Approximation and Online Algorithms for Selected Network Optimization Problems

Von der Fakultät für Mathematik, Informatik und Naturwissenschaften der
RWTH Aachen University zur Erlangung des akademischen Grades
eines Doktors der Naturwissenschaften genehmigte Dissertation

vorgelegt von

Diplom-Informatiker

Marcel Ochel

aus Frechen

Berichter: Universitätsprofessor Dr. Berthold Vöcking
Universitätsprofessor Dr. Rudolf Mathar

Tag der mündlichen Prüfung: 30. April 2013

Diese Dissertation ist auf den Internetseiten der Hochschulbibliothek online verfügbar.

Abstract

Optimizing energy efficiency is a fundamentally important issue in modern communication networks. In this thesis we study the objective of optimal energy usage on two general classes of wireless network optimization problems, namely *network lifetime maximization* and *OFDMA scheduling*.

Orthogonal frequency division multiplexing (OFDM) has become an increasingly popular technology in the area of broadband communication. In one of the associated fundamental scheduling problems, data has to be sent from a base station to a collection of terminals using a given set of communication channels where, however, each channel can be assigned to at most one designated terminal. Aside from determining an appropriate feasible assignment, each channel needs to be allocated a communication rate such that —summing over all channels assigned to a single terminal— the total data rate satisfies at least the demand of that terminal. In order to guarantee a successful transmission of data at a certain rate, the transmission power on the corresponding channel has to be large enough. The objective is to minimize the total sum of energy requirements while satisfying the demands of each terminal.

This and related problems (called *OFDMA scheduling problems*) have been intensively studied by the engineering sciences; however, we are the first to conduct a theoretical worst-case analysis of this practically relevant class of network optimization problems for which no lower nor upper bounds on the approximability were known before.

While we will show that —already on a very restricted subclass— the problem of minimizing the required transmission power for given data rate demands can not be approximated by any (polynomial time computable) factor, we will also see that the dual objective of proportionally maximizing the data rates for a given total transmission power is approximable arbitrarily close to the optimum on this subclass.

On general instances, OFDMA rate-maximization will be shown to be inapproximable by any factor better than $1/2$. Complementarily, a simple matching-based $1/m$-approximation algorithm will be derived (with m denoting the number of channels). Also a more involved convex programming approach is considered that (aside from its usefulness to derive better performance guarantees for special subclasses) can be used to compute $1/2$-approximate half-integral solutions in the general case.

In the second part of this dissertation, we continue with our theoretical study of energy minimization in a more implicit context: Motivated by the behavior of battery depletion in wireless sensor networks we introduce and analyze a novel online model for general linear packing problems, applicable to a broad class of *network lifetime maximization problems*.

Network lifetime maximization aims at sustaining a prespecified communication requirement as long as possible in a given communication network with limited power supplies. Though there exists a vast number of approximation approaches for a large variety of lifetime maximization problems, all previous formulations of network lifetime optimization assume perfect a priori knowledge about the status and future behavior of the energy supplies of each single network node. We relax this assumption by introducing an appropriate multiplicative uncertainty parameter which initially allows only a rough estimation of the energy capacities that becomes more and more accurate when exhausting the corresponding resources.

The definition of the online model and the competitive online algorithms developed in this work are independent of the underlying problem structure as long as the corresponding optimization problem can be formulated in terms of a general linear packing program. An advantageous property of the proposed online algorithms is that they do not require any explicit linear programming formulation; hence, they are also suited when algorithmic efficiency is an issue. As the algorithms additionally are approximation factor preserving they can be used as a framework for algorithmically intractable problem classes by applying appropriate offline approximation algorithms.

The proven competitive factors of the more sophisticated algorithms outperform naive online algorithms by at least a logarithmic factor with respect to the fixed uncertainty parameter. Complementarily, we show bounds on the competitiveness of arbitrary deterministic or randomized online algorithms in this online model.

Zusammenfassung

In modernen Kommunikationsnetzen kommt der Optimierung des Energiebedarfs eine hohe Bedeutung zu. In dieser Dissertation untersuchen wir das Ziel optimaler Energienutzung auf zwei fundamentalen Klassen von Netzwerkoptimierungsproblemen: der Netzwerk-Lebenszeitmaximierung und der OFDMA Kanal- und Ratenzuweisung.

Wir untersuchen Energieminimierungs- und Datenratenmaximierungsprobleme zunächst im Kontext von Orthogonal-Frequency-Division-Multiplexing (OFDM). In einer grundlegenden Variante des zugehörigen OFDM(A)-Scheduling-Problems sollen Informationen mit einer vorgegebenen Datenrate von einem Sender zu einer Menge von Empfängern auf möglichst energieeffiziente Weise übermittelt werden, wobei eine Anzahl von verschiedenen Frequenzbereichen (Kanälen) zur Verfügung steht. Da verschiedene Informationen an die jeweiligen Empfänger zu verschicken sind, können (aufgrund der entstehenden Interferenz) einzelne Frequenzbereiche nur jeweils einem Empfänger zugewiesen werden. Gesucht ist nun eine Zuweisung der Kanäle zu den Empfängern, so dass eine Datenratenverteilung auf die einzelnen Kanäle mit minimalem Energiebedarf bestimmt werden kann, welche die gegebenen Zieldatenraten für jeden Empfänger erfüllt.

In der vorliegenden Arbeit wird erstmalig eine theoretische Worst-Case-Analyse dieser praktisch relevanten Klasse von Netzwerkoptimierungsproblemen durchgeführt, zu der bisher noch keine Komplexitäts- und Approximationsschranken bekannt sind. Wir zeigen, dass obiges Problem bereits auf einer eingeschränkten Instanzklasse für keinen polynomialzeitberechenbaren Faktor approximierbar ist, dass jedoch die natürliche duale Variante dieses Problems, welche darin besteht, für ein gegebenes Gesamtenergiebudget die Zieldatenraten proportional zu maximieren, die optimale Lösung in dieser Klasse beliebig gut approximiert werden kann.

Im allgemeinen Fall der OFDMA-Ratenmaximierung beweisen wir Inapproximierbarkeit für jeden Faktor besser als $1/2$. Neben einem matching-basierten Algorithmus, welcher einen Approximationsfaktor von $1/m$ (linear in der Kanalanzahl) erreicht, wird ein Ansatz entwickelt, welcher durch Rundung konvexer Subprobleme eine halbintegrale Lösung der Güte $1/2$ berechnet.

Im zweiten Teil dieser Dissertation betrachten wir anschließend Energieoptimierung in einer verstärkt impliziten Form: Motiviert vom Entladeverhalten batteriegespeister Sendegeräte in Drahtlos-Sensornetzwerken untersuchen wir ein neuartiges Online-Modell für allgemeine lineare Pack- und Netzwerk-Lebenszeit-Optimierungsprobleme. Von den Ungenauigkeiten in der Bestimmung der noch zur Verfügung stehenden Energiemenge inspiriert, entwickeln wir ein Online-Modell, welches (im Gegensatz zu bereits existierenden Modellen) erlaubt, Ungewissheiten in Bezug auf die verfügbaren Kapazitäten zu modellieren.

Im Rahmen kompetitiver Online-Analyse analysieren wir Algorithmen mit garantierter Lösungsgüte. Die entworfenen Online-Algorithmen weisen die Eigenschaft auf, dass sie auch auf kombinatorischen Packproblemen, welche nicht polynomiell durch ein lineares Programm darstellbar sind, angewendet werden können, da sie nur allgemeine Eigenschaften der zugrunde liegenden linearen Struktur ausnutzen. Dies macht sie insbesondere interessant für den Einsatz bei Netzwerk-Lebenszeit-Optimierungsproblemen. Da sie zudem (gegebenenfalls bekannte) Offline-Approximationsalgorithmen als Subroutine einbinden können und die Online-Gütegarantie dabei nur um den entsprechenden Faktor abnimmt, sind sie auch effizient auf NP-vollständige Probleme anwendbar.

Die gezeigten Gütegarantien der entsprechenden Online-Algorithmen übersteigen naivere Herangehensweisen mindestens um einen logarithmischen Faktor (bzgl. des gegebenen Ungenauigkeits-Parameters). Komplementär zeigen wir eine obere Schranke für beliebige deterministische und randomisierte Online-Algorithmen in diesem Online-Modell.

Acknowledgements

I am very grateful to my advisor, Berthold Vöcking, without whom it would not have been possible to write this thesis. His valuable motivation, ideas, and support always encouraged me to tackle the challenges occurring during the research and writing process. Thanks also go to Rudolf Mathar for his interest in this work and acting as my co-referee.

Furthermore, I wish to thank all my colleagues at the algorithms and complexity group at RWTH Aachen University, especially Sascha Geulen, Klaus Radke, Benjamin Ries, Erika Schlebusch, and Walter Unger for proofreading and many helpful suggestions. I would also like to express thanks to Piotr Krysta at the University of Liverpool for several fruitful discussions on machine scheduling and related optimization problems.

Contents

1 Introduction **1**
- 1.1 Previous Work and Our Results . . . 4
  - 1.1.1 Related Online Models for General Linear Packing Problems . . . 4
  - 1.1.2 Our Results on Online Packing with Gradually Improving Capacity Estimations . . . 5
  - 1.1.3 Related Work on OFDM/OFDMA Scheduling . . . 6
  - 1.1.4 Our Results on OFDMA Scheduling . . . 8
- 1.2 Outline and Bibliographical Notes . . . 8

2 Orthogonal Frequency Division Multiplexing and Related Optimization Problems **11**
- 2.1 Single-User OFDM . . . 11
- 2.2 Multi-User OFDMA . . . 13
- 2.3 Santa Claus Scheduling . . . 15
  - 2.3.1 Related Work on Santa Claus Scheduling . . . 16
- 2.4 Some Basic Properties of Convex Functions . . . 17

3 Algorithmic Tractability of OFDMA Scheduling **19**
- 3.1 Efficient Solvability of Single-User OFDM Scheduling . . . 20
  - 3.1.1 Optimal Solutions to Instances with the Shannon Rate-Power Function 20
  - 3.1.2 Water-filling for Single-User OFDM with Capacity Limits . . . 21
- 3.2 NP-completeness of OFDMA Scheduling . . . 26
- 3.3 Bounds on the Approximability of OFDMA Scheduling . . . 29
  - 3.3.1 Inapproximability of Min-Power OFDMA with Uniformly Related Channel Costs . . . 31
- 3.4 OFDMA Scheduling with Linear Rate-Power Functions . . . 33
- 3.5 OFDMA with Uniform and Binary Channel Gains . . . 36
- 3.6 A Convex Fractional Relaxation of OFDMA Scheduling . . . 41

4 Approximation Algorithms for Max-Rate OFDMA Scheduling **45**
- 4.1 Uniformly Related Channel Costs . . . 45
  - 4.1.1 A $1/2$-Approximation-Algorithm for Non-uniform Demands . . . 46
    - 4.1.1.1 Speed-Up for Uniform Demands . . . 49
  - 4.1.2 A $2/3$-Approximation-Algorithm for Uniform Demands . . . 50
  - 4.1.3 Polynomial Time Approximation Schemes . . . 51
    - 4.1.3.1 PTAS for Uniform Demands . . . 53
    - 4.1.3.2 Derived PTAS for Santa Claus Scheduling . . . 55
    - 4.1.3.3 Modular Reduction to Santa Claus Scheduling . . . 55
    - 4.1.3.4 PTAS for Non-Uniform Demands . . . 57
- 4.2 Restricted Assignment Model . . . 61
- 4.3 Unrelated Channel Costs . . . 62
  - 4.3.1 A Matching-based $1/m$-Approximation . . . 62
  - 4.3.2 An Approach Based on Rounding the Convex Programming Solution . 64
    - 4.3.2.1 A $1/2$-approximate Half-Integral Solution . . . 68

4.3.2.2 A $1/n$-Approximation for Cycle-free Assignment Graphs and Restricted Assignment Instances with Uniform Demands . . 70
4.4 Summary . . . 73

5 An Online Model Capturing Gradually Improving Capacity Estimations 75
5.1 Introduction to the Online Model . . . 75
5.1.1 Formal Model for the Competitive Analysis . . . 76
5.2 Combinatorial Optimization . . . 77
5.2.1 Total Load Minimization Problems . . . 78
5.2.2 Linear Programming . . . 79
5.2.3 Combinatorial Packing Problems . . . 79
5.2.3.1 Maximum Flow . . . 80
5.2.3.2 Packing Spanning Trees . . . 81
5.3 Network Lifetime Maximization . . . 82
5.3.1 Wireless Communication Networks . . . 83
5.3.2 Topology Requirements . . . 84
5.3.3 Integrality Aspects . . . 85
5.3.4 Relation between Lifetime Maximization and Power Minimization . . 85
5.3.5 Short Survey on Known Results . . . 88

6 Competitive Analysis of the Online Model 93
6.1 The GREEDY-Algorithm . . . 94
6.2 The EQUITABLE-Algorithm . . . 95
6.2.1 Competitiveness of EQUITABLE . . . 95
6.2.2 Upper Bound on the Competitiveness of EQUITABLE . . . 97
6.3 The ENVISAGE-Algorithm . . . 98
6.4 An $O(1/\sqrt{\alpha})$ Upper Bound for Deterministic Online Algorithms . . . 100
6.5 The BALANCING-Algorithm . . . 100
6.6 Applicability to Network Lifetime Maximization . . . 102
6.6.1 Approximation Algorithms . . . 103
6.6.2 Randomized Rounding . . . 103

7 Competitiveness for Classic Combinatorial Packing Problems 105
7.1 Packing Spanning Trees . . . 105
7.1.1 A General $O(1/\sqrt{\alpha})$ Upper Bound for Randomized Online Algorithms . 105
7.1.2 Worst-Case Instances for Specific Algorithms . . . 109
7.1.2.1 A $2/\alpha$ Upper Bound on GREEDY . . . 109
7.1.2.2 An $O(\ln(\alpha)/\alpha)$ Upper Bound on EQUITABLE . . . 110
7.1.2.3 Non-Competitiveness of BALANCING . . . 110
7.2 Maximum Flow . . . 111
7.2.1 A Deterministic $O(1/\sqrt{\alpha})$ Upper Bound . . . 111
7.2.2 Worst-Case Instances for Specific Algorithms . . . 112
7.2.2.1 A $1/\alpha$ Upper Bound on GREEDY . . . 113
7.2.2.2 An $O(\ln(\alpha)/\alpha)$ Upper Bound on EQUITABLE . . . 113
7.2.2.3 Non-Competitiveness of BALANCING . . . 114
7.3 Summary . . . 114

8 Conclusions and Open Problems 115

Chapter 1

Introduction

When considering energy optimization in modern wireless communication networks, there are a lot of practical optimization problems that are either algorithmically intractable, have uncertain side constraints or even incorporate both of these challenges.

A way to deal with the first kind of problems is to discard the goal of exact solvability and settle for finding good suboptimal solutions. Pragmatically, practitioners try to develop good heuristics that — while giving no guarantee on the quality of the solution in general — show good results in simulations. The more theoretical approach of determining provable worst-case bounds on the approximability of an optimization problem will be the basis for this thesis.

However, not only limitations in computational power have to be considered when analyzing the algorithmic solvability of practical problem instances. When dealing with problems where the input is not completely known a priori or may be subject to uncertainty, the definition of online models provides a basis for theoretically analyzing the impact of this lack of knowledge and allows to develop algorithms coping with the given kind of uncertainty.

This dissertation covers the described aspects in the context of two practically relevant classes of optimization problems with the common goal of energy optimization in (wireless) communication networks.

Optimizing energy efficiency is inherent in many network optimization problems and ranges from the straightforward minimization of total energy cost, noise, or interference to more implicit formulations like maximizing the duration (lifetime) a communication network with limited energy resources is able to operate.

One of the most fundamental energy optimization problems in the context of wireless networks arises when transmitting data at a fixed rate from a given sender to a designated receiver while aiming to minimize the total energy consumption. As modern high-speed communication systems are based on *multi-carrier transmission* in order to increase bandwidth efficiency, a rate allocation to the sub-carriers can dynamically be chosen, which results in a continuous or discrete optimization problem (depending on the considered model).

In classic *single-carrier* communication, data is transmitted as a *serial* sequence of communication symbols. Thereby, higher data rates imply shorter durations of each single communication symbol. However, short symbol durations render the transmission extremely susceptible to *inter-symbol interference*, which in wireless communication is mainly caused by the delay spread of a signal traveling on multiple paths from the sender to the receiver (e.g. due to reflection, diffraction, or scattering), resulting in an overlapping of subsequent communication symbols.

In order to admit longer symbol durations while keeping a high data rate in total, the assigned frequency spectrum can be split into *multiple sub-carriers* which are used for a *parallel* transmission of communication symbols, each with a relatively low rate (for technical details and an extensive survey of multi-carrier digital communication see, e.g., [BSE04]).

An advantageous property of *parallel multi-carrier transmission* is its robustness against frequency selective distortion. The distortion of a wideband signal through a narrow band noise, for instance, may cause lost of the complete transmission. In multi-carrier communica-

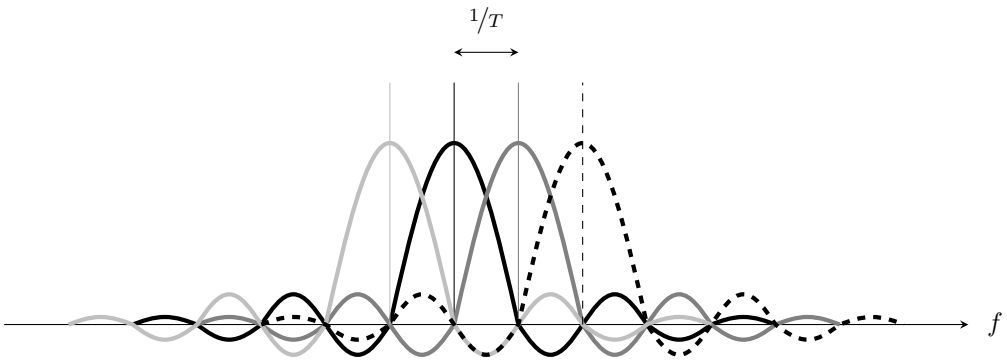


Figure 1.1 Orthogonal spectrum overlap of sub-carriers in OFDM

tion, noise or gain variations on the multiple sub-frequencies can be adaptively accounted for by increasing redundancy in the encoding or strengthening signal power only on the distorted channels (instead of modifying the whole transmission).

The described objective of adaptive optimal sub-carrier rate allocation is a basic sub-problem that has to be faced when implementing *orthogonal frequency division multiplexing* (OFDM) communication in practice, which has been recognized to offer an appealing basis to increase bandwidth and energy efficiency in modern high-speed communication [Kal89, Bin90] (see Section 1.1.3 or [BSE04] for a short history).

The principle of parallel multi-carrier communication was already applied in early communication systems but —when compared to single-carrier communication— the mutual interference of the parallel transmissions and the corresponding loss of bandwidth by the reservation of guard spaces results in an inefficient usage of the total frequency range. In OFDM this disadvantage is largely countered by assigning *orthogonal* frequency ranges to the sub-carriers (see Figure 1.1), so that, despite their overlap, the single channels can be easily separated at the receiver.

In the case of only one sender and one receiver the data rate allocation problem in OFDM systems can be modeled as a convex optimization problem and is very efficiently solvable by using simple water-filling solution methods [Kal89]. However, when more transceivers are involved, the complexity of the problem significantly increases.

In multi-user OFDM systems the sharing of sub-carriers by two or more users is generally precluded due to the high interference occurring when concurrently using the same frequency for different signals. By applying *frequency division multiple access (FDMA)* to avoid conflicts, the sub-carriers have to be exclusively distributed among the communication participants. When aiming to exploit the individual characteristics of each sub-carrier —with respect to the distinct transceivers in order to maximize bandwidth and energy efficiency— this translates to a complex (non-convex) mathematical optimization problem.

There is a broad class of associated problems which are called *OFDMA (OFDM + FDMA) scheduling problems*. In this thesis we will concentrate on the case of *downlink OFDMA*, that is, n many receivers (called *terminals* or *users*) requesting distinct data streams from a base station with specified quality-of-service requirements on the provided data rate that have to be fulfilled utilizing m many different *sub-carriers* (also called *channels*) with distinct frequency characteristics.

There have been numerous attempts to algorithmically solve downlink OFDMA scheduling, e.g., [WCLM99, RC00, KLL03, JL03, WSEA04, SAE05, SYKC06, KPL06, FMR08] (see Section 1.1.3), but because of the widely assumed intractability of the problem these were

mostly heuristics, not guaranteeing any bound on the quality of the obtained solutions or the running time. We are the first to conduct a theoretical study on the approximability of this practically relevant class of network optimization problems (for earlier attempts [WSEA04, VD06] on determining the complexity of OFDMA scheduling see Section 1.1.3).

At first sight, the obtained theoretical worst-case results might seem to be devastating and support the pursuit of heuristic solution methods that show to achieve good results on a fixed set of practical input instances. As we will see, the basic formulation of the OFDMA scheduling problem is not only NP-hard to solve but also inapproximable by *any* factor computable in polynomial time (unless $P = NP$, of course).

However, a nice feature of theoretical analysis is that it often delivers insights on what makes a problem hard and may point out possibilities to circumvent these obstacles. Luckily, OFDMA turns out to be a glaring example for this advantageous property.

While we will see that —already on a very restricted subclass— the problem of minimizing the required transmission power for given data rates (called Min-Power OFDMA) can not be approximated by any (polynomial time computable) factor, we will also see that the dual objective of proportionally maximizing the data rates for a given total transmission power (Max-Rate OFDMA) is approximable arbitrarily close to the optimum on this subclass. This remarkable gap in the approximability of the two problem formulations (with identical decision variants) motivated the analysis of a number of further interesting subclasses and variants of OFDMA scheduling. Depending on the algorithmic tractability of the particular subclass we develop efficient exact or approximative algorithms for the considered OFDMA scheduling variants. Also, for general unrestricted Max-Rate OFDMA several approximation algorithms will be proposed.

While in OFDMA scheduling the objective of energy minimization (resp. rate maximization) is self-evident, it is more hidden in network lifetime optimization problems. In a nutshell, *network lifetime maximization* aims at sustaining a prespecified communication as long as possible in a given communication network with limited power supplies. While the requirements of the communication are fixed, there may be numerous ways to establish it each resulting in different energy demands.

A simple example for this broad class of optimization problems arises in the context of *wireless sensor networks*. Sensor networks consist of a number of *sensor nodes* monitoring their nearby environment (e.g. by measuring temperature or seismic activity). In order to evaluate and analyze the aggregated data, the corresponding readings of the single sensor nodes are periodically collected at a *base station*. Since the energy consumption of a wireless transmission grows super-linearly with increasing distance between sender and receiver, it might be preferable for a sensor node to send its data to an intermediate node, which relays the transmission in order to decrease the required energy. As each sensor node is equipped only with a *limited power supply* (typically a battery), the goal is to keep the sensor network "alive" as long as possible, where the *lifetime* of a network ends when one of the nodes is unable to transmit the required data due to depletion of its energy supply.

However, often there are multiple ways to route and transmit a message through the network, and always picking a communication topology with minimal energy demands may lead to an unnecessarily early depletion of some parts of the network which may be critical to support further communications. Thus, network lifetime maximization requires a more global approach than just optimizing energy efficiency.

There exists a vast number of approaches in order to solve the problem of optimizing the usage of limited energy supplies while sustaining given communication requirements. The existing literature on this topic can mainly be partitioned into either the greedy minimization

of the total energy needed to realize the communication topologies [ČHE02, CH89, CHP+02, KKKP97, LN01, MPF96, RM99, WCLF01, WNE00, WNE01] or the direct maximization of network lifetime [CT99, CT00, KP03, CKOZ03, OY04, OY05, KP05, PS05, ELN+11] (see Section 5.3.5 for a brief overview).

However, to the best of our knowledge, all formulations of network lifetime assume perfect a priori knowledge about the status and future behavior of the energy supplies of each single network node. In real-world applications, this seems an unrealistic assumption. The measured residual energy after a (partial) execution of the transmission scheme may deviate from the initially predicted ideal behavior. In addition to the non-perfect discharge characteristics of batteries (for a brief introduction see, e.g., [RVR03]), the energy budget can also be influenced by varying communication costs caused from temporal interference (e.g. by rain or transient noise).

To exploit the updates on the residual energy levels during the execution of a transmission scheme, we introduce a new online model for general linear packing problems that can be used to increase the lifetime of wireless ad-hoc networks.

When dealing with problems where the input is not completely known a priori or may be subject to uncertainty, the definition of online models provides a basis for theoretically analyzing the impact of this lack of knowledge and allows to develop algorithms coping with the given kind of uncertainty (for an extensive introduction to online analysis see, e.g., [BEY98]).

While in classic models for online packing (see Section 1.1.1) the capacities of the resources are assumed to be known and the packable objects together with their load requirements and profits are revealed in an online fashion, we consider the opposite case where the requirements and profits are known but the capacities are uncertain.

Abstracting from various technological details, we assume that at any time an upper and a lower bound on the remaining capacity of every resource can be obtained and these bounds deviate at most by a factor of α, for a given parameter $\alpha \geq 1$. This way, while the guarantee on the relative error regarding the remaining capacity stays constant, the estimations on the true, initial capacity get more and more accurate with depletion.

Our online model is motivated by the behavior of battery depletion in wireless sensor networks and general network lifetime problems but it may also be applied to completely different contexts. We develop algorithms that exploit only general properties of the defined online model achieving a competitive factor that is independent of the underlying problem structure as long as it can be formulated in terms of a general linear packing program.

1.1 Previous Work and Our Results

There exists an abundance of literature on (offline) network lifetime optimization and we will present a short survey of known results later (Section 5.3.5). However, as we are mainly concerned with the study of online algorithms and competitive analysis for this problem class, we give a brief comparison to related online models in Section 1.1.1 followed by an outline of our results (Section 1.1.2).

An brief survey of existing works and an overview of our results on downlink OFDMA scheduling will be presented in Sections 1.1.3 and 1.1.4, respectively.

1.1.1 Related Online Models for General Linear Packing Problems

Though possibly not formulated in terms of a linear program and despite the fact that energy costs of wireless communication links behave in a non-linear fashion, network lifetime

problems are representable as combinatorial packing problems which —when considering their inflated representation— behave linearly with respect to the capacity constraints. In order to develop a general framework for this type of problems, we define our online model for general linear packing programs.

A distinct yet related online model for linear packing programs has been introduced by Buchbinder and Naor [BN05, BN09]. As in our online model they also consider general linear packing problems in the context of competitive online analysis. However, the assumptions on the revealed online information are exactly the opposite: In [BN09] packing problems are considered where the constraint capacities are assumed to be known a priori but the packing coefficients are revealed in an online fashion. In every round a new packing variable together with its corresponding load and profit coefficients is exposed and can be increased only in this particular round.

Considering the offline formulation, the stepwise introduction of new variables to the LP may seem artificial but in a dual view on the packing problem it comes naturally. There, the online revelation of the primal packing variable corresponds to the revelation of an additional constraint in the dual covering LP. In [BN09] this primal-dual structure is exploited to develop an online algorithm that is $O(\log n + \log \frac{a_{\max}}{a_{\min}})$-competitive where n is the number of resources and $a_{\max}$ and $a_{\min}$ are the largest (resp. smallest) coefficients in the constraint matrix. Also tight asymptotic lower bounds for this model were shown. When comparing these to the bounds shown in Chapter 6 it can easily be seen that the model from [BN09] and the formal model defined in Section 5.1.1 not only differ in their definition but also bear fundamental structural differences.

There are two substantial properties that distinguish our model from the model in [BN09] and other classic online models (see, e.g., [Alb03] for a survey). One distinctive feature of our model is the ability of the online player to influence the length/duration of the adversarial game by adjusting the size of the increase in the packing variables. This way, the online algorithm is not forced to be competitive in any prefix of the game, which allows more foresighted long-term strategies.

Another difference to other models is that the revelation of online information is not carried out in an atomic/binary fashion. In classic models a single information (e.g. a value or a request) is either known or completely unknown while in our model the online values are vaguely known from the beginning, getting more and more precise in the course of the adversarial game (as initially completely oblivious capacities would result in unbounded competitive ratios, see Chapter 6).

A somewhat similar advantage to the online algorithm is given in stochastic programming [BL97, GGLS08, IKMM04, SS04]. There, one assumes a stochastic model on the uncertain parameters and seeks solutions that optimize an objective function in expectation. The problems are often studied in the form of two-stage or multi-stage recourse models where corrective actions at higher prices are admitted as more and more of the random events occur. In our case though, we do not make any assumptions on the probability distribution of the outcomes and corrective actions are not allowed.

1.1.2 Our Results on Online Packing with Gradually Improving Capacity Estimations

We introduce a novel online model which formalizes the aspect of gradually improving capacity estimations experienced in the context of practical network lifetime optimization, using the theoretical framework of competitive online analysis. Several competitive online algorithms will be developed and evaluated for the proposed model and general upper and lower bounds

on their competitiveness on general linear packing problems will be derived. An advantageous property of the proposed online algorithms is that they do not require any explicit linear programming (LP) formulation of the optimization problem and, hence, are also suited for combinatorial packing problems which do not allow for a concise LP-representation when algorithmic efficiency is an issue. As the algorithms additionally are approximation factor preserving they can be used as a framework for algorithmically intractable problem classes by applying appropriate offline approximation algorithms as a subroutine. Furthermore, we will see that this property also allows for the efficient computation of good integral LP-solutions from fractional ones under reasonable conditions on the corresponding capacities.

The proven competitive factors of the more sophisticated algorithms outperform naive online algorithms by at least a logarithmic factor with respect to the fixed uncertainty parameter α. Complementarily we show bounds on the competitiveness of arbitrary randomized online algorithms on worst-case instances of several natural network optimization problems.

We also define a simplistic problem class which allows the construction of a deterministic online algorithm matching the general competitiveness bound of $1/\sqrt{\alpha}$ which is also holding (for deterministic online algorithms) in this subclass, thus proving a tight analysis of competitiveness for instances of this special form.

The competitive factors of the proposed general online algorithms will additionally be tightly analyzed for the classic combinatorial packing problems of network flow maximization and packing spanning trees (which is a relevant subproblem in a broad class of network lifetime problems).

1.1.3 Related Work on OFDM/OFDMA Scheduling

One of the early works in modern OFDM system design (for earlier history see [BSE04]) is by Kalet [Kal89] who considers a single-user Max-Rate OFDM problem for real-world modulation schemes and compares it to the performance of a single-carrier transmission using linear equalization. Kalet points out that the multi-carrier approach can significantly outperform the single-carrier approach, and that for Max-Rate OFDM "the solution of the optimum power distribution resembles the water-pouring solution of information theory".

Binghams survey [Bin90] of multi-carrier transmission techniques summarizes the early results in adaptive data rate allocation on different sub-carriers entitling multi-carrier modulation as "an idea whose time has come". When picking on the work of [Kal89] he argues that, although the analytical water-filling approach assumes the data rates to be continuous while in reality they are integral, this does not have much effect on the quality of a solution in practice.

Nevertheless, many algorithms for discrete data rate allocations were developed. Necessary and sufficient conditions for optimal discrete solutions to general rate maximizing and power minimizing single-user multi-carrier modulation problems are derived in [Cam98].

In [WCLM99] a heuristic algorithm for the Min-Power OFDMA problem is proposed by the authors (Wong, Cheng, Letaief, and Murch) based on relaxing the integrality condition on the variables x_{ij} (and some further modifications of the solution space in order to ensure convexity). As an interpretation for the cost summand $x_{ij} \cdot p_{ij} \cdot \psi_i(r_j)$ in the objective function (which may seem unintuitive for fractional x_{ij}) they suggest to view it as the cost-share of user i sustaining the collective data rate $r_j = \sum_{i=1}^{n} r_{ij}$ on channel j for its time-share $x_{ij} = r_{ij}/r_j$. The optimal solution to the relaxed optimization problem is converted into a feasible OFDMA schedule by greedily assigning each channel to the user with the highest share x_{ij}. Subsequently optimal single-user solutions are computed for the resulting channel assignment. In simulations the quality of the heuristic solution is compared (amongst others)

to the lower bound defined by the optimal fractional solution and was assessed to perform considerably well.

A similar approach of allowing fractional channel assignments is used by Rhee and Cioffi [RC00] for the Max-Rate OFDMA problem. However, the authors do not consider the solution obtained by relaxation of their original integral formulation to be suboptimal as they take sharing of a subchannel with respect to a multiple access technology as a natural approach in practice. Furthermore, they refer to an arguable claim in [YC00] stating that as long as the number of shared channels is relatively small the overall data rate should not be influenced too much if the channels are assigned to any of the sharing users.

In [KLL03] Kivanc, Li, and Liu propose a different heuristic for the Min-Power OFDMA problem which works by separating the OFDMA scheduling into the following two phases. First, using the average channel gain of each user over all channels to define an OFDMA scheduling problem with (user-specific) uniform channel costs, an optimal solution to this restriction is used to determine an appropriate number of sub-carriers for each user. Afterwards, a good assignment of channels to users satisfying the obtained number constraint is tried to be found by local improvement steps on a greedy allocation.

A similar heuristic but without the multi-phase approach is developed by Feiten, Mathar, and Reyer [FMR08] who give a local search algorithm exploring arbitrary channel assignments with optimal single-user rate allocations computed by water-filling. They evaluate it to perform better than the heuristic in [KLL03] in simulations.

A multi-user OFDM system without the restriction of exclusive channel allocation is examined in [JL03] by Jang and Lee. They show that under consideration of the mutual interference of users sharing the same sub-carrier, the maximum total data rate in an energy restricted downlink OFDM system is reached when any sub-carrier is exclusively assigned only to the user with the highest channel gain on it. Thus, the maximum throughput problem in OFDM/OFDMA is solvable in polynomial time by greedy channel allocation and subsequent water filling.

In [SAE05] a generalization of the model in [RC00] is considered by Shen, Andrews, and Evans. Instead of just maximizing the minimum data rate achieved by any terminal, proportionality with respect to the individual demands has to be satisfied. This optimization problem is equivalent to the Max-Rate OFDMA problem considered in this work. Shen et al. propose a heuristic where the channels are suboptimally distributed with respect to the following assignment rule: At each iteration the terminal with the currently lowest proportional total data rate may pick the channel with highest individual gain. The rate on this channel is set with respect to a fixed uniform power level (equal for each of the channels) and the total rate of the terminal is updated accordingly. Subsequently, after the assignment phase, an optimal rate-allocation for the obtained channel assignment is computed by convex optimization methods.

Seong, Yu, Kim, and Cioffi [SYKC06] consider OFDM systems with superposition coding as the applied multiple access technique. The sharing of sub-carriers between multiple users is explicitly allowed in this setting and interference is countered by imposing a proper encoding/decoding order. They find a nontrivial formulation of the corresponding rate-maximization and power-minimization problem in terms of geometric programming which allows the usage of efficient solution methods (see, e.g., [BV04] for an introduction to geometric programming). By using a known duality result from [JVG04] for their model the authors are also able to transfer their results from downlink OFDM to the uplink case.

In [KPL06] Min-Power and Max-Rate OFDMA (the latter one restricted to uniform demands) are considered by the authors Kim, Park, and Lee for the case of bounded discrete

domains of allowed data rates on each channel. The domain restriction allows for formulating both problems as pure integer linear programs by admitting the definition of indicator variables for each possible rate on each channel. Computing optimal solutions to the ILP for simulated OFDMA instances they observe that channels assigned to the same user tend to be allocated roughly the same data rate. Based on this observation they derive a corresponding equal-rate assumption and develop a heuristic for Min-Power OFDMA, which they also translate to the case of Max-Rate OFDMA. However, they were not able to exploit this condition directly to efficiently solve the scheduling problem (indeed we can show the problem to remain NP-hard, see Section 3.2), so in order to compute appropriate data rates for each user they additionally require the channel gains to be uniform (which renders the equal-rate assumption to be redundant, see Section 3.5).

Pointing out that hardness of OFDMA scheduling was only assumed but not formally proven before, Vemulapalli and Dasgupta [VD06] provide a reduction from the classic Subset Sum problem to Min-Power OFDMA. While the authors notice a previous claim of NP-hardness [WSEA04] to be formally incorrect, they overlook an exponential blowup in their own construction (nevertheless the problem is still NP-hard, as we will see in Sections 3.2 and 3.3).

1.1.4 Our Results on OFDMA Scheduling

We are the first to conduct a theoretical study of this practically relevant class of network optimization problems for which no lower nor upper bounds on the approximability were known before. We show that only severely restricted subclasses of this problem can be solved in polynomial time and develop efficient combinatorial algorithms for each of those classes. After proving NP-completeness for basic subclasses of the OFDMA scheduling problem we derive several inapproximability results for the power-minimization variant as well as for the rate-maximization problem. In general, OFDMA rate-maximization will be shown to be inapproximable by a factor of more than $1/2$, however, allowing for better approximation factors on subclasses where power minimization is not approximable by any polynomial time computable factor.

In particular, we design a polynomial time approximation scheme (PTAS) computing approximate solutions to the OFDMA scheduling problem in the rate-maximization variant arbitrarily close to the optimum for the case of uniformly related channel costs (resp. gains), that is, instances where the normalized signal-to-noise ratio of each sub-carrier is independent of the assigned terminal. As the running-time of those $(1-\epsilon)$-approximation algorithms depends exponentially on the chosen parameter $\epsilon > 0$, we additionally give efficient constant-factor approximation algorithms for this problem class which are very easy to implement.

For the general OFDMA rate-maximization problem a simple matching-based $1/m$-approximation algorithm is constructed (with m denoting the number of given channels). Also a more involved convex programming approach is considered that (aside from its usefulness to derive better performance guarantees for special subproblems) can be used to compute $1/2$-approximate half-integral solutions for the general case.

1.2 Outline and Bibliographical Notes

In the following three chapters we conduct or study on the algorithmic tractability and approximability of the OFDMA scheduling problem. NP-hardness and inapproximability results will be presented in Chapter 3 together with efficient algorithms for several polynomial

time solvable subclasses. In Chapter 4 we develop approximation algorithms for special restricted variants of Max-Rate OFDMA scheduling as well as for the general version.

Part of the results presented in Chapters 3 and 4 have already been published at the following conference.

- [OV09] Marcel Ochel and Berthold Vöcking. Approximability of OFDMA scheduling. In Amos Fiat and Peter Sanders, editors, Algorithms - ESA 2009, volume 5757 of Lecture Notes in Computer Science, pages 385–396. Springer Berlin / Heidelberg, 2009. doi:10.1007/978-3-642-04128-0_35.

In Chapters 5, 6, and 7 we introduce and analyze our new online model for general linear packing problems and develop several competitive online algorithms. Besides the formal definition of the online model, Chapter 5 will contain a short survey of network lifetime optimization. The competitive analysis of the online model on general linear packing problems is conducted in Chapter 6, whereas in Chapter 7 the competitiveness on classic combinatorial packing problems is considered.

Some of the presented results appeared in preliminary form in the following conference proceedings.

- [ORV12] Marcel Ochel, Klaus Radke, and Berthold Vöcking. Online packing with gradually improving capacity estimations and applications to network lifetime maximization. In Artur Czumaj, Kurt Mehlhorn, Andrew Pitts, and Roger Wattenhofer, editors, Automata, Languages, and Programming, volume 7392 of Lecture Notes in Computer Science, pages 648–659. Springer Berlin / Heidelberg, 2012. doi:10.1007/978-3-642-31585-5_57.

Chapter 2

Orthogonal Frequency Division Multiplexing and Related Optimization Problems

In the current chapter we will give formal definitions of the OFDMA scheduling problems considered in this work. After introducing basic variants of Min-Power/Max-Rate OFDMA scheduling, both for single (Section 2.1) and multiple users (Section 2.2), we also shortly discuss the problem of Santa Claus scheduling (Section 2.3) which is closely related to the well-known makespan scheduling problem and which received much attention by the theoretical computer science community within the last decade. It bears strong resemblances to Max-Rate OFDMA, however, exhibiting a simpler combinatorial structure. We will later exploit those similarities to develop appropriate approximation algorithms for OFDMA scheduling. An overview of known results in Santa Claus scheduling can be found on Section 2.3.1. At the end of the chapter we also present a few useful properties of convex functions which we use when studying the complexity of the introduced problems in Chapter 3.

2.1 Single-User OFDM

In quality-of-service environments where a minimal data rate has to be ensured, sub-carrier utilization in OFDM results in the following inherent optimization problem. Given for each channel j the *effective channel gain* g_j (which denotes the signal-to-noise ratio normalized by sending power), find an optimal assignment of data rates to the single sub-carriers which sum up to at least the required minimum data rate such that the overall transmission power is minimized.

