}(S) + \ell_t(S) \geq Kn \geq D_{t-1}(S^*)$, so $D(S) + \ell(S) \geq D(S^*)$.

A combination of these two inequalities with coefficients 1 and ρ gives the claimed ratio. □

Note that the analysis is tight when $\rho = 1$, even with $T = 2$ and in the homogeneous case. Consider this example with $T = 2$, $K = 1$, with 2 agents and 2 resources. At time 1 resources have respective valuations 0,2 while at time 2 they have valuations 2,2. Suppose that the A gives resource

1 to agent 1 and resource 2 to agent 2 at time 1, and vice-versa at time 2. Then the on-line algorithm produces a solution of value 2, while of course maintaining the same resource to each agent gives value 4.

As in the offline case, this trivial example suggests the same improvement for the homogeneous case: when the algorithm applies A on instance I_t, it performs the matching technique and permutes the allocation in order to maximize the number of resources not reallocated, leading to Algorithm $2'$. Then at least a fraction of $1/m$ of the resources are not reallocated, and we get the following result (details omited).

Theorem 4. *If A is a ρ-approximation algorithm, in the homogeneous case Algorithm $2'$ is a $\frac{\rho}{\rho+1-1/m}$-approximation algorithm.*

If m is unbounded, then we get ratio $\frac{\rho}{\rho+1}$. Note that this is tight for $\rho = 1$ even in the homogeneous case, even with $T = 2$: take $n = m^2$ resources, and $K = 1$. At time 1, the first m resources have valuation 1, while the others have 0. Suppose that the algorithm chooses to give all the last $m^2 - m$ resources to agent 1. At time 2, all the resources have valuation m. If the algorithm chooses to keep the same solution it gets transition revenue Km^2, and Santa Claus revenue $1 + m$, so the value is $Km^2 + m + 1 = m^2 + m + 1$. An optimal solution at time 2 gives m resources to each agent, with Santa Claus revenue m^2. At best the transition revenue if the algorithm chooses such an optimal solution is $K(2m-1) = 2m-1$. So in all the solution computed by the algorithm has value at most $m^2 + 2m$. An optimal solution sequence consists of giving the same m resources at time 1 and 2 for each agent, with a value $m + m^2 + m^2$. The ratio goes to $1/2$ when m goes to infinity.

For the on-line case, we get:

Corollary 2. *For any $\epsilon > 0$,* Over-Time Max-Min Fair Allocation *in the on-line setting is approximable within ratio:*

- $\frac{1}{2-1/m} - \epsilon$ *in the homogenenous case;*
- $\frac{1}{7+2\sqrt{10}} - \epsilon$ *in the case where $\ell_t(j,i) \in \{\ell_t(j), 0\}$.*

3.2 The case With Restrictions

We consider the variant where some resources cannot be allocated to some agents at some point of the time. In that case, for every $i \in M$, $j \in N$, we have $l_t(j,i) = -\infty$ if the agent i cannot receive the resource j at time t.

3.2.1 Off-line case

As in the case without restrictions, we would like to build a first solution S_0 with good Santa Claus revenue $\ell(S_0)$, and another solution S_1 with good transition revenue $D(S_1)$. However, due to the restrictions, the maximal transition revenue between two consecutive instances is not always nK as it was in the case without restrictions. A first question is then to devise, if possible, a procedure which computes a solution sequence maximizing the transition revenue.

To do this, let us consider Algorithm 3 which greedily affects each resource to the agent where it can stay for the longest period of time. More precisely, let $Q(j, i, t)$ be the number of (consecutive) time steps during which resource j can be allocated to agent i starting at time t ($Q(j, i, t) = 0$ if j cannot be given to i at time t). Then Algorithm 3 initially allocates resource j to the agent i on which it can stay the longer ($\text{argmax}_i\, Q(j, i, 1)$), and keep it as long as possible (i.e. $t = Q(j, i, 1)$ steps). Then at time $t + 1$, j has to change, and Algorithm 3 allocates it to $\text{argmax}_i\, Q(j, i, t + 1)$, and keeps it as long as possible (i.e. $Q(j, i, t + 1)$ steps), and so on.

We have the following property.