The needed transmission powers for each sub-carrier are influenced by the actual physical implementation; however, with the Shannon Capacity Law the theoretical power requirements can be computed as follows. Depending on the effective channel gain g_j and the assigned data rate r_j, the transmission power required on the respective channel corresponds to $f(r_j) = \frac{2^{r_j}-1}{g_j}$. We call $\psi(r) = 2^r - 1$ the *Shannon rate-power function* and will refer to *channel costs* $p_j = g_j^{-1}$ for each channel j instead of effective channel gains for a more convenient notation.

Note that $\psi(r) = 2^r - 1$ is a convex function, that is a function whose epigraph (all points above the graph of the function) forms a convex set (for useful basic properties of convex functions see Section 2.4). More formally, we use the following definition (cf. [BV04]).

▶ **Definition 2.1** (Convex Function)**.** A function $f\colon \mathbb{R}^n \to \mathbb{R}$ is *convex* if the domain $\mathrm{dom}(f)$ of f is a convex set and if for all $x, y \in \mathrm{dom}(f)$ and $0 \leq t \leq 1$, we have

$$f(tx + (1-t)y) \leq tf(x) + (1-t)f(y) \ .$$

A function f is *strictly convex* if the above inequality is strict for all $x \neq y$ and $0 < t < 1$.

In general, OFDM scheduling is not restricted to the usage of the Shannon rate-power function. We use the following definition to allow a more general formulation.

▶ **Definition 2.2** (Rate-Power Function)**.** A function $\psi\colon \mathbb{R}_{\geq 0} \to \mathbb{R}_{\geq 0}$ which maps a data rate to the required transmission power will be called *rate-power function*.

We require rate-power functions to satisfy the following commonly imposed conditions: to be *convex*, monotonically *strictly increasing*, and *non-negative* with $\psi(0) = 0$.

For analytical reasons we also assume rate-power functions to be *continuously differentiable*.

Now it is easy to formalize the previously described sub-carrier utilization problem in terms of a convex program, since the objective function is defined by the sum of the convex energy requirements for sustaining the corresponding data rate on each selected sub-carrier.

▶ **Definition 2.3** (Single-User OFDM Scheduling)**.** Let $\psi\colon \mathbb{R}_{\geq 0} \to \mathbb{R}_{\geq 0}$ be a rate-power function (e.g. $\psi(r) = 2^r - 1$) and d be the required total data rate. Also for each channel $j \in [m]$ let its channel cost $p_j \in \mathbb{R}_{\geq 0}$ (corresponding to the inverse effective channel gain) be given.

Find an optimal assignment of data rates r_j minimizing the following convex program.

$$\begin{aligned} \text{minimize} \quad & \sum_{j=1}^{m} p_j \cdot \psi(r_j) \\ \text{subject to} \quad & \sum_{j=1}^{m} r_j \geq d \\ & r_j \geq 0 \qquad \forall j \in [m] \end{aligned}$$

The above problem can of course be solved in polynomial time by convex optimization methods. Due to the simple constraint structure even an analytical solution is possible which corresponds to the widely used water-filling principle (see Section 3.1.2).

Instead of minimizing the transmission power while guaranteeing a certain data rate, the somewhat dual objective of maximizing the data rate while satisfying a given energy constraint can be considered.

▶ **Definition 2.4** (Max-Rate Single-User OFDM Scheduling)**.** Given channel costs p_j for each channel $j \in [m]$ and a rate-power function $\psi\colon \mathbb{R}_{\geq 0} \to \mathbb{R}_{\geq 0}$. Also let E be an upper bound on the feasible transmission energy.

$$\begin{aligned} \text{maximize} \quad & \sum_{j=1}^{m} r_j \\ \text{subject to} \quad & \sum_{j=1}^{m} p_j \cdot \psi(r_j) \leq E \\ & r_j \geq 0 \qquad \forall j \in [m] \end{aligned}$$

The decision variant, namely if there is a feasible rate allocation satisfying a given demand d and energy bound E, is the same for both problems. So both optimization problems are dual in the following sense.

▶ **Observation 2.5.** *Let $E^{\mathrm{opt}}(d)$ denote the objective value of an optimal rate allocation for the power minimization problem stated in Definition 2.3 with data rate demand d and let $R^{\mathrm{opt}}(E)$ be the value of an optimal solution to the rate maximization problem according to Definition 2.4 with energy bound E. Then*

$$R^{\mathrm{opt}}(E^{\mathrm{opt}}(d)) = d \qquad \text{and} \qquad E^{\mathrm{opt}}(R^{\mathrm{opt}}(E)) = E$$

for all $d \in \mathbb{R}_{\geq 0}$ and $E \in \mathbb{R}_{\geq 0}$.

2.2 Multi-User OFDMA

When there is more than one sender intending to communicate, medium access control on the transmission channel (either a single carrier or a multi-carrier system) is an important aspect of the system design. Aside from random access or token based media sharing there are three main multiple access techniques, namely code division multiple access (CDMA), frequency division multiple access (FDMA), and time division multiple access (TDMA).

While any of these techniques can be applied in OFDM systems, the inherently subdivided structure of OFDM channels offers an appealing basis for FDMA as an access control mechanism. Note that the above medium access techniques are also useful for multiplexing data streams when one sender has to distribute different informations to distinct receivers.

The adaptive application of FDMA to OFDM where the sub-carriers are exclusively distributed among the communication participants (under consideration of the corresponding individual characteristics of each sub-carrier with the aim of maximizing bandwidth efficiency) is called *orthogonal frequency division multiple access (OFDMA)*.

We will concentrate on the case of *downlink OFDMA* that is $n \in \mathbb{N}$ many receivers (called *terminals* or *users*) requesting distinct data streams from a base station with specified quality-of-service requirements on the provided data rate. The related (many-to-one) uplink OFDMA problem is similar in structure and many-to-many connections can be seen as a composition of the two above communication types (as long as the OFDMA condition of exclusive assignment of a channel to only one sender-receiver pair is guaranteed).

When considering downlink OFDMA where data rate requirements d_i of each user are given and the objective is to minimize the total required sending power, it can be formulated as the following (non-convex mixed-integer) optimization problem.

▶ **Definition 2.6** (Min-Power OFDMA)**.** Let ψ be a rate-power function. Given terminals $\{1, \ldots, n\}$ with demands $d_1, \ldots, d_n$ and channels $\{1, \ldots, m\}$ with costs p_{ij} for $i \in [n]$, $j \in [m]$.

Find a rate allocation $(r_j)_{j \in [m]}$ and a channel assignment $(x_{ij})_{i \in [n], j \in [m]}$ optimizing the following objective function with respect to the given constraints.

$$\begin{aligned}
\text{minimize} \quad & \sum_{i=1}^{n} \sum_{j=1}^{m} x_{ij} \cdot p_{ij} \cdot \psi(r_j) & & \\
\text{subject to} \quad & \sum_{j=1}^{m} x_{ij} r_j \geq d_i & & \forall i \in [n] \\
& \sum_{i=1}^{n} x_{ij} \leq 1 & & \forall j \in [m] \\
& x_{ij} \in \{0,1\}, \quad r_j \geq 0 & & \forall i \in [n] \quad \forall j \in [m]
\end{aligned}$$

A feasible solution (X, r) with the *channel assignment matrix* $X = (x_{ij})_{i\in[n],j\in[m]}$ and the *rate allocation vector* $r = (r_1, \ldots, r_m)$ will be called a *schedule*.

Note that even when relaxing the integrality conditions on the assignment variables x_{ij} the problem formulation above does not have the form of a convex program. However, as we will see in Section 3.6, a mixed-integer convex programming representation of OFDMA scheduling can be given.

As for the single-user scheduling problem we can also consider a throughput maximization variant of the above OFDMA problem. Again, the total energy is bounded by a constraint and the achievable total rate has to be maximized.

▶ **Definition 2.7** (Max-Throughput OFDMA)**.** Let ψ be a rate-power function. Given an energy bound E, terminals $\{1, \ldots, n\}$ and channels $\{1, \ldots, m\}$ with costs p_{ij} for $i \in [n]$, $j \in [m]$.

$$\begin{aligned}
\text{maximize} \quad & \sum_{i=1}^{n}\sum_{j=1}^{m} x_{ij} \cdot r_j \\
\text{subject to} \quad & \sum_{i=1}^{n}\sum_{j=1}^{m} x_{ij} \cdot p_{ij} \cdot \psi(r_j) \leq E \\
& \sum_{i=1}^{n} x_{ij} \leq 1 && \forall j \in [m] \\
& x_{ij} \in \{0,1\}, \quad r_j \geq 0 && \forall i \in [n] \quad \forall j \in [m]
\end{aligned}$$

The above formulation relates to the downlink model. In the uplink model, individual energy bounds for each sender could be added. However, also a global energy bound in the uplink model can make sense in order to limit the overall noise.

As a theoretical justification for the widely used exclusiveness constraint in channel-to-user assignments in practice, Jang and Lee [JL03] showed that even when allowing distinct data streams to share the same subchannel in the problem above, the exclusive OFDMA assignment is optimal (due to interference on the shared channels). Trivially, this result only holds for the throughput maximization problem and not for the power minimization variant (just consider a pathological instance with more users than channels).

While Max-Throughput OFDMA is easy to solve (see [JL03]), throughput maximization may not always be the objective of choice, because of its lack of fairness. Also, in contrast to the single-user case, throughput maximization and power minimization in the multi-user setting are not dual to each other in the sense of Observation 2.5.

However, the following problem definition of Max-Rate OFDMA (Definition 2.8) satisfies these conditions and coincides with throughput maximization when restricted to the single-user case. In Max-Rate OFDMA the goal is not to find a schedule with maximum total data rate but to maximize the (proportional) minimum data rate that can be guaranteed on each terminal.

The relation of Max-Rate and Min-Power OFDMA was exploited in [KPL06] to derive suboptimal solutions for both problems. While the decision variants of the problems may be equivalent, we will see in Chapter 4 that both problems severely differ in their approximability. Intuitively, for steep rate-power functions the objective value of Min-Power OFDMA is very sensitive to changes in the data rate allocation whereas Max-Rate OFDMA allows for relatively large changes in the energy constraint without affecting the objective too much.

▶ **Definition 2.8** (Max-Rate OFDMA)**.** Let ψ be a rate-power function. Given an energy bound E, terminals $\{1,\dots,n\}$ with demands $d_1,\dots,d_n$ and channels $\{1,\dots,m\}$ with costs p_{ij} for $i \in [n]$, $j \in [m]$. The objective is to maximize the (proportional) demand that can be satisfied for each terminal.

$$\begin{aligned}
&\text{maximize} && \lambda && && \\
&\text{subject to} && \sum_{i=1}^{n}\sum_{j=1}^{m} x_{ij} \cdot p_{ij} \cdot \psi(r_j) \leq E && && (2.1)\\
& && \sum_{j=1}^{m} x_{ij} r_j \geq \lambda d_i && \forall i \in [n] && (2.2)\\
& && \sum_{i=1}^{n} x_{ij} \leq 1 && \forall j \in [m] && (2.3)\\
& && x_{ij} \in \{0,1\}, \quad r_j \geq 0 && \forall i \in [n] \quad \forall j \in [m] && (2.4)
\end{aligned}$$

2.3 Santa Claus Scheduling

A problem with strong similarities to Max-Rate OFDMA scheduling but of purely combinatorial nature is the *Santa Claus scheduling problem* also known as *max-min machine scheduling* or *max-min (fair) allocation of indivisible goods*. In this problem, m jobs have to be distributed among n machines where the processing time of each job may depend on the machine it is allocated to. The goal is to maximize the time until the first machine becomes idle. Formally we have the following integer linear programming formulation.

▶ **Definition 2.9** (Santa Claus Scheduling)**.** Given a set of *machines* $\{1,\dots,n\}$ and a set of *jobs* $\{1,\dots,m\}$ with *processing time* (also called *size*) $s_{ij} \in \mathbb{R}_{\geq 0}$ for job j on machine i. Find an optimal solution to the following integer linear program which maximizes the *minimum completion time* T.

$$\begin{aligned}
&\text{maximize} && T && && \\
&\text{subject to} && \sum_{j=1}^{m} s_{ij} x_{ij} \geq T && \forall i \in [n] && (2.5)\\
& && \sum_{i=1}^{n} x_{ij} \leq 1 && \forall j \in [m] && (2.6)\\
& && x_{ij} \in \{0,1\} && \forall i \in [n] \quad \forall j \in [m] && (2.7)
\end{aligned}$$

Note that the exclusive assignment of jobs to machines expressed by Constraints 2.6 and 2.7 is equivalent to the exclusive assignment of channels to terminals in the Max-Rate OFDMA problem (Constraints 2.3 and 2.4 of Definition 2.8). Also for $T = \lambda$ and $s_{ij} = r_j/d_i$ Condition 2.5 becomes equivalent to the demand satisfaction Constraint 2.2.

Despite these similarities, the Santa Claus problem presents itself as a less complex optimization problem than Max-Rate OFDMA, as no convex energy constraints over a continuous variable domain have to be considered. However, once appropriate data rates have been chosen for an OFDM scheduling instance, techniques from Santa Claus scheduling can help to find a good channel assignment (although Constraint 2.1 can still be problematic

for general unrelated channel costs). In Chapter 4 this relation will be exploited to develop several approximation algorithms for the Max-Rate OFDMA problem.

What makes Santa Claus scheduling in some cases more accessible than OFDMA is its special combinatorial structure that allows for a partial separation of subproblems. For instance, for each machine the set of feasible job assignments satisfying the covering Constraint 2.5 depends only on the respective machine and the jobs assigned to this machine, whereas in OFDMA the decision if a given channel assignment to a terminal is feasible also depends on the energy used by channels assigned to other terminals.

Santa Claus scheduling is closely related to the classic makespan minimization problem, where an assignment of jobs to machines minimizing the maximum completion time has to be found (instead of maximizing the minimum). Most results for Santa Claus scheduling have their origin in work on the makespan minimization problem and similarly there are several subclasses that received much attention by the community.

One of the most basic classes is the *identical machines* model. There each job has the same size on all of the machines, i.e. $\forall i \in [n]\ s_{ij} = s_j$ for some machine independent value $s_j \in \mathbb{R}_{\geq 0}$ for each job j. The indifference of the machines makes the problem easier to approximate, though it still remains NP-hard.

A slight generalization of this model is the case of *(uniformly) related machines* where the processing time of a job does not only depend on the machine independent value s_j but also multiplicatively on the speed of the machine assigned to, i.e. $s_{ij} = s_j \cdot v_i$ for some machine specific value $v_i \in \mathbb{R}_{\geq 0}$. It behaves similar to the identical machines model.

Another well-studied generalization of the identical machines model is the *restricted assignment* case where each job still has the same size on all machines but machines may be unable to schedule some jobs. Formally, this is expressed by defining machine dependent sizes $s_{ij} \in \{0, s_j\}$ for all $i \in [n]$ and $j \in [m]$.

The general case of unrestricted $s_{ij} \in \mathbb{R}_{\geq 0}$ is called the *unrelated machines* model to distinguish it from the special cases described above.

2.3.1 Related Work on Santa Claus Scheduling

Santa Claus scheduling has first been considered in [Woe97] for the case of identical machines by Woeginger who developed the first polynomial time approximation scheme (PTAS) for this problem.

Using ideas from the context of makespan scheduling presented in [HS88], Azar and Epstein [AE98] were able to construct a PTAS for the more general case of uniformly related machines. Subsequently in [ES04] a general framework for a broad class of scheduling problems on uniformly related machines (including makespan and Santa Claus scheduling) is given by Epstein and Sgall.

General Santa Claus scheduling for unrelated machines was first considered in [BD05] albeit in a game theoretic context. Besides from studying mechanism design related questions to this problem the authors Bezáková and Dani give two different approximation algorithms for Santa Claus scheduling in the (omniscient) unrelated machines model. The first one is a simple matching-based algorithm which achieves an approximation factor of at least $\frac{1}{m-n+1}$. The second algorithm is based on the LP-Rounding technique of Lenstra, Shmoys and Tardos [LST90] for the makespan scheduling problem. However, in contrast to makespan scheduling no approximation factor can be guaranteed and only an additive guarantee of $\mathrm{OPT} - s_{\max}$ is given where $s_{\max}$ denotes the maximal size of any job on any machine in the instance.

In [Gol05] Golovin extends the LP-rounding algorithm of [BD05] to give a $\frac{1}{n}$-approximation algorithm for unrelated Santa Claus Scheduling. He also formulates a bicriteria approach where $\lceil (1-\frac{1}{k})n \rceil$ many machines can be guaranteed a minimum load of at least $\frac{\text{opt}}{k}$ for any $k \in \mathbb{N}$. Furthermore, he considers the big goods/small goods variant and develops a $\frac{1}{\sqrt{n}}$-approximation algorithm for it. In the big goods/small goods model each job j either has size $s_j = 1$ on all machines or size $s_j = B > 1$ for some arbitrary $B \in \mathbb{R}_{\geq 0}$. However scheduling a given job may not be possible on all machines. This is formalized by allowing $s_{ij} \in \{0, s_j\}$ to express the size of job $j \in [m]$ on machine $i \in [n]$ for $s_j \in \{1, B\}$. Thus, it can be seen as a special variant of the restricted assignment model.

For the general form of restricted assignment Bansal and Sviridenko [BS06] developed an $\Omega(\frac{\log\log\log n}{\log\log n})$-approximation algorithm. For their result they introduce a new LP-relaxation formulation for the Santa Claus scheduling problem which they call 'the configuration LP'. The (exponentially sized) configuration LP encodes each feasible assignment of a subset C of jobs (called a configuration) assigned to machine i as a variable $x_{i,C}$ and enforces disjoint assignment of configurations to machines. It is up to now the strongest known LP-formulation for Santa Claus and makespan scheduling on unrelated machines.

Recently Haeupler, Saha and Srinivasan [HSS11] could show how to compute a constant factor approximation of the restricted assignment case in polynomial time by constructivizing a proof of [Fei08] based on the Lovász Local Lemma.

In [AS07] the configuration LP is used to construct a randomized $\Omega(\frac{1}{\sqrt{n}\log^3 n})$-approximation algorithm for the general case of unrelated machines which up to a polylogarithmic factor matches the lower bound of $\Omega(\sqrt{n})$ on the integrality gap for the configuration LP.

Despite the large integrality gap, Chakrabarty, Chuzhoy and Khanna [CCK09] were able to iteratively use the configuration LP to develop an approximation algorithm that for any fixed $\epsilon \in \Omega(\frac{\log\log m}{\log m})$ computes an $\Omega(\frac{1}{m^\epsilon \log m})$-approximation in time $m^{O(1/\epsilon)}$ improving the result of [AS07] for m polynomial in n.

In [VW10, VW11] Verschae and Wiese study the impact of using the configuration LP formulation in the context of makespan scheduling. They identify a very restricted class of instances for which the configuration LP still has the same integrality gap as for the general problem. In this class called the *unrelated graph balancing* case each job has at most two machines which are able to schedule this job (with possibly different processing times on each of them). The authors also consider the problem of Santa Claus scheduling and are able to give a simple $\frac{1}{2}$-approximation algorithm for the unrelated graph balancing case which is tight to a corresponding upper bound by [BCG09] and beats the more complicated LP-based constant factor approximation algorithms given by [BCG09] and [CCK09] for this special class. For general Santa Claus scheduling with unrelated machines they adapt the LP-rounding technique from [LST90] to compute half-integral solutions which are $\frac{1}{2}$-approximate.

2.4 Some Basic Properties of Convex Functions

In the following we will frequently use some common properties of convex and concave functions which we briefly state here. An extensive overview of convex functions and general convex optimization can be found in [BV04], where most of our definitions on this topic are from.

▶ **Lemma 2.10.** *Any convex function satisfies Jensen's inequality, i.e.*

$$f(\sum_{i=1}^{k} t_i x_i) \leq \sum_{i=1}^{k} t_i f(x_i)$$

for all $x_1, \dots, x_k \in \mathrm{dom}(f)$ *and* $t_1, \dots, t_k \geq 0$ *with* $\sum_{i=1}^{k} t_i = 1$.

This well-known property of convex functions follows directly from their definition.

The following simple result will be useful when bounding convex functions. We call a function ψ *(strictly) concave* if $-\psi$ is (strictly) convex.

▶ **Lemma 2.11.** *Let* ψ *be a convex function with* $\psi(0) = 0$. *Then for any* $a \in \mathbb{R}_{\geq 0}$ *and* $x \in \mathbb{R}_{>0}$ *we have*

$$a\psi(x) \leq \psi(ax) \qquad \text{if } a > 1 \tag{2.8}$$

$$a\psi(x) \geq \psi(ax) \qquad \text{if } a < 1 \tag{2.9}$$

where the inequalities become strict for strictly convex functions ψ.

If ψ *is a concave function with* $\psi(0) = 0$ *the above inequalities hold for the respective complementary cases* $a < 1$ *(Inequality 2.8) and* $a > 1$ *(Inequality 2.9).*

Proof. We only prove the claim for convex functions as for concave functions ψ we just have to consider $-\psi$.

The cases $a = 1$ and $a = 0$ are trivial. For $a > 1$ by convexity we have

$$\psi\left(\frac{1}{a} \cdot x + \frac{a-1}{a} \cdot 0\right) \leq \frac{1}{a} \cdot \psi(x) + \frac{a-1}{a} \cdot \psi(0)$$

(which is strict if the function is strictly convex) so by scaling x by the factor a and using the property $\psi(0) = 0$ we get $\psi(x) \leq \frac{1}{a}\psi(ax)$ showing Inequality 2.8. For $0 < b < 1$ the derived inequality $\psi(x) \leq \frac{1}{a}\psi(ax)$ yields $\psi(x) \leq b\psi(\frac{x}{b})$ which by substituting x by $y \cdot b$ proves Inequality 2.9. ◀

There are several operations that preserve convexity like nonnegative weighted sums, composition with an affine mapping, pointwise maximum and supremum and many more (see [BV04]). Another very useful operation is the so called *perspective* of a function.

▶ **Lemma 2.12.** *Let* $\psi \colon \mathbb{R} \to \mathbb{R}$. *Then for* $x \in \mathbb{R}$ *and* $y \in \mathbb{R}_{>0}$ *the function*

$$\xi(x, y) = y \cdot \psi\left(\frac{x}{y}\right)$$

which is called the perspective *of the function* ψ *is convex (resp. concave) if and only if* ψ *is convex (concave).*

Proof. We follow the simple proof from [DM08] and show the property for convex functions. For concave functions the inequality below has to be inverted.

Trivially, already for $y = 1$ the function ξ can not be convex if ψ is not convex. So for the inverse direction let $z_1 = (x_1, y_1)$, $z_2 = (x_2, y_2)$. Then we derive

$$\begin{aligned}
\xi(tz_1 + (1-t)z_2) &= \xi(tx_1 + (1-t)x_2,\ ty_1 + (1-t)y_2) \\
&= (ty_1 + (1-t)y_2) \cdot \psi\left(\frac{tx_1 + (1-t)x_2}{ty_1 + (1-t)y_2}\right) \\
&= (ty_1 + (1-t)y_2) \cdot \psi\left(\frac{ty_1}{ty_1 + (1-t)y_2} \cdot \frac{x_1}{y_1} + \frac{(1-t)y_2}{ty_1 + (1-t)y_2} \cdot \frac{x_2}{y_2}\right) \\
&\leq ty_1\psi\left(\frac{x_1}{y_1}\right) + (1-t)y_2\psi\left(\frac{x_2}{y_2}\right) \\
&= t\xi(z_1) + (1-t)\xi(z_2)
\end{aligned}$$

for all $t \in [0, 1]$ and $(x_1, y_1), (x_2, y_2) \in \mathbb{R} \times \mathbb{R}_{>0}$. ◀

Chapter 3

Algorithmic Tractability of OFDMA Scheduling

In this chapter we analyze the complexity of algorithmically solving the OFDMA scheduling problem in general as well as in several tractable subclasses.

We will first consider the single-user OFDM scheduling problem in Section 3.1, which can easily be solved in polynomial time by general convex optimization methods, and derive closed-form characterizations of optimal solutions for the Shannon rate-power function in Section 3.1.1. While this is only a special case of the commonly known water-filling characterization of optimal OFDM schedules, the determined explicit description of optimal energy costs will be needed for the inapproximability result in the uniformly related channel costs model (see Section 3.3.1). The general version of water-filling will be considered in Section 3.1.2. We extend the single-user OFDM problem by allowing capacity restrictions on each channel and derive the water-filling property for this generalized variant which will be useful for both the construction of worst-case instances and the development of approximation algorithms for the multi-user OFDMA scheduling problem.

In Section 3.2 we study the algorithmic tractability of multi-user OFDMA scheduling. As we will see, this problem is already NP-complete for very restricted subclasses. Our reduction, based on the NP-hard 3-dimensional matching problem, points out cases of multi-user OFDMA scheduling that might be more accessible —which we will subsequently consider in Sections 3.4 and 3.5.

The given hardness proofs will also be extended to show bounds on the approximability of Min-Power and Max-Rate OFDMA. In Section 3.3 we will see that, while Max-Rate OFDMA can not be approximated by any factor better than $1/2$ in general, the situation for Min-Power OFDMA is much worse: we give a reduction proving that, even for the very restricted case of uniformly related channel costs, Min-Power OFDMA can not be approximated by any factor computable in polynomial time.

If rate-power functions are restricted to be linear, very efficient solution methods can be derived for OFDMA scheduling which have running times less than quadratic in the input size. Also for arbitrary rate-power functions (even allowing different rate-power functions for distinct users) we can identify a non-trivial restriction that allows for efficient solvability. In this class, the cost coefficients of all channels for each user are uniform (though they might be different to other users), but terminals may be restricted from using certain channels. As this can be encoded by choosing the channel gains (inverse channel costs) to be from the binary domain $\{0, 1\}$ while incorporating the actual channel cost of a user into the corresponding user-specific rate-power function we call it the *binary channel gain* model. OFDMA with binary channel gains can be efficiently solved and extends the simplified case of purely uniform channel costs assumed in some heuristic solution approaches.

At the end of this chapter we propose a fractional formulation of Min-Power OFDMA which is polynomial time solvable. Trivially relaxing the integrality constraints on the assignment variables x_{ij} is not enough, since the natural problem formulation is not convex. We derive an alternative convex formulation similar to the ones given in [RC00] and [SAE05]

but without restricting the assignment variables to be non-zero and with additional properties that are useful in the design of approximation algorithms for the original integral problem (see Chapter 4).

3.1 Efficient Solvability of Single-User OFDM Scheduling

From the convex programming formulation in Definition 2.3 it is immediately obvious that the single-user OFDM scheduling problem is polynomial time solvable. In Section 3.1.2 we will describe a practically common approach to solve this and similar problems with a more efficient solution method than general convex optimization. While the applicability of the so called *water-filling* method is a general structural property of the solution space of single-user OFDM we will sometimes need more explicit solution characterizations in order to immediately determine the required energy cost of an optimal solution.

3.1.1 Optimal Solutions to Instances with the Shannon Rate-Power Function

For single-user OFDM with the Shannon rate-power function, we can directly derive an analytical solution which will especially be useful for hardness proofs. Though it implicitly equals the water-filling solution, the proof is simpler and can be done without using results from general convex optimization theory.

▶ **Lemma 3.1.** *Given an instance of the single-user OFDM scheduling problem according to Definition 2.3 with the Shannon rate-power function and channels ordered by increasing costs* $0 < p_1 \leq p_2 \leq \cdots \leq p_m$. *For demand* d *let* ℓ_i *be defined as*

$$\ell_i = \frac{d + \sum_{j=1}^{i} \log_2 p_j}{i}$$

and let $k \geq 1$ *be the smallest index such that* $\log_2 p_{k+1} \geq \ell_k$. *Then the optimal rate allocation* r^{opt} *for each channel* j *is given by*

$$r_j^{opt} = \max\{0, \ell_k - \log_2 p_j\} \,.$$

Proof. As a first step we remove the additive constant $-\sum_{j=1}^{m} p_j$ from the objective function $\sum_{j=1}^{m} p_j(2^{r_j} - 1)$ in order to determine its minimum. Equivalent to the task of allocating the rates $r_j \geq 0$ with $\sum_{j=1}^{m} r_j = d$ such that $\sum_{j=1}^{m} 2^{r_j + \log_2 p_j}$ is minimal the following optimization problem can be considered instead.

$$\begin{aligned} \text{minimize} \quad & \sum_{j=1}^{m} 2^{\tilde{r}_j} \\ \text{subject to} \quad & \sum_{j=1}^{m} \tilde{r}_j = d + \sum_{j=1}^{m} \log_2 p_j \\ & \tilde{r}_j \geq \log_2 p_j \qquad\qquad \forall j \in [m] \end{aligned} \tag{3.1}$$

By Constraint 3.1 the sum of the rates on channels $\{k+1, \ldots, m\}$ for any k is at least $\sum_{j=k+1}^{m} \log p_j$ so the total sum on channels $\{1, \ldots, k\}$ is at most $d + \sum_{j=1}^{k} \log p_j$. Thus, by arithmetic mean, there is a channel in any set $\{1, \ldots, k\}$ with rate at most $\frac{1}{k}(d + \sum_{j=1}^{k} \log_2 p_j) = \ell_k$.

For any fixed $r > 0$ and parameter $0 \leq \epsilon \leq r$ the global minimum of $2^{r+\epsilon} + 2^{r-\epsilon}$ is reached for $\epsilon = 0$. So if $\tilde{r}_j > \log_2 p_j$ for some channel $j > k$ with $\log_2 p_j \geq \ell_k$, a portion

of $\tilde{r}_j$ can be swapped to the designated channel in $\{1,\dots,k\}$ with rate at most ℓ_k thereby decreasing the energy cost. Hence, $\tilde{r}_j = \log_2 p_j$ for all $j \geq k+1$ where k is the smallest index with $\log_2 p_{k+1} \geq \ell_k$.

For the remaining channels $\{1,\dots,k\}$ the unique global optimum of the function

$$f(r_1,\dots,r_{k-1}) = 2^{d-\sum_{i=1}^{k-1}\tilde{r}_i} + \sum_{i=1}^{k-1} 2^{\tilde{r}_i}$$

is given by $r_j = \ell_k$ for each $j \in \{1,\dots,k\}$ which satisfies Constraint 3.1 as the minimality of the chosen k implies

$$\frac{k-1}{k}\log_2 p_k \leq \frac{k-1}{k}\ell_{k-1} \quad \Rightarrow \quad \log_2 p_k \leq \frac{k-1}{k}\ell_{k-1} + \frac{1}{k}\log_2 p_k = \ell_k \ .$$

Re-substituting $r_j = \tilde{r}_j - \log_2 p_j$ proves the claim. ◀

▶ **Corollary 3.2.** *Given an instance of the single-user OFDM scheduling problem according to Definition 2.3 with the Shannon rate-power function and channels ordered by increasing costs* $0 < p_1 \leq p_2 \leq \cdots \leq p_m$.

Let $k \geq 1$ *be the smallest index such that* $\log_2 p_{k+1} \geq \frac{1}{k}\cdot(d + \sum_{1\leq j\leq k}\log_2 p_j)$. *Then the optimal solution has a total cost of*

$$\text{optcost}^{(d)}(C) := k \cdot \sqrt[k]{2^d \prod_{1\leq j\leq k} p_j} - \sum_{1\leq j\leq k} p_j \quad .$$

Proof. Using Lemma 3.1 the cost for sustaining the optimal data rate on each channel $c \in \{1,\dots,k\}$ is exactly

$$p_c \cdot (2^{\frac{1}{k}\cdot(d+\sum_{1\leq j\leq k}\log_2 p_j)-\log_2 p_c} - 1) = p_c \cdot \left(\sqrt[k]{2^d \prod_{1\leq j\leq k} p_j}\cdot\frac{1}{p_c} - 1\right)$$

which in total results in the claimed cost. ◀

Considering the optimal data rates stated in Lemma 3.1 it is not surprising that Kim, Park, and Lee [KPL06] observed that optimal solutions to simulated practical OFDMA instances exhibit the property of having roughly the same data rate on each channel assigned to a single user.

3.1.2 Water-filling for Single-User OFDM with Capacity Limits

For the case of single-user OFDM it has already early been recognized [Kal89] that the set of optimal schedules exhibits the *water-filling* property, which is also very common in some classic problems of signal processing and information theory [CT91, MS00].

The term *water-filling* refers to the geometric interpretation (see Figure 3.1) of a special structure shared by optimal solutions of affected optimization problems. Depending on the context, it is used to denote the structure, the optima, or the method to derive such optimal solutions.

In basic water-filling solutions the value of each variable x is fully determined by a mapping f_x (or a relation specifying a range of solutions) evaluated on a common global parameter ℓ called the *water level*. For example, as we have seen in Section 3.1.1, all optimal

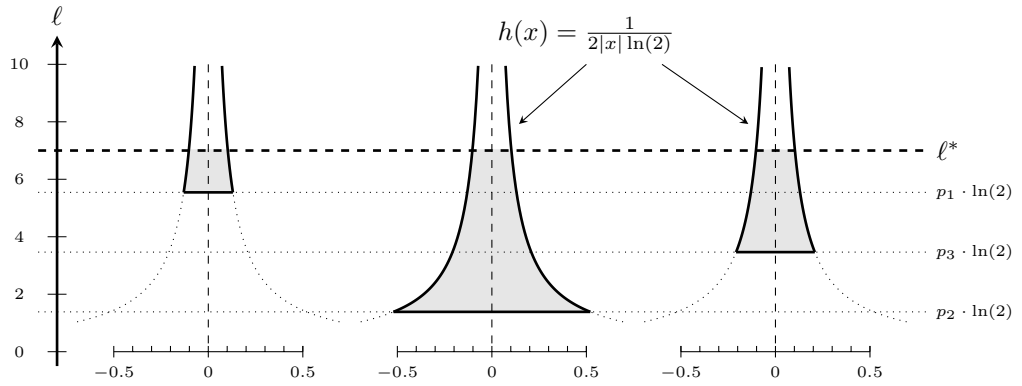


Figure 3.1 Water-filling bins for the Shannon rate-power function with water level $\ell^* = 7$. From left to right: channel 1 with $p_1 = 8$, channel 2 with $p_2 = 2$ and channel 3 with $p_3 = 5$. The data rates for water level ℓ^* on each channel in the water-filling solution correspond to the filled area in each bin.

rate assignments $(r_j)_{j\in[m]}$ to the single-user OFDM scheduling problem with the Shannon rate-power function have the form

$$r_j = \max\{0, \ell - \log_2(p_j)\} \qquad \forall j \in [m]$$

for some "water level" ℓ independent of j.

Intuitively, one can imagine the variables in a water-filling solution as bins (of possibly different shapes and sizes) arranged at different altitudes which have to be filled keeping a globally equalized water level. The amount of water contained in a bin is directly proportional to the value of the corresponding variable. Also, if the bottom of a bin lies above the current water level, the value of the variable is zero. Though the bins are only mentioned for picturing the basic concept they can also be explicitly computed. Figure 3.1 is derived from a calculation of Reyer [Rey08] who computed the water-filling bins for the single-user OFDM scheduling problem with the Shannon rate-power function, which can be mathematical defined as the area between the function $h(x) = \frac{1}{2x\ln(2)}$ and its mirror image at the vertical axis. The bottom of the bin for each channel j starts at water level $\ell = p_j \cdot \ln(2)$ where p_j denotes the cost of the channel.

In order to find an optimal solution to a given input instance with known water-filling property one can continuously increase the water level until all side constraints are satisfied. In the case of single-user OFDM scheduling the only non-trivial constraint is implied by the minimum data rate demand.

The water-filling property of single-user OFDM is well known and proofs are commonly omitted since the result is considered to be folklore. A short proof using directional derivatives can be found in [FMR08]. We will not state this particular result here as we provide a proof for the water-filling structure of a more generalized version of single-user OFDM scheduling which we will use as a subproblem for our polynomial time approximation schemes in Section 4.1.3. We extend the scheduling problem by allowing channels to have limits on the maximal supportable data rate.

Though more general, this problem is still trivially polynomial time solvable by general convex optimization. Furthermore, the problem is actually equivalent to a very restricted single-commodity convex cost flow problem using only parallel links. Thus, there are very efficient solution methods for this problem even when the data rates r_j are restricted to be

integral (e.g. the cycle-canceling method [KM97] or capacity-scaling [AMO93]).

▶ **Definition 3.3** (Single-User OFDM Scheduling with Capacity Limits)**.** Let ψ be a rate-power function. Given channels $\{1, \dots, m\}$ with costs p_j and capacities c_j for $j \in [m]$.

$$\begin{aligned} \text{minimize} \quad & \sum_{j=1}^{m} p_j \cdot \psi(r_j) \\ \text{subject to} \quad & \sum_{j=1}^{m} r_j \geq d \\ & r_j \geq 0 \qquad \forall j \in [m] \\ & r_j \leq c_j \qquad \forall j \in [m] \end{aligned}$$

To prove the water-filling property we use a general result on convex programming, namely the characterization of optimal solutions by the Karush-Kuhn-Tucker conditions. We first define the standard form of a convex program (cf. [BV04]).

▶ **Definition 3.4** (Standard Form of a Convex Program)**.** A *convex program in standard form* is formulated as the following mathematical optimization problem.

$$\begin{aligned} \text{minimize} \quad & f_0(x) \\ \text{subject to} \quad & f_i(x) \leq 0 \qquad \forall i \in [m] \qquad (3.2) \\ & h_i(x) = 0 \qquad \forall i \in [q] \qquad (3.3) \end{aligned}$$

where $h_1, \dots, h_q$ are linear and $f_0, f_1, \dots, f_m$ are convex functions over the domain $\mathbb{R}^n$.

The Karush-Kuhn-Tucker (KKT) Conditions stated below (Definition 3.5) apply Langrangian Dualization methods to explicitly characterize optimal solutions by a system of convex inequalities on a primal-dual solution pair. In a nutshell, the conditions combine complementary slackness with analytical gradient-based optimality criteria to enforce a duality gap of zero.

▶ **Definition 3.5** (Karush-Kuhn-Tucker Conditions)**.** Consider a standard convex program and let λ_i and ν_i be the Lagrange multipliers associated to the constraints 3.2 and 3.3, respectively. Also let the functions $f_0, f_1, \dots, f_m, h_1, \dots, h_q$ be differentiable.

Then for a primal-dual solution pair x and (λ, ν) the following conditions are called the *Karush-Kuhn-Tucker (KKT) Conditions*:

$$\forall i \in [m] \qquad f_i(x) \leq 0 \qquad (3.4)$$

$$\forall i \in [q] \qquad h_i(x) = 0 \qquad (3.5)$$

$$\forall i \in [m] \qquad \lambda_i \geq 0 \qquad (3.6)$$

$$\forall i \in [m] \qquad \lambda_i f_i(x) = 0 \qquad (3.7)$$

$$\nabla f_0(x) + \sum_{i=1}^{m} \lambda_i \nabla f_i(x) + \sum_{i=1}^{q} \nu_i \nabla h_i(x) = 0 \qquad (3.8)$$

where $\nabla f(r) = (\frac{\partial f(r)}{\partial r_j})_{j \in [m]}$ denotes the gradient of the function f.

▶ **Lemma 3.6.** *A primal-dual solution pair x^* and (λ^*, ν^*) for a convex optimization problem is optimal if and only if it satisfies the KKT Conditions.*

For proof of the lemma and general details on the KKT Conditions see [BV04].

Now we will define data rates fulfilling the KKT conditions, thereby proving their optimality. Data rates not equivalent to the following definition will shown to be suboptimal.

▶ **Lemma 3.7.** *A rate assignment $\hat{r}$ to the single-user OFDM scheduling problem with capacity constraints and strictly positive channel costs is optimal if and only if*

$$\hat{r}_j = \begin{cases} 0 & \text{if } \ell/p_j < \psi'(0) \\ x \in \{r \in [0, c_j] \mid \psi'(r) = \ell/p_j\} & \text{if } \psi'(0) \leq \ell/p_j \leq \psi'(c_j) \\ c_j & \text{if } \psi'(c_j) < \ell/p_j \end{cases} \tag{3.9}$$

for some $\ell \in \mathbb{R}$ such that $\sum_{j=1}^{m} \hat{r}_j = d$.

Proof. As rate-power functions are monotonically strictly increasing we know that any rate allocation $(r_j)_{j\in[m]}$ with $\sum_{j=1}^{m} r_j > d$ for the problem according to Definition 3.3 can not be optimal. So we analyze the optimal solutions of the following convex program.

$$\begin{aligned} \text{minimize} \quad & \sum_{j=1}^{m} p_j \cdot \psi(r_j) \\ \text{subject to} \quad & r_j \leq c_j \qquad \forall j \in [m] && (3.10) \\ & r_j \geq 0 \qquad \forall j \in [m] && (3.11) \\ & \sum_{j=1}^{m} r_j = d && (3.12) \end{aligned}$$

In order to characterize the optimal solutions of this program we consider the corresponding KKT conditions with the dual variables ν, μ_j, λ_j associated to the constraints 3.12, 3.10 and 3.11.

For $f(r) = \sum_{j=1}^{m} p_j \cdot \psi(r_j)$ we derive

$$\frac{\partial f(r)}{\partial r_j} = p_j \psi'(r_j) \,,$$

also

$$\frac{\partial g_k(r)}{\partial r_j} = \begin{cases} -1 & \text{if } j = k \\ 0 & \text{else} \end{cases} \qquad \frac{\partial e_k(r)}{\partial r_j} = \begin{cases} 1 & \text{if } j = k \\ 0 & \text{else} \end{cases}$$

for $g_j(r) = -r_j$ and $e_j(r) = r_j - c_j$. Furthermore using

$$\frac{\partial h(r)}{\partial r_j} = 1 \qquad \forall j \in [m]$$

for $h(r) = -d + \sum_{j=1}^{m} r_j$ we can explicitly state the KKT conditions for the considered program.

It is easy to see that KKT condition 3.8 on the gradient of the objective and constraint functions translates to the system of equations

$$\forall j \in [m] \qquad p_j \psi'(r_j) - \lambda_j + \mu_j + \nu = 0 \tag{3.13}$$

which together with the remaining KKT conditions

$$\forall j \in [m] \qquad r_j \leq c_j \tag{3.14}$$

$$\forall j \in [m] \qquad r_j \geq 0 \tag{3.15}$$

$$\sum_{j=1}^{m} r_j = d \tag{3.16}$$

$$\forall j \in [m] \qquad \lambda_j \geq 0 \tag{3.17}$$

$$\forall j \in [m] \qquad \mu_j \geq 0 \tag{3.18}$$

$$\forall j \in [m] \qquad \lambda_j r_j = 0 \tag{3.19}$$

$$\forall j \in [m] \qquad \mu_j (r_j - c_j) = 0 \tag{3.20}$$

yields a characterization of the optimal solutions.

We will first show sufficiency and afterwards necessity of the water-filling rates defined by equation 3.9.

Sufficiency: We define a corresponding feasible dual solution $(\lambda_j)_{j\in[m]}, (\mu_j)_{j\in[m]}, \nu$. First, note that any rates corresponding to the water-filling equation 3.9 and the rate condition $\sum_{j=1}^{m} r_j = d$ are feasible (with respect to the primal program) by definition.

In the dual solution we set $\nu = -\ell$. If $r_j = 0$ we choose $\lambda_j = p_j \psi'(r_j) + \nu$ which is non-negative since $-\nu/p_j \leq \psi'(0)$. Setting $\mu_j = 0$, any KKT condition containing μ_j and λ_j is satisfied. Analogously we can set $\lambda_j = 0$ and $\mu_j = -p_j \psi'(r_j) + \nu$ if $r_j = c_j$.

Thus, the only possibly violated conditions could be the ones containing μ_j and λ_j for some channel j with $0 < r_j < c_j$. The inequalities $0 < r_j < c_j$ require $\lambda_j = \mu_j = 0$ in order to satisfy the complementary slackness conditions 3.19 and 3.20, but since $\psi'(r_j) = \frac{-\nu}{p_j}$ holds it is easy to see that Condition 3.13 is satisfied.

Necessity: By strong duality of convex programming we know that for any optimal rate allocation $(r_j)_{j\in[m]}$ there are values $\nu, (\mu_j)_{j\in[m]}, (\lambda_j)_{j\in[m]}$ such that Conditions 3.13 to 3.20 are satisfied. If some channel $j \in [m]$ has zero capacity $c_j = 0$ then Equation 3.9 is equivalent to $r_j = 0$ independent of the water level ℓ. This constraint is trivially necessary so w.l.o.g. $c_j > 0$ for all $j \in [m]$.

Consider any given optimal primal-dual solution pair and suppose $-\nu/p_j > \psi'(0)$. We will show that in this case $r_j > 0$ will be necessary. Otherwise, for $r_j = 0$, the complementary slackness condition 3.20 implies $\mu_j = 0$, so together with $-\nu/p_j > \psi'(0) = \psi'(r_j)$ and the non-negativity of λ_j we get $p_j \psi'(r_j) - \lambda_j + \mu_j + \nu < 0$ contradicting Condition 3.13. Similarly, for $-\nu/p_j < \psi'(c_j)$ we can show $r_j < c_j$. Hence, in any optimal solution $\psi'(0) < -\nu/p_j < \psi'(c_j)$ implies $0 < r_j < c_j$. But in that case $\lambda_j = \mu_j = 0$ and thereby $p_j \psi'(r_j) + \nu = 0$ which is equivalent to $-\nu/p_j = \psi'(r_j)$.

Now, consider the case $-\nu/p_j = \psi'(0)$. By monotonicity of ψ' we have $p_j \cdot \psi'(r_j) \geq 0$ which together with Condition 3.13 implies $\lambda_j \geq \mu_j$. So for $\lambda_j = 0$ we have $\mu_j = 0$ and $p_j \psi'(r_j) + \nu = 0$. If $\lambda_j > 0$ the equality $r_j = 0$ has to hold and thereby trivially $-\nu/p_j = \psi'(r_j)$. Analogously $-\nu/p_j = \psi'(r_j)$ follows from $-\nu/p_j = \psi'(c_j)$. So the only remaining cases are $-\nu/p_j < \psi'(0)$ and $-\nu/p_j > \psi'(c_j)$.

If $\psi'(0) > -\nu/p_j$ we have $p_j \psi'(r_j) + \nu > 0$ for any (non-negative) data rate r_j as by convexity of the rate-power function the derivative ψ' is monotonically increasing. Evaluating Condition 3.13 this implies $\lambda_j > 0$ since μ has to be non-negative (Condition 3.18). Together

with Equation 3.19 this yields $r_j = 0$ in that particular case. Thus, it is a necessary optimality condition that $r_j = 0$ holds if $\psi'(0) > -\nu/p_j$.

Similarly, for $\psi'(c_j) < -\nu/p_j$, inequality $p_j\psi'(r_j)+\nu < 0$ implies $\mu_j > 0$ by Conditions 3.13 and 3.17, which by the complementary slackness condition 3.20 translates to $r_j - c_j = 0$. ◀

▶ **Lemma 3.8.** *A rate assignment $\hat{r}$ to the single-user OFDM scheduling problem with capacity constraints, strictly positive channel costs and a strictly convex rate-power function is optimal if and only if*

$$\hat{r}_j = \begin{cases} 0 & \text{if } \ell/p_j \leq \psi'(0) \\ \psi'^{-1}(\ell/p_j) & \text{if } \psi'(0) < \ell/p_j < \psi'(c_j) \\ c_j & \text{if } \psi'(c_j) \leq \ell/p_j \end{cases} \tag{3.21}$$

for ℓ such that $\sum_{j=1}^{m} \hat{r}_j = d$.

Proof. Any continuous and monotonically strictly increasing function is invertible, so for strictly convex rate-power functions the function ψ'^{-1} is well-defined on the domain of non-negative real numbers. Trivially $\psi'^{-1}(\ell/p_j)$ is the only element in $\{r \mid \psi'(r) = \ell/p_j\}$. For $\ell/p_j = \psi'(0)$ and $\ell/p_j = \psi'(c_j)$ this function evaluates to 0 and c_j, respectively, which proves the claim. ◀

3.2 NP-completeness of OFDMA Scheduling

After studying efficient solution methods for single-user OFDM we consider the algorithmic complexity of solving multi-user OFDMA scheduling.

While the throughput maximization variant is known to be polynomial time solvable [JL03] Min-Power and Max-Rate OFDMA scheduling is often claimed to be NP-hard in existing literature, though up to our knowledge no valid proof for its hardness has been given before (attempts can be found in [WSEA04] and [VD06]).

In the current section we will show NP-completeness of Min-Power (and thereby also Max-Rate) OFDMA by a reduction from the NP-complete 3-dimensional matching problem.

From the efficient solvability of the single-user OFDM scheduling problem immediately follows that the multi-user variant is contained in the class NP, since for a given channel assignment the problem of deciding if a rate guarantee on all terminals can be satisfied while not violating a given energy bound decomposes into independent single-user subinstances.

However, unlike single-user scheduling the multi-user OFDMA problem can shown to be NP-hard —already for the special case of uniform demands and channel costs restricted to $p_{ij} \in \{p, q, \infty\}$. Thereby the problem can not be solved in polynomial time (unless $P = NP$).

We first observe the intuitive result that in any optimal solution each terminal only distributes its total data rate over more than one channel if the cost discrepancy among those channels is not too large. In the following lemma a channel is said to be *used* in a schedule if its corresponding data rate is strictly positive.

▶ **Lemma 3.9.** *Let (X, r) be any optimal OFDMA schedule with $x_{ij} = x_{ik} = 1$ for a user i with demand d and strictly convex rate-power function ψ where the channel costs p_{ij} and p_{ik} satisfy $p_{ik} > p_{ij} = 0$ or $p_{ik} \geq \frac{\psi'(d)}{\psi'(0)} \cdot p_{ij}$ for $p_{ij}, p_{ik} \in \mathbb{R}_{>0}$ and $\psi'(0) > 0$. Then channel k is not used in the schedule (i.e. $r_k = 0$).*

Proof. We denote $p = p_{ij}$ and $q = p_{ik}$. First consider the trivial case $p = 0$ and $q > 0$. Clearly channel k incurs strictly positive cost for any rate $r_k > 0$ as the rate-power function

is non-negative and strictly increasing. However scheduling the whole demand on channel j is free of cost, thus $r_k = 0$ holds in any optimal schedule.

Now assume $p, q \in \mathbb{R}_{>0}$ with $q \geq \frac{\psi'(d)}{\psi'(0)} \cdot p$ and suppose $r_k > 0$. Lemma 3.8 implies $r_k = \psi'^{-1}(\ell/q)$ for some $\ell > q\psi'(0)$ and $r_j = \psi'^{-1}(\ell/p)$ for the same water level ℓ. However by $\ell > q\psi'(0)$ and $q \geq \frac{\psi'(d)}{\psi'(0)} \cdot p$ we have

$$\frac{\ell}{p} > \frac{q}{p} \cdot \psi'(0) \geq \psi'(d)$$

which by strict convexity of ψ yields $r_j = \psi'^{-1}(\ell/p) > d$ and thereby contradicts optimality of the schedule. ◀

This property allows us to conclude that in instances where distinct channel costs differ by a sufficiently large multiplicative factor each terminal only uses channels of the same cost in any optimal solution. The following lemma implies that in this case the energy cost for each terminal is easily computable once the uniform channel assignment has been fixed.

▶ **Lemma 3.10.** *If user $i \in [n]$ with demand d gets assigned $k \geq 1$ many channels in total which all have the same channel cost p in the Min-Power OFDMA problem the optimal total energy cost for this user is exactly $p \cdot \varphi(k)$ with $\varphi(k) = k \cdot \psi(d/k)$.*

Proof. For any convex function ψ Jensen's inequality (Lemma 2.10) directly implies

$$\sum_{j=1}^{k} \frac{1}{k} \cdot \psi(r_j) \geq \psi\left(\sum_{j=1}^{k} \frac{1}{k} \cdot r_j\right)$$

and thereby

$$\sum_{j=1}^{k} p \cdot \psi(r_j) \geq k \cdot p \cdot \psi(d/k)$$

for any feasible rate allocation $(r_j)_{j\in[m]}$ with $\sum_{j=1}^{k} r_j = d$. ◀

Note that the statement of Lemma 3.9 holds independently of the number of available channels of cost $p = p_{ij}$ and $q = p_{ik}$ for terminal i. Even for arbitrary many channels of cost q, terminal i would rather prefer sustaining its total rate demand on a single channel of cost p. We conclude the following lemma.

▶ **Lemma 3.11.** *Given a strictly convex rate-power function ψ. Then for any $m \geq 1$, $d \in \mathbb{R}_{>0}$ and $p, q \in \mathbb{R}_{\geq 0}$ with $q > p = 0$ or $q \geq \frac{\psi'(d)}{\psi'(0)} \cdot p$ for $p, q \in \mathbb{R}_{>0}$ and $\psi'(0) > 0$ the inequality*

$$p \cdot \psi(d) < m \cdot q \cdot \psi(d/m)$$

holds.

Proof. For $q > p = 0$ the inequality trivially follows by strict monotonicity of the function ψ. So consider the claim for $p > 0$. By strict convexity, we have $\psi(d) < d \cdot \psi'(d)$ and $\psi(d/m) \geq \frac{d}{m} \cdot \psi'(0)$. Thereby

$$m \cdot q \cdot \psi(d/m) \geq \frac{\psi'(d)}{\psi'(0)} \cdot p \cdot m \cdot \psi(d/m) \geq p \cdot d \cdot \psi'(d) > p \cdot \psi(d)$$

holds, which proves the claim. ◀

We are now ready to derive the NP-hardness result.

▶ **Theorem 3.12.** *For any strictly convex rate-power function ψ and $p, q \in \mathbb{R}_{\geq 0}$ with $q > p = 0$ or $q \geq \frac{\psi'(d)}{\psi'(0)} \cdot p$ for $p, q \in \mathbb{R}_{>0}$ and $\psi'(0) > 0$ the Min-Power OFDMA scheduling problem with uniform demand $d > 0$ and channel costs $p_{ij} \in \{p_j, \infty\}$ with $p_j \in \{p, q\}$ is NP-hard.*

Proof. We give a reduction from the NP-complete 3-dimensional matching problem (inspired by the proof of [LST90] for makespan scheduling). The proof is independent of the absolute demand which may take any constant non-zero value, so w.l.o.g. we assume $d = 1$.

Given a tripartite 3-dimensional (hyper-)graph $G = (A \cup B \cup C, E)$ with $|A| = |B| = |C| = s$ and $|E| = t$. We define an OFDMA instance with t many terminals and $t + 2s$ many channels. For each of the nodes in G we identify a corresponding channel. We will call this set of $3s$ many channels the set of *basic channels*. The remaining $t - s$ many channels are called *dummy channels*.

For each (hyper-)edge $e \in E$ we construct a corresponding user which has cost q on the three basic channels corresponding to the nodes incident to e. The dummy channels have uniform cost of p independent of the user. Any other p_{ij} will be set to infinity.

Consider the case where there is a perfect 3-dimensional matching in G. Then there is an assignment of the $3s$ many basic channels to a subset of s many terminals such that each user in the subset gets three basic channels with individual channel cost q. Each of the remaining $t - s$ many terminals gets a dummy channel of cost p so that the total cost of this assignment (with optimally chosen data rates) is

$$E_{\text{matching}} := (t - s) \cdot p \cdot \psi(d) + 3s \cdot q \cdot \psi(d/3) \ .$$

Now suppose that no perfect 3-dimensional matching exists. Consider an optimal schedule and let H denote the subset of terminals which are not assigned any of the dummy channels. As $|H| \geq s$ holds we may choose a subset H' of those terminals of size $|H'| = s$. The cost incurred by the remaining $t - s$ many terminals $K = \{1, \ldots, t\} \setminus H'$ is at least the optimal cost of a *single*-user OFDM instance with demand $(t - s)d$ given $(t - s)$ many channels of cost p and an arbitrary number of channels of cost q. Lemma 3.9 implies that only the channels of cost p are used in any optimal solution to this single-user instance so together with Lemma 3.10 we conclude that the energy $E(K)$ needed by the terminals in K is at least

$$E(K) \geq (t - s) \cdot p \cdot \psi(d) \ .$$

In order to bound the energy $E(H')$ needed by the terminals in H' we note that those s many terminals may use at most $k \leq 3s - 1$ many basic channels in total as otherwise this assignment defines a perfect 3-dimensional matching. So by Lemma 3.10 we have

$$E(H') \geq k \cdot q \cdot \psi(ds/k) \qquad \text{for some } k < 3s - 1.$$

Using Lemma 2.11 for $a := \frac{k}{3s} < 1$ and exploiting the strict convexity of ψ we get $\frac{k}{3s} \cdot \psi(r) > \psi(\frac{k}{3s} \cdot r)$ and thereby

$$k \cdot q \cdot \psi\Big(\frac{ds}{k}\Big) = \frac{k}{3s} \cdot 3s \cdot q \cdot \psi\Big(\frac{ds}{k}\Big) > 3s \cdot q \cdot \psi\Big(\frac{d}{3}\Big)$$

which implies that $E(K) + E(H')$ is strictly greater than E_{matching}.

Thus, any OFDMA scheduling algorithm finding an optimal solution in polynomial time would yield a decision algorithm for the NP-complete 3-dimensional matching problem. ◀

Note that in Chapter 2 the OFDMA scheduling problem was formally defined only for finite channel costs $p_{ij} \in \mathbb{R}_{\geq 0}$. However, allowing $p_{ij} \in \mathbb{R}_{\geq 0} \cup \{\infty\}$ is an interesting extension in its own which will be discussed in a restricted form in Section 3.5.

Intuitively, allowing infinite channel costs should not make the scheduling problem any harder, since any sufficiently large value could be used as a substitute. We will formalize this by the following lemma allowing us to translate the NP-hardness result to the domain of purely finite channel costs.

▶ **Lemma 3.13.** *If Min-Power OFDMA scheduling with n terminals, m channels, uniform demand d and strictly convex rate-power function ψ with $\psi'(0) > 0$ is NP-complete for $p_{ij} \in \{p_1, \ldots, p_z\}$ with $p_k \in \mathbb{R}_{\geq 0} \cup \{\infty\}$ for all $1 \leq k \leq z$ then there exists some $q \in \mathbb{R}_{>0}$ such that the problem is also NP-complete for $p_{ij} \in \{\bar{p}_1, \ldots, \bar{p}_z\}$ with $\bar{p}_k = \min\{p_k, q\}$ for all $1 \leq k \leq z$.*

Proof. For $p = \max\{p_k \in \mathbb{R}_{\geq 0} \mid 1 \leq k \leq z\}$ choose any $q \geq n \cdot \frac{\psi'(d)}{\psi'(0)} \cdot p$. Using the bound $E = n \cdot p \cdot \psi(d)$ we can show that a feasible schedule (of finite cost) to the original instance exists if and only if a schedule of cost at most E can be found to our modified instance with $p_{ij} \in \{\bar{p}_1, \ldots, \bar{p}_z\}$.

Trivially, if there is a finite cost schedule to the original instance, assigning to each terminal one of the channels for which it has a cost of at most p and setting the rates accordingly yields a schedule of total cost at most E (both in the original and the modified instance). If, however, a feasible schedule to our modified instance exists but not to the original one, then there is a terminal using only channels of cost q. The cost incurred by this terminal is at least $m \cdot q \cdot \psi(d/m)$ which by Lemma 3.11 (for $p' = n \cdot p$) is strictly greater than $n \cdot p \cdot \psi(d)$ for the chosen q. ◀

Note that q may increase the size of the input instance. For the Shannon rate-power function, however, this increase is only linear in the original input size when d is constant.

▶ **Corollary 3.14.** *The Min-Power OFDMA problem with n terminals having uniform demand $d = 1$ and m channels using the Shannon rate-power function is NP-complete for $p_{ij} \in \{1, 2, 4n\}$.*

So already for unary coded numbers the problem is NP-hard.

▶ **Corollary 3.15.** *Min-Power OFDMA scheduling is strongly NP-complete.*

While the corollaries above are based on the usage of the Shannon rate-power function the following special case of Theorem 3.12 holds generally.

▶ **Corollary 3.16.** *For any $p \in \mathbb{R}_{>0}$ and any strictly convex rate-power function Min-Power OFDMA with uniform demands $d \in \mathbb{R}_{>0}$ is NP-complete for $p_{ij} \in \{0, p, \infty\}$.*

Of course, the results given in this section directly carry over to the case of Max-Rate OFDMA as the decision variants of both scheduling problems are the same.

3.3 Bounds on the Approximability of OFDMA Scheduling

The proof of Theorem 3.12 can also be adapted to derive inapproximability results. We will use a more advanced reduction from the 3-dimensional matching problem in order to show $1/2$-inapproximability for the Max-Rate OFDMA problem which is based on an idea originally developed by Lenstra, Shmoys and Tardos [LST90] for the makespan scheduling problem. First we state a useful result derived from Lemma 3.11.

▶ **Corollary 3.17.** *Given a strictly convex rate-power function ψ. Then for any $\epsilon, d \in \mathbb{R}_{>0}$ and $p, q \in \mathbb{R}_{\geq 0}$ with $q > p = 0$ or $q \geq \frac{\psi'(d)}{\psi'(0)} \cdot p \cdot \frac{1}{\epsilon}$ for $p, q \in \mathbb{R}_{>0}$ the inequality*

$$p \cdot \psi(d) < q \cdot \psi(\epsilon d)$$

holds.

Proof. Follows directly by Lemma 3.11 with $m \geq 1/\epsilon$ and $p' = p \cdot 1/\epsilon$. ◀

Of course in order to show inapproximability the rate-power function has to be steep enough. We study the case of the Shannon rate-power function $\psi(r) = 2^r - 1$ which is widely used in practice.

▶ **Theorem 3.18.** *For any constant $\epsilon > 0$ the Max-Rate OFDMA problem with the Shannon rate-power function with uniform demand $d \geq 2(1 + \log_2 n)/\epsilon$ and channel costs $p_{ij} \in \{p_j, u\}$ with $p_j \in \{p, q\}$ for $q > p$ where $q \geq \frac{\psi'(d)}{\psi'(0)} \cdot p$ and $u \geq \frac{\psi'(d)}{\psi'(0)} \cdot q \cdot \frac{2}{\epsilon}$ can not be approximated by a factor of $\frac{1}{2} + \epsilon$ in polynomial time (unless $P = NP$).*

Proof. Again we transform the 3-dimensional matching instance $G = (A \cup B \cup C, E)$ with $|A| = |B| = |C| = s$ and $|E| = t$ into an OFDMA instance with t many terminals. However, only $t + s$ many channels will be created.

For a modified construction of dummy channels the triples $e \in E$ will be partitioned into different categories. A triple $e = (a_x, b_y, c_z)$ with $e \in E$ containing a node $a_x \in A$ for $x \in [s]$ will be called *a triple of type x*. For each node $a_x \in A$ let t_x denote the number of triples of type x. We construct $t_x - 1$ many dummy channels of type x having cost p for each terminal representing a triple of type x and cost u otherwise.

So while the number of dummy channels is again $t - s$ we reduce the number of basic channels to $2s$, one for each node in $B \cup C$. Each triple containing a node $b \in B$ or $c \in C$ has cost q on this node. All other costs are set to u.

It is easy to see that if a matching exists then with a total energy bound

$$E_{\text{matching}} := (t - s) \cdot p \cdot \psi(d) + 2s \cdot q \cdot \psi(d/2)$$

every terminal can sustain a data rate of d.

Choosing $d \geq 2(1 + \log_2 t)/\epsilon$ and imposing E_{matching} as energy bound we show that any channel of cost q can sustain a data rate of strictly less than $\frac{d}{2}(1 + \epsilon)$. This is because

$$\psi\Big(\frac{d}{2}(1+\epsilon)\Big) + 1 = 2^{d\epsilon/2} \cdot 2^{d/2} \geq 2t \cdot \Big(\psi\Big(\frac{d}{2}\Big) + 1\Big)$$

for the Shannon rate-power function and the chosen d which together with Lemma 3.11 implies

$$q \cdot \psi\Big(\frac{d}{2}(1+\epsilon)\Big) > 2(t-s) \cdot q \cdot \psi\Big(\frac{d}{2}\Big) + 2s \cdot q \cdot \psi\Big(\frac{d}{2}\Big) > (t-s) \cdot p \cdot \psi(d) + 2s \cdot q \cdot \psi\Big(\frac{d}{2}\Big)$$

for $q \geq \frac{\psi'(d)}{\psi'(0)} \cdot p$. Furthermore, by Corollary 3.17 the data rate sustainable on a channel of cost u with the given energy bound is strictly less than $\epsilon d/2$.

We will now show that if no matching exists there is at least one terminal using at most one channel of cost q, at most one of cost u and none of cost p.

Given $t + s$ many channels and t many terminals there are at least $t - s$ many terminals getting only one channel. If one of those channels has cost p or q on the assigned terminal we are done. Otherwise all channels assigned to those terminals are dummy channels and

the remaining $2s$ many channels are basic channels which have to be distributed among a set H of s many terminals. Since we already removed the set of terminals with at most one channel, each terminal in H gets at least and thereby exactly two channels. If all of those channels would have cost q this would imply a matching, so there is at least one terminal using at most one q-channel and one u-channel.

Thus, we can conclude that at least one terminal can sustain a data rate of strictly less than $(1+2\epsilon)\frac{d}{2} = (\frac{1}{2}+\epsilon)d$ if no matching exists. Thereby any solution to the OFDMA instance with a minimum data rate of $(\frac{1}{2}+\epsilon)d$ shows the existence of a perfect matching in the original 3-dimensional matching instance, so no polynomial time $(1/2+\epsilon)$-approximation algorithm for the Max-Rate OFDMA problem can exist (unless $P = NP$). ◀

Intuitively the situation for Min-Power OFDMA is significantly worse than for the Max-Rate variant, as for the Shannon rate-power function the required energy grows exponentially with the allocated data rate.

▶ **Theorem 3.19.** *Let $\alpha(x)$ be any polynomial time computable function. Then Min-Power OFDMA for n terminals having uniform demand d and m channels with associated costs $p_{ij} \in \{p_j, u\}$ for $p_j \in \{p, q\}$ using the Shannon rate-power function can not be approximated up to a factor of $\alpha(nm)$.*

Proof. Let a 3-dimensional matching instance with t many edges and $3s$ many nodes be given. Like in Theorem 3.18 we construct $n = t$ many terminals and $m = t+s$ many channels and fix $\alpha = \alpha(nm)$. We set $d = 2(\lceil \log_2(\alpha) \rceil + \lceil \log_2(2t) \rceil)$, $p = 1$, $q = 2^d$ and $u = 4^d$. Note that $d \in O(\log(\alpha t))$ and $p, q, u \in O((\alpha t)^4)$ so the binary encoding size of the values is linear in $\log(\alpha t)$ and thereby polynomial in nm.

Analogously to Theorem 3.18 we can show that if no 3-dimensional matching exists at least one terminal has to use at most one channel of cost q and none of cost p. The energy requirement for this terminal to sustain a data rate of d is by Lemma 3.11 (independently of the number of available channels of cost u) at least $q \cdot \psi(d)$. For the Shannon rate-power function and the specified d we have

$$\psi(d) > 2^{d/2}\psi(d/2) \geq \alpha \cdot 2t \cdot \psi(d/2)$$

so $q \cdot \psi(d) > \alpha \cdot E_{\text{matching}}$, proving that no approximation with a factor α is possible (unless $P = NP$). ◀

This shows that Min-Power OFDMA is not approximable within any factor computable in polynomial time (including factors exponential in n and m). Even if we allow the approximation factor to be considered as a function $\alpha'(x)$ of the total encoding size $|I|$ of the respective input instance I (including all cost values), no approximation factor $\alpha'(|I|)$ for any polynomial function α' is possible.

Note that Theorems 3.18 and 3.19 directly translate to the case where $u = \infty$ and thereby already imply approximation hardness for the OFDMA scheduling subclass equivalent to the big goods/small goods case of Santa Claus scheduling (where the sizes s of the jobs/goods are either small or uniformly big on all machines, i.e. $s_j \in \{1, B\}$, but may not be assignable to all of them, $s_{ij} \in \{0, s_j\}$).

3.3.1 Inapproximability of Min-Power OFDMA with Uniformly Related Channel Costs

The above inapproximability result implies that we can not expect to develop any reasonable approximation algorithm for the Min-Power OFDMA problem in general. While the restricted

subclasses studied in Sections 3.4 and 3.5 (linear rate-power functions, binary channel gains) allow exact solvability in polynomial time, each step further away from uniform channel costs for general rate-power functions seems to lead to an intractable and not approximable optimization problem. As we will see below already the subclass of Min-Power OFDMA corresponding to the most basic class considered in the context of machine scheduling, namely the identical machines model, is hard to approximate by any (even exponential) factor computable in polynomial time. In this subclass of OFDMA scheduling, we restrict instances to have *uniformly related channel costs*. In contrast to purely uniform costs different channels may have different costs but each channel j has to have the same cost p_j on all terminals (without assignment restrictions).

We will show approximation hardness of Min-Power OFDMA in the uniformly related channel costs model for any polynomial computable factor α even for the case of uniform demands. This will be done by a reduction from the classic NP-complete 3-Partition problem. As an intermediate step we reduce 3-Partition to an optimization problem we call Min-3π-Partition which is defined below.

▶ **Definition 3.20** (Min-3π-Partition)**.** Given a set $A = \{a_1, \ldots, a_{3n}\}$ of nonnegative integers. Find a partition P of A into triples such that $\sum_{(x,y,z)\in P} xyz$ is minimal.

▶ **Lemma 3.21.** *Let $\alpha(k)$ be any polynomial time computable function. Then Min-3π-Partition can not be approximated in polynomial time within a factor of $\alpha(|A|)$ (where $|A|$ denotes the cardinality of instance A), unless $P = NP$.*

Proof. Assume there is a polynomial time algorithm which approximates Min-3π-Partition within a factor of α. We reduce 3-Partition to Min-3π-Partition in a gap-introducing way.

Let $B = \{b_1, \ldots, b_{3n}\}$ be an instance of 3-Partition. Since the problem is known to be strongly NP-hard, we can assume the b_i to be unary coded. For every integer b_i we construct an element $a_i = (\alpha n)^{b_i}$. Note that for every a_i the size of the binary representation is bounded by $b_i \cdot \log_2(\alpha n)$ and thereby polynomial in the size of the (unary coded) 3-Partition instance. The corresponding Min-3π-Partition instance is given by $A = \{a_1, \ldots, a_{3n}\}$.

If B is a yes-instance of 3-Partition, the cost of an optimal solution to Min-3π-Partition is bounded by $c_{\text{yes}} \leq n \cdot (\alpha n)^s$ where $s := \sum_{i=1}^{3n} b_i/n$. A no-instance B however, would incur a cost of $c_{\text{no}} > (\alpha n)^{s+1} \geq \alpha \cdot c_{\text{yes}}$. Thus the α-approximation algorithm for Min-3π-Partition would solve 3-Partition, which leads to a contradiction (unless P = NP). ◀

Now we show approximation-hardness of Min-Power OFDMA by a gap-preserving reduction from Min-3π-Partition.

▶ **Theorem 3.22.** *Min-Power OFDMA with uniform demands and uniformly related channel costs using the Shannon rate-power function can not be approximated within a factor of α, where α is any polynomial time computable function in the cardinality of the input.*

Proof. Let $A = \{a_1, \ldots, a_{3n}\}$ be an instance of Min-3π-Partition. For every element of A we will add a channel to our constructed OFDMA instance with cost-coefficient $p_i := a_i^3$. The number of terminals is given by n. We choose the uniform demand d to be at least $6 \cdot (\log_2 m + \log_2(p_{\max}/p_{\min}) + \log_2 \alpha)$.

Distributing all channels equally among the terminals incurs a cost less than $m \cdot p_{\max} \cdot 2^{d/3}$, so this can be used as an upper bound on the optimal solution. Any solution where there

is a terminal which gets at most two channels has cost larger than $2 \cdot p_{\min} \cdot 2^{d/2} - \sum_{j=1}^{m} p_j$ which is more than

$$2 \cdot 2^{3 \cdot (\log_2 m + \log_2(p_{\max}/p_{\min}) + \log_2 \alpha)} \cdot p_{\min} - \sum_{j=1}^{m} p_j \geq 2 \cdot m \cdot \frac{p_{\max}}{p_{\min}} \cdot \alpha \cdot 2^{d/3} \cdot p_{\min} - \sum_{j=1}^{m} p_j$$

for the specified d which we can further bound from below by

$$\alpha \cdot m \cdot p_{\max} \cdot 2^{d/3} + m \cdot p_{\max} - \sum_{j=1}^{m} p_j \geq \alpha \cdot m \cdot p_{\max} \cdot 2^{d/3}$$

Thus any solution which is within a factor of at most α times the optimal solution has the property that every terminal gets exactly three channels.

Together with Corollary 3.2 it follows, that the optimal cost of a terminal is given by $3 \cdot \sqrt[3]{2^d \cdot p_i \cdot p_j \cdot p_k} - (p_i + p_j + p_k)$ if the terminal gets assigned channels i, j and k. Thus, the optimal solution yields a partition P of the channel set into triples such that

$$\text{cost}(P) := 3 \cdot 2^{d/3} \cdot \sum_{(i,j,k) \in P} \sqrt[3]{p_i \cdot p_j \cdot p_k} - \sum_{i=1}^{m} p_i$$

is minimal.

Using this equivalence is it easy to see that an α-approximate solution to the OFDMA instance defines an α-approximate solution to the original Min-3π-Partition instance as

$$\sum_{(i,j,k) \in P'} a_i \cdot a_j \cdot a_k > \alpha \cdot \min_{P} \sum_{(i,j,k) \in P} a_i \cdot a_j \cdot a_k$$

for a suboptimal channel partition P' implies

$$\text{cost}(P') + \sum_{i=1}^{m} p_i > \alpha \cdot \text{cost}(\arg\min_P \sum_{(i,j,k) \in P} a_i \cdot a_j \cdot a_k) + \sum_{i=1}^{m} p_i \geq \alpha \cdot \text{cost}(P^*) + \sum_{i=1}^{m} p_i$$

compared to an optimal Min-Power OFDMA solution P^*. ◀

3.4 OFDMA Scheduling with Linear Rate-Power Functions

While rate-power functions according to Definition 2.2 are allowed to be generally convex, the NP-hardness proof in Theorem 3.12 requires strict convexity thereby not covering the class of linear functions. In the current section we will see that this is not just a weakness of the analysis: Min-Power OFDMA scheduling for linear rate-power functions is polynomial time solvable.

Consider the Min-Power OFDMA scheduling problem for linear rate-power functions of the form $\psi(r) = a \cdot r$. It is easy to see that all instances of this form trivially reduce to the simpler case where the rate-power functions equal the identity function $r \mapsto r$ so we will use the definition below.

▶ **Definition 3.23** (Min-Power OFDMA with Linear Rate-Power Functions).

$$\begin{aligned}
\text{minimize} \quad & \sum_{i=1}^{n}\sum_{j=1}^{m} x_{ij} \cdot p_{ij} \cdot r_j \\
\text{subject to} \quad & \sum_{j=1}^{m} x_{ij} r_j \geq d_i && \forall i \in [n] \\
& \sum_{i=1}^{n} x_{ij} \leq 1 && \forall j \in [m] \\
& x_{ij} \in \{0,1\}, \quad r_j \geq 0 && \forall i \in [n] \quad \forall j \in [m]
\end{aligned}$$

Note that this formulation also incorporates the case of different rate-power functions $\psi_{ij}(r) = a_{ij} \cdot r$ for each user-channel pair $i \in [n]$, $j \in [m]$ using appropriate cost coefficients $p'_{ij} = p_{ij} \cdot a_{ij}$.

We will show that this problem can be reduced to a min-cost matching problem by the following straightforward transformation. The key is, that with linear rate-power functions we know the existence of simple optimal solutions where each terminal uses exactly one channel.

▶ **Definition 3.24** (Min-Cost Matching Instance for OFDMA with Linear Rate-Power Functions). Let a Min-Power OFDMA instance with linear rate-power functions (Definition 3.23) with n users, m channels, channel costs p_{ij}, and demands d_i be given.

We construct the (undirected) complete bipartite graph $K_{n,m}$ with edges between all users $u_i \in U$ and channels $c_j \in C$, where the cost of edge $\{u_i, c_j\}$ is defined by $w_{ij} = p_{ij} \cdot d_i$.

First we show that in the graph defined above for any given feasible OFDMA schedule an appropriate matching of total cost not more than the cost of the schedule exists.

▶ **Lemma 3.25.** *For any feasible OFDMA schedule with total cost w there exists a matching of size n and cost not more than w in the bipartite graph according to Definition 3.24.*

Proof. The cost of a given feasible schedule (X, r) for OFDMA with linear rate-power functions is

$$w = \sum_{i=1}^{n}\sum_{j=1}^{m} x_{ij} \cdot p_{ij} \cdot r_j = \sum_{i=1}^{n} \sum_{\{j|x_{ij}=1\}} p_{ij} \cdot r_j \geq \sum_{i=1}^{n} \sum_{\{j|x_{ij}=1\}} \min\{p_{ij} \mid x_{ij} = 1\} \cdot r_j$$

So we can bound w by

$$w \geq \sum_{i=1}^{n} \min\{p_{ij} \mid x_{ij} = 1\} \cdot \sum_{\{j|x_{ij}=1\}} r_j$$

which by feasibility of the data-rates implies

$$w \geq \sum_{i=1}^{n} d_i \cdot \min\{p_{ij} \mid x_{ij} = 1\} = \sum_{i=1}^{n} \min\{w_{ij} \mid x_{ij} = 1\} \ .$$

However each feasible assignment matrix $X = (x_{ij})_{i\in[n],j\in[m]}$ defines a subgraph of the bipartite graph according to Definition 3.24 by removing the edges $\{u_i, c_j\}$ corresponding to variables x_{ij} with $x_{ij} = 0$. Furthermore, the term $\sum_{i=1}^{n} \min\{w_{ij} \mid x_{ij} = 1\}$ equals the cost of

a min-cost matching of size n in the subgraph induced by the edges corresponding to entries $x_{ij} = 1$ (which exists by feasibility of the channel assignment). Since a min-cost matching of size n in the original graph can not be of greater cost this proves the claim. ◀

The inverse direction is immediately obvious but for sake of completeness we will give a formal proof.

▶ **Lemma 3.26.** *Given a (not necessarily min-cost) matching M of size n and total cost w for the graph according to Definition 3.24 with respect to a given Min-Power OFDMA instance. Then there also is a corresponding OFDMA schedule with total cost w.*

Proof. We define a schedule (X, r) for a given matching M in the following way: for every edge $\{u_i, c_j\} \in M$ we set $x_{ij} = 1$ and $r_j = d_i$. Since M is a matching each channel is assigned to at most one terminal, so the rates are well-defined and the exclusiveness constraints $\sum_{i=1}^{n} x_{ij} \leq 1$ with $x_{ij} \in \{0, 1\}$ are satisfied. All yet undefined values x_{ij} and r_j will be set to zero. Non-negativity trivially holds and the constructed schedule fulfills the demand constraints because for $|M| = n$ each terminal gets exactly one assigned channel with data rate equal to its demand. So only the cost-equivalence is left to be shown.

Each channel j with positive rate of r_j incurs a cost of $p_{ij} \cdot d_i = w_{ij}$ where i is the unique terminal assigned to j in the matching M. Summing up yields the total cost $w = \sum\{w_{ij} \mid \{u_i, c_j\} \in M\}$ of the given matching M. ◀

Having shown that the reduction is sound, we can derive that the Min-Power OFDMA problem for linear rate-power functions can be solved in polynomial time by applying min-cost matching algorithms.

▶ **Theorem 3.27.** *Min-Power OFDMA with linear rate-power functions on m channels and n users, with largest/smallest demand $d_{\max}/d_{\min}$ and largest/smallest channel cost $p_{\max}/p_{\min}$, can be solved (after a linear time precomputation) in time $\mathrm{MCM}(n + m, n \cdot m, n, \frac{d_{\max}}{d_{\min}} \cdot \frac{p_{\max}}{p_{\min}})$, where $\mathrm{MCM}(|V|, |E|, w, k)$ denotes the time to compute a min-cost matching of size w in a bipartite graph with $|V|$ many nodes and $|E|$ many edges each having a cost of at most k.*

Proof. Follows directly by the transformation according to Definition 3.24 and Lemmas 3.25 and 3.26. ◀

▶ **Corollary 3.28.** *Min-Power OFDMA for linear rate-power functions on m channels and n users can be solved in time $O(m(n^2 + \log m))$.*

Proof. Using the Hungarian method the time needed to compute a min-cost matching of size n can be bounded by $O(n(|E| + |V| \log |V|))$ (see, e.g., [Sch03]) which results in the claimed running time for $|E| \in O(nm)$ and $|V| \in O(m)$. ◀

▶ **Corollary 3.29.** *Min-Power OFDMA for linear rate-power functions on m channels and n users can be solved in time $O(n^2 m \cdot \log\log(\frac{d_{\max} p_{\max}}{d_{\min} p_{\min}}))$.*

Proof. We again apply the Hungarian method. However by using the algorithm of Karlsson and Poblete [KP83] as a subroutine a shortest path can be found in $O(|E| \log\log k)$ for k denoting the largest edge cost which yields a total running time of $O(n \cdot nm \log\log k)$. ◀

3.5 OFDMA with Uniform and Binary Channel Gains

In this section we study the restriction of Min-Power OFDMA to uniform channels, i.e. every channel having the same gain or cost. This problem will be shown to be polynomial time solvable, even if we consider a more general formulation where a channel j may either be suitable for a user i with normalized uniform gain $g_{ij} = p_{ij}^{-1} = 1$ or a transmission of user i over channel j is not possible. As we can express the restriction that a channel j is not usable by a terminal i by setting the corresponding channel gain g_{ij} to zero (equivalent to $p_{ij} = \infty$), we call this the *binary channel gain* model.