Lemma 1. *Algorithm 3 outputs a solution sequence with maximum transition revenue.*

Proof. We prove that this is true for each resource. Let S be the solution returned by Algorithm 3. Let $D_t(S, j)$ be the contribution of the resource j to the transition revenue of S up to time t, and $D_t(S', j)$ the contribution for another solution sequence S'.

- At $t = 1$, $D_t(S, j) = 0 = D_t(S', j)$.
- Suppose that $D_p(S, j) \geq D_p(S', j)$ for $p = 1, \ldots, t$, and consider time $t+1$. If j remains to the same agent i in S between time t and $t+1$, or if j moves in S' between t and $t+1$, then clearly $D_{t+1}(S, j) \geq D_{t+1}(S', j)$. Otherwise, let k be the last time at which j moves in S' (between solution at time $k - 1$ and k). Since j can stay in S' at time $t + 1$ and not in S, by principle of the algorithm, j has been given to the same agent i in S before time k, it was already given to i at time $k - 1$. Since $D_{k-1}(S, j) \geq D_{k-1}(S', j)$, and between $k - 1$ and k, j moves in S' but not in S, we have $D_k(S, j) \geq D_k(S', j) + K$. Then the next modification is between t and $t + 1$, so $D_t(S, j) \geq D_t(S', j) + K$ and then $D_{t+1}(S, j) \geq D_{t+1}(S', j)$.

This is true for each resource, so the result follows. □

Then, consider Algorithm 3′ which computes two solutions S_1 and S_2, and outputs the best one. S_1 is the solution given by Algorithm 3, and $S_2 = (\tilde{O}_1, ..., \tilde{O}_T)$, where $\tilde{O}_t$ is obtained by applying a subroutine algorithm A for the static case of the Max-Min Fair Allocation problem on instance I_t.

Theorem 5. *If A is a (polytime) ρ-approximation algorithm for the static case, then Algorithm 3′ is a $\frac{\rho}{\rho+1}$-approximation for the off-line problem with restrictions.*

Proof. Let $S^* = (O_1^*, \ldots, O_T^*)$ be an optimal solution sequence. By Lemma 1, $D(S_1)$ is maximum so $D(S_1) \geq D(S^*)$. Consequently, $f(S_1) \geq D(S^*)$. We also have $\ell_t(\tilde{O}_t) \geq \rho \cdot \ell_t(O_t^*)$ for every $t \in \{1, ..., T\}$, so $\ell(S_2) = \sum_{t=1}^{T} \ell_t(\tilde{O}_t) \geq \rho \cdot \sum_{t=1}^{T} \ell_t(O_t^*) = \rho \cdot \ell(S^*)$, and this gives $f(S_2) \geq \rho \cdot \ell(S^*)$. This implies $\rho \cdot f(S_1) + f(S_2) \geq \rho \cdot (\ell(S^*) + D(S^*)) \geq \rho \cdot f(S^*)$. Then, $(\rho + 1) \cdot f(S) \geq \rho \cdot f(S^*)$.

So, S is a $\frac{\rho}{\rho+1}$-approximate solution sequence.

□

Let us consider an instance with $T = 2$ time steps, three agents and three resources. There is no restriction at time 1, but at time 2 resource i cannot be given to agent i. Their respective valuations are 0,0,0 at time 1, and 3,3,3 at time 2. $K = 1$. Then Algorithm 3′ may choose a solution S_1 where resources 1 and 3 are given to agent 2 and resource 2 to agent 1, both at time 1 and 2. It has transition revenue 3, but $\ell(S_1) = 0$. S_2 may give resource i to agent i at time 1, and then at time 2 a solution of value 3 (one resource per agent) but with no transition revenue (because of the constraints). An optimal solution would be to give resource 1 to agent 2, 2 to 3 and 3 to 1 both at time 1 and 2, with value $3 + 3 = 6$. The ratio is $1/2$ (with $\rho = 1$).

3.2.2 On-line case

Now, we consider the on-line case where at time t we have not any knowledge on what will be happen at time $t' > t$. Interestingly, while in the unrestricted case we can devise an on-line algorithm with (nearly) the same ratio as in the off-line case, we show that in the case with restrictions being blind makes it impossible to get any algorithm with guaranteed competitive ratio.