Note that this restriction is similar to the big goods/small goods case of Santa Claus scheduling where each job has either uniform size 1 on all machines or size B but some machines may be unable to schedule some jobs. We have already seen in Section 3.2 that restricting OFDMA scheduling instances to channel costs $p_{ij} \in \{p_j, \infty\}$ with $p_j \in \{1, B\}$ still leads to an NP-complete optimization problem. However, enforcing $p_{ij} \in \{p, \infty\}$ allows for polynomial time solvability.

Furthermore, general user-specific rate-power functions ψ_i for each user $i \in [n]$ may be allowed without rendering the problem to be computationally infeasible, which leads to the following generalized problem formulation. For soundness, the suitability of each channel-user pair is encoded in the demand constraint (instead of $p_{ij} \in \{1, \infty\}$ in the objective function), i.e. only when the gain is nonzero the transmitted rate is accounted for.

▶ **Definition 3.30** (General Min-Power OFDMA with Binary Channel Gains)**.** Given n terminals with user-specific rate-power functions ψ_i and data rate demands d_i. Also let m channels with binary channel gains $g_{ij} \in \{0, 1\}$ for each terminal-channel pair be given.

$$
\begin{aligned}
\text{minimize} \quad & \sum_{i=1}^{n} \sum_{j=1}^{m} x_{ij} \cdot \psi_i(r_j) & & \\
\text{subject to} \quad & \sum_{j=1}^{m} x_{ij} \cdot g_{ij} \cdot r_j \geq d_i & & \forall i \in [n] \\
& \sum_{i=1}^{n} x_{ij} \leq 1 & & \forall j \in [m] \\
& x_{ij} \in \{0, 1\}, \quad r_j \geq 0 & & \forall i \in [n] \quad \forall j \in [m]
\end{aligned}
$$

For binary/uniform channel gains the data rate allocation for each single user becomes trivial, since in that case the equal distribution of data rates on the assigned available channels is optimal (see Lemma 3.10).

In [KPL06] the authors conjecture that assuming equally distributed rates for each terminal might help for designing polynomial time algorithms even for Min-Power OFDMA with general channel costs. However, the condition of equal-rate distribution alone is not enough to guarantee polynomial time solvability. As we have seen Section 3.2, even when in every optimal solution the data rates of each terminal are equally distributed among the assigned channels, the problem remains NP-hard.

However, for the binary channel gain model the channel assignment subproblem can be formulated as a linear cost flow problem. We use $\varphi_i(k) = k \cdot \psi_i(d_i/k)$ to compute the energy cost incurred by terminal i if k channels of non-zero gain are assigned to it. Note that the functions φ_i have the same form as the perspective function introduced in Section 2.4, so it immediately follows by Lemma 2.12 that the functions are convex (when extended

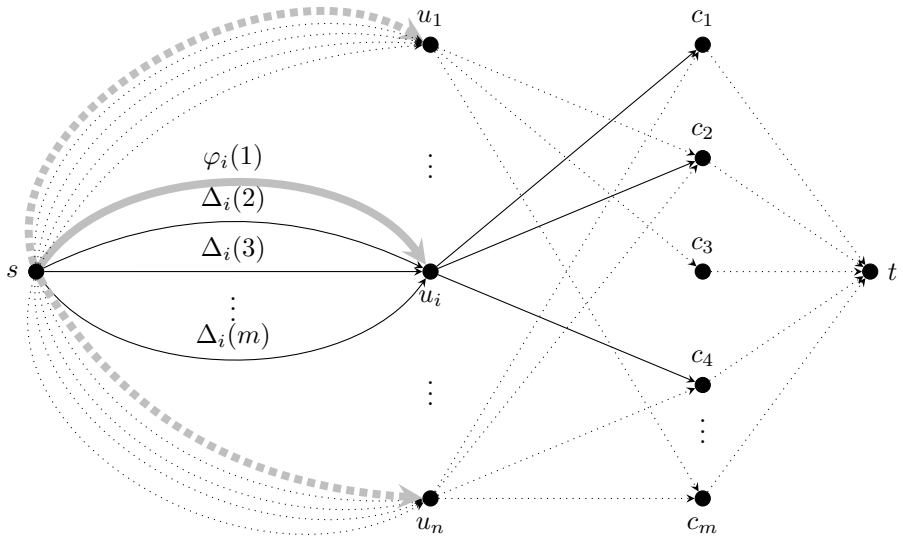


Figure 3.2 Terminal-cost representation network for OFDMA with binary channel gains where user i is able to use channels 1, 2 and 4. (Only edges incident to user node u_i are drawn solid. Thick gray lines indicate a minimum flow of 1.)

to the domain of strictly positive real numbers). We use the property that the marginal benefit (decrease in cost) of assigning one additional channel to a terminal decreases with an increasing number of already assigned channels to it, that is, the (negative valued) function

$$\Delta_i(k) := \varphi_i(k) - \varphi_i(k-1) \quad \text{for } k \geq 2,\ k \in \mathbb{N}$$

is monotonically increasing for each user $i \in [n]$ which follows directly by the convexity of φ_i.

Using the functions Δ_i to define appropriate edge costs we can construct a (linear cost) flow network that is able to represent feasible OFDMA schedules by corresponding integral flows where an integral min-cost flow is equivalent to an optimal OFDMA schedule.

▶ **Definition 3.31** (Terminal-Cost Representation Network)**.** Given n users and m channels with binary channel gains and user-specific rate-power functions ψ_i.

We define a linear cost flow network containing a directed bipartite graph as a subgraph, consisting of nodes $u \in U$ representing the users and nodes $c \in C$ representing the channels with an edge from user to channel if the corresponding user-channel pair has nonzero gain.

Each channel node has an outgoing edge to the sink and each user node is directly fed via m many parallel edges by the source. The (linear) costs of the incoming edges $e_1^i, \ldots, e_m^i$ for each user node $u_i \in U$ are defined by $\text{cost}(e_k^i) = \Delta_i(k)$ for $2 \leq k \leq m$ and $\text{cost}(e_1^i) = \varphi_i(1)$. All other edge-costs are set to zero. Every edge has a uniform capacity of 1 and additionally there is a lower bound of 1 on the minimum flow over e_1^u for each user u.

For a graphical representation see Figure 3.2.

Using trivial reductions (see [AMO93]), flow problems with negative edge costs and bounds on the minimum flow over an edge can be transformed into a standard min-cost flow formulation without increasing the number of nodes or edges. Parallel edges can be replaced

by paths over intermediate nodes, increasing the number of nodes and edges each by the number of parallel edges.

The correspondence between an integer flow solution and an OFDMA schedule will be defined straightforward: channel j is assigned to user i if and only if there is a flow of 1 on the edge between nodes u_i and c_j. By integrality, any channel is assigned to at most one user and the flow value is equivalent to the number of used channels.

Using the integrality property of (linear) min-cost flow problems (see, e.g., [AMO93]) we know that any terminal-cost representation network (Definition 3.31) has an optimal flow which is integral. We make the following observation.

▶ **Lemma 3.32.** *Any feasible flow with minimum total cost in a given terminal-cost representation network is also a maximum flow in this network.*

Proof. Suppose f is not a maximum flow. Then there is a path from s to t in the residual network. Consider the last three nodes visited by the path, i.e. u_i, c_j, t for some $u_i \in U$ and $c_j \in C$. Since the value of f is smaller than m there is an edge (s, u_i) with non-empty residual capacity which has negative cost (as, by feasibility, this can not be e_1^i). So augmenting f along (s, u_i, c_j, t) yields a flow with smaller cost leading to a contradiction. ◀

▶ **Lemma 3.33.** *The cost of any integral min-cost flow in a terminal-cost representation network is equivalent to the cost of the corresponding OFDMA schedule.*

Proof. We consider an integral min-cost flow f of maximum value z. It is easy to see that f can be decomposed into z many paths of the form (s, u, c, t) for $u \in U$ and $c \in C$ each with flow 1.

We verify the claim of cost-equivalence for any user $i \in [n]$. Let k be the number of paths (of the flow decomposition) leading through node u_i. This corresponds to k many assigned channels to user i in the corresponding OFDMA solution (since all edges have capacity 1) with a total cost of $\varphi_i(k)$.

Let $e_{\pi(1)}, \ldots, e_{\pi(k)}$ (for some injective mapping $\pi\colon [k] \to [m]$ with $\pi(1) = 1$) denote the k distinct incoming edges to u_i which are contained in the flow decomposition (note that e_1 has to be contained because of the corresponding minimum flow bound). Then the total cost of the flow through node u_i equals $\varphi_i(1) + \sum_{j=2}^{k} \Delta_i(\pi(j))$ which by monotonicity of Δ_i is at least $\varphi_i(1) + \sum_{j=2}^{k} \Delta_i(j)$. However, the cost through node u_i can not be larger. Otherwise, redirecting flow over $\{e_1^i, \ldots, e_k^i\}$ instead of $e_{\pi(1)}, \ldots, e_{\pi(k)}$ would yield a feasible flow of the same value but of smaller cost. Thus, we have

$$\varphi_i(1) + \sum_{j=2}^{k} \Delta_i(\pi(j)) = \varphi_i(1) + \sum_{j=2}^{k} \Delta_i(j) = \varphi_i(k)$$

which proves the claim. ◀

▶ **Lemma 3.34.** *If there is a solution of total cost w to the general Min-Power OFDMA scheduling problem with binary channel gains (according to Definition 3.30), then there is also a flow in the corresponding terminal-cost representation network of total cost w.*

Proof. Consider a feasible schedule for the OFDMA problem. For any user $i \in [n]$ let $c_{j_1}, \ldots, c_{j_k}$ be the set of assigned channels in this schedule. Then for each edge e_ℓ^i with $1 \leq \ell \leq k$ the path (s, u_i, c_{j_ℓ}, t) leading over this edge can support a flow of value 1. The total cost of those path flows is $\varphi_i(1) + \sum_{\ell=2}^{k} \Delta_i(\ell) = \varphi_i(k)$. Since the channels are assigned

exclusively to the users, the defined flows for each user do not interfere so that together they result in a feasible flow of total cost w. ◀

Combining Lemmas 3.32, 3.33 and 3.34 we get that any (integral) min-cost maximum flow directly translates into a minimum power OFDMA schedule. Since the corresponding terminal-cost representation network has $n+m+2$ many nodes and at most $2nm+m$ many edges (resp. $nm+n+m+2$ nodes and $3nm+m$ many edges after replacing the parallel edges) we can immediately derive the following theorem.

▶ **Theorem 3.35.** *General Min-Power OFDMA with binary channel gains on m channels and n users can be solved (after a linear time precomputation) in time* $\mathrm{MCF}(O(nm), O(nm), m)$*, where* $\mathrm{MCF}(|V|, |E|, w)$ *denotes the time to compute an (integral) min-cost flow of value w on a standard min-cost flow network with $|V|$ many nodes and $|E|$ many edges.*

A complexity survey of known algorithms for minimum-cost flow can be found in [Sch03]. For $w < |V|$ one of the asymptotically fastest algorithms is the enhanced successive shortest path algorithm by Edmonds and Karp [EK72] and Tomizawa [Tom71] which needs at most time $O(w \cdot (|E| + |V| \log |V|))$ using the Fredman and Tarjan [FT87] implementation of the Dijkstra algorithm.

In order to avoid the blowup in the number of nodes caused by the removal of parallel edges we use the following observation. As we have seen in the proof of Lemma 3.33, any minimum cost path visiting node u_i for some $i \in [n]$ directly after node s will pick the unused edge e^i_j with smallest cost, so of the m parallel edges for each user i only one has to be considered for each path augmentation. Thus, the complexity of the shortest path search in each iteration of the successive shortest path algorithm can be reduced to $O(mn + m \log m)$ when considering only $O(m)$ many nodes, leading to the following result.

▶ **Corollary 3.36.** *General Min-Power OFDMA with binary channel gains on m channels and n users can be solved in time $O(m^2(n + \log m))$.*

We deliberately considered only *strongly* polynomial min-cost flow algorithms here, as the costs for the arcs in the terminal-cost representation network defined by the functions φ_i may evaluate to real numbers with arbitrary precision. The complexity of dealing with these numbers could be very high, so using algorithms which involve only comparison the given values are advantageous. Of course, using appropriate min-cost flow algorithms, the method above can be easily adapted to exploit the special structure of instances with a very small domain of possible cost values implied by the rate-power functions.

The Multiple-Assignment Problem. In [WM64] Walkup and MacLaren consider a matching problem similar to the above OFDMA scheduling variant. Their multiple-assignment problem aims to find a maximum total benefit matching between a set of men and a set of tasks where the benefit of a task is measured by a concave function on the numbers of men assigned to it. Using Lemma 3.10 this directly applies to the OFDMA problem defined in Definition 3.30 just with inverted signs —instead of maximizing the sum of concave benefit functions we consider the equivalent problem of minimizing the sum of the convex cost functions φ_i.

The algorithm proposed in [WM64] for the concave benefit assignment problem is basically equivalent to the successive shortest path algorithm for the corresponding min-cost flow problem.

To summarize the algorithm from [WM64] we assume w.l.o.g. that the given benefit functions are positive-valued. The undirected bipartite graph $G = (M \cup T, E)$ representing

the qualification of men to tasks is augmented by an 'idle' task t^* with zero benefit. An initial matching with all men assigned to task t^* is assumed. Then for any man $1 \leq k \leq |M|$ the algorithm successively searches for an alternating path involving man k (which at that time is still assigned to task t^*) with a maximum increase in benefit. The symmetric difference of the new path and the old matching defines the new matching. After $|M|$ many iterations an optimal assignment is found.

As the marginal benefit of the alternating path searched in each iteration of the algorithm can be expressed solely in terms of the benefit function of the end node different to t^*, the search for the optimal path can be reduced to a reachability search with subsequent selection of the best reachable node, which can be done linearly in the number of edges.

Exploiting this fact we can further improve the upper bound of the running time stated in Corollary 3.36.

▶ **Corollary 3.37.** *General Min-Power OFDMA with binary channel gains on m channels and n users can be solved in time $O(m^2 n)$.*

Note that, although we speak of uniform/binary channel gains, distinct channel gains for different users are possible, as this can be encoded as a factor into the user-specific rate-power functions ψ_i.

This could provide a helpful addition to existing heuristic solution methods. Several heuristics assume each user to experience the same gain on all available channels (with possibly different gains for different users) to simplify the computation. Our algorithm allows excluding channels which are severely disturbed and thereby increasing the quality of the assignment without significantly increasing the complexity (allowing binary channel gains instead of uniform ones basically increases the running time only by a factor of m). Thus, it could also be practically relevant for finding good heuristic solutions to OFDMA with general channel gains.

Purely Uniform Channel Gains. When all channel gains are purely uniform, that is, an instance of the OFDMA problem according to Definition 3.30 satisfies $g_{ij} = 1$ for all users i and channels j, the algorithms above iteratively assign yet unallocated channels greedily to the user with maximum improvement $\varphi_i(k+1) - \varphi_i(k)$ (without having to care about any augmenting path structure). This directly corresponds to the discrete water-filling approach used as a subroutine in some heuristic OFDM scheduling algorithms (e.g. [KLL03]).

In total, the running time needed to find an optimal solution to the OFDMA problem with unrestricted uniform channel costs can be bounded by $O(nm)$.

As also the input size is reduced (to $O(m)$ plus the encoding size of $\varphi_1, \ldots, \varphi_n$) in the case of purely uniform demands (because no channel costs or restrictions have to be encoded) we can even further reduce the running time for $m \gg n$ by applying capacity scaling (see, e.g., [AMO93]) on the convex cost integral flow problem described by the convex program

$$\begin{aligned} \text{minimize} \quad & \sum_{i=1}^{n} \varphi_i'(k_i) \\ \text{subject to} \quad & \sum_{i=1}^{n} k_i = m \\ & k_i \in \{1, \ldots, m\} \qquad \forall i \in [n] \end{aligned}$$

with

$$\varphi_i'(k) := \begin{cases} \varphi_i(k) & \text{for } k > 0 \\ \infty & \text{for } k = 0 \end{cases}$$

which can be done in $O(n^2 \log(n) \cdot \log(m))$.

3.6 A Convex Fractional Relaxation of OFDMA Scheduling

Let us consider the mathematical programming formulation of the standard Min-Power OFDMA problem according to Definition 2.6.

$$\begin{aligned} \text{minimize} \quad & \sum_{i=1}^{n}\sum_{j=1}^{m} x_{ij} \cdot p_{ij} \cdot \psi(r_j) \\ \text{subject to} \quad & \sum_{j=1}^{m} x_{ij} r_j \geq d_i && \forall i \in [n] \\ & \sum_{i=1}^{n} x_{ij} \leq 1 && \forall j \in [m] \\ & x_{ij} \in \{0,1\}, \quad r_j \geq 0 && \forall i \in [n] \quad \forall j \in [m] \end{aligned}$$

Even when relaxing the integrality condition for variables x_{ij} this is not a convex optimization problem, as neither the objective function $x \cdot \psi(r)$ nor the term $x \cdot r$ appearing in the demand constraint is convex in (x, r). As a simple counterexample consider the points $A = (0, 1, 0)$, $B = (1, 0, 0)$ and $C = (1/2, 1/2, 1/4)$ contained in the graph of the function $x \cdot \psi(r)$ for $\psi(r) = r$. Then C lies strictly above the line segment defined by A and B.

It is easy to replace the term $x_{ij} r_j$ in the constraint by a variable r_{ij} representing the data rate of terminal i on channel j and enforce $r_j = \sum_{i=1}^{n} r_{ij}$. However there is no direct way to remove the variables x_{ij} in the objective function without simplifying the problem too much. Replacing $x_{ij}\psi(r_j)$ by $\psi(r_{ij})$ would not penalize the multiple usage of a channel in any way, so in most cases solutions derived from this formulation would not be very useful.

The authors in [WCLM99] try to counter this by replacing the functions $x_{ij}\psi(r_j)$ by $x_{ij}\psi(r_{ij}/x_{ij})$ which is convex for $r_{ij} \in \mathbb{R}_{\geq 0}$ and $x_{ij} > 0$ (see Lemma 2.12). However, extending the function on the whole domain of $\mathbb{R}^2_{\geq 0}$ would either result in non-zero costs for any unused terminal-channel pair with $x_{ij} = 0$ or render the objective function to be non-convex.

Nevertheless, an appropriate fractional relaxation of the problem can be derived. For each terminal-channel pair we introduce an individual energy budget represented by a variable e_{ij} and corresponding constraints enforcing $x_{ij} p_{ij} \psi(r_j) \leq e_{ij}$ for all $i \in [n]$ and $j \in [m]$. The total needed energy can then be characterized as the sum of the single energy budgets $\sum_{i=1}^{n}\sum_{j=1}^{m} e_{ij}$ which replaces the non-convex objective function.

However, the new constraints $x_{ij} p_{ij} \psi(r_j) \leq e_{ij}$ still have to be translated into a convex form. W.l.o.g. we assume $p_{ij} > 0$ for all $i \in [n]$ and $j \in [m]$ in the transformation below, since for $p_{ij} = 0$ condition $x_{ij} p_{ij} \psi(r_j) \leq e_{ij}$ is trivially fulfilled. We will first reformulate the constraint for the case of $x_{ij} = 1$ and check later that it is also compatible for $x_{ij} = 0$.

Instead of the rate-power function ψ we can also work with an inverse *power-rate function* ρ without changing the set of feasible solutions. Since ψ is convex, strictly increasing, and

continuous, the inverse $\psi^{-1}\colon \mathbb{R}_{\geq 0} \to \mathbb{R}_{\geq 0}$ is well defined and also continuous and strictly increasing. Furthermore ψ^{-1} is concave.

For each terminal i we define the function $\rho_i(e) = \min\{\psi^{-1}(e), d_i\}$ which —as a pointwise minimum of concave functions— is concave, too. Note that imposing $r_j \leq d_i$ for each $i \in [n]$ and $j \in [m]$ with $x_{ij} = 1$ does not change the value of an optimal solution to the OFDMA scheduling problem. Thus, we can replace the condition $p_{ij}\psi(r_j) \leq e_{ij}$ by the stronger inequality $r_j \leq \rho_i(e_{ij}/p_{ij})$.

By $\rho_i(e) \leq d_i$ for all values $e \in \mathbb{R}_{\geq 0}$ we have $\lim_{y\to 0} y \cdot \rho_i(x/y) = 0$ for each $i \in [n]$. Hence, the function $\xi_i(e, x)$ defined by

$$\xi_i(e,x) = \begin{cases} 0 & x = 0 \\ x\rho_i(e/x) & x > 0 \end{cases}$$

is a continuous extension of the concave perspective function of ρ_i and is concave, too. We call ξ_i the *extended power-rate function* for terminal i. A more direct characterization of this function can be given by the equivalent definition $\xi_i(e, x) = \min\{xd_i, x\psi^{-1}(e/x)\}$.

Trivially, for $x = 1$ we have $\xi_i(e, x) = \rho_i(e)$ so we may freely substitute $r_j \leq \rho_i(e_{ij}/p_{ij})$ by the constraint $r_j \leq \xi_i(e_{ij}/p_{ij},\ x_{ij})$ for $x_{ij} = 1$. For $x_{ij} = 0$ we do not want r_j to be restricted by the energy budget of terminal i in any way. Luckily $x_{ij}r_j \leq \xi_i(e_{ij}/p_{ij},\ x_{ij})$ is appropriate for both $x_{ij} = 1$ and $x_{ij} = 0$. Replacing $x_{ij}r_j$ by r_{ij} we get the convex constraint

$$r_{ij} - \xi_i(e_{ij}/p_{ij},\ x_{ij}) \leq 0 \qquad \forall i \in [n] \quad \forall j \in [m]$$

which results in the convex program stated in Definition 3.38.

A similar approach to formulate a fractional version of the Max-Rate OFDMA scheduling problem (using the Shannon rate-power function) as a convex optimization problem has been used in [RC00] and [SAE05]. However, those formulations implicitly assume data rates to be strictly positive for *all* terminal-channel pairs $i \in [n]$, $j \in [m]$ which may not be realistic in general.

▶ **Definition 3.38** (Relaxed Convex Programming Formulation of Min-Power OFDMA)**.**

$$\begin{aligned}
\text{minimize} \quad & \sum_{i=1}^{n}\sum_{j=1}^{m} e_{ij} \\
\text{subject to} \quad & r_{ij} - \xi_i(e_{ij}/p_{ij},\ x_{ij}) \leq 0 && \forall i \in [n] \quad \forall j \in [m] \\
& \sum_{j=1}^{m} r_{ij} \geq d_i && \forall i \in [n] \\
& \sum_{i=1}^{n} x_{ij} \leq 1 && \forall j \in [m] \\
& x_{ij} \geq 0, \quad r_{ij} \geq 0, \quad e_{ij} \geq 0 && \forall i \in [n] \quad \forall j \in [m]
\end{aligned} \tag{3.22}$$

Note that the set of optimal solutions for *integral* x_{ij} is equivalent to the set of optimal solutions for the original Min-Power OFDMA problem according to Definition 2.6.

Our formulation already filters out a lot of inappropriate fractional solutions by a technique similar to the *job truncation* idea from [LST90] for the makespan scheduling problem. By the definition of ξ_i in Constraint 3.22 we enforce $r_{ij} \leq x_{ij}d_i$. In our intuitive interpretation, with r_{ij} representing $x_{ij}r_j$, this corresponds to the condition that the total data rate r_j on

each channel j should not be larger than the demand of any terminal this channel is assigned to —which is implicit in any optimal integral solution. However, we have to be careful with this interpretation as we can not enforce the variables r_{ij} to evaluate to the intended value of $x_{ij}r_j$ with $r_j = \sum_{i=1}^{n} r_{ij}$. Nevertheless, as we will see in Section 4.3, $r_{ij} \leq x_{ij}d_i$ is a useful property when aiming to approximate the (integral) OFDMA scheduling problem, because it rules out many pathological fractional schedules with unlimited integrality gap like, for instance, those using only one channel to satisfy the total demand of all terminals.

Chapter 4

Approximation Algorithms for Max-Rate OFDMA Scheduling

The inapproximability results in Chapter 3 imply that we can not expect to develop any reasonable approximation algorithm for the Min-Power OFDMA problem, even in the basic case of uniformly related channel costs with uniform demands (Theorem 3.22) as well as in the restricted big goods/small goods case (Theorem 3.19).

However, the huge gap between the inapproximability results given in Section 3.3 suggests that the corresponding rate maximization problem might be more susceptible to approximation approaches. Intuitively, for steep rate-power functions the objective value of Min-Power OFDMA is very sensitive to changes in the data rate allocation, whereas Max-Rate OFDMA allows for relatively large changes in the energy constraint without affecting the objective too much.

While Min-Power OFDMA with uniformly related channel costs was shown to be not approximable by any polynomial time computable factor, we will see in Section 4.1 that the formulation of Max-Rate OFDMA not only allows for efficient constant-factor approximation algorithms but also admits the construction of a PTAS for this subclass. To the best of our knowledge no polynomial time approximation algorithm for Max-Rate OFDMA with any provable theoretical worst-case bound on any non-trivial subclass has been known before.

Considering the Max-Rate OFDMA problem on more general classes (like the restricted assignment model, see Section 4.2, or general unrelated channel costs, see Section 4.3), a similarly strong approximability result is not possible anymore —as implied by the upper bound of $1/2$ proven in Chapter 3. In Section 4.2 a reduction will show that in order to achieve constant or logarithmic approximation factors, the restricted assignment case is already as hard to approximate as the general unrelated cost model.

We will develop several approximation algorithms for the general Max-Rate OFDMA scheduling problem. A simple matching-based $1/m$-approximation algorithm is given in Section 4.3.1. Using the relaxed convex programming formulation introduced in Section 3.6 we also derive appropriately rounded solutions that can be used to compute better approximations for special cases like restricted assignment instances with uniform demands. Furthermore, relaxing the integrality constraint to half-integrality allows us to determine $1/2$-approximations for general Max-Rate OFDM (with arbitrary unrelated channel costs) in Section 4.3.2.

4.1 Uniformly Related Channel Costs

In the following we derive efficient constant factor approximation algorithms for the uniformly related channel costs model (both for the case of uniform as well as for non-uniform demands). These algorithms are presented in Sections 4.1.1 and 4.1.2.

Subsequently, in Section 4.1.3.1 we present a polynomial time approximation scheme (PTAS) for the case of uniform demands in the uniformly related channel cost model. The PTAS will be constructed by adapting techniques from the area of makespan scheduling [HS87].

As a byproduct a PTAS for the Santa Claus Scheduling problem on identical machines (different from existing ones [Woe97, AE98, ES04]) will be given in Section 4.1.3.2.

A more modular reduction of OFDMA scheduling with uniform demands and uniformly related channels to the Santa Claus scheduling problem with identical machines is given in Section 4.1.3.3, which allows us to adapt techniques [AE98] from machine scheduling to derive a PTAS for the case of non-uniform demands in Section 4.1.3.4.

4.1.1 A $1/2$-Approximation-Algorithm for Non-uniform Demands

We will develop an algorithm that for a given λ either returns a feasible schedule guaranteeing an objective value of $\lambda/2$ for the considered Max-Rate OFDMA instance with uniformly related channel costs or determines that no solution with minimum data rate λd_i on each terminal $i \in [n]$ can exist. Thus, if the optimal objective value λ^* is known, the algorithm yields a 1/2-approximation.

In case of unknown optimal objective values it is common to determine a sufficiently precise estimate of this value via binary search. In order to guarantee a quick completion of the binary search we will prove that our approximation algorithm for fixed λ achieves a rate of slightly more than 1/2 (i.e. $1/2+\epsilon$ for $\epsilon^{-1} \in \Theta(n \cdot \frac{d_{\max}}{d_{\min}})$). As we will see later this will allow us to stop the execution of the binary search after a linear number of iterations (with respect to the size of the given OFDMA instance) while already guaranteeing an approximation factor of at least 1/2.

The approximation algorithm will be based on the identification of a special substructure that contains at least one good approximate OFDMA schedule which then can be efficiently searched by dynamic programming.

Let us shortly recall the model of uniformly related channel costs. There, each channel $j \in [m]$ has the same cost $p_{ij} = p_j$ for all terminals $i \in [n]$, but different channels may have different costs.

We begin by showing that optimal schedules for Max-Rate OFDMA with uniformly related channel costs behave monotonically in the following way.

▶ **Lemma 4.1.** *Consider an optimal rate vector $r = (r_1, \ldots, r_m)$ for the Max-Rate OFDMA problem with m many uniformly related channels. Pick any two channels $i, j \in [m]$. If $r_i > r_j$, then $p_i \leq p_j$.*

Proof. Assume $r_i > r_j$ and $p_i > p_j$. By strict monotonicity of the rate-power function ψ we have $a \cdot \psi(r_i) > a \cdot \psi(r_j)$ for any $a \in \mathbb{R}_{>0}$ including $a = p_i - p_j$. Thus, swapping the rates r_i and r_j (together with the respective terminal assignment) strictly decreases the energy cost of the schedule by $\psi(r_i)(p_i - p_j) + \psi(r_j)(p_j - p_i)$ (without changing the data rates on the respective terminals). The decrease in energy cost allows us to increase the data rate on the minimum rate terminals by a non-zero fraction thereby contradicting optimality. ◀

Using this property we know that, when the channels are sorted non-decreasingly with respect to their cost coefficients, the rate allocation to channels in this order is monotonically non-increasing in any optimal solution. However, this does not allow any direct inference on the assignment of channels to terminals. For this purpose we will show that any channel assignment can be transformed into another one where the channels sustaining the largest data rates are assigned to the terminals with largest demands without losing more than a factor of 1/2 in the total data rate on each terminal. Combined with the property of Lemma 4.1 we may deduce that for any Max-Rate OFDMA instance approximate solutions exist which satisfy the following ordering constraint.

▶ **Definition 4.2** (Ordering Constraint). Given a Min-Power/Max-Rate OFDMA instance. Assume the channels and terminals to be ordered such that $p_1 \leq \cdots \leq p_m$ and $d_1 \geq \cdots \geq d_n$ holds. Let X_i denote the set of channels assigned to terminal $i \in [n]$. Then we say that an assignment $X = (X_1, \ldots, X_n)$ with $X_i \subseteq [m]$ for $i \in [n]$ satisfies the *ordering constraint* if for all terminals $i \in [n]$ and all channels $c \in X_i$, $c' \in \bigcup_{j=i+1}^{n} X_j$ we have $c < c'$.

A schedule with an assignment satisfying the ordering constraint is said to be *ordered*.

Exploiting this structural property, an algorithm just has to search the severely reduced solution space of ordered schedules for an optimal one to achieve a good approximation.

In order to prove that ordered schedules exist which are at least $(1/2 + \epsilon)$-approximate we use a lemma which on first sight might seem a bit technical. It basically states that, if the total sum R of rates of a given rate allocation $(r_j)_{j\in[m]}$ is large enough to cover the sum D of the given demands, then the terminal with largest demand (resp. $1/2 + \epsilon$ of its demand) can be covered by assigning only the channels with largest data rates to this terminal, while guaranteeing the subinstance of the remaining terminals and channels to stay feasible (for slightly reduced demands).

Of course, certain technical side-constraints have to be enforced. For covering the demand d_i of a terminal i we are only allowed to consider channels $c \in C_i$ with rates not larger than d_i. We define the corresponding channel set C_i by

$$C_i := \{c \in [m] \mid r_c \leq d_i\} .$$

For convenience of notation let

$$D_i := \sum_{t=i}^{n} d_t$$

denote the sum of the demands of the remaining terminals $\{i, \ldots, n\}$, and for each subset $C \subseteq [m]$ of channels

$$R(C) := \sum_{c\in C} r_c \qquad \text{for } C \subseteq [m]$$

represent the sum of corresponding data rates.

▶ **Lemma 4.3.** *Given any rate allocation $(r_j)_{j\in[m]}$ to the Max-Rate OFDMA scheduling problem with uniformly related channels $\{1, \ldots, m\}$ for $n \geq 2$ many terminals $\{1, \ldots, n\}$ with demands $d_1 \geq \cdots \geq d_n$. Let $r = (r_1, \ldots, r_m)$ satisfy $r_1 \geq \cdots \geq r_m$.*

If there is an $a \in \mathbb{R}_{\geq 0}$ such that $R(C_i) \geq D_i - a$ holds for all $i \in [n]$ then for any $b \in \mathbb{R}_{\geq 0}$ with $b \leq D_2 - a$ there exists an index $k \geq 1$ such that $R([k]) \geq (d_1 + b)/2$ is satisfied while $R(C'_i) \geq D_i - a - b$ holds for all $2 \leq i \leq n$ with $C'_i := C_i \setminus [k]$.

Proof. Let k be the smallest index such that $\sum_{j=1}^{k} r_j \geq (d_1 + b)/2$.

If $k \geq 2$ we know that $\sum_{j=1}^{k} r_j \leq d_1 + b$ holds as by minimality $\sum_{j=1}^{k-1} r_j < (d_1 + b)/2$ is satisfied which implies $r_k \leq r_{k-1} < (d_1 + b)/2$. For $k = 1$ and $1 \in C_1$ we have $r_1 \leq d_1$. So either $R(C'_i) \geq D_i - a$ for all $i \in [n]$ trivially holds (for $k \notin C_1$) or $R([k]) \leq d_1 + b$ is satisfied.

Now let $t \geq 1$ be the largest index with $C_t \cap [k] \neq \emptyset$. Then $C_1 \setminus C_t \subsetneq [k]$. Thus $C_t \setminus [k] = C_1 \setminus [k]$ and thereby $R(C'_t) \geq R(C_1) - R([k]) \geq D_1 - a - (d_1 + b) = D_2 - a - b$. For any $i > t$ we have $C'_i = C_i$ so $R(C'_i) \geq D_i - a - b$ holds for all $2 \leq i \leq n$. ◀

By using induction this leads to the following result.

▶ **Lemma 4.4.** *Let $(r_j)_{j\in[m]}$ be any rate allocation with data rates $r_1 \geq \cdots \geq r_m$ to the Max-Rate OFDMA scheduling problem for terminals $\{1,\ldots,n\}$ with demands $d_1 \geq \cdots \geq d_n$ on uniformly related channels $\{1,\ldots,m\}$.*

If $R(C_i) \geq \lambda D_i$ for all $i \in [n]$ then there is an ordered schedule with the same rate allocation sustaining a total data rate of $(\frac{d_i}{2} + \frac{d_n}{4n})\lambda$ on each terminal i.

Proof. W.l.o.g. we may assume $\lambda = 1$. The case $n = 1$ is trivial so assume $n \geq 2$. We inductively apply Lemma 4.3 with $b = d_n/(2n)$ to show that terminals $\{1,\ldots,t\}$ can be covered to $d_i/2+d_n/(4n)$ by rates $\{r_1,\ldots,r_k\}$ for some $k \in [m]$ such that $R(C_i\backslash[t]) \geq D_i - \frac{td_n}{2n}$ holds for all $i > t$. Finally for terminal n we have $R(C_n) \geq d_n - \frac{n-1}{2n}d_n = (\frac{1}{2} + \frac{1}{2n})d_n$. ◀

Note that Lemma 4.4 holds independently of any corresponding channel assignment $X = (x_{ij})_{ij\in[n]\times[m]}$ and especially does not impose the existence of any feasible schedule satisfying demands $d_1,\ldots,d_n$ with rates $r_1,\ldots,r_m$. We only consider the channel assignment of an optimal solution to verify that indeed $R(C_i) \geq \lambda D_i$ holds for the respective rates. By combining Lemmas 4.1 and 4.4 we immediately get.

▶ **Corollary 4.5.** *For any Max-Rate OFDMA instance with n terminals having maximal demand $d_{\max}$ and minimal demand $d_{\min}$ there exists an ordered schedule (according to Definition 4.2) which is at least $\frac{1}{2} + \epsilon$ approximate for $\epsilon = \frac{1}{4n} \cdot \frac{d_{\min}}{d_{\max}}$.*

We use the following algorithm to compute an optimal Min-Power OFDMA schedule satisfying the ordering constraint with data rates $((\frac{1}{2} + \epsilon)\lambda d_i)_{i\in[n]}$ for a given λ and the ϵ specified above. If the computed minimal required energy does not violate the energy bound, a feasible Min-Power/Max-Rate schedule for the chosen data rates is found, otherwise by Corollary 4.5 no feasible schedule for demands $(\lambda d_i)_{i\in[n]}$ can exist. The running time of the algorithm is dominated by $O(m^2n)$ many calls of a single-user OFDM optimization procedure (like, e.g., the water-filling method described in Section 3.1.2).

Algorithm OPT-ORDERED

Let $p_1 \leq \cdots \leq p_m$ and $d_1 \geq \cdots \geq d_n$. For given λ set $\bar{\lambda} = (\frac{1}{2}+\epsilon)\lambda$ with $\epsilon = \frac{1}{4n} \cdot \frac{d_{\min}}{d_{\max}}$.
Compute the matrix $F = (f(i,j))_{i\in[n],j\in[m+1]}$ according to

$$\begin{aligned} f(i,j) &:= \min_{1\leq k\leq m-j}\{f(i+1,j+k) + \text{optcost}^{(\bar{\lambda}d_i)}(\{j,\ldots,j+k-1\})\} \\ f(i,m+1) &:= \infty \\ f(n,j) &:= \text{optcost}^{(\bar{\lambda}d_n)}(\{j,\ldots,m\}) \end{aligned}$$

where $\text{optcost}^{(d)}\{C\}$ denotes the energy value of an optimal Min-Power schedule of the single-user subproblem using channel set C to satisfy demand d.
If $f(1,1) \leq E$ holds for the given Max-Rate OFDMA energy bound E, a feasible schedule for $\bar{\lambda}$ can be found by backtracking the corresponding assignments.

We now show how to determine a sufficiently precise estimate of the optimal objective value λ^* if it is not known. The following trivial lemma implies that the number of needed iterations of the binary search is relatively small.

▶ **Lemma 4.6.** *Given an algorithm that either asserts that for a given λ no feasible schedule exists or returns a feasible schedule which is at least $(\alpha+\epsilon)$-approximate. Then in order to compute an α-approximate schedule for unknown λ at most $O(\log(\Lambda \cdot \delta^{-1}))$ many binary search iterations are needed, where $\delta = \epsilon/(1+\epsilon)$ and $\Lambda = \lambda_{\text{sup}}/\lambda_{\text{inf}}$ denotes the ratio of an initial upper bound λ_{sup} and lower bound λ_{inf} on the maximal feasible λ.*

Proof. By scaling of the demands we may assume $\lambda_{\inf} \geq 1$. In every step the binary search maintains an upper bound $\hat{\lambda}$ for λ which is known to be infeasible and a lower bound $\check{\lambda}$ for which an $(\alpha+\epsilon)$-approximation could be derived. After $O(\log(\Lambda \cdot \delta^{-1}))$ many iterations we have $\check{\lambda} \geq \hat{\lambda} - \delta \geq \hat{\lambda}(1-\delta)$. So a fixed-rate $(\alpha+\epsilon)$-approximation for $\check{\lambda}$ returns an approximative schedule with approximation factor at least $(\alpha+\epsilon)\cdot(1-\delta)$. However by $\alpha \leq 1$ we have $(\alpha+\epsilon)\delta \leq (1+\epsilon)\delta = \epsilon$ for $\delta = \epsilon/(1+\epsilon)$ which proves the claimed approximation factor of α. ◀

If no more sophisticated bounds for $\lambda_{\inf}$ and $\lambda_{\sup}$ are known we can use the following simple initialization.

▶ **Lemma 4.7.** *The optimal objective value λ^* for a given Max-Rate OFDMA instance is contained in the interval $[\lambda_{\inf}, \lambda_{\sup}]$ for efficiently computable values $\lambda_{\inf}$ and $\lambda_{\sup}$ with $\lambda_{\sup} \leq mn \cdot \frac{p_{\max}}{p_{\min}} \cdot \lambda_{\inf}$.*

Proof. It is easy to give a solution where every terminal can sustain an absolute rate of r for any r with $p_{\max} \cdot \psi(r) \leq \frac{E}{n}$ at the same time (of course only if the instance is feasible, i.e. $n \leq m$). So let $\delta_{\min}$ denote the largest minimal total rate we can guarantee for every terminal on this way. Then $\psi(\delta_{\min}) = \frac{E}{n \cdot p_{\max}}$.