Proposition 2. *There is no on-line algorithm with bounded competitive ratio even for the homogeneous case with restrictions.*

Proof. Consider two agents, one single resource, and K the transition reward. Let us also consider two time-steps, $T = 2$. At time $t = 1$, the valuation of the resource is 0 for both agents. The resource could be allocated to any agent. Suppose w.l.o.g. that it is allocated to the first agent by an on-line algorithm. At $t = 2$, the resource is allowed to be allocated only to the second agent, and has valuation 0 for this agent. Hence, the on-line algorithm is forced to allocate the resource to agent two. In this way, it gets a revenue of 0, while the optimum would be to allocate the resource to the second agent in both timesteps resulting to a total revenue of K. □

3.2.3 1-lookahead case

As mentioned in the introduction, in the setting of over-time optimization it seems reasonable to consider limited lookahead: we may have rather good information on what will happen tomorrow or in a few days (time steps) from today, while we may have no clue on what will be the situation one month from now. Motivated by the dramatic difference on what can be obtained in the off-line case (ratio $\rho/(\rho + 1)$, Theorem 5) and in the on-line case (no competitive ratio), we consider now the situation where we have lookahead 1: at time t when we have to decide the allocation of the resources for instance I_t, we know also I_{t+1}. We show in the following that this is sufficient to get a good (constant) approximation ratio.

We use the same idea as previously, namely at each time we try to maximize either the Santa Claus revenue, or the transition revenue. However, because of the restrictions, we introduce a third option: choose an allocation maximizing the transition revenue between the current instance and the instance that will follow. We choose the third option when the potential transition revenue is at least two times larger than the revenue of the two other options.

Let us now present Algorithm 4, where A is, as previously, an algorithm for the static problem. Also, for every $t \in \{1, \ldots, T-1\}$, P_t^1 and P_t^2 denote two allocations for I_t and I_{t+1} respectively, such that $D_t(P_t^1, P_t^2)$ is maximum (simply affect a resource to the same agent at time t and $t+1$ if possible). We denote by $D_t^{\max} = D_t(P_t^1, P_t^2)$. The solution $S = (O_1, \ldots, O_T)$ is built in the following way:

- At $t = 1$: compute $\tilde{O}_1 = A(I_1)$ and two allocations P_1^1 on I_1 and P_1^2 on I_2. We compute $\max\left(l_1(\tilde{O}_1), \frac{D_1^{\max}}{2}\right)$ and we choose for O_1 between $\tilde{O}_1$ and P_1^1 accordingly ($O_1 = \tilde{O}_1$ if $l_1(\tilde{O}_1) \geq \frac{D_1^{\max}}{2}$, otherwise $O_1 = P_1^1$).

- At time $t = 2, \ldots, T-1$: Similarly, compute $\tilde{O}_t = A(I_t)$ and two allocations P_t^1 on I_t and P_t^2 on I_2. We distinguish two cases:
  - If at time $t-1$ we have selected $O_{t-1} = P_{t-1}^1$, then, at t, we compute $\max\left(D_{t-1}^{\max}, \ell_t(\tilde{O}_t), \frac{D_t^{\max}}{2}\right)$ and we choose for O_t between P_{t-1}^2, $\tilde{O}_t$ and P_t^1 accordingly.
  - If not, then we compute $\max\left(l_t(\tilde{O}_t), \frac{D_t^{\max}}{2}\right)$ and we choose for O_t between $\tilde{O}_t$ and P_t^1 accordingly.
- At time T, we do the same up to the fact that there is no $D_T^{\max}$ (no P_T^1, P_T^2).

Theorem 6. *If A is a (polytime) ρ-approximation algorithm for the static case, then Algorithm 4 has an approximation ratio of $\frac{\rho}{4\rho+2-\frac{2\rho+1}{2^{T-1}}} \xrightarrow{T\to\infty} \frac{\rho}{4\rho+2}$.*

Proof. Let V be the set of time-steps where we selected $O_t = P_{t-1}^2$ or $O_t = \tilde{O}_t$ (but not $O_t = P_t^1$). Note that $T \in V$. In the proof we partition the time period into $|V|$ periods which end at some time $t \in V$. Intuitively, we prove the claimed ratio in each of these sub-periods.

Formally, let $u, u+1, ..., v-1, v$ be consecutive time-steps such that $u, u+1, ..., v-1 \notin V$, and $v \in V$. Suppose first that $u \leq v-1$ (meaning $v-1 \notin V$). For every $t \in \{u, ..., v-1\}$, we have selected $O_t = P_t^1$. This means that:

- $\frac{D_t^{\max}}{2} \geq D_{t-1}^{\max}$, implying:
$$D_{v-1}^{\max} \geq 2D_{v-2}^{\max} \geq 2^2 D_{v-3}^{\max} \geq ... \geq 2^{v-u-1} D_u^{\max} \tag{4}$$
- $D_t^{\max} \geq 2\ell_t(\tilde{O}_t)$. Since from the previous item $D_{v-1}^{\max} \geq 2^{v-t-1} D_t^{\max}$, we have :
$$D_{v-1}^{\max} \geq 2^{v-t} \ell_t(\tilde{O}_t) \tag{5}$$

Summing Equation (4) for t from u to $v-1$ we get:

$$\sum_{t=u}^{v-1} D_t^{\max} \leq D_{v-1}^{\max} \sum_{t=u}^{v-1} \frac{1}{2^{v-t-1}} = D_{v-1}^{\max}\left(2 - \frac{1}{2^{v-u-1}}\right) \tag{6}$$

Doing the same with Equation (5) we get:

$$\sum_{t=u}^{v-1} \ell_t(\tilde{O}_t) \leq D_{v-1}^{\max} \sum_{t=u}^{v-1} \frac{1}{2^{v-t}} = D_{v-1}^{\max}\left(1 - \frac{1}{2^{v-u}}\right) \tag{7}$$

At time v, the algorithm chooses either P^2_{v-1} or $\tilde{O}_v$, with a reward r_v at least $\max\left(D^{max}_{v-1}, \ell_v(\tilde{O}_v), \frac{D^{max}_v}{2}\right)$. This reward r_v verifies thanks to Equation (6) (and using $v-u \leq T-1$)

$$\sum_{t=u}^{v} D^{\max}_t \leq D^{\max}_{v-1}\left(2 - \frac{1}{2^{v-u-1}}\right) + D^{\max}_v \leq r_v\left(4 - \frac{1}{2^{T-2}}\right) \tag{8}$$

Thanks to Equation (7) we have also

$$\sum_{t=u}^{v} \ell_t(\tilde{O}_t) \leq D^{\max}_{v-1}\left(1 - \frac{1}{2^{v-u}}\right) + \ell_v(\tilde{O}_v) \leq r_v\left(2 - \frac{1}{2^{T-1}}\right) \tag{9}$$

Let $S^* = (O^*_1, ..., O^*_T)$ be an optimal solution. Let us denote $f_{u,v}(S) = \sum_{t=u}^{v}(D_t(S^*) + \ell_t(O^*_t))$ (with by convention $D_T(S^*) = 0$). Since $D_t(S^*) \leq D^{\max}_t$ and $\ell_t(O^*_t) \leq \frac{\ell_t(\tilde{O}_t)}{\rho}$, we have:

$$f_{u,v}(S^*) \leq \sum_{t=u}^{v}\left(D^{\max}_t + \frac{\ell_t(\tilde{O}_t)}{\rho}\right) \leq r_v\left(4 + \frac{2}{\rho} - \frac{2+1/\rho}{2^{T-1}}\right) \tag{10}$$

Note that this is also trivially true if $u = v$.[1] Hence, in each of our sub-periods, we get a reward which is a fraction of at least $\frac{1}{4+\frac{2}{\rho}-\frac{2+1/\rho}{2^{T-1}}}$ of the reward of the optimal solution on the same period. The result follows. □

Consider an example with T time steps, such that for $t \in \{1, ..., T-1\}$, $\ell^{\max}_t = \frac{D^{\max}_t}{2} = 2^t$, $\ell^{\max}_T = 2^T + 1$, and $D^{\max}_T = 0$. We have then $\frac{D^{\max}_t}{2} \geq \ell^{\max}_t$ and $\frac{D^{\max}_t}{2} \geq D^{\max}_{t-1}$ for $t \in \{1, ..., T-1\}$ and also $\ell^{\max}_T > D^{\max}_{T-1}$ and $\ell^{\max}_T > D^{\max}_T$. Using an exact algorithm A, the solution obtained by the algorithm would be $S = (P^1_1, ..., P^1_{T-1}, \tilde{O}_T)$. This would give $1 = u, ..., T-1 \notin V$ and $T = v \in V$. The approximation ratio of S in this case tends to $\frac{1}{6}$ when $T \to \infty$, indicating that the previous analysis is tight.