Now consider the maximal single data rate $r_{\max}$ that can be sustained on any channel in any feasible solution and suppose $r_{\max} \geq a \cdot \delta_{\min}$ for some $a \geq 1$. Then $a \leq n \cdot \frac{p_{\max}}{p_{\min}}$ holds as otherwise Lemma 2.11 implies that this channel would violate the energy bound by $\psi(r_{\max}) \geq a \cdot \psi(\delta_{\min}) > \frac{E}{p_{\min}}$.

Since the maximal total data rate $\delta_{\max}$ for any terminal can trivially be bounded by $\delta_{\max} \leq m \cdot r_{\max}$ we get $\delta_{\max} \leq m \cdot n \cdot \frac{p_{\max}}{p_{\min}} \cdot \delta_{\min}$. As the binary search has to be performed only on the interval $[\delta_{\min}/d_i, \delta_{\max}/d_i]$ for an arbitrarily chosen terminal $i \in [n]$ this proves the claim. ◀

▶ **Corollary 4.8.** *The number of iterative calls of algorithm OPT-ORDERED in order to compute a $1/2$-approximation for unknown λ is bounded linearly in the size of the given OFDMA instance.*

Proof. For $\epsilon = \frac{1}{4n} \cdot \frac{d_{\min}}{d_{\max}}$ we may use $\log(\delta^{-1}) \in O(\log(n) + \log(\frac{d_{\max}}{d_{\min}}))$ in Lemma 4.6 so together with $\log(\Lambda) \in O(\log(m) + \log(\frac{p_{\max}}{p_{\min}}))$ for $m \geq n$ the claim follows. ◀

4.1.1.1 Speed-Up for Uniform Demands

If the demands are uniform the binary search can also be performed independently of the above approximation algorithm by using the following simple decision procedure. For given λ just compute an optimal single-user OFDM schedule with channel capacities λd satisfying a data rate of $n\lambda d$. Clearly the existence of a feasible multi-user schedule for the given λ requires the existence of the corresponding single-user instance. On the other hand, if a single-user schedule exists we can apply Lemma 4.4 as $R(\{c \in [m] \mid r_c \leq \lambda d\}) = R([m]) \geq \lambda D_i$ for each $i \in [n]$ using the determined single-user rates. This guarantees the existence of a $(1/2+\epsilon)$-approximate ordered multi-user schedule which can be derived by distributing the single-user rates according to the proof of Lemma 4.4. Thus the total running time for the case of unknown λ when considering uniform demands reduces to $O(T \cdot (\log m + \log \frac{p_{\max}}{p_{\min}}))$ where T denotes the time to solve the corresponding single-user instances.

4.1.2 A $2/3$-Approximation-Algorithm for Uniform Demands

For the case of uniform demands we can improve the approximation factor of algorithm OPT-ORDERED by a small modification. Instead of searching for an optimal ordered schedule, initially each of the terminals gets assigned one of the n cheapest channels. Assume $p_1 \leq p_2 \leq \cdots \leq p_m$. Then we assign channel n to terminal 1, channel $n-1$ to terminal 2 and so forth. Independently of the actual data rates on these channels, by Lemma 4.1 this assignment ensures that the remaining data rate gap of the terminals in their initial order $1, \ldots, n$ is non-increasing in any optimal solution.

We use the following trivial property to ensure that we do not have to consider gaps that are larger than a certain threshold.

▶ **Lemma 4.9.** *Consider a partial schedule with data rates $r_1 \geq \cdots \geq r_m$ where the $k-1$ largest channels are already distributed and each terminal i sustains a total data rate of $d_i - g_i$ for $g_i \in \mathbb{R}_{\geq 0}$ on the already assigned channels. Let $R(\{k, \ldots, m\}) \geq \sum_{i=1}^{n} g_i$. Then greedily assigning the remaining channels to terminals with total rate not more than $d_i - r_k$ at the time of the assignment results in a schedule where each terminal has a data rate of at least $d_i - r_k$.*

Proof. Just use the premise of the lemma as an invariant when iteratively extending the channel assignment of the schedule. By assigning a channel j to a terminal i the set C of remaining channels and thereby $R(C)$ is reduced. As $R(C) \geq \sum_{i=1}^{n} g_i$ was satisfied before we now have $R(C \setminus \{j\}) \geq (\sum_{i=1}^{n} g_i) - r_j$. However, since i had a total rate of at most $d_i - r_k$ before assigning channel j to i the updated remaining gap g_i' (which is by $r_j \leq r_k$ still non-negative) also is reduced by rate r_j such that $R(C') \geq \sum_{i=1}^{n} g_i'$ for $C' = C \setminus \{j\}$ and the updated gaps g_i' is satisfied. This can be iteratively applied until $\max_i g_i' < r_k$ holds which guarantees a minimum data rate of $d_i - g_i' > d_i - r_k$ on each terminal i. ◀

▶ **Lemma 4.10.** *The initial channel assignment of each channel j for $1 \leq j \leq n$ to terminal $n+1-j$ can be extended by an ordered schedule (approximately) filling the remaining gaps by using the set of yet unassigned channels $C' = [m] \setminus [n]$ such that the final schedule is $(\frac{2}{3} + \frac{1}{18n+3})$-approximate.*

Proof. By scaling we may assume $\lambda d = 1$. Consider the sets $\tilde{C}_i := \{c \in C' \mid r_c \leq g_i\}$ where g_i denotes the remaining gap of terminal i. By monotonicity of the data rates (cf. Lemma 4.1) and the definition of the initial assignment we have $g_1 \geq \cdots \geq g_n$ for any optimal rate allocation on the channels $1, \ldots, n$. Also we may assume that $g_n > \frac{1}{3} - \epsilon$ for $\epsilon = \frac{1}{18n+3}$ as any terminal with smaller gap is already filled with a total data rate of at least $\frac{2}{3} + \epsilon$. Furthermore if $g_i \geq \frac{2}{3} + \epsilon$ for at least one terminal $i \in [n]$ we know that the data rates on channels $\{n+1, \ldots, m\}$ are at most $\frac{1}{3} - \epsilon$ which by Lemma 4.9 trivially allows covering the terminal demands by at least $\frac{2}{3} + \epsilon$. So w.l.o.g. we may assume $\frac{1}{3} - \epsilon < g_i < \frac{2}{3} + \epsilon$ for each terminal i.

If we can prove $R(\tilde{C}_i) \geq \sum_{t=i}^{n} g_t$ for all $i \in [n]$ we can apply Lemma 4.4 and show that the remaining gaps can be at least filled up by an additive rate of $\frac{g_i}{2} + \frac{g_n}{4n}$ which leads to a total rate of at least $1 - \frac{g_i}{2} + \frac{g_n}{4n}$ on each terminal i.

Trivially $R(C') \geq \sum_{i=1}^{n} g_i$. We consider an optimal solution where the total rate of channels assigned to each terminal equals exactly its demand. If in this optimal solution another one of the n cheapest channels is assigned to the same terminal as channel 1, i.e. there is a terminal i which gets assigned channels 1 and j for $1 < j \leq n$, then for our initial assignment we have $g_n \geq r_j \geq r_c$ for all $c \in C'$ and thereby $\tilde{C}_n = C'$. As the gaps are

non-increasing we trivially have $R(\tilde{C}_i) \geq \sum_{t=1}^{n} g_t$ for all $i \in [n]$ in this case. If however there is an optimal solution (with exact demand satisfaction) where channel 1 is assigned to terminal n which does not use any other channel from $[n]$ this implies that a subset $G_n \subseteq \tilde{C}_n$ exists with $R(G_n) = g_n$ which allows us to recursively apply the argument on terminals $\{1, \ldots, n-1\}$ with channels $C' \setminus G_n$. It follows by induction that $R(\tilde{C}_i) \geq \sum_{t=i}^{n} g_t$ for all $i \in [n]$.

What is left to show is that applying Lemma 4.4 to fill the remaining gaps leads to a $(\frac{2}{3} + \epsilon)$-approximate solution. The inequality $g_n > \frac{1}{3} - \epsilon$ with $\epsilon \leq \frac{1}{18n+3}$ yields $\frac{g_n}{2n} > 3\epsilon$. Thereby the total data rate on each terminal is at least $1 - \frac{g_i}{2} + \frac{g_n}{4n} > 1 - \frac{g_i}{2} + \frac{3}{2}\epsilon$. Together with $\frac{g_i}{2} < \frac{1}{3} + \frac{1}{2}\epsilon$ the claim follows. ◀

Testing all feasible ordered schedules extending the initial assignment of the n cheapest channels we get the following algorithm.

Algorithm CHEAPEST-FIRST

Let $p_1 \leq \cdots \leq p_m$. Set $\bar{\lambda} = (\frac{2}{3} + \epsilon)\lambda$ for $\epsilon = \frac{1}{18n+3}$. Compute the matrix $F = (f(i,j))_{i \in [n], j \in \{n+1, \ldots, m+1\}}$ according to

$$f(i,j) := \min_{0 \leq k \leq m-j} \{f(i{+}1, j{+}k) + \text{optcost}^{(\bar{\lambda} d)}(\{n{+}1{-}i\} \cup \{j, \ldots, j{+}k{-}1\})\}$$

$$f(i, m{+}1) := \infty$$

$$f(n,j) := \text{optcost}^{(\bar{\lambda} d)}(\{1\} \cup \{j, \ldots, m\})$$

Check if $f(1, n+1) \leq E$. Then a feasible schedule for $\bar{\lambda} = (\frac{2}{3} + \epsilon)\lambda$ can be constructed. Otherwise no feasible schedule for λ can exist.

For unknown λ we can again use the decision procedure discussed in Section 4.1.1.1 to restrict the estimate on λ to the interval $[\lambda^*/2, \lambda^*]$ in time $O(T \cdot (\log m + \log \frac{p_{\max}}{p_{\min}}))$. From that point on only $O(\log n)$ calls of algorithm CHEAPEST-FIRST are needed to determine an appropriate 2/3-approximation.

4.1.3 Polynomial Time Approximation Schemes

In this section we will develop several polynomial time approximation schemes for the Max-Rate OFDMA problem with uniformly related channel costs. Our approach builds upon the ideas of Hochbaum and Shmoys [HS87] and Azar and Epstein [AE98] originally developed for makespan and Santa Claus scheduling problems.

Following the close relation between OFDMA and Santa Claus scheduling, using existing techniques for the latter problem seems to be a natural approach to cope with OFDMA scheduling instances. Once data rates have been fixed for a given OFDMA instance the assignment of channels to terminals resembles the scheduling of jobs on machines. In its basic version the objective is the same for both problems: to maximize the minimum rate/load on each terminal/machine.

In general this is only a loose analogy and can not directly be exploited because the needed energy to sustain a fixed rate on a given channel depends on the specific terminal it is assigned to, while furthermore the terminals compete with each other in order to not violate the given global energy bound. So even for fixed rates the channel assignment problem in OFDMA scheduling is more involved than in the corresponding Santa Claus variant.

For uniformly related channels, however, the global energy requirements of a given data rate allocation are independent of the actual channel assignment. Thus, once feasibility of data rates with respect to the energy bound has been verified the channel assignment problem indeed reduces to the problem of minimum load maximization without any additional side constraints.

We exploited this fact in Sections 4.1.1 and 4.1.2 to prove the existence of specially structured channel assignments considering the rate allocation vector of an optimal assignment. In the current section we will first use this property to derive a PTAS for the Max-Rate OFDMA problem with uniformly related channel costs and uniform demands.

Existing polynomial time approximation schemes [Woe97, AE98, ES04] for the Santa Claus scheduling problem rely on the simple fact that for a constant number of possible different job sizes an optimal schedule can be computed in polynomial time.

It is easy to see that this property also holds for OFDMA with uniformly related channel costs as for a constant number of different data rates all possible rate configurations could be generated and tested in polynomial time —provided that an appropriate compact representation is used. (This should, however, not be confused with a limitation of the cost coefficients to a constant number of different values which could still result in a polynomial number of different rates that have to be considered for each channel.) For each rate allocation the minimal energy needed to support those rates and thereby feasibility with respect to the energy bound can be determined by single-user OFDM optimization whereas feasibility with respect to the demand constraints can be tested by solving the corresponding Santa Claus instance.

▶ **Observation 4.11.** *If for each channel the set of supported data rates is a subset of a common rate set R of constant cardinality then Min-Power/Max-Rate OFDMA for uniformly related channel costs is polynomial time solvable.*

A general technique to exploit this property in order to design a PTAS for a non-constant number of possible rates is to define a coarse grid where the rates may be rounded to without losing more than a small multiplicative factor. Problematic in this context are rates that are smaller than the step-size of the grid as they might suffer from an unbounded loss when rounded.

Fortunately the treatment of small data rates can be (at least partially) decoupled from the assignment procedure that distributes the channels with large data rates. As observed by Lemma 4.9 distributing the small rates can be used to equalize a previously computed partial schedule. However, when using this two-phase approach and trying to combine Observation 4.11 and Lemma 4.9, the partial schedule computed on the large data rates (rounded to a constant number of different values) has to anticipate this kind of behavior.

In [OV09] we developed an appropriate dynamic programming formulation for this approach in order to derive a PTAS for Max-Rate OFDMA with uniformly related channels and uniform demands. The algorithm will be described in Section 4.1.3.1 and a derived alternative formulation to solve Santa Claus scheduling on identical machines (compared to existing ones in [Woe97, AE98, ES04]) is briefly stated in Section 4.1.3.2. As the formulation heavily relies on the identical representation of all terminals (requiring uniform demands) it can not be modified to be applicable in case of non-uniform demands.

However, a surprisingly simple method to reduce the Max-Rate OFDMA problem to Santa Claus scheduling in a more modular way can be given (which was unfortunately overlooked in [OV09]). Though this method has a larger running time than the algorithm derived in Section 4.1.3.1 it can be very easily described and allows for adapting the techniques of Azar

and Epstein [AE98] to the case of Max-Rate OFDMA (on uniformly related channels) with non-uniform demands. The respective algorithms for the case of uniform and non-uniform are described in Sections 4.1.3.3 and 4.1.3.4.

4.1.3.1 PTAS for Uniform Demands

Let a Max-Rate OFDMA instance with known optimal objective value λ^{opt} be given. Then for any fixed constant $\epsilon > 0$ we will be able to derive a schedule with objective value $(1-\epsilon)\lambda^{\text{opt}}$. The case of unknown λ^{opt} can be treated similarly to Section 4.1.1 by executing a binary search on λ. Also we can quickly narrow down the interval of this binary search in a preprocessing phase by using the 1/2-approximate value of the previously described approximation algorithm for initializing the corresponding bounds.

When speaking of demands in the following sections, we will mean the already scaled proportional demands $d'_i := \lambda d_i$. Now let $d' = d'_1 = \cdots = d'_n$ denote the uniform demand of the given OFDMA instance.

Adapting the ideas of Hochbaum and Shmoys [HS87] for makespan scheduling we will distinguish two types of channels: channels $j \in [m]$ that are able to sustain a relatively large data rate in an optimal solution, namely $r_j \geq \epsilon d'$, called *large channels*, and the remaining channels (supporting only small rates) called *small channels*.

If channels are known to have a large data rate $r_j \geq \epsilon d'$ in a fixed optimal solution those rates could be rounded to integral multiples of $\epsilon^2 d'$ without losing more than a small multiplicative factor. The total loss of data rate due to the rounding experienced by any terminal is at most $\epsilon d'$ as the number of large channels assigned to each terminal in any optimal solution guaranteeing a rate of exactly d' is bounded by at most $1/\epsilon$. Since normally we do not know the data rate allocation of an optimal schedule the data rates can not directly be rounded. However the number of different possible rounded data rates is bounded by $1/\epsilon^2$ and thereby constant. Thus for each large channel all possibilities may be tried.

Small channels can not be rounded without losing a significant portion of their data rate. However, given a partial optimal schedule where the large channels are already optimally assigned we know by Lemma 4.9 that the small channels can be greedily distributed such that for each terminal a data rate of at least $(1-\epsilon)d'$ can be guaranteed. As already noted, this result holds for any small rate allocation which satisfies the condition that the total volume of small data rates is enough to cover the total gap left by the assignment of large channels to reach a rate of d' on each terminal, so we do not have to use the exact rates from the originally considered optimal schedule.

Combining these observations we may conclude that trying all possible rounded data rates on large channels eventually leads to a partial schedule that can be filled up by small data rates to guarantee a data rate of at least $(1-2\epsilon)d'$ on each terminal.

An obstacle that still has to be overcome is that the classification of small and large channels is not known a priori as it depends on an optimal rate allocation. Using Lemma 4.4, however, we know that in any optimal solution the data rate of a channel decreases with increasing cost of the channel. Thus when the channels are ordered non-decreasingly with respect to their cost coefficients this implies the existence of a split index $\bar{m}$ such that for a fixed optimal solution all channels $\{1, \ldots, \bar{m}\}$ of relatively low cost have large data rates and all more costly channels $\{\bar{m}+1, \ldots, m\}$ only sustain small data rates. By exhaustive search the index can be efficiently determined, so we may assume the split index $\bar{m}$ to be known for stating our approximation algorithm.

For a fixed ϵ an integer $1/\bar{\epsilon}$ can be chosen such that $\epsilon/4 \leq \bar{\epsilon} \leq \epsilon/2$. The following definition will allow us to compactly represent the total rate assigned to each terminal in a

partial schedule where only rates rounded to integral multiples of $\bar{\epsilon}$ have been distributed by classifying the terminals into $z = (1-\bar{\epsilon})/\bar{\epsilon}^2$ many categories.

▶ **Definition 4.12** (Terminal-Rate Configuration)**.** For a Max-Rate OFDMA instance we call a vector $(n_0, \ldots, n_z)$ with $z = (1-\bar{\epsilon})/\bar{\epsilon}^2$ representing a partial schedule with n_ℓ many terminals having a total rate of $R_\ell := \ell \cdot \bar{\epsilon}^2 d'$ a *terminal-rate configuration.*

Given a terminal-rate configuration $(n_0, \ldots, n_z)$ and a channel index $j \in [m]$ which denotes the smallest index of a channel not already assigned in the partial schedule represented by $(n_0, \ldots, n_z)$ we can immediately determine for each $t \in \{0, \ldots, z\}$ the energy $p_j\psi(t\bar{\epsilon}^2 d')$ needed to sustain a rate of $t \cdot \bar{\epsilon}^2 d'$ on channel j. Computing an appropriate allocation of small data rates on channels $\{\bar{m}+1, \ldots, m\}$ to fill the remaining gaps via single-user optimization can also be efficiently done. Thus, we can use the following dynamic programming formulation to try all possible rounded data rate allocations in polynomial time.

Dynamic Programming Formulation: Our dynamic program searches for the minimum amount $e(n_0, \ldots, n_z, j)$ of energy that is needed to schedule the as yet unassigned large channels $\{j, \ldots, \bar{m}\}$ on the partial schedule implied by $(n_0, \ldots, n_z)$ while ensuring that the total rate on the small channels is big enough to cover the total gap $V_g = n(1-\bar{\epsilon})d' - \sum_{\ell=1}^{z} n_\ell \cdot \ell\bar{\epsilon}^2 d'$ left to reach a minimum rate of $(1-\bar{\epsilon})d'$ on every terminal. Therefore $e(n_0, \ldots, n_z, j)$ is computed by

$$\min_{\ell \in \{0,\ldots,z\}, 1/\bar{\epsilon} \le t \le z-\ell} \{p_j\psi(t\bar{\epsilon}^2 d') + e(n_0, \ldots, n_\ell - 1, \ldots, n_{\ell+t}+1, \ldots, n_z, j+1) \mid n_\ell > 0\}$$

where the starting elements $e(n_0, \ldots, n_z, \bar{m}+1)$ indicate the energy needed to fill the total remaining gap with small data rates on the costly channels and are defined as the optimal solution (or infinity if the program is infeasible) to the corresponding single-user OFDM instance with demand V_g and capacity limit $\bar{\epsilon}d'$ on each channel.

After execution of the dynamic program it is checked if the final entry $e(n, 0, \ldots, 0, 1)$ satisfies the energy bound E. In that case a feasible schedule guaranteeing a minimum data rate of $(1-2\bar{\epsilon})d'$ can be efficiently constructed. The required backtracking information has to be stored independently during the execution of the dynamic program.

▶ **Theorem 4.13.** *The above approximation scheme computes a feasible* $(1-\epsilon)$*-approximation of the optimal Max-Rate OFDMA solution.*

Proof. Remember that we have chosen $\bar{\epsilon}$ such that $\bar{\epsilon} \le \epsilon/2$. As argued before, we can efficiently find $\bar{m}$ and a sufficiently precise estimate for $d^{\text{opt}} = \lambda^{\text{opt}} d$ and thus may assume them to be given. Note that if the dynamic program finds a schedule, it always has a minimum data rate of $(1-2\bar{\epsilon})d^{\text{opt}} \ge (1-\epsilon)d^{\text{opt}}$. Also, after the execution of the dynamic program only feasible solutions are accepted. So we only have to show that the dynamic program always finds a solution if it is possible to reach a minimum data rate of d^{opt}.

Consider any feasible schedule S with minimum data rate d^{opt} and split index $\bar{m}$. By rounding the large rates of S and keeping the exact small rates we know that there is a schedule with large data rates being multiples of $\bar{\epsilon}^2 d^*$ and small data rates with total size at least as big as the total remaining gap in order to reach rates of at least $(1-\bar{\epsilon})d^{\text{opt}}$ on each terminal. Furthermore Lemma 4.9 implies that independent of the actual small data rates on the costly channels we can always schedule them to reach a minimum data rate of $(1-2\bar{\epsilon})d^{\text{opt}}$, so we can use the possibly different rates computed by the corresponding single-user solution. Note that the computed rates do not need more energy than is used

by the small rates on costly channels in schedule S, since S satisfies all the constraints of the single-user instance. Only the data rates on large channels could violate the energy constraint, but because we try all possible rounded rates, eventually we find a solution in which the energy bound holds. ◀

If we consider d^{opt} and $\bar{m}$ to be given, the running time of the above algorithm can be bounded by $O(n^{(\bar{\epsilon}^{-2})} \cdot (T + \bar{\epsilon}^{-4} \cdot m))$. The exhaustive search for the split index causes an extra factor of m and the number of iterations of the binary search for d^{opt} starting with the 1/2-approximate solution computed in Section 4.1.1 is logarithmic in ϵ^{-1}.

4.1.3.2 Derived PTAS for Santa Claus Scheduling

Note that the dynamic programming approach used above directly translates to an alternative PTAS for the Santa Claus scheduling problem on identical machines (different from [Woe97, AE98, ES04]).

Assume that the optimal minimum processing time T is known and the jobs are ordered non-increasingly with respect to their processing times. W.l.o.g. let the index set of large jobs be $\{1, \ldots, \bar{m}\}$ (with rounded sizes $\tilde{r}_1, \ldots \tilde{r}_{\bar{m}}$) and the index set of small jobs be $\{\bar{m}+1, \ldots, m\}$. There are at most $z := (1-\bar{\epsilon})/\bar{\epsilon}^2$ many possible non-zero loads for a machine considering only rounded large jobs. The important information that is kept track of in our dynamic program is, whether a schedule of n_ℓ many machines with load $\ell \cdot \bar{\epsilon}^2 T$ for $0 \leq \ell \leq z$ can be extended by scheduling the remaining large jobs $\{j, \ldots, \bar{m}\}$ to have minimum processing time at least $(1-2\bar{\epsilon})T$ after filling it up with small jobs. More precisely, the dynamic program computes the entries of a boolean matrix B with index set $\{0, \ldots, n\}^{z+1} \times [\bar{m}+1]$ sequentially by the recursive rule

$$b(n_0, \ldots, n_z, j) := \max_{\ell \in \{0,\ldots,z\}} \{b(n_0, \ldots, n_\ell - 1, \ldots, n_{\ell+\bar{r}_j} + 1, \ldots, n_z, j+1) \mid n_\ell > 0\}$$

where $\bar{r}_j := \min\{\tilde{r}_j, z - \ell\}$. The actual schedule can be reconstructed by recording the corresponding assignments in a second matrix. If $b(n, 0, \ldots, 0, 1) = 1$, then a feasible schedule satisfying a minimum load of $(1-\epsilon)T$ is found.

Let $V_s := \sum_{j=\bar{m}+1}^{m} r_j$ denote the total processing time of all small jobs, $V_k := \sum_{\ell=1}^{z} n_\ell \ell \bar{\epsilon}^2 T$ denote the total load of all machines according to $n_0, \ldots, n_z$, and let $V_g := n(1-\bar{\epsilon})T - V_k$ be the corresponding total gap that needs to be filled to reach $(1-\bar{\epsilon})T$ on each machine. Then the starting entries $b(n_0, \ldots, n_z, \bar{m}+1)$ for the dynamic program are defined by

$$b(n_0, \ldots, n_z, \bar{m}+1) := \begin{cases} 1 & \text{if } V_s \geq V_g, \\ 0 & \text{otherwise,} \end{cases}$$

for all $n_\ell \in \{0, \ldots, n\}$, $0 \leq \ell \leq n$.

The running time of the algorithm for known d is bounded by $O(n^{(\bar{\epsilon}^{-2})} \cdot \bar{\epsilon}^{-2} \cdot m)$.

4.1.3.3 Modular Reduction to Santa Claus Scheduling

Our goal in this section is to include the treatment of small data rates directly into the allocation of large data rates thereby allowing the usage of known approximation algorithms for the Santa Claus scheduling problem as a black box in order to approximate Max-Rate OFDMA.

When generating a collection of Santa Claus instances to test all possible (rounded) data rates for each channel, only a constant number of different rate steps can be considered

without introducing a super-polynomial blowup. If a channel, however, is assumed to have a rate smaller than the chosen step-size ϵ^* the exact value of the rate is crucial for the Santa Claus scheduling algorithm. Yet, the exact value depends on the a priori unknown total data rates of each terminal after finishing the assignment of large channels. As there is an infinite number of possible small rates they can not all be tested. The idea used in our PTAS in Section 4.1.3.1 is that the number of possible total rates defined by the assignment of rounded large data rates on each terminal is constant. This leads to a polynomial number of possible total gaps (summed over all terminals) that have to be filled by small rates. For each of those gap-configurations an exact allocation of small rates of total volume V that covers these gaps can be efficiently computed. However, the polynomial number of computed small rates is still too large to add the corresponding values to the set of possible data rates.

Though overcoming this obstacle can be done in a surprisingly simple way, this approach can be easily overlooked. When designing the first PTAS for the Santa Claus scheduling problem on identical machines Woeginger [Woe97] stated: "Rounding large jobs to a constant number of distinct job sizes of course simplifies the problem, but there seems to be no way of integrating the small jobs into a dynamic program." The idea to circumvent this problem is to abstract from specific small rate allocations and only consider the required total volume. As already exploited in the PTAS in Section 4.1.3.1, Lemma 4.9 does not impose any assumption about the structure of the set of small channels other than that their individual sizes are bounded by a given threshold $\epsilon^* := \epsilon\lambda d$. This even allows us to replace them by a set of $q := \lfloor V/\epsilon^* \rfloor$ many channels of size ϵ^* and one channel of size $V - \epsilon^* q$ without affecting the approximation guarantee. We may further omit this last channel and lose at most an additive term of ϵ^* in the total data rate of at most one terminal.

Thus, for each possible gap-volume we can encode the relevant information about the small data rate allocation in terms of a number of channels sustaining a data rate of exactly ϵ^* while losing only an insignificant portion of the total data rate volume, which allows finding an $(1-\epsilon)$-approximate Max-Rate OFDMA schedule for given λ by solving a polynomial number of Santa Claus instances.

This simple reduction is based upon an idea of Azar and Epstein [AE98] originally introduced for the Santa Claus scheduling problem. We adapt their method to give an alternative PTAS for Max-Rate OFDMA (on uniformly related channels) with uniform demands in the current section and a PTAS for the case of non-uniform demands in Section 4.1.3.4.

We choose $\bar{\epsilon} \leq \epsilon/3$ such that $1/\bar{\epsilon}$ is integral.

▶ **Definition 4.14** (Channel-Rate Configuration)**.** A *channel-rate configuration* is a vector $(m_0, m_1, \ldots, m_z)$ for $z = (1-\bar{\epsilon})/\bar{\epsilon}^2$ representing a rounded rate allocation where m_ℓ denotes the number of channels having a data rate of $(\bar{\epsilon} + \ell\bar{\epsilon}^2)d'$ for $d' = \lambda d$.

In a channel-rate configuration the set of allowed values for each m_ℓ is $\{0, \ldots, m\}$. Furthermore, $z+1$ can be trivially bounded from above by $1/\bar{\epsilon}^2$. Thus, there are at most $(m+1)^{\bar{\epsilon}^{-2}}$ many different channel-rate configuration vectors.

Instead of computing the exact small data rates to cover the total gap of a given *terminal*-rate configuration representing a partial schedule of large channels, we just enumerate all possibilities of rounded total volumes $\tilde{V}$ of small rates with $\tilde{V} = q \cdot \bar{\epsilon}d'$, by adding q many large channels of rate exactly $\bar{\epsilon}d'$ to the *channel*-rate configuration.

Each configuration vector can be tested if it supports a Santa Claus schedule of minimum load $T = (1-2\bar{\epsilon})d'$ (already accounting for the possible error introduced by rounding the individual large rates and the volume of small rates). For checking this property we can

either use existing techniques [Woe97, AE98, ES04] for the Santa Claus scheduling problem or apply the dynamic program defined in Section 4.1.3.2 (having only to consider the special case without small jobs).

In order to enumerate all relevant channel-rate configurations and their respective minimum energy cost (with respect to the corresponding OFDMA schedule), we introduce an additional parameter j in the configuration vector, that is $(j, m_0, \ldots, m_z)$, which denotes the largest index of channels whose rates have already been fixed/rounded and construct a graph using these vectors as node identifiers. We assume the set of jobs to be ordered such that $p_1 \geq \cdots \geq p_m$ holds, thereby guaranteeing $r_1 \leq \cdots \leq r_m$ (note that this is inverse to the order considered in Section 4.1.3.1).

For any vector $(j-1, m_0, \ldots, m_z)$ with $j \in [m]$ and any index $\ell \in \{0, \ldots, z\}$ there is an edge to vector $(j, m_0, \ldots, m'_\ell, \ldots, m_z)$ with $m'_\ell = m_\ell + 1$ labeled with the energy cost needed to sustain a rate of $(\bar{\epsilon} + \ell\bar{\epsilon}^2)d'$ on channel j. Additionally, from the initial starting node $(0, 0, \ldots, 0)$ there are edges to any vector $(j, m_0, 0, \ldots, 0)$ with $j \in [m]$ and $m_0 \in [m]$. The cost of those edges equals the required energy value of an optimal single-user solution for a demand of $m_0 \cdot \epsilon d'$ on channels $\{1, \ldots, j\}$ with capacity limits $\epsilon d'$ and is set to infinity if no feasible schedule exists.

Note that in contrast to [AE98] we round down and not up, so that we can guarantee that the energy levels are indeed sustainable for the chosen rates. By our rounding rule we can ensure that, if there is a path in the graph leading from the initial node to a node allowing a feasible Santa Claus schedule, the energy needed to sustain these data rates does not exceed the cost of that path.

If both the feasibility of the needed energy as well as the computation of an appropriate Santa Claus schedule for a minimum load of T can be guaranteed then a feasible schedule with a minimum data rate of at least $T - \bar{\epsilon}d' = (1 - 3\bar{\epsilon})d' \geq (1 - \epsilon)\lambda d$ can be constructed by adopting the assignment of large channels/jobs of the considered Santa Claus schedule with fixed data rates according to the corresponding path in the previously defined graph and subsequently distributing the small channels by the greedy allocation rule according to Lemma 4.9. A more formal proof will be given for the following generalization to non-uniform demands considered in the next section.

4.1.3.4 PTAS for Non-Uniform Demands

When considering non-uniform demands, an a priori rounding to channel-rate configurations of constant size is not possible anymore as for each terminal i the notion of small and large data rates and thereby also the rounding steps (depending on d'_i) differ. (Though this also holds for Santa Claus scheduling on uniformly related machines, in this context the fixed job sizes allow for an initial rounding compatible to the different machine speeds, see [ES04].) As already mentioned, trying all possibilities for rounded rates would result in a super-polynomial running time. Nevertheless, following the approach from [AE98] we can derive a PTAS for Max-Rate OFDMA on uniformly related channels with non-uniform demands.

Let λ and thereby $d'_i = \lambda d_i$ for each terminal i be given. We assume $p_1 \geq \cdots \geq p_m$ and $d'_1 \leq \cdots \leq d'_n$. Channel sizes will be rounded depending on the demand of the corresponding terminal. More precisely, we map each terminal to a bin range B_k which contains all terminals i with demand d'_i such that $d'_1 \cdot 2^k \leq d'_i < d'_1 \cdot 2^{k+1}$ holds. Clearly, the number of considered bin ranges is bounded by $D = 1 + \log_2(d'_n/d'_1)$. (Note that we can not omit empty bin ranges like in [AE98] as the corresponding scaling phases may be needed for fixing small channel rates.)

For a fixed ϵ an integer $1/\bar{\epsilon}$ can be chosen such that $\epsilon/8 \leq \bar{\epsilon} \leq \epsilon/4$. Depending on $\bar{\epsilon}$ the values $\epsilon_k = 2^k \cdot \bar{\epsilon} d'_1$ and $\delta_k = 2^k \cdot \bar{\epsilon}^2 d'_1$ for $k \in \{0, \ldots, D\}$ are defined. We set the maximal number z of possible different sizes for large channels in each bin range to be $z = (2 - \bar{\epsilon})/\bar{\epsilon}^2$ as we have to ensure both $\epsilon_k \leq \bar{\epsilon} d'_i$ as well as $\epsilon_k + z \cdot \delta_k \geq d'_i$ for all d'_i in the interval $[2^k d'_1, 2^{k+1} d'_1]$.

For each bin range B_k the vector $(i, j, m_0, m_1, \ldots, m_z)$ will correspond to a configuration where i denotes the largest index of terminals already covered, j denotes the index of channels $\{1, \ldots, j\}$ considered so far, m_ℓ for $\ell \in \{1, \ldots, z\}$ represents the number of (remaining) channels rounded down to $s_{k,\ell} := \epsilon_k + \ell \cdot \delta_k$ and m_0 represents the rounded volume $\tilde{V} = m_0 \cdot \epsilon_k$ of (remaining) small channels rounded to an integral multiple of ϵ_k and normalized by ϵ_k. Note that all large channels with rates within the first interval $[\epsilon_k, \epsilon_k + \delta_k]$ will also be considered as part of the set of small channels.

In order to allow for scaling and rounding of the channel-rate configuration to the distinct bin ranges we extend the configuration vector by a number specifying the active layer L corresponding to bin range B_L. From one layer L to the next one $(L + 1)$ the size of the bin range increases from $2^L d'_1$ to $2^{L+1} d'_1$.

Definition of the Rounded-Rates Graph: We construct a graph with D many layers where in any layer L the nodes $(L, i, j, m_0, \ldots, m_z)$ corresponding to the channel-rate configurations of the already rounded channels $\{1, \ldots, j\}$ with respect to the bin range B_L are created for every terminal $i \in \{0, \ldots, n\}$, every channel $j \in \{0, \ldots, m\}$ and every $m_\ell \in \{0, \ldots, m\}$ with $\ell \in \{0, \ldots, z\}$.

Note that, as z is constant, the graph has size polynomial in n, m, and $\log_2(d'_n/d'_1)$. We will define edges with appropriate edge costs such that the cost of a path from $(0, 0, 0, 0, \ldots, 0)$ to a node $(L, i, j, m_0, \ldots, m_z)$ corresponds to the energy cost for a feasible OFDMA schedule covering the demands of all terminals from $\{1, \ldots, i\}$ (contained in bin ranges $B_0, \ldots, B_L$) on channels $\{1, \ldots, j\}$ while the as yet unused channels define the channel-rate configuration $(m_0, \ldots, m_z)$.

There will be an edge from each node $(L, i - 1, j, m_0, \ldots, m_z)$ to $(L, i, j, m'_0, \ldots, m'_z)$ if terminal i is contained in bin range B_L and the channel-rate configuration $(a_0, \ldots, a_z)$ with $a_j = m_j - m'_j$ for $j \in \{0, \ldots, m\}$ allows a feasible covering of terminal i up to a total rate of $(1 - 3\bar{\epsilon})d'_i$, where this total rate is chosen according to Lemma 4.17 in order to guarantee its feasibility whenever an optimal schedule for demands d'_i exists. We call those edges *covering edges* and they will be of zero cost. If we only add covering edges where the channel-rate configuration $(a_0, \ldots, a_z)$ with $a_j = m_j - m'_j$ is minimal, that is, no configuration $(a'_0, \ldots, a'_z)$ with $a'_j \leq a_j$ for all $j \in \{0, \ldots, m\}$ exists, the number of total outgoing edges from each node is significantly reduced (from polynomial to constant, see Observation 4.15).

Analogously to Section 4.1.3.3 we define an edge from any node $(L, i, j - 1, m_0, \ldots, m_z)$ to $(L, i, j, m_0, \ldots, m'_\ell, \ldots, m_z)$ with $m'_\ell = m_\ell + 1$ of cost $p_j \cdot \psi(\epsilon_L + \ell \cdot \delta_L)$ for any $\ell \in \{0, \ldots, z\}$ and an edge from the initial node $(0, 0, 0, 0, \ldots, 0)$ to any $(0, 0, j, m_0, 0, \ldots, 0)$ of cost equal to an optimal single-user schedule on channels $\{1, \ldots, j\}$ with capacity limits ϵ_0 and demand $m_0 \cdot \epsilon_0$.

Additionally, edges of zero cost for changing the layer from nodes $(L, i_L, j, m_0, \ldots, m_z)$ to $(L + 1, i_L, j, m'_0, \ldots, m'_z)$ will be introduced where i_L denotes the largest index of a terminal contained in bin range B_L. Entering layer $L + 1$ the large rates have to be rounded down to integral multiples of δ_{L+1} or to be considered as small rates. We translate $(L, i_L, j, m_0, \ldots, m_z)$ to $(L + 1, i_L, j, m'_0, \ldots, m'_z)$ in the following way. In layer L the variable m_ℓ denotes the number of channels with (rounded) rate $\epsilon_L + \ell\delta_L$. So all channels previously mapped to m_ℓ

with $\ell \leq 1/\bar{\epsilon}$ become small. For $\ell > 1/\bar{\epsilon}$ the value of m_ℓ is added to m'_c with $c = \lfloor \frac{\ell - 1/\bar{\epsilon}}{2} \rfloor$. In other terms, m'_c is set to the sum of the values m_ℓ with $\lfloor \frac{\ell-1/\bar{\epsilon}}{2} \rfloor = c$. (So this affects only the two entries $m_{2c+1/\bar{\epsilon}}$ and $m_{2c+1+1/\bar{\epsilon}}$.)

For complexity considerations we make the following observations bounding the size of the rounded-rates graph.

▶ **Observation 4.15.** *For each graph node the number of outgoing covering edges when restricted to minimal covers can be bounded by $\Delta = (z+2)^{2/\bar{\epsilon}}$ which is constant for constant $\bar{\epsilon}$.*

This observation follows directly from the fact that any terminal with $d'_i \leq 2^{k+1} d'_1$ can be covered by at most $2/\bar{\epsilon}$ many channels of rate at least $\epsilon_k = \bar{\epsilon} 2^k d'_1$ (using collections of small channels in chunks of size ϵ_k). Each of those channels can be fixed to a rate of zero or one of the $z+1$ many possible non-zero rate steps. Note that Δ is also a (very ample) bound on the number of total outgoing non-covering edges for each node.

The complexity for the construction of covering edges dominates the complexity of constructing edges of other types and can be bounded by $O(1/\bar{\epsilon})$ for each edge (when using the compact generation procedure described above).

▶ **Observation 4.16.** *The number of total nodes in the rounded-rates graph is at most $N = n \cdot m^{z+2} \cdot D$ and the total number of edges can be bounded by $M = 2\Delta N$.*

Thus, we may conclude that the construction of the graph and a subsequent shortest path search can be done in time $O(\Delta N/\bar{\epsilon}) \subseteq O\big(\big(\frac{m}{\epsilon}\big)^{O(1/\epsilon^2)} \cdot \log \frac{d_n}{d_1}\big)$. We will now show that searching a path in the graph can be used to efficiently compute a $(1-\epsilon)$-approximate OFDMA schedule.

▶ **Lemma 4.17.** *If there is an optimal OFDMA schedule assigning to each terminal i a channel set C_i satisfying its total demand d'_i using total energy at most E, then there exists a path of cost at most E in the rounded-rates graph that allows a covering of each terminal i with channels C_i up to a demand of at least $(1-3\bar{\epsilon})d'_i$ using the corresponding rounded rates.*

Proof. Consider the set C_i of channels assigned to terminal i in a given optimal solution with optimal data rates r_j for $j \in [m]$. We partition C_i into the two sets $C_i^{>}$ and $C_i^{\ll}$ denoting the subsets of large (resp. small) channels with respect to bin range B_k containing terminal i.

First note that for any bin range B_k there is a path in the graph such that the accumulated multiplicative rounding error of any large channel j with respect to B_k can be bounded by at most $\bar{\epsilon}$. In order to see this, let ℓ be chosen such that $\epsilon_k + \ell\delta_k \leq r_j < \epsilon_k + (\ell+1)\delta_k$ holds. If for channel j fixing the rounded rate $\tilde{r}_j$ can be delayed until layer k is reached, then by setting $\tilde{r}_j := \epsilon_k + \ell \cdot \delta_k$ a multiplicative error of not more than $\delta_k/\epsilon_k = \bar{\epsilon}$ is introduced. If j has to be fixed in a smaller layer h, that is, there is a channel $j' > j$ which is used to cover a terminal i' contained in bin range B_h for some $h < k$ in the optimal solution, this bound does still hold. Applying Lemma 4.1 to this case implies $r_j \leq r_{j'} \leq d'_{i'} < 2^{h+1} d'_1 = \epsilon_h + z\delta_h$ so there is an appropriate $\ell' \leq z$ with $\epsilon_h + \ell'\delta_h = \epsilon_k + \ell\delta_k$ for the previously defined ℓ. Fixing this rate ensures that $\tilde{r}_j$ stays the same in all layers $B_h, \ldots, B_k$ and the total multiplicative rounding error comparing $\tilde{r}_j$ to the optimal rate r_j is bounded by at most $\bar{\epsilon}$.

Using this property we get

$$\sum_{j \in C_i^{>}} \tilde{r}_j \geq (1-\bar{\epsilon}) \sum_{j \in C_i^{>}} r_j$$

for the rounded large rates assigned to terminal i.

Now consider the set $C^{\ll}$ of small channels with respect to terminal i in the optimal solution. When rounding down the total volume of small channels to integral multiples of ϵ_h we may lose an additive term of ϵ_h in each layer h (including the initial layer $h = 0$) summing up to at most $\sum_{h=0}^{k} \epsilon_h = \bar{\epsilon} d'_1 \sum_{h=0}^{k} 2^h = \bar{\epsilon} d'_1 2^{k+1} - 1 < 2\epsilon_k$ when reaching layer k. This covers the case of all channels $j \in C^{\ll}$ with $r_j \leq \epsilon_0$ by choosing an appropriate edge from the initial node $(0, 0, 0, 0, \ldots, 0)$ to a node $(0, 0, c, m_0, 0, \ldots, 0)$ for $c \geq j$.

If $j \in C_i^{\ll}$ but $r_j > \epsilon_0$ holds, a rounded rate $\tilde{r}_j$ is fixed for channel j in some layer $h < k$ in which j is considered to be large. The initial rounding error when fixing $\tilde{r}_j$ for the first time (which by the argument used before can be chosen to stay fixed until j is considered to be small) can again be bounded by a multiplicative factor of $\bar{\epsilon}$. Thus we get the following guarantee for the rounded total volume of channels contained in $C_i^{\ll}$.

$$\sum_{j \in C_i^{\ll}} \tilde{r}_j \geq \left(\sum_{j \in C_i^{\ll}} (1 - \bar{\epsilon}) r_j \right) - \sum_{h=0}^{k} \epsilon_h > \left((1 - \bar{\epsilon}) \sum_{j \in C_i^{\ll}} r_j \right) - 2\epsilon_k$$

Combining the bounds for both subsets we conclude

$$\begin{aligned} \sum_{j \in C_i^{\ll}} \tilde{r}_j + \sum_{j \in C_i^{>}} \tilde{r}_j &\geq \left((1 - \bar{\epsilon}) \sum_{j \in C_i^{\ll}} r_j \right) - 2\epsilon_k + (1 - \bar{\epsilon}) \sum_{j \in C_i^{>}} r_j \\ &= \left((1 - \bar{\epsilon}) \sum_{j \in C_i} r_j \right) - 2\epsilon_k \geq (1 - \bar{\epsilon}) d'_i - 2\epsilon_k \\ &\geq (1 - 3\bar{\epsilon}) d'_i \, . \end{aligned}$$

To see that the energy bound is not exceeded by the total cost of the path we just note that fixing any channel j to $\tilde{r}_j \leq r_j$ may only decrease the required energy (which is equal to the corresponding edge cost) to sustain this rate. Also initially fixing the rates on channels contained in $C_0 = \{j \in [m] \mid r_j < \epsilon_0 + \delta_0\}$ to the solution of an optimal single-user schedule of total rounded volume $\tilde{V} \leq \sum_{j \in C_0} r_j$ does not increase the cost. ◀

Thus, the multiplicative rounding error on each terminal i with respect to its demand d'_i amounts to at most $3\bar{\epsilon}$. Subsequently determining a greedy assignment of exact small data rates may introduce another additive error of ϵ_k when B_k is the bin range containing terminal i.

▶ **Lemma 4.18.** *If there is a path of cost at most E in the rounded rates graph covering each terminal i to a total rate of $(1 - 3\bar{\epsilon})d'_i$ then a feasible OFDMA schedule satisfying a demand of $(1 - 4\bar{\epsilon})d'_i$ on each terminal $i \in [n]$ can be efficiently computed.*

Proof. Follow the path in the graph and fix the rates of each channel j to the value determined by the corresponding edge connecting node $(L, i, j - 1, m_0, \ldots, m_z)$ with its successor node $(L, i, j, m_0, \ldots, m'_\ell, \ldots, m_z)$ on this path. If j is smaller than the initial split index c defined by the transition $(0, 0, 0, 0, \ldots, 0)$ to $(0, 0, c, m_0, 0, \ldots, 0)$ taken by the path, set r_j according to the corresponding optimal single-user schedule for demand $m_0\epsilon_0$ and capacities ϵ_0.

Whenever a covering edge $(L, i - 1, j, m_0, \ldots, m_z)$ to $(L, i, j, m'_0, \ldots, m'_z)$ for some terminal i is taken we assign channels to this terminal according to the vector $(a_0, \ldots, a_z)$ with $a_j = m_j - m'_j$ for $j \in \{0, \ldots, m\}$. For all coordinates a_j with $j \geq 1$ we can immediately identify the specified number of channels with rate $r_j = \epsilon_L + j\delta_L$ and assign a subset of this cardinality to terminal i. In order to fill the gap covered by coordinate a_0 we select

from all (yet remaining) channels with rate less than ϵ_L a maximal prefix $C_i^{\ll}$ such that $\sum_{j \in C_i^{\ll}} r_j \leq a_0 \cdot \epsilon_L$. By Lemma 4.9 this results in a cover losing at most an additive term of $\epsilon_L \leq \bar{\epsilon} d'_i$ in the total rate of terminal i guaranteeing an overall rate of $(1 - 4\bar{\epsilon})d'_i$ for each terminal i. ◀

▶ **Theorem 4.19.** *If there is a path of cost at most E in the defined rounded-rates graph from node $(0, 0, 0, 0, \ldots, 0)$ to a node $(D, n, m, m_0, \ldots, m_z)$ for any $m_0, \ldots, m_z \in \{0, \ldots, m\}$ then a feasible OFDMA schedule with respect to the energy bound E guaranteeing a demand of $(1 - \epsilon)d'_i$ for each terminal $i \in [n]$ can be efficiently found. If no such paths exists then also no feasible OFDMA schedule satisfying demands d'_i for each terminal $i \in [n]$ exists.*

Proof. Follows directly by Lemmas 4.17 and 4.18 using $\bar{\epsilon} \leq \epsilon/4$. ◀

4.2 Restricted Assignment Model

The restricted assignment model in Santa Claus scheduling is a subclass that received a lot of attention by the community in the last years [BS06, AFS08, Fei08, HSS11] because it allows for easier approximation approaches as the general Santa Claus variant while seemingly bearing the same hardness properties. In a sequence of improvement steps it could be shown that restricted assignment instances of Santa Claus scheduling are approximable by a constant factor (for latest results see [HSS11]).

Translated to the context of OFDMA scheduling, the restricted assignment model is similar to the case of uniformly related channel costs considered in Section 4.1. As before, each channel has its own uniform cost coefficient p_j which is the same on all terminals. However, unlike in uniformly related channel instances, terminals may be excluded from using some channels which may be encoded by allowing the more general cost coefficients $p_{ij} \in \{p_j, \infty\}$.

By Theorem 3.18 we already know that the restricted assignment case is not approximable by any factor better than $1/2$ even in the special case of uniform demands and restricted universe $\{p, q\} \ni p_j$ for all $j \in [m]$. In general we can show that restricted assignment instances can not be approximated significantly better than instances with unrelated costs.

▶ **Theorem 4.20.** *Let $\alpha \in \Omega(\log(mn)^{-1})$ be the approximation factor of an algorithm approximating the Max-Rate OFDMA problem in the restricted assignment model. Then also an $\Omega(\log(mn)^{-1})$-approximation algorithm for the unrelated channel cost model can be given.*

Proof. We will briefly sketch the reduction from general (unrelated channel) instances to the restricted assignment model. Let k be the number of different cost values $p_{ij} \in \{\bar{p}_1, \ldots, \bar{p}_k\}$ of a channel j in the given unrelated Max-Rate OFDMA instance. In the restricted assignment instance there will be k copies $j^{(1)}, \ldots, j^{(k)}$ of channel j each with its own uniform cost $p_{j^{(i)}} = \bar{p}_i$ for $1 \leq i \leq k$. Each copy $j^{(i)}$ may only be assigned to terminals t which have $p_{tj} = \bar{p}_i$ in the original instance and a small selection of the dummy terminals defined below. For channel j we create $k - 1$ many dummy terminals $t_1, \ldots, t_{k-1}$ (each with neglectable demand compared to the smallest demand of the original instance) where terminal t_h is only allowed to use channels $j^{(h)}$ and $j^{(h+1)}$. The demand of t_h is small enough to be completely satisfied by any of those two channels with only marginal energy requirements.

It is easy to see that by this construction only one copy of channel j can be used by one of the original terminals, as independently of the demand at least one channel has to be assigned to each dummy terminal. Also any solution to the original instance can be translated to a solution for the transformed instance with a neglectable decrease in data

rates (since only a marginal amount of energy has to be used for satisfying the demand of the dummy terminals).

As the size of the new instance is polynomial in the size of the original instance, any approximation factor logarithmic in the number of terminals and channels would also imply a logarithmic factor for the original unrelated cost instance. ◀

While the theorem suggests that developing algorithms achieving a constant or logarithmic approximation factor for the restricted assignment model may not be easier than for the general case it does not allow any conclusion about the difficulty of finding linear factor approximations. Also the reduction requires the terminals to have non-uniform demands. So restricting to uniform demands might allow for easier approximation. Indeed in Section 4.3.2.2 we exploit uniform demands to derive a $1/n$-approximation in the restricted assignment model.

4.3 Unrelated Channel Costs

When not restricting the Max-Rate OFDMA problem according to Definition 2.8 in any way we speak of the unrelated channel cost model. Intuitively, approximating this general variant is much more difficult than in the special cases considered before. However, up to now, no stronger inapproximability result than the one given in Section 3.3 is known, which is based on a specially structured restricted assignment instance (corresponding to the big goods/small goods case of Santa Claus scheduling). In principle, deriving an approximation algorithm achieving an approximation factor of $1/2$ might still be possible.

The related problem of Santa Claus scheduling shows a similar huge gap between lower and upper bounds on the approximation factor. While the best known approximation algorithm [CCK09] for general instances of this problem achieves a factor of $\Omega(1/(m^\epsilon \log m))$ in time $m^{O(1/\epsilon)}$ the best known upper bound [BD05] is $1/2$.

All recent approaches for deriving good approximation algorithms for the Santa Claus scheduling problem are based on the formulation of the so called configuration-LP introduced by [BS06]. This (exponentially sized) linear program encodes each feasible assignment of a subset (configuration) C of jobs assigned to machine i as a variable $x_{i,C}$ imposing a fractionally relaxed exclusive assignment constraint when considering configuration-to-machine assignments. Unfortunately, this approach is not applicable to OFDMA scheduling as the configurations suited for each terminal can not be defined by a local property but depend on the global distribution of channels to terminals.

A more basic approximation technique by [BD05] can however be adapted to derive a simple and fast $1/m$-approximation algorithm for the general Max-Rate OFDMA problem.

4.3.1 A Matching-based $1/m$-Approximation

Our algorithm will be based on the following simple observation which directly follows from the fact that no terminal can get more than $m-n-1$ many channels in any feasible solution.

▶ **Observation 4.21.** *For any Max-Rate OFDMA instance with n terminals and $m \geq n$ many channels there is a feasible $1/(m-n+1)$-approximate solution where each terminal uses exactly one channel.*

Choosing for each terminal exactly one of the assigned channels with largest data rate in a given optimal schedule would immediately result in such a one-channel solution. As an optimal general Max-Rate OFDMA schedule is hard to compute we just restrict ourselves to

find an optimal one-channel solution (which by this property is then known to be at least $1/(m-n+1)$-approximate).

The simplified problem can be solved by the execution of a minimum cost matching algorithm nested into a binary search for the maximal λ. For each fixed λ we can compute the energy e_{ij} needed to route data at a rate of λd_i for terminal i over channel j. For each terminal-channel pair (i,j) this energy is completely determined by λd_i and p_{ij} independent of any other terminals. If the minimum cost matching of cardinality n has total cost at most the value of the energy bound E, the considered λ is achievable, otherwise no one-channel solution is able to sustain this rate.

▶ **Theorem 4.22.** *The min-cost matching algorithm above computes a schedule that is at least* $\frac{1}{m-n+1}$*-approximate.*

Proof. Follows directly by Observation 4.21. ◀

For fixed λ the running time is dominated by the computation of the min-cost matching of cardinality n in the constructed weighted assignment graph. As seen in Corollary 3.28 for $|V| \in O(m)$ and $|E| \in O(nm)$ this can be done in time $O(m(n^2 + \log m))$. For maximizing λ we can initialize the binary search with the same upper and lower bounds already specified in Lemma 4.7 which also hold in the unrelated channel case. So a binary search up to a precision of ϵ needs at most $O(\log(m) + \log(\frac{p_{\max}}{p_{\min}}) + \log(\epsilon^{-1}))$ many iterations.

Like in [BD05] finding an optimal one-channel solution can also be successively repeated to improve the quality of the approximate schedule. Of course, the costs of the matching edges have to be updated for each terminal $i \in [n]$ to reflect the actual energy requirements of the proposed k-channel assignment on i given the $(k-1)$-channel assignment fixed before. If C^i_{k-1} denotes the set of channels assigned to terminal i in iteration $k-1$, then the cost for edge (i,j) in iteration k will be defined by the optimal solution to the single-user OFDM instance with demand λd_i for terminal i on channel set $C^i_{k-1} \cup \{j\}$.

This repetition renders the algorithm to be superior to the following simple assignment rule which is (in a more advanced form) implemented by some heuristic solution methods designed for practical use [WSEA04, SAE05], however with the additional benefit of achieving a worst-case approximation guarantee. The most basic form of the heuristic assignment rule uses the following round-robin distribution of channels to terminals with decreasingly ordered preference lists: The first terminal greedily takes his favorite channel, the second terminal chooses the best of the remaining ones and so on. After n channels are assigned, the process starts all over again with the remaining channels repeating this distribution until no channel remains. Since worst-case instances could always be constructed in a way such that the critical terminal with minimum rate is the last in the assignment procedure the heuristic basically can ensure only the following performance guarantee.

▶ **Observation 4.23.** *For every terminal i let $\pi(i,j)$ denote the j-th cheapest channel from the perspective of this terminal, that is, the individual channel costs $p_{i,\pi(i,1)}, \ldots, p_{i,\pi(i,m)}$ are increasing for each i. Then the above heuristic and also the repeated min-cost matching algorithm yield a solution which is at least as good as any solution assigning to each terminal every n-th channel in its preference list $p_{i,\pi(i,1)}, \ldots, p_{i,\pi(i,m)}$ (allowing multiple use of multiply assigned channels).*

It is easy to see that our approximation algorithm always finds a solution that is at least as good as the guarantee stated above. In the repeated matching algorithm each terminal i is assigned a channel of cost at most $q_{i,k} = p_{i,\pi(i,kn)}$ in iteration k. Suppose this is not the case,

then there is a terminal i being assigned a channel j of cost $u_{i,k} = p_{i,j} > q_{i,k}$. As the other terminals block at most $n-1$ of the channels corresponding to $p_{i,\pi(i,(k-1)n)}, \ldots, p_{i,\pi(i,kn)}$ there is at least one channel with cost at most $q_{i,k}$ for terminal i in iteration k which is not incident to the matching. Substituting channel j by this channel in the assignment to terminal i would decrease the cost of the matching by $u_{i,k} \cdot \psi(\lambda d_i) - q_{i,k} \cdot \psi(\lambda d_i) > 0$.

Note that the whole repeated min-cost matching algorithm stated above can also be used in a more generalized model allowing not only unrelated channel cost coefficients but also unrelated rate-power functions for each individual terminal-channel pair. Of course, the computation of the corresponding single-user optima leads to an increase in algorithmic complexity in this setting.

4.3.2 An Approach Based on Rounding the Convex Programming Solution

It is not trivial to design approximation algorithms achieving an approximation ratio more than $\Theta(1/m)$ for Max-Rate OFDMA scheduling in the general case of unrelated channel cost. In terms of a fixed multiplicative guarantee relative to the value of an optimal solution we were not able to break this bound. The following approach, however, provides an additive bound on the performance guarantee which —though being arbitrarily bad in worst-case— could be useful for a large number of practical input instances. As a direct consequence of the additive approximation guarantee we get that relaxing the integrality condition on the exclusive channel assignment constraint to half-integrality allows us to derive a $1/2$-approximate solution. Furthermore, this approach can also be extended to derive a $1/n$-approximation algorithm for the restricted assignment model in case of uniform demands.

Using the relaxed natural LP formulation of the Santa Claus scheduling problem, Bezáková and Dani [BD05] were able to construct a suboptimal Santa Claus schedule satisfying an additive approximation guarantee of $\mathrm{OPT} - s_{\max}$ where $s_{\max}$ denotes the maximal size of any job on any machine in the instance.

A similar approach can also be used in OFDMA scheduling. Analogously to Definition 3.38 of the relaxed Min-Power OFDMA problem we are able to formulate a fractional version of Max-Rate OFDMA scheduling. There is a small obstacle, though, as incorporating the variable λ into the extended power-rate function $\xi_i(e,x) := \min\{x\lambda d_i, x\psi^{-1}(e/x)\}$ may render it to be non-concave. For fixed λ, however, this is not a problem, so —nested inside a binary search to maximize λ— testing feasibility of the following system of convex constraints is efficiently possible.

▶ **Definition 4.24** (Feasibility Constraints for Fractionally Relaxed Max-Rate OFDMA)**.**

$$\sum_{i=1}^{n}\sum_{j=1}^{m} e_{ij} \leq E \tag{4.1}$$

$$\forall i \in [n] \quad \forall j \in [m] \qquad r_{ij} - \xi_i(e_{ij}/p_{ij},\ x_{ij}) \leq 0 \tag{4.2}$$

$$\forall i \in [n] \qquad \sum_{j=1}^{m} r_{ij} \geq \lambda d_i \tag{4.3}$$

$$\forall j \in [m] \qquad \sum_{i=1}^{n} x_{ij} \leq 1 \tag{4.4}$$

$$\forall i \in [n] \quad \forall j \in [m] \qquad x_{ij} \geq 0, \quad r_{ij} \geq 0, \quad e_{ij} \geq 0 \tag{4.5}$$

Thus, by using convex optimization techniques we can compute a fractional solution to Max-Rate OFDMA in polynomial time.

The inflated data rate $\widehat{r_{ij}}$ of a terminal-channel pair defined below represents terminal i's estimate of the total data rate on channel j.

▶ **Definition 4.25** (Inflated Data Rate)**.** We define the *inflated data rate* $\widehat{r_{ij}}$ *perceived by terminal* i *on channel* j to be

$$\widehat{r_{ij}} := \begin{cases} r_{ij}/x_{ij} & \text{for } x_{ij} > 0 \\ 0 & \text{else} \end{cases}$$

As discussed in Section 3.6 the variable r_{ij} was introduced to substitute the non-convex term $x_{ij}r_j$. So in this context $\widehat{r_{ij}}$ indicates the assumed value of r_j for terminal i. Note that, however, $\widehat{r_{ij}} \neq \widehat{r_{tj}}$ for $t \neq i$ can not be precluded in general.

We will give a transformation to construct an integral schedule from any fractional solution according to Definition 4.24 while losing at most a data rate of $\max_j \widehat{r_{ij}}$ on each terminal i. Since the technique of [BD05] is not applicable in our context we will adapt the approach of Shmoys and Tardos [ST93] originally introduced for the generalized assignment problem, modeling the integrality transformation as a bipartite matching problem.

In order to show the existence of an appropriate integral schedule we exploit some properties of fractional matchings.

▶ **Definition 4.26** (Fractional Matching)**.** A *fractional matching* in an undirected graph $G = (V, E)$ is a mapping $f\colon E \to \mathbb{R}_{\geq 0}$ such that $\sum_v f(\{u, v\}) \leq 1$ holds for each node $u \in V$. For simplicity of notation we write $f(u, v)$ instead of $f(\{u, v\})$.

We say that a matching *covers* node u if $\sum_v f(u, v) = 1$. The *cost* of a fractional matching in a graph with edge costs $w\colon E \to \mathbb{R}_{\geq 0}$ is defined by $\sum_{e \in E} f(e)w(e)$.

We construct a bipartite graph in which we can represent any fractional OFDMA schedule satisfying the convex constraints given in Definition 4.24 in terms of a fractional matching. This will be used to show the existence of an appropriate integral matching in the constructed graph —which corresponds to an integral OFDMA schedule for the original instance.

▶ **Definition 4.27** (Extended Fractional Assignment Graph)**.** Given a fractional OFDMA schedule $(x_{ij})_{ij \in [n] \times [m]}$, $(r_{ij})_{ij \in [n] \times [m]}$, $(e_{ij})_{ij \in [n] \times [m]}$ satisfying Constraints 4.2, 4.3, 4.4, and 4.5.

We define the *extended fractional assignment graph* to be an undirected bipartite graph $G = (U \cup C, E)$ where node partition C consists of the nodes $\{1, \ldots, m\}$ each representing a channel, whereas node partition U representing the terminals contains $k_i := \lceil \sum_j x_{ij} \rceil$ many nodes $\{v_1^i, \ldots, v_{k_i}^i\}$ for each terminal $i \in [n]$.

The edge set will be defined in the following way: Consider the node set $\{v_1^i, \ldots, v_{k_i}^i\}$ corresponding to a terminal i. Let the channel set be ordered non-increasingly with respect to the inflated data rates perceived by this terminal. For simplicity of notation when considering a terminal i we assume $\widehat{r_{i1}} \geq \cdots \geq \widehat{r_{im}}$. For any terminal i and any $s \in \mathbb{N}_0$ with $s < k_i$ let $\nabla_s^i \in \mathbb{N}$ denote the minimal index with $\sum_{j=1}^{\nabla_s^i} x_{ij} \geq s$. For $s = k_i$ we define ∇_s^i to be the largest index with $x_{i\nabla_s^i} > 0$. Then from node v_s^i there will be an edge to all channels in the interval $\{\nabla_{s-1}^i, \ldots, \nabla_s^i\}$. The cost $w(e)$ of an edge $e = \{v_s^i, j\}$ is defined by $w(e) = p_{ij}\psi(\widehat{r_{ij}})$.

For intuition on the construction of the extended fractional assignment graph and the corresponding fractional matching consider for each terminal i the total fraction $F_i := \sum_j x_{ij}$ of channels assigned to it (in the given fractional OFDMA schedule). Our intention is to divide the channels contained in $C^i := \{j \mid x_{ij} > 0\}$ into groups $C^i_1, \ldots, C^i_{\lceil F_i \rceil - 1}$, each representing a total fraction of exactly $\sum_{j \in C^i_s} x_{ij} = 1$ for all $1 \leq s \leq \lceil F_i \rceil - 1$, and one group $C^i_{\lceil F_i \rceil}$ representing the remaining fraction $F_i - (\lceil F_i \rceil - 1)$. We then derive the existence of an appropriate integral OFDMA schedule assigning to each terminal i at least one of the channels contained in C^i_s for each $1 \leq s \leq \lceil F_i \rceil - 1$.

Of course, this approach is only possible if the fraction of some channels may be distributed between more than one group. If we have $\sum_{j \in C^i_s} x_{ij} > 1$ but $\sum_{j \in C^i_s \setminus \{c\}} x_{ij} < 1$ for a chosen channel $c \in C^i_s$, then we split up x_{ic} into the two fractions $x'_{ic} = 1 - \sum_{j \in C^i_s \setminus \{c\}} x_{ij}$ and $x''_{ic} = \sum_{j \in C^i_s} x_{ij} - 1$ and account for the fraction x'_{ic} in C^i_s and add the remaining fraction x''_{ic} to C^i_{s+1}.

More formally, using the described splitting approach, we show the existence of an appropriate fractional matching.

▶ **Lemma 4.28.** *Any fractional energy-minimal OFDMA schedule satisfying Constraints 4.2, 4.3, 4.4, and 4.5 with total energy cost $w = \sum_{i=1}^{n} \sum_{j=1}^{m} e_{ij}$ defines a fractional matching of cost w covering all nodes from $U' = \{v^i_s \in U \mid s < k_i, i \in [n]\}$ in the corresponding extended fractional assignment graph $G = (U \cup C, E)$.*

Proof. We define a feasible fractional matching in the following way.

Let $c \in \{\nabla^i_{s-1}, \ldots, \nabla^i_s\}$ be one of the channels adjacent to v^i_s. Note that $\nabla^i_s \geq \nabla^i_{s-1} + 1$ for all $2 \leq s < k_i$ since $x_{ij} \leq 1$ for all i, j. If $\nabla^i_{s-1} < c < \nabla^i_s$ then c is matched to v^i_s by the entire fraction $f(v^i_s, c) = x_{ic}$ given by the OFDMA schedule. Also for $c = \nabla^i_{k_i}$ we set $f(v^i_{k_i}, c) = x_{ic}$. For $c = \nabla^i_{s-1}$, however, channel c is assigned to v^i_s by the fraction $f(v^i_s, c) := \sum_{j=1}^{c} x_{ij} - (s-1)$ and for $c = \nabla^i_s$ with $s < k_i$ by $f(v^i_s, c) = s - \sum_{j=1}^{c-1} x_{ij}$. Thereby, $f(v^i_{s-1}, c) + f(v^i_s, c) = x_{ic}$ for all channels c and $2 \leq s \leq k_i$. Furthermore, by definition of $f(v^i_s, \nabla^i_{s-1})$ and $f(v^i_s, \nabla^i_s)$ we have $\sum_c f(u, c) = 1$ for each $u \in U'$ and $\sum_{v'} f(v, v') \leq 1$ for each node $v \in U \cup C$.

To prove cost equivalence of the matching we note that by energy minimality of the OFDMA schedule the (by Constraint 4.2) imposed inequalities

$$x_{ij} \psi^{-1}\left(\frac{e_{ij}}{p_{ij} x_{ij}}\right) \geq \xi_i\left(\frac{e_{ij}}{p_{ij}}, x_{ij}\right) \geq r_{ij}$$

have to be tight (otherwise e_{ij} can be decreased). Thus, $\frac{r_{ij}}{x_{ij}} = \psi^{-1}(\frac{e_{ij}}{p_{ij} x_{ij}})$ or equivalently $x_{ij} p_{ij} \psi(\widehat{r_{ij}}) = e_{ij}$ which equals exactly the (fractional) cost of the corresponding fractional matching edge. ◀

By matching/flow theory we can directly derive the following lemma.

▶ **Lemma 4.29.** *In the extended fractional assignment graph of a feasible fractional OFDMA schedule with total energy cost w there exists an integral matching covering all nodes from $U' = \{v^i_s \in U \mid s < k_i, i \in [n]\}$ of cost at most w.*

Proof. If the given fractional OFDMA schedule is not energy-minimal we transform it into an appropriate schedule of less energy cost. Now, consider the fractional matching defined by Lemma 4.28 and remove all nodes from $U \setminus U'$. If we use the standard transformation of the bipartite graph into a flow network we observe that the fractional matching defines

a (fractional) flow of value $|U'|$ and cost w which by the node elimination above is also a maximum flow in the network. As there is always an integral min-cost maximum flow in any network with linear flow costs and integral edge capacities we know that there exists an integral flow of value $|U'|$ and cost at most w which defines an integral matching of the same cost covering all nodes from U'. ◀

In order to actually determine an integral matching with the desired properties we can just ignore the nodes in $U \setminus U'$ and instead consider the reduced fractional assignment graph defined below.

▶ **Definition 4.30** (Reduced Fractional Assignment Graph)**.** The induced subgraph $G' = (U' \cup C, E')$ of the extended fractional assignment graph $G = (U \cup C, E)$ obtained by eliminating all nodes (and incident edges) from $U \setminus U'$ with $U' = \{v_s^i \in U \mid s < k_i, i \in [n]\}$ will be called the *reduced fractional assignment graph.*

Before stating our algorithm CP-ROUNDING we have to ensure that appropriate matchings in the reduced fractional assignment graph can indeed be retranslated to a corresponding OFDMA schedule. At this point we will exploit the chosen ordering of the set of channels assigned to each terminal $i \in [n]$.

▶ **Lemma 4.31.** *Any integral matching of size $|U'|$ and cost w in the reduced fractional assignment graph $G' = (U' \cup C, E')$ corresponds to an integral OFDMA schedule of total energy cost w with data rate at least $\lambda d_i - \max\{r_{ij}/x_{ij} \mid j \in [m], x_{ij} > 0\}$ on each terminal $i \in [n]$.*

Proof. When interpreting a (fractional/integral) matching of the extended fractional assignment graph as a (fractional/integral) OFDMA schedule, the data rate of terminal i on channel j is defined by $\sum_{s=1}^{k_i} f(v_s^i, j)\widehat{r_{ij}}$. For each node $v_s^i \in U$ let $C(v_s^i)$ denote the set of channels adjacent to v_s^i. Also we define $\sigma_s^i = \sum_{j \in C(v_s^i)} f(v_s^i, j)\widehat{r_{ij}}$ as the data rate guaranteed in a matching by the edges incident to node v_s^i. Then for the fractional matching according to Lemma 4.28 we have $\sum_{s=1}^{k_i} \sigma_s^i \geq \lambda d_i$.

By definition of the extended fractional assignment graph we know that $\widehat{r_{ij}} \geq \widehat{r_{ij'}}$ for all $j \in C(v_s^i)$ and $j' \in C(v_{s'}^i)$ with $s' > s$. So it directly follows that $\widehat{r_{ij}} \geq \sigma_{s+1}^i$ for any $j \in C(v_s^i)$. Thus, any integral matching covering all nodes from U' in the reduced fractional assignment graph guarantees a data rate of at least

$$\sum_{s=1}^{k_i-1} \sigma_{s+1}^i \geq \lambda d_i - \sigma_1^i \geq \lambda d_i - \max_{j \in [m]} \widehat{r_{ij}}$$

for each terminal i.

The cost of any edge (v_s^i, j) used in the integral matching is defined by $p_{ij}\psi(\widehat{r_{ij}})$ and thereby exactly equals the energy needed for terminal i to sustain a rate of $\widehat{r_{ij}}$ on channel j. So the data rates of the schedule induced by the integral matching incur a total energy cost equal to the cost of the matching, which finally proves the claim. ◀

Summarizing the single steps to construct an integral OFDMA schedule we get the following algorithm.

Algorithm CP-ROUNDING

1. Determine the maximal value λ^* for which Constraints 4.1, 4.2, 4.3, 4.4, and 4.5 can be satisfied and compute a feasible fractional schedule $(x_{ij})_{ij\in[n]\times[m]}$, $(r_{ij})_{ij\in[n]\times[m]}$, $(e_{ij})_{ij\in[n]\times[m]}$ for λ^*.
2. Use this solution to construct the corresponding reduced fractional assignment graph according to Definition 4.30.
3. Find a minimum cost maximum cardinality matching in this bipartite graph. If for any $s \in [k_i - 1]$ edge (v_s^i, j) is in the matching fix $\bar{x}_{ij} = 1$ otherwise $\bar{x}_{ij} = 0$.
4. Output channel assignment $(\bar{x}_{ij})_{ij\in[n]\times[m]}$ with data rate allocation $(\bar{r}_j)_{j\in[m]}$ where $\bar{r}_j = \widehat{r_{ij}}$ if channel j is assigned to terminal i. If channel j is not assigned to any terminal set $\bar{r}_j = 0$.

▶ **Theorem 4.32.** *Let λ^* denote the maximal objective value satisfying the fractional convex constraints given in Definition 4.24 and let $(x_{ij})_{ij\in[n]\times[m]}$ and $(r_{ij})_{ij\in[n]\times[m]}$ denote the fractional channel assignment and rate allocation of a corresponding optimal fractional OFDMA schedule.*

Then algorithm CP-ROUNDING computes a feasible integral OFDMA schedule guaranteeing a data rate of $\lambda^* d_i - \max\{r_{ij}/x_{ij} \mid j \in [m], x_{ij} > 0\}$ on each terminal $i \in [n]$.

Proof. Follows directly by Lemmas 4.28, 4.29, and 4.31 ◀

The theorem trivially implies the following corollary which can be useful in practical instances.

▶ **Corollary 4.33.** *If for the fractional OFDMA schedule computed in Step 1 of algorithm CP-ROUNDING there is an $\alpha \leq 1$ such that $r_{ij} \leq \alpha x_{ij} \lambda d_i$ holds for all $i \in [n]$, $j \in [m]$, then the algorithm outputs a feasible integral $(1-\alpha)$-approximate OFDMA schedule.*

4.3.2.1 A $1/2$-approximate Half-Integral Solution

As a way to interpret fractional solutions of the OFDMA scheduling problem under consideration of the interference restrictions of real-world OFDMA scheduling, the authors of [WCLM99] propose the notion of time-sharing.

The OFDMA scheduling problem introduced in Chapter 2 aims to optimize the OFDMA subchannel usage over a given time-frame. A natural and atomic unit to measure the length of a time-frame is the duration of a single OFDM symbol. While short time-frames may support only one OFDM symbol, this size might not be reasonable from a complexity point of view when optimizing the subchannel utilization. Since even the computation of an adequate suboptimal OFDMA schedule can be a time-consuming task this should only be done when significant changes in the subchannel gains occur.

If a time-frame ranges over the duration of more than one OFDM symbol, then in principle TDMA sharing of subchannels is possible. Of course, the relation between required energy and acquired data rate has to be adapted to this modification.

We assume that the data rates are normalized with respect to the length of a time-frame, that is, a data rate of r corresponds to submitting a total volume of r units of data over the whole duration of the time-frame. If a (sub)channel is shared equally between two terminals, each of those owns only half a time-frame for the exclusive transmission of data over this channel. So in order for one of the terminals to reach its desired data rate of r (with respect to the whole time-frame duration) it has to utilize the assigned part of the time-frame at a

data rate of $2r$. As the energy cost for a terminal i to sustain a data rate of r on channel j over one time-frame is defined by $p_{ij}\psi(r)$ the energy needed to support the double rate for half of that time is $\frac{1}{2}p_{ij}\psi(2r)$.

In general let $0 \leq x_{ij} \leq 1$ denote the time-share of terminal i on channel j. Then, in order for terminal i to sustain a total data rate of r_{ij} (over the duration of a whole time-frame) on channel j using only its time-share x_{ij}, the required energy amounts to $x_{ij}p_{ij}\psi(r_{ij}/x_{ij})$ which is exactly the energy relation used in the fractional formulation of the OFDMA scheduling problem used in Definitions 3.38 and 4.24.

Thus, if the duration of a time-frame is large enough, say L many OFDM symbols, such that $x_{ij}L$ is integral for each terminal-channel pair $i \in [n]$, $j \in [m]$ in an optimal fractional OFDMA schedule, this schedule could be realized by a TDMA sharing of OFDM subchannels. However, the authors of [WCLM99] discard their consideration to be impractical, as in most cases channels may not stay unchanged long enough in order to realize the large time-frames needed for small fractions x_{ij}.

Nevertheless, modifying the convex programming rounding approach from Section 4.3.2 may offer a solution to this problem as implied by the following theorem (based upon an idea from [VW10] for the Santa Claus scheduling problem).

▶ **Theorem 4.34.** *Let λ^* be the objective value of an optimal fractional Max-Rate OFDMA schedule with respect to Definition 4.24.*

Then we can efficiently construct a half-integral OFDMA schedule (i.e. $\forall i \forall j\ 2x_{ij} \in \mathbb{N}_0$) that satisfies the constraints from Definition 4.24 for $\lambda^*/2$.

Proof. The proof is analogous to the analysis of algorithm CP-ROUNDING. First we introduce for each channel j an identical copy j' setting $x_{ij'} = x_{ij}$ for each terminal i. Of course, if w is the energy requirement for the original schedule the new one has energy cost $2w$.

Now we construct the extended and reduced fractional assignment graph for the modified schedule. According to Lemma 4.29 an integral matching covering all $k_i - 1 = \lceil 2\sum_j x_{ij} \rceil - 1$ many nodes from U' in the extended fractional assignment graph exists. So analogously to Lemma 4.31 the data rates of the implied integral OFDMA schedule $(\bar{x}_{ij})_{ij\in[n]\times(C\cup C')}$, $(\bar{r}_{ij})_{ij\in[n]\times(C\cup C')}$ with $C = \{1,\dots,m\}$ and $C' = \{1',\dots,m'\}$ requiring total energy cost $\sum_i \sum_j \bar{e}_{ij} + \bar{e}_{ij'} \leq 2w$ sum up to at least $\sum_{s=2}^{k_i} \sigma_s^i \geq 2\lambda d_i - \max\{r_{ij}/x_{ij} \mid j \in [m], x_{ij} > 0\}$ for each terminal i.

In order to get a feasible (half-integral) schedule to the original OFDMA instance we set $\tilde{x}_{ij} := \frac{1}{2}(\bar{x}_{ij} + \bar{x}_{ij'})$ and $\tilde{e}_{ij} = \frac{1}{2}(\bar{e}_{ij} + \bar{e}_{ij'})$ before removing the channel copies $\{1',\dots,m'\}$. Trivially, the $\tilde{x}_{ij}$ are half-integral and $\sum_i \sum_j \tilde{e}_{ij} \leq w$ holds. As $\bar{e}_{ij} + \bar{e}_{ij'} \geq (\bar{x}_{ij} + \bar{x}_{ij'})p_{ij}\psi(\widehat{r_{ij}})$ is satisfied, the individual energy bound $\tilde{e}_{ij}$ is enough to support a data rate of $\widehat{r_{ij}}$ for terminal i for an $\tilde{x}_{ij}$-fraction of the time-frame —or equivalently $\tilde{x}_{ij}\widehat{r_{ij}}$ averaged over the whole time-frame.

Since $\sum_j (\bar{x}_{ij} + \bar{x}_{ij'})\widehat{r_{ij}} \geq \sum_{s=2}^{k_i} \sigma_s^i$ we have $\sum_j \tilde{x}_{ij}\widehat{r_{ij}} \geq \lambda d_i - \frac{1}{2}\max_{j\in[m]} \widehat{r_{ij}}$ which by Constraint 4.2 is at least $\lambda d_i/2$ for each terminal i. Thus, the half-integral schedule defined by $\tilde{x}_{ij}$ and $\tilde{r}_{ij} = \tilde{x}_{ij}\widehat{r_{ij}}$ is at least half as good as the initially computed fractional schedule. ◀

Note that Theorem 4.34 only compares the quality of the constructed half-integral solution to an optimal fractional OFDMA schedule satisfying Constraint 4.2 which implicitly imposes the inequality $r_{ij} \leq x_{ij}\lambda d_i$ that is trivially fulfilled by every integral solution. Especially, the approximation guarantee is not with respect to arbitrary fractional schedules optimizing the non-concave Max-Rate OFDMA problem according to the originally integral Definition 2.8 but compares to optimality with respect to the fractional formulation given in Definition 4.24.

We use the technique described in Theorem 4.34 to derive an appropriate half-integral approximation algorithm. By a duplication of the set of fractionally assigned channels it can use the same standard matching techniques already applied in algorithm CP-ROUNDING and can be stated analogously.