From Theorem 6 we get:

Corollary 3. *For any $\epsilon > 0$,* Over-Time Max-Min Fair Allocation *with restriction and lookahead 1 is approximable within (constant) ratio:*

- $\frac{1}{6} - \epsilon$ *in the homogenenous case;*
- $\frac{1}{16+4\sqrt{10}} - \epsilon$ *in the case where $\ell_t(j,i) \in \{\ell_t(j), 0\}$.*

[1] we get $r_v \geq \max\left(l_v(\tilde{O}_v), \frac{D^{max}_v}{2}\right)$, so $f_{v,v}(S^*) \leq (2 + \frac{1}{\rho})r_v$.

4 Conclusion

We have provided several constant ratio approximation algorithms for an over-time version of the Max-Min Fair Allocation, based on a recently introduced [GTW14] model which combines quality and stability of allocations over time. One striking point of this work is the introduction in this setting of the notion of k-lookahead, which allows tradeoffs between on-line and off-line settings. We show an application of this concept for the problem we consider, since 1-lookahead is sufficient to reach constant approximation algorithm for the problem with restriction, while no ratio can be guaranteed in the on-line case. We do believe that this notion is worth being studied for other multiagent optimization problems over time. Another direction that might be studied deals with the agregation function: in this work we made a sum of the value of solutions at each time step, and of the transition revenues. There are many other ways to combine two objectives (bounding the number of modifications, finding Pareto-efficient solutions,...) that could be investigated. Finally, dealing with allocation of resources, other criteria or properties (fairness in average and not step by step, envy-freeness, non manipulability,...) could be also considered.

Acknowledgements

This work has been supported by the project ANR-14-CE24-0007-01 "CoCoRICo-CoDec" funded by ANR. It benefited also from the support of the FMJH program "Gaspard Monge in optimization and Operation Research" and from the support to this program from EDF, via the project 2016-1760H/C16/1507 "Stability versus Optimality in Dynamic Environment Algorithmics".

References

[AKS17] Chidambaram Annamalai, Christos Kalaitzis, and Ola Svensson. Combinatorial algorithm for restricted max-min fair allocation. *ACM Trans. Algorithms*, 13(3):37:1–37:28, 2017.

[AS10] Arash Asadpour and Amin Saberi. An approximation algorithm for max-min fair allocation of indivisible goods. *SIAM J. Comput.*, 39(7):2970–2989, 2010.

[BCG09] MohammadHossein Bateni, Moses Charikar, and Venkatesan Guruswami. Maxmin allocation via degree lower-bounded ar-

borescences. In *Proceedings of the 41st Annual ACM Symposium on Theory of Computing, STOC 2009, Bethesda, MD, USA, May 31 - June 2, 2009*, pages 543–552, 2009.

[BCM16] Sylvain Bouveret, Yann Chevaleyre, and Nicolas Maudet. Fair allocation of indivisible goods. In Felix Brandt, Vincent Conitzer, Ulle Endriss, Jérôme Lang, and Ariel D. Procaccia, editors, *Handbook of Computational Social Choice*, pages 284–310. Cambridge University Press, 2016.

[BD05] Ivona Bezáková and Varsha Dani. Allocating indivisible goods. *SIGecom Exchanges*, 5(3):11–18, 2005.

[BLFL05] Sylvain Bouveret, Michel Lemaître, Hélène Fargier, and Jérôme Lang. Allocation of indivisible goods: a general model and some complexity results. In Frank Dignum, Virginia Dignum, Sven Koenig, Sarit Kraus, Munindar P. Singh, and Michael Wooldridge, editors, *4th International Joint Conference on Autonomous Agents and Multiagent Systems (AAMAS 2005), July 25-29, 2005, Utrecht, The Netherlands*, pages 1309–1310. ACM, 2005.

[BS06] Nikhil Bansal and Maxim Sviridenko. The santa claus problem. In *Proceedings of the 38th Annual ACM Symposium on Theory of Computing, Seattle, WA, USA, May 21-23, 2006*, pages 31–40, 2006.

[CCK09] Deeparnab Chakrabarty, Julia Chuzhoy, and Sanjeev Khanna. On allocating goods to maximize fairness. In *50th Annual IEEE Symposium on Foundations of Computer Science, FOCS 2009, October 25-27, 2009, Atlanta, Georgia, USA*, pages 107–116, 2009.

[CKW92] János Csirik, Hans Kellerer, and Gerhard Woeginger. The exact lpt-bound for maximizing the minimum completion time. *Operations Research Letters*, 11(5):281–287, 1992.

[DFL82] Bryan L. Deuermeyer, Donald K. Friesen, and Michael A. Langston. Scheduling to maximize the minimum processor finish time in a multiprocessor system. *SIAM Journal on Algebraic Discrete Methods*, 3(2):190–196, 1982.

[EMS14] David Eisenstat, Claire Mathieu, and Nicolas Schabanel. Facility location in evolving metrics. In *Automata, Languages, and Programming - 41st International Colloquium, ICALP 2014, Copenhagen, Denmark, July 8-11, 2014, Proceedings, Part II*, pages 459–470, 2014.

[ES04] Leah Epstein and Jirí Sgall. Approximation schemes for scheduling on uniformly related and identical parallel machines. *Algorithmica*, 39(1):43–57, 2004.

[GJ79] M. R. Garey and David S. Johnson. *Computers and Intractability: A Guide to the Theory of NP-Completeness*. W. H. Freeman, 1979.

[GP10] Boris Golden and Patrice Perny. Infinite order lorenz dominance for fair multiagent optimization. In Wiebe van der Hoek, Gal A. Kaminka, Yves Lespérance, Michael Luck, and Sandip Sen, editors, *9th International Conference on Autonomous Agents and Multiagent Systems (AAMAS 2010), Toronto, Canada, May 10-14, 2010, Volume 1-3*, pages 383–390. IFAAMAS, 2010.

[GTW14] Anupam Gupta, Kunal Talwar, and Udi Wieder. Changing bases: Multistage optimization for matroids and matchings. In *Automata, Languages, and Programming - 41st International Colloquium, ICALP 2014, Copenhagen, Denmark, July 8-11, 2014, Proceedings, Part I*, pages 563–575, 2014.

[JMD05] Geert Jonker, John-Jules Ch. Meyer, and Frank Dignum. Towards a market mechanism for airport traffic control. In Carlos Bento, Amílcar Cardoso, and Gaël Dias, editors, *Progress in Artificial Intelligence, 12th Portuguese Conference on Artificial Intelligence, EPIA 2005, Covilhã, Portugal, December 5-8, 2005, Proceedings*, volume 3808 of *Lecture Notes in Computer Science*, pages 500–511. Springer, 2005.

[LMMS04] Richard J. Lipton, Evangelos Markakis, Elchanan Mossel, and Amin Saberi. On approximately fair allocations of indivisible goods. In *Proceedings 5th ACM Conference on Electronic Commerce (EC-2004), New York, NY, USA, May 17-20, 2004*, pages 125–131, 2004.

[Lus12] Hanan Luss. *Equitable Resource Allocation: Models, Algorithms and Applications*. Wiley, 2012.

[LVB99] Michel Lemaître, Gérard Verfaillie, and Nicolas Bataille. Exploiting a common property resource under a fairness constraint: a case study. In Thomas Dean, editor, *Proceedings of the Sixteenth International Joint Conference on Artificial Intelligence, IJCAI 99, Stockholm, Sweden, July 31 - August 6, 1999. 2 Volumes, 1450 pages*, pages 206–211. Morgan Kaufmann, 1999.

[OV09] Marcel Ochel and Berthold Vöcking. Approximability of OFDMA scheduling. In *Algorithms - ESA 2009, 17th Annual European Symposium, Copenhagen, Denmark, September 7-9, 2009. Proceedings*, pages 385–396, 2009.

[SS10] Barna Saha and Aravind Srinivasan. A new approximation technique for resource-allocation problems. In *Innovations in Computer Science - ICS 2010, Tsinghua University, Beijing, China, January 5-7, 2010. Proceedings*, pages 342–357, 2010.

[Woe97] Gerhard J. Woeginger. A polynomial-time approximation scheme for maximizing the minimum machine completion time. *Operations Research Letters*, 20(4):149–154, 1997.

[WS11] David P. Williamson and David B. Shmoys. *The Design of Approximation Algorithms*. Cambridge University Press, 2011.