Algorithm HALF-INTEGRAL-ROUNDING

1. Determine the maximal value λ^* for which the constraints from Definition 4.24 can be satisfied and compute a feasible fractional schedule $(x_{ij})_{ij\in[n]\times[m]}$, $(r_{ij})_{ij\in[n]\times[m]}$, $(e_{ij})_{ij\in[n]\times[m]}$ for λ^*.
2. Add channels $\{1',\dots,m'\}$ to the channel set and fix $x_{ij'} = x_{ij}$.
3. Use this schedule to construct the corresponding reduced fractional assignment graph according to Definition 4.30.
4. Find a minimum cost maximum cardinality matching in this bipartite graph. If for any $s \in [k_i - 1]$ edge (v_s^i, j) with $j \in \{1,\dots,m,1',\dots,m'\}$ is in the matching fix $\bar{x}_{ij} = 1$ otherwise $\bar{x}_{ij} = 0$.
5. Set $\tilde{x}_{ij} = \frac{1}{2}(\bar{x}_{ij} + \bar{x}_{ij'})$ for each $i \in \{1,\dots,n\}$ and $j \in \{1,\dots,m\}$ and delete channels $\{1',\dots,m'\}$
6. Output the half-integral channel assignment $(\tilde{x}_{ij})_{ij\in[n]\times[m]}$ with data rate allocation $(\tilde{r}_{ij})_{ij\in[n]\times[m]}$ where $\tilde{r}_{ij} = \tilde{x}_{ij}\widehat{r_{ij}}$.

The dependency on fractionality can even be further reduced. By a simple coloring argument the half-integral solution computed above can also be transformed into a (half-integral) solution with all except $n/2$ many channels being exclusively assigned, however, by possibly losing an additional factor of 2 in the approximation guarantee.

The corresponding theorem from [VW10] is directly applicable since the channel assignment and the respective data rates may remain fixed throughout the transformation where only the selection of an appropriate *sub-assignment* is required. We call an assignment $(y_{ij})_{ij\in[n]\times[m]}$ a *sub-assignment* of a given channel assignment $(x_{ij})_{ij\in[n]\times[m]}$ if $y_{ij} \le x_{ij}$ holds for all i, j. So the required total energy cost does not increase.

4.3.2.2 A $1/n$-Approximation for Cycle-free Assignment Graphs and Restricted Assignment Instances with Uniform Demands

We will study another approach to compute feasible integral schedules with provable worst-case approximation guarantees in the current section. Consider an optimal fractional Max-Rate OFDMA schedule feasible to the constraints given in Definition 4.24. Instead of analyzing the extended fractional assignment graph, sometimes the following natural representation of the channel assignment can be useful.

▶ **Definition 4.35** (Simple Fractional Assignment Graph)**.** Given a feasible fractional channel assignment $(x_{ij})_{ij\in[n]\times[m]}$ together with a rate allocation $(r_{ij})_{ij\in[n]\times[m]}$. We define the node set $V = T \cup C$ where the nodes $\{t_1,\dots,t_n\} = T$ correspond to the terminals in the given assignment and the nodes $\{c_1,\dots,c_m\} = C$ to the channels with an (undirected) edge between t_i and c_j if $x_{ij} > 0$. Edge $\{t_i, c_j\}$ will have a capacity of 1 and a cost of $p_{ij}\psi(\widehat{r_{ij}})$.

Golovin [Gol05] found out (in the context of Santa Claus scheduling) that if the simple fractional assignment graph is cycle-free, then an integral schedule which is at least $1/n$-approximate to the given fractional solution can directly be derived by a greedy rounding procedure.

More precisely (and translated to OFDMA scheduling terminology), we have the following lemma.

▶ **Lemma 4.36.** *Let $(x_{ij})_{ij\in[n]\times[m]}$ be a (fractional) cycle-free assignment of inflated data rates $(\widehat{r_{ij}})_{ij\in[n]\times[m]}$ to terminals $\{1,\dots,n\}$ satisfying the following properties*

$$\sum_{i=1}^{n} x_{ij} = 1 \qquad \forall j \in [m]$$

$$\sum_{j=1}^{m} x_{ij}\widehat{r_{ij}}/d_i \geq \lambda \qquad \forall i \in [n]$$

$$x_{ij}\widehat{r_{ij}}/d_i \leq x_{ij}\lambda \qquad \forall i \in [n] \quad \forall j \in [m]$$

Then there is a sub-assignment y_{ij} with $y_{ij} \leq x_{ij}$ for all i, j which satisfies

$$\sum_{i=1}^{n} \lceil y_{ij} \rceil \leq 1 \qquad \forall j \in [m]$$

$$\sum_{j=1}^{m} y_{ij}\widehat{r_{ij}} \geq \lambda d_i/n \qquad \forall i \in [n]$$

Proof. Analogously to [Gol05]. ◀

The assignment is computed by rooting each tree-component of the simple fractional assignment graph at an arbitrary terminal node. Then a bottom-up greedy assignment of channels to terminals is done by assigning each terminal i all its (remaining) children $C(i)$ and —if necessary (that is $\sum_{j\in C(i)} x_{ij}\widehat{r_{ij}} < \lambda d_i/n$)— also its parent (thereby removing it from the set of children of its ancestor). The proof in [Gol05] shows that no conflicts due to competing children can occur.

The above assignment procedure is advantageous as it does not increase the individual data rates for each terminal-channel pair and thereby may only decrease the required total energy for the schedule. Thus, this approach also works in the case of OFDMA scheduling.

However, in contrast to OFDMA, the simpler structure of Santa Claus scheduling admits to convert arbitrary feasible fractional assignments into cycle-free ones which allows to derive general $1/n$-approximations. This seems not to be possible for OFDMA scheduling even in the restricted assignment case where the cost of each channel is the same for all assignable terminals.

Though a given fractional assignment for restricted assignment Max-Rate OFDMA scheduling can be easily translated into a fractional schedule which is cycle-free (by fixing the total channel rate $r_j = \sum_{i\in[n]} r_{ij}$, redefining $x'_{ij} := r_{ij}/r_j$ accordingly and applying techniques from generalized network flow theory using flow-gain/loss multipliers on the edges), the obtained schedule loses the critical property of rate truncation, that is, $x_{ij}\widehat{r_{ij}} \leq x_{ij}\lambda d_i$ may not be satisfied anymore —which, however, (together with $\sum_{i=1}^{n} x_{ij} = 1$) is essential for the approach in [Gol05].

If we impose uniform demands in the restricted assignment model, though, the above rounding technique indeed can be used to derive a $1/n$-approximation. We briefly sketch the conversion into a cycle-free assignment in Theorem 4.38.

For the restricted assignment model we can use the nice property that in any feasible fractional solution the total data rate allocated on a channel $j \in [m]$ (summed over all terminals) can always be sustained by the sum of the corresponding energy budgets fixed by the fractional schedule. More formally, we get the following lemma.

▶ **Lemma 4.37.** *Let $p_j \in \mathbb{R}_{\geq 0}$ denote the uniform cost coefficient of channel j on all terminals where it can be assigned to in the restricted assignment model. Then in any fractional schedule $(x_{ij})_{ij\in[n]\times[m]}$, $(r_{ij})_{ij\in[n]\times[m]}$, $(e_{ij})_{ij\in[n]\times[m]}$ according to Definition 4.24 the inequality $p_j\psi(\sum_{i\in[n]} r_{ij}) \leq \sum_{i\in[n]} e_{ij}$ holds for each channel $j \in [m]$.*

Proof. For each $i \in [n]$ with $r_{ij} > 0$, Constraint 4.2 implies $x_{ij} > 0$ and $r_{ij} \leq x_{ij}\psi^{-1}(\frac{e_{ij}}{p_{ij}x_{ij}})$, thus $p_{ij} = p_j < \infty$. Thereby, $r_{ij} \leq x_{ij}\psi^{-1}(\frac{e_{ij}}{p_{ij}x_{ij}})$ is equivalent to $x_{ij}p_j\psi(\widehat{r_{ij}}) \leq e_{ij}$. W.l.o.g. we may assume $\sum_{i\in[n]} x_{ij} = 1$ as otherwise we can just add dummy elements i' with $x_{i'j} = 1 - \sum_{i\in[n]} x_{ij}$ and $r_{i'j} = 0$. Together with Jensen's inequality (Lemma 2.10) we derive

$$p_j\psi(\sum_{i\in[n]} r_{ij}) = p_j\psi(\sum_{i\in[n]} x_{ij}\widehat{r_{ij}}) \leq p_j \sum_{i\in[n]} x_{ij}\psi(\widehat{r_{ij}}) \leq \sum_{i\in[n]} e_{ij}$$

which proves the lemma. ◀

▶ **Theorem 4.38.** *From an optimal fractional OFDMA schedule according to Definition 4.24 we can efficiently compute a feasible integral $1/n$-approximation for restricted assignment instances with uniform demands.*

Proof. Consider the normalization $x'_{ij} := r_{ij}/r_j$ for $r_j = \sum_{i\in[n]} r_{ij}$ in a given fractional channel assignment $(x_{ij})_{ij\in[n]\times[m]}$ with rate allocation $(r_{ij})_{ij\in[n]\times[m]}$.

Note that by fixing the overall rate of each channel $j \in [m]$ to r_j we can guarantee that for any modified fractional channel assignment $(x''_{ij})_{ij\in[n]\times[m]}$ with $\sum_{i\in[n]} x''_{ij} \leq 1$ the total required energy for the corresponding fractional OFDMA schedule with data rates $r''_{ij} = x''_{ij}r_j$ does not exceed the total energy budget $\sum_{i=1}^{n}\sum_{j=1}^{m} e_{ij}$ of the original fractional schedule (follows directly by Lemma 4.37).

By uniformity of the demands and Constraints 4.2 and 4.4 we have

$$r_j = \sum_{i\in[n]} r_{ij} \leq \sum_{i\in[n]} x_{ij}\lambda d \leq \lambda d$$

so all premises of Lemma 4.36 with exception of cycle-freeness are fulfilled.

We extend the simple fractional assignment graph according to Definition 4.35 to a generalized flow network with flow gain multipliers on each edge $e \in E$ (for an introduction to generalized flows see, e.g., [AMO93]). We orient the edges from channels to terminals, that is, we have $e = (c_j, t_i)$ for $c_j \in C$ and $t_i \in T$. The gain multiplier for each edge $e = (c_j, t_i)$ for some $j \in [m]$ will be set to r_j for all $i \in [n]$, which means that when routing z_{ij} many units of flow along edge (c_j, t_i) a flow of $z_{ij} \cdot r_j$ many units arrives at node t_i. Using the inverse edge in the residual network, of course, reduces the flow by a factor of r_j.

The normalized channel assignment $x'_{ij} := r_{ij}/r_j$ naturally defines a generalized flow f in the constructed network where each channel supplies a flow of 1 to the network and each terminal $i \in [n]$ receives a total flow of $\sum_{j\in[m]} r_{ij}$.

Suppose f implies an undirected cycle K on the edge set of the graph. As only channels assigned to at least two different terminals can be on a cycle, we have $0 < f(e) < r_j$ for each edge $e = (c_j, t_i)$ contained in the cycle. Thus, in the residual network there exist exactly two directed cycles corresponding to K. Each cycle in the residual network is either gainy, lossy or neutral, meaning that when sending one unit of flow along the direction of the cycle more than one (resp. less than or exactly one) unit arrives at the arbitrarily chosen starting node. Clearly, if one direction of the underlying undirected cycle is lossy, the other direction must be gainy. So for each cycle K we can pick a direction that is either neutral or gainy. As the property trivially holds for each node contained in the cycle, we may augment flow along

this directed cycle without reducing the total flow received by each terminal on the cycle. Filling the bottleneck capacity of this cycle, at least one of the cycle edges disappears in the residual network without adding any new residual edge. After augmenting we remove all edges with zero flow from the corresponding simple assignment graph.

By iterating this cycle-augmentation approach a linear number of times, we get a cycle-free schedule, that is, the corresponding simple fractional assignment graph is cycle free. Now we can apply Lemma 4.36 on the obtained fractional assignment and derive the claimed $1/n$-approximative integral schedule. ◀

4.4 Summary

In this chapter we tried to circumvent the strong inapproximability of OFDMA scheduling observed in Chapter 3 by restricting the problem to special subclasses.

Even for one of the most basic classes, namely the case of uniformly related channel costs, Min-Power OFDMA could be shown to be inapproximable by any polynomial time computable factor. However, switching to the dual Max-Rate variant allowed us to derive approximation guarantees arbitrarily close to the optimum on this class. We derived several very efficient constant-factor approximation algorithms for the uniformly related channel cost model (with or without uniform demands), before finally constructing appropriate polynomial time approximation schemes.

For more general cases, approximation ratios arbitrarily close to 1 are not expectable (unless $P = NP$) as implied by the $1/2$-inapproximability result for Max-Rate OFDMA given in Chapter 3. In fact, it seems that even the construction of constant-factor approximation algorithms for the unrelated cost model could be out of reach. Similar gaps between the best known approximation guarantees and complementary inapproximability results can be observed in the related problem of Santa Claus scheduling.

Using a simple matching-based approach we were able to give an easy and efficient $1/m$-approximation algorithm for the general case of unrelated channel costs. Exploiting the structure of fractional OFDMA schedules we were able to see that relaxing the exclusiveness constraint on each channel to half-integrality admits the computation of a schedule which results in a $1/2$-approximation with respect to the optimal integral solution.

For the restricted assignment model with uniform demands, integral $1/n$-approximations could be derived by translating optimal fractionally relaxed OFDMA schedules into appropriate integral solutions.

Chapter 5

An Online Model Capturing Gradually Improving Capacity Estimations

We now proceed to the second major topic of this thesis and conduct our study of online optimization in network lifetime and general linear packing problems. Motivated by the behavior of battery depletion in wireless sensor networks we introduce and analyze a novel *online model for general linear packing problems* that formalizes the aspect of gradually improving capacity estimations.

Though possibly not formulated in terms of a linear program and despite the fact that energy costs of wireless communication links behave in a non-linear fashion, network lifetime problems are representable as combinatorial packing problems which —when considering their inflated representation— behave linearly with respect to the capacity constraints. Using this more universal problem formulation allows us to exploit general properties of the underlying linear packing structure without changing the notion of algorithmic efficiency with respect to the commonly used concise non-linear representation.

In the following section we introduce and formally define our online model using general linear programming terminology. Subsequently, we characterize our notion of combinatorial packing problems in Section 5.2 before presenting a short survey on network lifetime problems in Section 5.3. In Chapters 6 and 7 we will then develop and evaluate several online algorithms for the proposed online model showing provable worst-case bounds on the competitiveness of these algorithms on general and special linear packing problems.

5.1 Introduction to the Online Model

In every-day life there are numerous occasions where one has to make decisions depending on the vague knowledge of parameters that are not known exactly and can only be roughly estimated. However, after some decisions have been taken, often the estimates get better and better so that later decisions can be based on better estimates.

This situation is familiar to every scientist who wants to explain a topic to an audience consisting of a number of different listeners with distinct backgrounds and temper. Some approaches of explanation may be more suitable for one part of the audience while others might prefer different ways.

At the beginning of your talk you perhaps already know some of your listeners very well while others could be complete strangers whose preferences might be difficult to assess. However, by carefully observing your audience you should be able to pick up some clues of how much of a persons patience for a given subtopic is left. Usually those hints become clearer the closer you get to overstrain someones endurance. At the latest, you know that the patience of a listener has been depleted when he furiously leaves the room.

If your goal is to provide as much information as possible before the first person leaves the audience, you perhaps should skip that detailed linear programming formulation when one of your listeners shows growing annoyance and instead spend more time on the intuition of your approach. On the other hand, this might seem verbose to others which like to see a

precise definition. After all, explanation can be more efficiently given when a formal model is defined. In hindsight, one could determine an optimal strategy, but in most cases winning your audience for a second attempt may not be an option.

The described problem is a typical example of an online problem where decisions under uncertainty have to be made which can not be undone later. It fits naturally into the online model defined in the following section. The original motivation for our study on gradually improving capacity estimations is also intrinsically online: once a transmission in a given instance of a network lifetime maximization problem has been executed, the limited power of the corresponding network nodes is irreversibly reduced by the required transmission energy.

While solving online problems in every-day life is based mostly on experience and expectation of future events, the predominant theoretical framework for studying online problems is given by competitive analysis (an extensive introduction and survey on this topic can be found in [BEY98]). Similar to worst-case analysis in classic complexity theory, the input is assumed to be chosen by a malicious adversary, to reflect the worst input the online algorithm can be exposed to when comparing its final performance to an optimal (offline) solution. Though this approach might sometimes be overly pessimistic, it can be used to derive absolute guarantees on the performance of specific algorithms and may also lead to general bounds on the competitiveness of any possible online algorithm in a given model.

In order to study the consequences of gradually improving capacity estimations we will introduce a new online model for general optimization problems that can be described in terms of linear packing programs, thereby including the classes of combinatorial packing and network lifetime problems. While in classic models for online packing (see Section 1.1.1) the capacities of the resources are assumed to be known and the packable objects together with their load requirements and profits are revealed in an online fashion, we consider the opposite case where the requirements and profits are known but the capacities are uncertain.

To the best of our knowledge, all formulations of network lifetime assume perfect a priori knowledge about the status and future behavior of the energy supplies of each single network node. In real-world applications, however, this seems an unrealistic assumption. The measured residual energy after a (partial) execution of the transmission scheme may deviate from the initially predicted ideal behavior. In addition to the non-perfect discharge characteristics of batteries (for a brief introduction see, e.g., [RVR03]), the energy budget can also be influenced by varying communication costs caused from temporal interference (e.g. by rain or transient noise). To exploit the updates on the residual energy levels during the execution of a transmission scheme, we introduce a new online model for general linear packing problems that can be used to increase the lifetime of wireless ad-hoc networks.

Abstracting from various technological details, we assume that at any time an upper and a lower bound on the remaining capacity of every resource can be obtained and these bounds deviate at most by a factor of α, for a given parameter $\alpha \geq 1$. This way, while the guarantee on the relative error regarding the remaining capacity stays constant, the estimations on the true, initial capacity get more and more accurate with depletion.

5.1.1 Formal Model for the Competitive Analysis

The proposed online model for packing problems with gradually improving estimations on the packing constraints considers general linear packing programs of the form $\max p^\top x$ subject to $Ax \leq c$ and $x \geq 0$, with $p \in \mathbb{R}^n_{\geq 0}$, $A \in \mathbb{R}^{m \times n}_{\geq 0}$, and $c \in \mathbb{R}^m_{\geq 0}$. The coefficients in the vector c are called *(true) capacities*.

In our model, there are two players, an *online algorithm* and an *adversary*. The algorithm aims to find a feasible solution to the packing program maximizing the objective function $p^\top x$.

It is equipped with the coefficients in A and p but only receives estimates on the capacities. The estimates are given in form of lower and upper bounds specified by the adversary.

In particular, there is a parameter $\alpha \geq 1$ that controls the quality of these estimates. The algorithm may always allocate a portion of the *remaining capacities*, i.e. of the capacities that have not been used previously. At the beginning of each allocation step the online algorithm receives an upper and a lower bound on the remaining capacities from the adversary. The only restriction is that the two bounds deviate at most by the factor α. Based on this imprecise information, the algorithm has to decide which amounts of the capacities shall be used in the current step.

We consider online algorithms that run in stages. Let N denote the number of stages. Initially, the online player starts with the solution $x^{(0)} = 0 \in \mathbb{R}^n_{\geq 0}$. It can increase the variables x from stage to stage. Each stage $s \in \{1, \ldots, N\}$ proceeds as follows:

- The adversary reveals bound vectors $\ell^{(s)}, u^{(s)} \in \mathbb{R}^m_{\geq 0}$ on the remaining slack such that for each constraint $i \in \{1, \ldots, m\}$ we have

$$\ell_i^{(s)} \leq c_i - (Ax^{(s-1)})_i \leq u_i^{(s)} \quad \text{and} \quad u_i^{(s)} \leq \alpha \cdot \ell_i^{(s)}.$$

- Based on p, A, $\ell^{(s)}$ and $u^{(s)}$ the online algorithm determines its additional allocations, i.e. it determines a $y^{(s)} \in \mathbb{R}^n_{\geq 0}$ and sets $x^{(s)} := x^{(s-1)} + y^{(s)}$.

We assume that the online algorithm has to guarantee that all solutions $x^{(1)}, \ldots, x^{(N)}$ are feasible. Note that we did not restrict the adversary in a way assuming that the most recently presented lower bound on the slack yields the best lower bound on the true capacity. Those presented in earlier stages might translate into better lower bounds on the true capacity.

To simplify the model, we assume w.l.o.g. that $u^{(s)} = \alpha \cdot \ell^{(s)}$ as by telling any better upper bound the adversary would provide more information than necessary. In other words, the adversary specifies only a lower bound vector satisfying $\ell_i^{(s)} \leq c_i - (Ax^{(s-1)})_i \leq \alpha \cdot \ell_i^{(s)}$. We will call such lower bounds *α-sharp*.

We evaluate online algorithms in terms of their *competitive ratio* [BEY98], i.e. the worst-case ratio between the objective value obtained by the online algorithm and the best possible objective value under the true constraints.

5.2 Combinatorial Optimization

The goal in combinatorial optimization is to deliver methods to find *optimal* solutions among a set of *combinatorial objects* given in an input instance. Thereby optimality refers to objects with maximum (resp. minimum) value which is typically measured by a real-valued mapping on the domain of the feasible objects.

What distinguishes combinatorial optimization problems from other (general) optimization problems is that the set of feasible objects is normally given by a finite set with a concise representation (see [PS82, Sch03]). We will call these sets and the contained objects *combinatorial*.

When studying the algorithmic complexity of a combinatorial problem, it is referred to the size of this representation rather than the size of the search space. Typically in many combinatorial optimization problems there is an exponential gap between the number of feasible structures and the size of their representation. Often the underlying structure of a combinatorial problem not only allows for a concise representation but also for efficient computation of an optimal solution (e.g. finding shortest paths in a graph structure). However, there are also many algorithmically intractable combinatorial problems (e.g. finding longest paths).

We will not impose any restriction to the representation of a combinatorial set other than that it should offer efficient methods to test feasibility and to compute the evaluation value of a combinatorial object.

▶ **Definition 5.1** (Combinatorial Minimization/Maximization Problem)**.** Given a combinatorial set $\mathcal{H}$ with an efficiently computable evaluation function $p\colon \mathcal{H} \to \mathbb{R}$. Then the corresponding *combinatorial minimization problem* $\mathcal{P} = \langle \mathcal{H}, p \rangle$ asks for finding an $H \in \mathcal{H}$ with $p(H) \leq p(H')$ for all $H' \in \mathcal{H}$. A combinatorial *maximization* problem is defined analogously with the condition $p(H) \geq p(H')$ for all $H' \in \mathcal{H}$.

Algorithmic complexity is measured with respect to $\text{size}(\langle \mathcal{H}, p \rangle)$ which is defined as the length of the encoding of the concise representation of $\mathcal{H}$ and p.

Formally, the above definition just describes *one instance* of a combinatorial optimization problem whereas normally we mean a whole collection of input instances sharing some common properties when speaking of optimization problems.

Simple Example: Minimum Weight Spanning Tree. A classic combinatorial optimization problem is the one of finding a minimum weight spanning tree.

For a given graph $G = (V, E)$ with edge weights $w\colon E \to \mathbb{R}_{\geq 0}$, let $\mathcal{H}$ be the set of all spanning trees of G. Note that the set $\mathcal{H}$ is implicitly defined by G, as the feasibility check "Is a given edge collection $H \subseteq E$ a spanning tree of G?" can easily be done in polynomial time with respect to $|G| = |V| + |E|$. The evaluation function defined by $\bar{w}(H) = \sum_{e \in H} w(e)$ for a spanning tree $H \in \mathcal{H}$ is efficiently computable with respect to the size of G and the encoding of w. Thus, the problem of finding $\arg\min_{H \in \mathcal{H}} \bar{w}(H)$ is an instance of a combinatorial minimization problem with representation $G = (V, E)$, w and $\text{size}(\langle \mathcal{H}, \bar{w} \rangle) = |V| + |E| + \sum_{e \in E} |w(e)|$.

The *minimum weight spanning tree problem* is the collection of instances of this form for arbitrary graphs G and weights w.

5.2.1 Total Load Minimization Problems

The minimum weight spanning tree problem is an example for a special subclass of combinatorial optimization problems which we will call *total load minimization problems*.

▶ **Definition 5.2** (Total Load Minimization Problem)**.** Given a set $\mathcal{H}$ of combinatorial objects and a set $\mathcal{R}$ of elements $R \in \mathcal{R}$ which we will call *resources*. Each resource $R \in \mathcal{R}$ is charged by the object $H \in \mathcal{H}$ depending on the efficiently computable positive load function $\ell_R\colon \mathcal{H} \to \mathbb{R}_{\geq 0}$. Additionally, each resource $R \in \mathcal{R}$ has an assigned *weight* $w(R)$ with $w\colon \mathcal{R} \to \mathbb{R}_{\geq 0}$.

The *(weighted) total load* $L(H)$ of an object $H \in \mathcal{H}$ is defined by the sum $L(H) = \sum_{R \in \mathcal{R}} w(R) \cdot \ell_R(H)$, and the problem of finding an object $H \in \mathcal{H}$ with minimum total load is the *total load minimization problem* $\langle \mathcal{H}, (\mathcal{R}, w), (\ell_R)_{R \in \mathcal{R}} \rangle$.

We define $\text{size}(\langle \mathcal{H}, (\mathcal{R}, w), (\ell_R)_{R \in \mathcal{R}} \rangle) = |\mathcal{R}| + \text{size}(\mathcal{H}) + \sum_{R \in \mathcal{R}} (|w(R)| + \text{size}(\ell_R))$ where $\text{size}(\mathcal{H})$ and $\text{size}(\ell_R)$ refer to the size of the concise representation as in Definition 5.1.

Total load minimization problems can play an important role when trying to solve network lifetime problems. In Section 5.3.4 we will see by general linear programming theory that (approximately) solving appropriate power minimization problems can be used to (approximately) solve the corresponding lifetime maximization problems and vice versa.

5.2.2 Linear Programming

In contrast to combinatorial optimization, *linear programming* asks for an optimal solution over an *infinite* domain of possible solutions. This domain of feasible solutions is defined by a set of *linear inequalities*, where a maximum/minimum solution with respect to a *linear evaluation function* has to be found (see Definition 5.3). When only integral values are allowed for the variables we speak of an *integer linear program (ILP)*.

While the definition of feasible LP-solutions is not combinatorial it can easily be seen that the subset defined by the vertices of the convex polytope represented by $Ax \leq c$ indeed is combinatorial. Also it contains an optimal solution with respect to the profit vector p whenever the polytope does, thus linear programming can be described as a combinatorial problem.

▶ **Definition 5.3** (Standard Linear Program)**.** The *standard maximization form of a linear program (LP)* of dimension $m \times n$ with variables $x \in \mathbb{R}^n$ is

$$\begin{aligned} \text{maximize} \quad & \sum_{j=1}^{n} p_j x_j & \\ \text{subject to} \quad & \sum_{j=1}^{n} a_{ij} x_j \leq c_i & \forall i \in [m] \\ & x_j \geq 0 & \forall j \in [n] \end{aligned}$$

or in short

$$\max p^\top x \text{ subject to } Ax \leq c, x \geq 0,$$

where $p = (p_j)_{j\in[n]}$ is called the *(linear) profit function*, $c = (c_i)_{i\in[m]}$ the vector of *capacities* and $A = (a_{ij})_{i\in[m],j\in[n]}$ the *constraint coefficient matrix*.

A linear *packing* program has the following more restricted form.

▶ **Definition 5.4** (Linear Packing Program)**.** A standard linear program

$$\max p^\top x \text{ subject to } Ax \leq c, x \geq 0,$$

with *positive* $A \in \mathbb{R}^{m\times n}_{\geq 0}$ and $p \in \mathbb{R}^n_{\geq 0}$ is called a *linear packing program* (or *packing LP*).

5.2.3 Combinatorial Packing Problems

Packing Problems in their most simple form ask —instead of just searching for an optimal element in a given set— to determine a subset of elements that maximize total profit while not exceeding certain *packing constraints*.

This can again result in a typical (non-packing) combinatorial optimization problem. For example, packing a maximum number of edges of a graph with the restriction that for any node at most one of its outgoing edges may be selected, results in the well-known maximum cardinality matching problem. The set of all matchings is a combinatorial set and, though the number of matchings can be exponential in the size of a graph, a maximum matching can be found in polynomial time.

More general packing problems may allow objects to be used multiple (possibly a fractional number of) times. The information on the selected objects in a proposed solution and their multiplicities will be called a *packing*.

▶ **Definition 5.5** (Packing)**.** A *packing* of a set S is a mapping $\mu\colon S \to \mathbb{R}_{\geq 0}$. We call $\mu(s)$ the *multiplicity* of $s \in S$ and a packing *integral* if $\mu(s) \in \mathbb{N}_0$ for every $s \in S$.

When speaking of *combinatorial packing problems*, we assume the domain of the packing to be combinatorial, thus tightening the requirements for algorithmic efficiency. In general, of course, the concise representation of the packing domain does not guarantee that the packing itself is efficiently representable.

▶ **Definition 5.6** (Combinatorial Packing Problem (CPP))**.** Given a combinatorial set $\mathcal{H}$ with an efficiently computable positive profit function $p\colon \mathcal{H} \to \mathbb{R}_{\geq 0}$ and a set $\mathcal{R}$ of resources $R \in \mathcal{R}$ with limited capacities $c\colon R \to \mathbb{R}_{\geq 0}$ which are charged by the objects $H \in \mathcal{H}$ depending on the efficiently computable positive load functions $\ell_R\colon \mathcal{H} \to \mathbb{R}_{\geq 0}$. Then the corresponding *combinatorial packing problem* $\mathcal{P} = \langle(\mathcal{H}, p), (\mathcal{R}, c), (\ell_R)_{R\in\mathcal{R}}\rangle$ asks for an optimal solution to the following linear program.

$$\begin{aligned} \text{maximize} \quad & \sum_{H\in\mathcal{H}} p(H)\cdot x_H & & \\ \text{subject to} \quad & \sum_{H\in\mathcal{H}} \ell_R(H)\cdot x_H \leq c(R) & & \forall R \in \mathcal{R} \\ & x_H \geq 0 & & \forall H \in \mathcal{H} \end{aligned}$$

We define $\text{size}(\langle(\mathcal{H}, p), (\mathcal{R}, c), (\ell_R)_{R\in\mathcal{R}}\rangle) = |\mathcal{R}| + \text{size}(\langle\mathcal{H}, p\rangle) + \sum_{R\in\mathcal{R}}(|c(R)| + \text{size}(\ell_R))$ and call the above linear packing program the *inflated representation* of a combinatorial packing problem.

When the variables x_H are required to be integral we speak of an *integral combinatorial packing problem (ICPP)*.

Though the inflated representation of a combinatorial packing problem can be useful for deriving structural properties, the algorithmic complexity will still be measured with respect to the concise representation of $(\mathcal{H}, p)$, $\mathcal{R}$, c, and $(\ell_R)_{R\in\mathcal{R}}$.

Note that the functions p and ℓ_R for $R \in \mathcal{R}$ are just general functions whose concise representations may not be linear at all.

5.2.3.1 Maximum Flow

One of the most basic combinatorial packing problems (and one of the problems considered in the competitive analysis for specific combinatorial packing problems in Chapter 7) is the task of finding a maximum flow in a capacitated graph between given source-sink pairs. We just define a flow as a packing of paths.

▶ **Definition 5.7** (Flow)**.** Given a graph $G = (V, E)$ with edge capacities $c\colon E \to \mathbb{R}_{\geq 0}$ and k-many source-sink pairs $(s_i, t_i)_{1\leq i\leq k}$. Let $\mathcal{H}_i$ be the set of all paths between s_i and t_i and $\mathcal{H}$ denote the union of all those sets.

A *flow* is a packing of paths contained in $\mathcal{H}$. It is *feasible* if for all edges $e \in E$ the sum of the multiplicities of paths containing e does not exceed its capacity $c(e)$.

We speak of *single-commodity flows* if only one source-sink pair (*commodity*) is given, opposed to *multi-commodity flows* for the case $k > 1$.

▶ **Definition 5.8** (Maximum Flow Problem)**.** Given a graph $G = (V, E)$ with edge capacities $c\colon E \to \mathbb{R}_{\geq 0}$ and k-many source-sink pairs $(s_i, t_i)_{1 \leq i \leq k}$.

The goal is to find a feasible flow of maximum total multiplicity.

Using the general terms from Definition 5.6 the combinatorial domain $\mathcal{H}$ of the maximum flow problem is the set of all source-sink paths with uniformly defined profit function $p(H) = 1$ for all $H \in \mathcal{H}$. The resources are the edges of the given graph with corresponding capacities while the load functions are defined by

$$\ell_e(H) = \begin{cases} 1 & \text{if edge } e \text{ is contained in path } H \\ 0 & \text{else.} \end{cases}$$

Aside from the inflated representation of this packing problem, the maximum flow problem is representable by the much more compact linear program

$$\begin{aligned}
\text{maximize} \quad & \sum_{i=1}^{k} \left(\sum_{\substack{e \in E \\ e=(s_i,v)}} x_{e,i} - \sum_{\substack{e \in E \\ e=(v,s_i)}} x_{e,i} \right) \\
\text{subject to} \quad & \sum_{i=1}^{k} x_{e,i} \leq c(e) && \forall e \in E \\
& \sum_{\substack{e \in E \\ e=(v,u)}} x_{e,i} - \sum_{\substack{e \in E \\ e=(u,v)}} x_{e,i} = 0 && \forall i \in [k] \quad \forall v \in V \setminus \{s_i, t_i\} \qquad (5.1) \\
& x_{e,i} \geq 0 && \forall e \in E \quad \forall i \in [k]
\end{aligned}$$

where Constraint 5.1 is called the flow conservation constraint.

Using the constraints above to define a feasible solution space, flows can alternatively be described by a function $f_i\colon E \to \mathbb{R}_{\geq 0}$ for each commodity $i \in [k]$. Note, however, that even when transformed to standard form the flow conservation constraint is not a valid constraint for a *packing* LP over the variables $x_{e,i}$ (requires positive $A \in \mathbb{R}_{\geq 0}^{m \times n}$).

The maximum flow problem has been extensively studied in numerous variants and there exists a vast variety of efficient algorithms for it. For a general survey see [AMO93] or [Sch03].

5.2.3.2 Packing Spanning Trees

Another classic combinatorial packing problem is the one of finding a maximum packing of spanning trees in a capacitated graph. It is a relevant subproblem in a broad class of network lifetime problems and can be efficiently solved even when the packing is required to be integral.

The definition of this packing problem is similar to the one of the maximum flow problem given above.

▶ **Definition 5.9** (Maximum Spanning Tree Packing Problem)**.** Let $\mathcal{H}$ be the set of all spanning trees in an undirected capacitated graph $G = (V, E)$ with edge capacities $c\colon E \to \mathbb{R}_{\geq 0}$.

Then the *maximum spanning tree packing problem* is the combinatorial packing problem induced by the inflated LP-representation

$$\begin{aligned} \text{maximize} \quad & \sum_{H \in \mathcal{H}} x_H \\ \text{subject to} \quad & \sum_{\substack{H \ni e \\ H \in \mathcal{H}}} x_H \leq c(e) && \forall e \in E \\ & x_H \geq 0 && \forall H \in \mathcal{H}\,. \end{aligned}$$

Efficient algorithms for this problem can be found in [Sch03].

5.3 Network Lifetime Maximization

We are now ready to formulate general network lifetime maximization in terms of a combinatorial packing problem.

Network lifetime maximization in general aims at sustaining a pre-specified communication as long as possible in a given communication network with limited power supplies. While the requirements of the communication are fixed, there may be numerous ways to establish it, each resulting in different energy demands.

Obviously, lifetime maximization requires a more global approach than just optimizing power efficiency, as always picking a communication topology with minimal energy demands may lead to an unnecessarily early depletion of some parts of the network which may be critical to support further communications. Since the lifetime of a network ends when the specified communication can no longer be sustained due to depletion of the given (limited) energy supplies, lifetime maximization naturally presents itself as a packing problem.

In the context of combinatorial packing problems, the limited energy supplies of the network nodes correspond to the resources and the communication topologies supporting the pre-specified communication requirements are described by a combinatorial set.

▶ **Definition 5.10** (Network Lifetime Maximization Problem)**.** Given a communication network $G = (V, E)$ with energy supplies at each network node $v \in V$ with capacities $c(v) \in \mathbb{R}_{\geq 0}$ and a communication requirement described by a combinatorial set $\mathcal{H}$ whose elements consume energy from the resources defined by the load functions $\ell_v\colon \mathcal{H} \to \mathbb{R}_{\geq 0}$ for each $v \in V$.

Find a maximum feasible packing to the combinatorial packing problem

$$\begin{aligned} \text{maximize} \quad & \sum_{H \in \mathcal{H}} x_H \\ \text{subject to} \quad & \sum_{H \in \mathcal{H}} \ell_v(H) \cdot x_H \leq c(v) && \forall v \in V \\ & x_H \geq 0 && \forall H \in \mathcal{H} \end{aligned}$$

in form of a *transmission schedule* (S, μ) with $S = \{H \in \mathcal{H} \mid x_H > 0\}$ and $\mu(s) = x_s$ for $s \in S$.

The related power efficiency problem of finding just one communication structure with minimal total power consumption is:

▶ **Definition 5.11** (Minimum Power Communication Problem)**.** Given a communication network $G = (V, E)$, a communication requirement described by a combinatorial set $\mathcal{H}$ whose elements consume energy from the network nodes defined by the load functions $\ell_v \colon \mathcal{H} \to \mathbb{R}_{\geq 0}$, and a linear weight coefficient $\lambda_v \in \mathbb{R}_{\geq 0}$ for each $v \in V$.

Find an element $H \in \mathcal{H}$ that minimizes $\sum_{v \in V} \lambda_v \cdot \ell_v(H)$.

One typical communication requirement in many network lifetime problems is to continuously being able to broadcast messages from a designated base node. So for a directed network $G = (V, E)$ with specified base node $b \in V$ the set $\mathcal{H}$ becomes just the set of all broadcast trees rooted at b in that case.

The load functions may depend on the applied transmission technology. However, there are only two relevant load functions in the literature on network lifetime maximization.

The standard case considered in most practical applications assumes the wireless network nodes to be radio transceivers with *omnidirectional antennas*. There, the radio signal is assumed to propagate in a disc/sphere shaped fashion, ideally having the same strength at all points in a specific distance from the sender.

Formally, let $w((u, v))$ denote the power needed for a transmission of node u to node v (which depends on the underlying network model, see Section 5.3.1), then, if transceiver u is sending with power p, the signal is received by all nearby neighbors v with $w((u, v)) \leq p$ in the omnidirectional antenna model. For given transmission power requirements $w(e) \in \mathbb{R}_{\geq 0}$ for each transmission link $e \in E$ this leads to load functions $\ell_u \colon \mathcal{H} \to \mathbb{R}_{\geq 0}$ with $\ell_u(H) = \max_{(u,v) \in E \cap H} w((u, v))$, which behave in a non-linear fashion with respect to the set of supported communication links by a node u.

In contrast to that, the load functions in the *unidirectional antenna* model are linear. Unidirectional antennas are assumed to support only communications from a sender to *one* specific receiver at a time, resulting in the load functions $\ell_u(H) = \sum_{(u,v) \in E \cap H} w((u, v))$.

5.3.1 Wireless Communication Networks

The study of network lifetime problems originated in the distinct requirements and growing importance of *wireless ad-hoc networks*.

On one side, these networks involve an implicit need for energy efficient communication protocols, since the devices participating in an ad-hoc setting are typically assumed to only have a limited energy supply (e.g. battery-driven sensor nodes). On the other side, the flexible structure of wireless communication networks allows for significant influence on the communication costs by an adaptive adjustment of transmission powers to the single network nodes.

In contrast to classic network analysis, communication links in wireless networks are not static. Since wireless network nodes normally use radio transceivers to transmit a message, in principle any node in the network can reach any other node by direct transmission (provided that the transmission power is large enough).

In the literature, there are numerous models to capture the characteristics of wireless networks, mainly covering two basic aspects: *path attenuation* and *interference*.

The power needed to directly transmit a single message from one transceiver to another is defined by the *path attenuation* (also called *path loss*). A common assumption is that in order to transmit a radio signal over a distance d the sender has to use a transmission power that is proportional to d^α (in practice typically $2 \leq \alpha \leq 4$).

Interference describes the disturbance of a signal caused by concurrent transmissions or background noise, for instance. There are numerous models with different consideration of specific fading characteristics, interference, noise, and signal occlusion.

To abstract from specific physical details, we assume that the power requirements are generally given in form of a weighted graph $G = (V, E, w)$, $w\colon E \to \mathbb{R}_{\geq 0}$, where the edge weights are set according to the underlying wireless network model, e.g., $w((u, v)) = d^\alpha$ when d is the (euclidean) distance between sender u and receiver v. This allows to express asymmetric power requirements $w(u, v) \neq w(v, u)$, forbidden/occluded communication links $(u, v) \notin E$, and global transmission limitations (e.g. by restricting the considered graph class to disc graphs).

Interference of concurrent transmissions is commonly disregarded in network lifetime problems as the concerned communication requirements are mainly stated as general topological constraints. Using multiplexing methods (e.g. TDMA) a topology can be decomposed into a sequence of single non-interfering transmission links. However, the CPP-formulation in Definition 5.10 does not exclude interference conditions, which could be encoded into the load functions $\ell_v\colon \mathcal{H} \to \mathbb{R}_{\geq 0}$.

An additional aspect that has impact on the energy cost of a specific transmission topology is the considered transceiver model. The complexity of the corresponding lifetime problem is largely influenced by the applied antenna type, chosen to be either *omnidirectional* or *(uni)directional*.

In practice, there are many other characteristics that are relevant to the power consumption in wireless networks. Some studies on network lifetime cover the aspect that not only sending but also receiving a signal draws energy from a transceiver. We will refer to this by the *receiver consumption* model.

5.3.2 Topology Requirements

Normally the considered communication requirements in network lifetime problems are described in form of connectivity constraints on the communication topologies that have to be packed.

One of the most basic requirements is to sustain a communication between a given pair of nodes, which we call the *unicast requirement*. In that case, the topologies to be packed have to contain a subgraph connecting these two nodes. With the goal of maximizing lifetime and thereby implicitly minimizing power, the set $\mathcal{H}$ of topologies can be restricted to the set of paths between the two nodes without ruling out relevant optimal solutions.

Generalizing this communication constraint we obtain the *multicast requirement* where a sender b has to reach a pre-specified set $V' \subseteq V$ of nodes instead of just one receiver. Aside from unicast, by setting $V' = V$, multicast covers another important special case, namely *broadcast*. For broadcast and general multicast the considered topologies in $\mathcal{H}$ are just all arborescences resp. steiner branchings rooted at the sending node.

Since we are considering directed graphs the above topology requirements can also be imposed with inversed directions. Then multicast becomes *gathering* and broadcast becomes *convergecast*, that is, the tree connecting b with the subset V' has to be directed towards b.

There are variants of network lifetime problems that are not entirely covered by Definition 5.10. One of the more prominent ones considers a severe restriction to the choice of feasible transmission schedules (S, μ). It requires S to be a singleton set, allowing only the static choice of a *single topology* to maximize lifetime. Opposed to this restriction (in literature sometimes referred to as *static*) the standard model according to Definition 5.10 will be called the *multi-topology* (or *dynamic*) model.

5.3.3 Integrality Aspects

Most studies on network lifetime (with only few exceptions, see [ELN+11]) consider the problem of finding optimal fractional transmission schedules (S, μ), allowing μ to be a probably non-integral valued function. This is attended by interpreting $\mu(s)$ to be the *duration* the communication topology s is supported.

Concerning practical implementations of lifetime maximization techniques, Park and Sahni [PS05] pointed out that in "a real application" the wireless network would have to perform an integral sequence of communications, rather than a fractional schedule.

This can be accomplished by trying to solve the integral formulation of the corresponding lifetime problem directly. As most problems become even harder when restricting to integral solutions there are only few attempts to do so. Besides the heuristics proposed in [PS05], the (to our knowledge) only theoretical analysis of solvability/approximability of integral network lifetime problems is performed in [ELN+11].

However, when the capacities of the resources are guaranteed to be sufficiently large, LP-rounding techniques can be used to derive good approximate integral solutions from corresponding fractional schedules. In Section 6.6.2 we will sketch an appropriate randomized rounding technique which transforms a fractional solution into an integral one —while losing at most an ϵ-fraction of the obtained objective value— in polynomial time even for exponential-sized LPs if the number of non-zero valued variables in the fractional solution is polynomially bounded.

5.3.4 Relation between Lifetime Maximization and Power Minimization

The following theorem establishes a direct correspondence between the approximability of a network lifetime problem and the corresponding power minimization problem which can be useful when designing approximation algorithms for those problems. It follows from general results in linear programming theory and polyhedral combinatorics.

▶ **Theorem 5.12.** *A network lifetime maximization problem with input network (V, E, c, w), communication structure set $\mathcal{H}$, and load-functions ℓ_v, $v \in V$, is f-approximable (for arbitrary node capacities) if and only if the corresponding minimum power communication problem is f-approximable (for arbitrary node weights).*

Proof. Follows by Theorem 5.13. ◀

While in the existing literature on network lifetime the first direction of this correspondence (approximate power minimization implies approximate lifetime maximization) was implicitly exploited to derive some positive results (e.g. [CKOZ03]), the only explicit mention can be found in [OY04]. In the context of network optimization the converse direction is commonly unaccounted for, which is why we provide a proof for this result for general combinatorial packing problems.

The equivalence of both problems is used in Section 5.3.5 to translate negative results from the minimum power communication problem to network lifetime maximization and thereby fill some open gaps when listing known bounds on the (offline) approximability of network lifetime problems.

In order to prove this result for general combinatorial packing problems we use the following definitions.

Given a combinatorial packing problem $\mathcal{P} = \langle(\mathcal{H}, p), (\mathcal{R}, c), (\ell_R)_{R\in\mathcal{R}}\rangle$ with inflated representation

$$\begin{aligned} \text{maximize} \quad & \sum_{H\in\mathcal{H}} p(H)\cdot x_H & & \\ \text{subject to} \quad & \sum_{H\in\mathcal{H}} \ell_R(H)\cdot x_H \le c(R) & \forall R\in\mathcal{R} & \qquad \text{(P)} \\ & x_H \ge 0 & \forall H\in\mathcal{H} & \end{aligned}$$

the dual to the above combinatorial packing program is the combinatorial covering problem

$$\begin{aligned} \text{minimize} \quad & \sum_{R\in\mathcal{R}} c(R)\cdot y_R & & \\ \text{subject to} \quad & \sum_{R\in\mathcal{R}} \ell_R(H)\cdot y_R \ge p(H) & \forall H\in\mathcal{H} & \qquad \text{(D)} \\ & y_R \ge 0 & \forall R\in\mathcal{R}\,. & \end{aligned}$$

For normalized load functions $\ell'_R(H) = \frac{\ell_R(H)}{p(H)}$ we define the corresponding total load minimization problem to be the problem of finding $\arg\min_{H\in\mathcal{H}} \sum_{R\in\mathcal{R}} w(R)\cdot\ell'_R(H)$ for arbitrary weights $w\colon \mathcal{R}\to\mathbb{R}_{\ge 0}$.

We say that load functions $\ell_R\colon \mathcal{H}\to\mathbb{R}_{\ge 0}$ are *efficiently invertible* if for a given vector of loads $(l_R)_{R\in\mathcal{R}}$ with $\forall R\in\mathcal{R} : l_R = \ell_R(H)$ for some $H\in\mathcal{H}$ an adequate H can be efficiently computed.

▶ **Theorem 5.13.** *A combinatorial packing problem is f-approximable (for arbitrary capacities) if and only if the corresponding total load minimization problem is f-approximable (for arbitrary weights) under the assumption of efficiently invertible load functions.*

Proof. We provide a proof following the sketch of Jain, Mahdian, and Salavatipour [JMS03] on the fractional Steiner tree packing problem which is based on a technique from Carr and Vempala [CV02]. Furthermore we use some ideas from [KV06].

First we show that f-approximability for the weighted load minimization problem implies the f-approximability of the packing problem.

load minimization ⇒ packing: In order to solve problem P we consider its dual D. Note that the covering constraints in D are equivalent to the condition that all $H\in\mathcal{H}$ have weighted total load of at least 1 (with respect to the normalized load functions ℓ') for resource weights y_R. Thus, any *exact* algorithm for the weighted load minimization problem could be used as a separation oracle finding violated constraints for D without having to check the (possibly exponential number of) inequalities one-by-one. If the algorithm returns an object H with weighted total load $\sum_{R\in\mathcal{R}} w(R)\cdot\ell'_R(H)$ smaller than 1 the corresponding constraint is trivially violated, if not, all constraints are satisfied because of the minimality of H. An f-approximation algorithm probably does not find all violated constraints, but if no object with weighted total load smaller than 1 is returned, the weighted load of each $H\in\mathcal{H}$ has to be at least $1/f$ thus a feasible solution can be achieved by scaling.

Let $\mathcal{A}$ be an f-approximation algorithm for the weighted load minimization problem. In order to approximate the optimal objective value of D we apply the ellipsoid method using $\mathcal{A}$ as an (approximate) separation oracle.

In a binary search a minimal value for B has to be found such that the linear inequalities in D together with

$$\sum_{R \in \mathcal{R}} c(R) \cdot y_R \leq B \tag{X1}$$

are still feasible. For each B the ellipsoid method starts with the initial ellipsoid containing the whole polytope defined by D+X1.

Let $\dot{y}$ be the center of the current ellipsoid. If $\dot{y}$ is infeasible we have to find a separating hyperplane whose corresponding constraint is violated by $\dot{y}$. Therefore X1 is checked with $\dot{y}$. If it is violated the ellipsoid is cut and shrunk by the corresponding hyperplane. Otherwise algorithm $\mathcal{A}$ is called. If $\mathcal{A}$ returns some $H' \in \mathcal{H}$ with weighted total load smaller than 1, the corresponding hyperplane is used for the cut. If an object of weighted total load at least 1 is returned the algorithm assumes $\dot{y}$ to be feasible. Though this might not be correct, the solution is at least *approximate feasible* as argued above.

When the binary search narrowed down a sufficiently small interval for which $B' - \epsilon$ is infeasible while B' is approximate feasible, we know that all basic solutions of D within this interval have the same objective value.

Let $\mathcal{H}' \subset \mathcal{H}$ be the set of separating hyperplanes acquired in the call of the ellipsoid method for $B' - \epsilon$ and let $\mathrm{D}(\mathcal{H}')$ denote the relaxed linear program D by restricting the constraints to the (polynomial sized) collection defined by $\mathcal{H}'$. Compute an optimal solution to $\mathrm{D}(\mathcal{H}')$ with the standard ellipsoid method. Let B^* denote the value of this optimum. We check (approximate) feasibility of this solution with respect to the whole set of inequalities in D by calling the ellipsoid method for D+X1 with B^* using $\mathcal{A}$ as a separation oracle. There are two cases:

- The call returns approximate feasibility. Then the optimum of D is at most f times the optimum of $\mathrm{D}(\mathcal{H}')$. Set $\mathcal{H}^* := \mathcal{H}'$.
- The call returns infeasibility for B^* certified by a set $\mathcal{H}^*$ of separating hyperplanes. For $B^* < B'$ we know that all basic solutions within the interval $[B' - \epsilon, B']$ have value B^*. Since this value is infeasible with respect to $\mathcal{H}^*$ the optimal solution of $\mathrm{D}(\mathcal{H}^*)$ would have an optimum greater than B'. If $B^* > B'$ this trivially holds.

In any case we obtain a polynomial sized set of separating hyperplanes $\mathcal{H}^*$ where the relaxed program $\mathrm{D}(\mathcal{H}^*)$ has an optimal solution which is at least $1/f$ times the optimal solution of D.

Also from strong duality it follows that there is a solution to P with the same objective value as $\mathrm{D}(\mathcal{H}^*)$ which assigns 0 to all variables $\{x_H \mid H \notin \mathcal{H}^*\}$. Since there is a feasible solution to D which is at most f-times larger, the optimum of P restricted to the variables $\{x_H \mid H \in \mathcal{H}^*\}$ is an f-approximation to the total optimum. Because this restricted linear program is of polynomial size its optimal solution can be computed in polynomial time.

packing ⇒ load minimization: In order to find for arbitrary given weights $w(R)$ an $H \in \mathcal{H}$ with total load $\sum_{R \in \mathcal{R}} w(R) \cdot \ell'_R(H) \leq f \cdot \arg\min_{H' \in \mathcal{H}} \sum_{R \in \mathcal{R}} w(R) \cdot \ell'_R(H')$ we search for the maximal γ such that the system of linear inequalities of D is violated for $y_R = w(R) \cdot \gamma$ in the call of an approximate separation oracle. By the assumption on the load functions the corresponding violated hyperplane can be used to determine an $H \in \mathcal{H}$ with the properties stated above.

We have to show that an f-approximate separation oracle can be given for D, that is, an oracle that for a given point $\dot{y}$ in the $|\mathcal{R}|$-dimensional space either returns a facet of the polytope defined by D which is violated or ensures that $f \cdot \dot{y}$ is contained in the polytope.

For this we consider the polytope Q congruent to the one defined by D but with $(0,\dots,0)$ as an interior point. Then its polar $Q^\circ = \{z \in \mathbb{R}^n \mid z^\top x \leq 1 \text{ for all } x \in Q\}$ also contains $(0,\dots,0)$ and Q equals $(Q^\circ)^\circ = \{x \mid z^\top x \leq 1 \text{ for all } z \in Q^\circ\}$ (see [KV06]). So in order to find a violated inequality of $Q = (Q^\circ)^\circ$ for a given point $\dot{y}$ it suffices to find a point z in Q° that (approximately) maximizes $\dot{y}^\top z$ and check the (approximate) feasibility. For this we must be able to optimize Q° in any given direction $\dot{y}$.

By the technique used in the first part of the proof, we can approximately optimize Q° when an approximate separation oracle for Q° is given. Again, this can be simulated by an algorithm which (approximately) optimizes Q in any given direction which is equivalent to (approximately) optimizing P for any given capacities. ◀

5.3.5 Short Survey on Known Results

The complexity of the general network lifetime problem is severely influenced by the considered antenna/power attenuation model, the feasible topologies, and the underlying graph structure.

Chang and Tassiulas [CT99, CT00] were the first to study the lifetime of wireless networks, defined as the time until the first node runs out of energy after satisfying several given communication requests. The communication requests in their model are given by a set of sensor nodes generating information which has to be routed to any of the specified destination nodes via unicast communication. They compared the lifetime of optimal solutions obtained by linear programming to several heuristics, one of which repeatedly uses communication paths with minimum total power requirements.

Not surprisingly, maximizing the network lifetime via the greedy selection of power minimal topologies (according to Definition 5.11) was shown to behave poorly on both worst-case and random instances. However, most of the early work [ČHE02, CH89, CHP$^+$02, KKKP97, LN01, MPF96, RM99, WCLF01, WNE00, WNE01] on energy efficient realizations of topological communication requirements in wireless networks concentrated on finding power minimal topologies —although the limited energy supplies of the single network nodes were explicitly mentioned as a key motivation for some of those studies [CHP$^+$02, LN01, RM99, WCLF01, WNE00, WNE01].

Revisiting the question on how well the power minimization of such a single communication topology extends the lifetime of a wireless network, Kang and Poovendran [KP03] defined the *static (single-topology) model* for the network lifetime problem. They showed that an optimal static lifetime schedule for *broadcast* with *omnidirectional antennas* can be computed in polynomial time by finding (for appropriately normalized edge weights) an arborescence minimizing the maximum edge weight.

At the same time, a general approach to solve the (dynamic) network lifetime problem according to Definition 5.10 was developed by Calinescu, Kapoor, Olshevsky, and Zelikovsky [CKOZ03]. They used a primal-dual method similar to the Garg-Könemann algorithm [GK98, GK07] to show that f-approximability of the minimum power communication problem implied $(1+\epsilon)f$-approximability of the corresponding network lifetime problem. As we observed in Section 5.3.4 this relation can be improved by general linear programming theory to *exact* correspondence, that is, network lifetime is f-approximable *if and only if* power minimization is f-approximable.

Calinescu et al. [CKOZ03] also gave an $O(\ln |V|)$-approximation algorithm for the minimum power broadcast problem by using methods for the node-weighted steiner tree problem. An asymptotically matching lower bound on the approximability of this problem was previously proven by a gap-preserving reduction from set cover [CCP$^+$01, ČHE02].

An examination of the predominant variants of the network lifetime problem under a

Minimum Power Communication

unidirectional antenna

Topology	Upper Bound	Lower Bound
Unicast	optimal equivalent to shortest path	optimal
Broadcast	optimal equivalent to minimum directed spanning tree	optimal
Multicast	equivalent to Directed Steiner Tree (thus at most $O(\|V\|^\epsilon)$ for any $\epsilon > 0$ by [CCC+98])	equivalent to Directed Steiner Tree (thus at least $\Omega(\ln \|V\|)$)
Convergecast	optimal equivalent to directed spanning tree	optimal

omnidirectional antenna

Topology	Upper Bound	Lower Bound
Unicast	optimal equivalent to shortest path	optimal
Broadcast	$O(\ln \|V\|)$ [CKOZ03, ČHE05]	$\Omega(\ln \|V\|)$ [CCP+01, ČHE02]
Euclidean metric	$(2.2d + 0.61)$-approximable for d dimensions and $\alpha \geq d$ [CFM07]	NP-hard [CCP+01]
Multicast	gap preserving reduction *to* DST reduction folklore, e.g., in [Lia02]	gap preserving reduction *from* DST indicated by [CKOZ03]
Convergecast	optimal (see unidirectional model)	optimal

Table 5.1 Best known bounds on the Minimum Power Communication Problem

unified algorithmic framework was done by Orda and Yassour [OY04, OY05]. They studied unicast, broadcast and general multicast for both omnidirectional and unidirectional antennas under the dynamic multi-topology model as well as the static single-topology model and contributed several of the bounds presented in Tables 5.2 and 5.3.

Orda and Yassour observed that, while the dynamic network lifetime problem is hard for most communication requirements in the omnidirectional antenna model and polynomial time solvable with unidirectional antennas, the opposite holds for the static case, i.e. lifetime with omnidirectional antennas is polynomial time solvable but hard with unidirectional antennas.

The exception here is unicast, of course, which is indifferent to the antenna model as only one receiver has to be considered for each direct transmission. So positive results from either antenna model can be carried over to the other one. Note that the same applies to convergecast/gathering topologies where the messages of multiple sources are routed to a single destination node. For each sending node only one outgoing transmission link has to be considered. Thus, convergecast (inverse broadcast) is optimally solvable in both the static and the dynamic case for any antenna model by a modification of the corresponding broadcast algorithms from [OY05, KP05].

Aside from the general graph version of network lifetime problems several special graph classes also received much attention. Especially the Euclidean distance metric for both its theoretical accessibility and its practical importance was intensively studied by the community [Amb05, ČHE02, ČHE05, CCP+01, CHP+02, CKOZ03, ELN+11, KKKP97, WCLF01]. In this model the wireless network is defined by a set of nodes distributed over a plane (or d-dimensional Euclidean space) with symmetric power requirements $w((u, v)) = w((v, u))$

Network Lifetime Multi-Topology

unidirectional antenna

Topology	Upper Bound	Lower Bound
Unicast	optimal max-flow/LP-formulation [OY05]	optimal
Broadcast	optimal [OY05] via packing arborescences	optimal
integral model	$(1, O(\log(n \cdot \mathrm{OPT})))$ using resource augmentation [ELN+11]	$\frac{1}{0}$ (unbounded) [ELN+11]
Multicast	same as Directed Steiner Tree *	as hard as Directed Steiner Tree *
Convergecast	optimal by a modification of broadcast	optimal

omnidirectional antenna

Topology	Upper Bound	Lower Bound
Unicast	optimal max-flow/LP-formulation [OY05]	optimal
Broadcast	$O(\ln \lvert V \rvert)$ [CKOZ03] via Garg-Könemann	$\Omega(\ln \lvert V \rvert)$ *
Euclidean metric	$(2.2d + 0.61)$-approximable *	NP-hard *
Multicast	same as Directed Steiner Tree *	as hard as Directed Steiner Tree *
Convergecast	optimal [KDN03] via packing arborescences	optimal

* bound from minimum power communication carries over due to Theorem 5.12

Table 5.2 Best known bounds on the Multi-Topology Network Lifetime Problem

Network Lifetime Single-Topology

unidirectional antenna

Topology	Upper Bound	Lower Bound
Unicast	optimal (translates to omnidirectional model)	optimal
Broadcast	-	NP-hard [OY05]
integral model	asymptotic $O(\log(n))$ for $\mathrm{OPT} \to \infty$ [ELN+11]	$\frac{1}{0}$ (unbounded) [ELN+11]
Multicast	-	NP-hard (generalization of broadcast)
Convergecast	optimal (see omnidirectional model)	optimal

omnidirectional antenna

Topology	Upper Bound	Lower Bound
Unicast	optimal (special case of multicast)	optimal
Broadcast	optimal [KP03, KP05]	optimal
Multicast	optimal [OY04] adapting [KP03]	optimal
Convergecast	optimal (e.g. [Seg08]) by trivial modification of broadcast	optimal

Table 5.3 Best known bounds on the Single-Topology Network Lifetime Problem

equal to the path loss $(d(u,v))^\alpha$ depending on the Euclidean distance $d(u,v)$ of the nodes u and v.

For minimum power broadcast Ambühl [Amb05] was able to provide the tight upper bound of 6 on the achievable approximation factor for the predominant minimum spanning tree heuristic in the (2-dimensional) Euclidean metric model. A smaller approximation factor of 4.2 can be achieved by a more sophisticated algorithm [CFM07] which also delivers good bounds on higher d-dimensional Euclidean spaces $(2.2d + 0.61)$. This results in improved bounds on the corresponding (dynamic) network lifetime problems as well.

Concerning practical implementations of methods for lifetime maximization, Park and Sahni [PS05] pointed out that in "a real application" the wireless network would have to perform an integral sequence of communications rather than a fractional schedule. They compared several heuristics in order to sequentially select power efficient broadcast trees.

More recently, Elkin, Lando, Nutov, Segal, and Shpunging [ELN+11] revisited the concept of integral network lifetime and developed approximation algorithms for several Euclidean broadcast and convergecast variants. They indicate that the integral broadcast lifetime problem for unidirectional antennas (for both the dynamic and the static model) is not approximable within any factor. To counter this, they consider approximation algorithms with *resource augmentation* (notion analogous to the online model introduced in [KP00]): they provide an algorithm that achieves the lifetime of an optimal schedule while violating the battery capacities by a factor of at most $O(\log(n))$ in the single-topology case and a factor of $O(\log(n \cdot t))$ in the multi-topology case, where t is the lifetime (number of broadcast operations) of an optimal solution. By scaling, this gives an *asymptotic* $O(\log(n))$-approximation-algorithm (in the meaning of [BEY98]) for the static/single-topology model.

Chapter 6

Competitive Analysis of the Online Model

In the following chapters we analyze the online model defined in Chapter 5 with respect to the generally achievable competitive ratio.

We begin (in Section 6.1) with evaluating a natural greedy approach: In each stage the algorithm GREEDY computes the best allocation with respect to the currently known lower bounds on the capacities. We show that this algorithm, which at first might look promising, only achieves the trivial competitive ratio of $\Theta(\frac{1}{\alpha})$.

In Section 6.2 we introduce an algorithm called EQUITABLE as it uses only an equitable portion of the capacities in each stage. Our main result in this chapter shows that its competitive ratio is at least $(1-\epsilon)\frac{\ln(\alpha)}{\alpha-1}$ on general packing problems if the algorithm runs for $N \geq \frac{\alpha-1}{\alpha^{\epsilon}-1}$ stages. This proves that the algorithm is at least $\frac{\ln(\alpha)}{\alpha}$-competitive already for a small number of stages, i.e. $N \geq \alpha^2/\ln(\alpha)$. In the context of network lifetime problems this is a favorable property as the efficiency gain is not overshadowed by the communication overhead needed to implement a multistage approach. In addition, we give an upper bound for the competitive ratio of EQUITABLE showing that our analysis is tight.

The presented framework is applicable to all problems that can be formulated in terms of a linear packing program and thereby especially covers all combinatorial packing problems and network lifetime problems according to their respective definitions given in Chapter 5. In fact, in every stage existing offline algorithms can be used as a black box without any modification. The actual LP formulation is only needed in the analysis. For hard problem variants where the underlying packing LPs are only approximately solvable, the competitiveness of our framework just decreases by the approximation factor of the black box algorithm. Those and related properties will be discussed in Section 6.6. If integral LP-solutions are prefered over fractional ones, we show that (under reasonable conditions on the capacities) in each stage of the online algorithm, fractional solutions can be converted to integral ones via a randomized rounding technique similar to [Sri99].

In Section 6.3 an algorithm related to EQUITABLE but with a more aggressive resource allocation procedure is introduced. The same lower bound as for EQUITABLE can be shown for this algorithm called ENVISAGE which, however, could possibly exhibit a strictly better worst-case behavior.

To complement our results, in Section 6.4, we analyze the limits of online algorithms using gradually improving capacity estimations. We prove an upper bound on the competitive ratio of any deterministic online algorithm showing that there does not exist a deterministic algorithm achieving a competitive ratio of $\frac{3}{\sqrt{\alpha}}$ or better (this upper bound will also be generalized in Chapter 7 to cover randomized online algorithms).

All upper bounds on the competitiveness of online algorithms in this chapter are shown for a simple problem class called $(1,k)$-Pack, which is defined by a small LP with only two constraints and two variables (see Definition 6.1 below). The simple structure of $(1,k)$-Pack allows the negative competitiveness results to carry over to many other problem classes as well. We will perform a competitive analysis on some classic combinatorial packing problems in Chapter 7.

▶ **Definition 6.1** ($(1,k)$-Pack)**.** For a fixed number k an instance of the $(1,k)$-Pack problem is specified by two capacities c_1 and c_2. The goal is to find an optimum solution to the following linear program.

$$\begin{aligned} \text{maximize} \quad & x_1 + x_2 \\ \text{subject to} \quad & k \cdot x_1 + x_2 \leq c_1 \\ & x_1 + k \cdot x_2 \leq c_2 \\ & x_1, x_2 \geq 0 \end{aligned}$$

For the simplistic $(1,k)$-Pack problem we will see that the upper bound of $O(\frac{1}{\sqrt{\alpha}})$ on the competitive ratio of any deterministic online algorithm is indeed asymptotically tight. This will be shown in Section 6.5 by constructing an algorithm that achieves a competitive ratio of at least $\frac{1}{\sqrt{\alpha}}$ on this special problem class (while, however, being non-competitive in general).

6.1 The GREEDY-Algorithm

The naive algorithm for solving a packing problem $\mathcal{P}$ of the form $\max p^\top x$ subject to $Ax \leq c$, $x \geq 0$, in our model works as follows. Take the initial lower bounds $l^{(1)}$ on the slack vector and solve the (completely known) linear packing problem $\mathcal{P}(l^{(1)})$ defined as $\max p^\top x$ subject to $Ax \leq l^{(1)}$, $x \geq 0$. Since the optimal value of a packing LP is linear in the constraint capacities and since the components of $l^{(1)}$ are at most a factor of α smaller than the true capacities c, this algorithm is $\frac{1}{\alpha}$-competitive.

▶ **Observation 6.2.** *The naive optimal one-stage exhaustion of lower bound capacities is exactly $\frac{1}{\alpha}$-competitive in the worst-case.*

Of course it is better to iterate this behavior. The GREEDY-algorithm uses the lower bounds on the resulting slacks again as new capacities for the considered offline problem, i.e. it solves the problem $\mathcal{P}(l^{(2)})$, and repeats this until the remaining capacities do not allow any further increase of the profit.

Algorithm GREEDY

Input: Online packing LP $\mathcal{P}$ of the form $\max p^\top x$ subject to $Ax \leq c$, $x \geq 0$, with unknown c.

1. Set $s := 0$ and $x^{(0)} = y^{(0)} := 0 \in \mathbb{R}^n_{\geq 0}$.
2. Repeat
   a. Increase the step number $s := s + 1$.
   b. Probe the lower bounds $\ell^{(s)} = (\ell_1^{(s)}, \ldots, \ell_m^{(s)})$ on the remaining capacities.
   c. Compute an optimal solution $y^{(s)}$ for the (offline) linear program $\max p^\top x$ subject to $Ax \leq \ell^{(s)}$, $x \geq 0$, and set $x^{(s)} := x^{(s-1)} + y^{(s)}$.

   until $y^{(s)} = 0$.
3. Output solution $x^{(s)}$.

As shown by the following lemma, the algorithm can be tricked into performing badly even on the simple $(1,k)$-Pack problem.

▶ **Lemma 6.3.** *GREEDY is less than $\frac{2}{\alpha}$-competitive in the worst-case.*

Proof. Consider an instance of $(1,k)$-Pack according to Definition 6.1. By summing the two constraints it is clear that the profit of any solution to this LP is at most $(c_1 + c_2)/(k+1)$.

Let the initial lower bounds be $\ell^{(1)} = (1; 1)$. In the first stage GREEDY computes a solution with profit at most $2/(k+1)$ and blocks capacity equal to 1 on both constraints. At the beginning of the second stage the lower bound on the remaining slacks turns out to be $\ell^{(2)} = (0; \alpha - 1)$. Note that this also determines the original capacities to be $c = (1; \alpha)$. As the first constraint is already tight, GREEDY is not able to perform any further improvements and stops with a profit of at most $2/(k+1)$. For $k = \alpha$ the optimal offline solution achieves a profit of exactly 1, which results in the claimed worst-case factor. ◀

6.2 The EQUITABLE-Algorithm

Now we will give an algorithm which is $\frac{\ln(\alpha)}{\alpha}$-competitive (after a polynomial number of stages) for every problem that can be formulated as a linear packing program. It avoids the drawbacks of GREEDY by restricting itself in each stage to a budget which ensures that in each forthcoming stage the solution can only improve.

Algorithm EQUITABLE

Input: Online packing LP $\mathcal{P}$ of the form $\max p^\top x$ subject to $Ax \leq c$, $x \geq 0$, with unknown c.

1. Fix the number N of stages to be large enough (e.g. $N := \lceil \sqrt{\alpha} \rceil + 1$).
2. Set $x^{(0)} = y^{(0)} := 0 \in \mathbb{R}^n_{\geq 0}$ and $\bar{\ell}^{(0)} := 0 \in \mathbb{R}^m_{\geq 0}$.
3. For $s := 1$ to N do
   a. Probe the lower bounds $\ell^{(s)} = (\ell_1^{(s)}, \ldots, \ell_m^{(s)})$ on the remaining capacities.
   b. For each constraint j set $\bar{\ell}_j^{(s)} := \max\{\ell_j^{(s)},\ \bar{\ell}_j^{(s-1)} - (Ay^{(s-1)})_j\}$ to be the best known lower bound on the remaining slack.
   c. Define the current capacity budget for stage s to be

$$b^{(s)} := \frac{\bar{\ell}^{(s)}}{N - (s-1)} \,.$$

   d. Compute an optimal solution $y^{(s)}$ for the (offline) linear program $\max p^\top x$ subject to $Ax \leq b^{(s)}$, $x \geq 0$, and set $x^{(s)} := x^{(s-1)} + y^{(s)}$.
4. Output solution $x^{(N)}$ or —if wanted— use the remaining slack with GREEDY.

Notice that for achieving the competitive ratio it is not necessary to use the remaining slack after finishing the fixed number of stages $1, \ldots, N$. As we will see later, there are worst-case instances where no usable slack remains.

The definition of $b^{(s)}$ is deliberately chosen to guarantee monotonicity in the vector components as will be shown in the proof of Lemma 6.4. Furthermore, since the adversary might return new bounds on the remaining capacities that are considerably worse than what can be estimated from the last allocation, we have to keep track of the currently best known lower bounds which is done by the vector $\bar{\ell}$.

6.2.1 Competitiveness of EQUITABLE

As before, when given a packing problem $\mathcal{P}$ of the form $\max p^\top x$ subject to $Ax \leq c$ and $x \geq 0$, we denote by $\mathcal{P}(c')$ the problem induced by substituting the capacities c by the vector c'. Furthermore, $\mathcal{P}^{\mathrm{OPT}}(c')$ shall denote the optimal objective value of $\mathcal{P}(c')$.

By linearity of the objective function the output $x := x^{(N)}$ computed for problem $\mathcal{P}$ by the online algorithm EQUITABLE has value $\mathcal{P}^{\mathrm{ALG}} := \sum_{s=1}^{N} \mathcal{P}^{\mathrm{OPT}}(b^{(s)})$.

As the lower bounds provided in every stage may deviate from the true slacks by a factor of α, it might first seem impossible to obtain a better competitive ratio than GREEDY. However, by allowing EQUITABLE only to use a particular fraction of the remaining capacities, we can show that these budgets are monotonically increasing from stage to stage. Hence, the marginal profits obtained in every stage are also increasing. Moreover, this restrictive behavior guarantees that EQUITABLE, in contrast to GREEDY, cannot be tricked into getting stuck too early and enables the algorithm to correct previous allocations to a certain extent when the estimates become better in later stages.

▶ **Lemma 6.4.** *The marginal profit computed by EQUITABLE in stage $s+1$ can be lower bounded by*

$$\mathcal{P}^{OPT}(b^{(s+1)}) \geq \frac{\mathcal{P}^{OPT}}{s + \alpha \cdot (N-s)}.$$

Proof. First, let us observe that the vector $b^{(s)}$ monotonically increases in every component: By definition we know $\bar{\ell}^{(s+1)} \geq \bar{\ell}^{(s)} - Ay^{(s)}$ and since $Ay^{(s)} \leq b^{(s)}$ holds we get

$$b^{(s+1)} = \frac{\bar{\ell}^{(s+1)}}{N-s} \geq \frac{\bar{\ell}^{(s)} - b^{(s)}}{N-s} = b^{(s)}.$$

The key to comparing the overall optimal solution with the partial solutions achieved in every stage of the algorithm lies in the simple fact that $Ax^{(s)} + \alpha \cdot \ell^{(s+1)}$ provides an upper bound on c. Furthermore, $Ax^{(s)}$ does not exceed $\sum_{u=1}^{s} b^{(u)}$ which by monotonicity of the $b^{(u)}$ is smaller than $s \cdot b^{(s)}$.

Combining these facts we get

$$\begin{aligned}\mathcal{P}^{\mathrm{OPT}}(c) &\leq \mathcal{P}^{\mathrm{OPT}}(Ax^{(s)} + \alpha \cdot \ell^{(s+1)}) \\ &\leq \mathcal{P}^{\mathrm{OPT}}(s \cdot b^{(s+1)} + \alpha \cdot \bar{\ell}^{(s+1)}) = \mathcal{P}^{\mathrm{OPT}}((s + \alpha \cdot (N-s)) \cdot b^{(s+1)}) \,.\end{aligned}$$

Because of the linearity of the packing problem this proves the lemma. ◀

The bound of Lemma 6.4 significantly increases in later stages. Exploiting this, we get the following result.

▶ **Theorem 6.5.** *The profit achieved by algorithm EQUITABLE with a total number of $N \geq \frac{\alpha-1}{\alpha^{\epsilon}-1}$ stages is at least*

$$\mathcal{P}^{ALG} \geq (1-\epsilon) \cdot \frac{\ln(\alpha)}{\alpha - 1} \cdot \mathcal{P}^{OPT} \,.$$

Proof. For each stage we add the bounds given by Lemma 6.4,

$$\mathcal{P}^{\mathrm{ALG}} = \sum_{s=0}^{N-1} \mathcal{P}^{\mathrm{OPT}}(b^{(s+1)}) \geq \sum_{s=0}^{N-1} \frac{\mathcal{P}^{\mathrm{OPT}}}{s + \alpha \cdot (N-s)} \,.$$

We separate the first summand and interpret the rest as a Riemann sum:

$$\begin{aligned}\frac{\mathcal{P}^{\mathrm{ALG}}}{\mathcal{P}^{\mathrm{OPT}}} &\geq \frac{1}{\alpha N} + \int_{x=0}^{N-1} \frac{1}{x + \alpha \cdot (N-x)} \,\mathrm{d}x \\ &= \frac{1}{\alpha N} + \frac{\ln(\alpha N) - \ln(N-1+\alpha)}{\alpha - 1} \\ &= \frac{\ln(\alpha) - \ln(1 + \frac{\alpha-1}{N})}{\alpha - 1} + \frac{1}{\alpha N} \,.\end{aligned}$$

Since for $N \geq \frac{\alpha-1}{\alpha^{\epsilon}-1}$ we have $\ln(1 + \frac{\alpha-1}{N}) \leq \epsilon \ln(\alpha)$, the claim follows. ◀

Not surprisingly, with an increasing number of stages the performance guarantee of the algorithm improves and converges to $\frac{\ln(\alpha)}{\alpha-1} \cdot \mathcal{P}^{\mathrm{OPT}}$. Even for small $\alpha > 1$ this gives a strict improvement to the greedy algorithm, since $\frac{\ln(\alpha)}{\alpha-1} > \frac{1}{\alpha}$.

Furthermore, Theorem 6.5 shows that already for a small number of steps, e.g., $N \geq \sqrt{\alpha}+1$, the algorithm is $\Theta(\frac{\ln(\alpha)}{(\alpha-1)})$-competitive. For a competitive ratio strictly greater than $\frac{\ln(\alpha)}{\alpha}$ (i.e. $\epsilon \leq 1/\alpha$) it is sufficient to have a total number of $N \geq \alpha^2/\ln(\alpha)$ many stages.

In the next section we will see that the bound of Theorem 6.5 is asymptotically tight.

6.2.2 Upper Bound on the Competitiveness of EQUITABLE

We will give an instance of $(1, k)$-Pack where the competitiveness of the algorithm is bounded by $O(\frac{\ln(\alpha)}{\alpha})$ for an arbitrary number of stages, even when the remaining slack after finishing stage N is optimally allocated. The following lemma will generally be useful when proving upper bounds for EQUITABLE.

▶ **Lemma 6.6.** *Consider the EQUITABLE algorithm with a total number of N stages. Let $q := \frac{\alpha-1}{\alpha} \cdot N$. When the lower bounds $\ell^{(1)}, \ldots, \ell^{(s)}$ are non-increasing, then the total profit of the algorithm achieved up to any stage $s \leq \lceil q \rceil$ is bounded by $\mathcal{P}^{ALG \leq s} \leq (1+\ln(\alpha)) \cdot \mathcal{P}^{OPT}(\ell^{(1)})$.*

Proof. By linearity of the problem and by definition of the algorithm we get

$$\mathcal{P}^{\mathrm{ALG} \leq s} = \sum_{i=1}^{s} \mathcal{P}^{\mathrm{OPT}}(b^{(i)}) = \sum_{i=0}^{s-1} \mathcal{P}^{\mathrm{OPT}}\left(\frac{\bar{\ell}^{(i+1)}}{N-i}\right) .$$

Now we use the fact, that by monotonicity of the lower bound vectors $\ell^{(1)} \geq \bar{\ell}^{(i)}$ holds for all $i \leq s$. So the sum of the profits in every stage can be bounded from above by

$$\mathcal{P}^{\mathrm{ALG} \leq s} \leq \mathcal{P}^{\mathrm{OPT}}(\ell^{(1)}) \cdot \sum_{i=0}^{s-1} \frac{1}{N-i} \leq \mathcal{P}^{\mathrm{OPT}}(\ell^{(1)}) \cdot (1 + \sum_{i=0}^{s-2} \frac{1}{N-i}) .$$

Interpreting the sum as a Riemann sum and using $s - 2 \leq q - 1$ we get

$$\begin{aligned} \mathcal{P}^{\mathrm{ALG} \leq s} &\leq \mathcal{P}^{\mathrm{OPT}}(\ell^{(1)}) \cdot (1 + \int_{x=0}^{q} \frac{1}{N-x} \mathrm{d}x) \\ &= \mathcal{P}^{\mathrm{OPT}}(\ell^{(1)}) \cdot (1 + \ln(N) - \ln(N-q)) . \end{aligned}$$

Substituting q by its defined value we get the claimed result. ◀

▶ **Theorem 6.7.** *EQUITABLE is not more than $O(\frac{\ln(\alpha)}{\alpha})$-competitive for the $(1, k)$-Pack problem.*

Proof. Consider and instance of $(1, k)$-Pack with $k = \alpha$. Let us fix $c_1 = 1$ and $c_2 = \alpha$ and define the corresponding lower bounds as functions in the constraint usage $\lambda_j = (Ax)_j$.

$$\ell_1(\lambda) = 1 - \lambda \quad \text{for } \lambda \leq 1 \qquad\qquad \ell_2(\lambda) = 1 - \frac{\lambda}{\alpha} \quad \text{for } \lambda \leq \alpha$$

Clearly these lower bounds are α-sharp (i.e. $\ell_i^{(s)} \leq c_i - (Ax^{(s)})_i \leq \alpha \cdot \ell_i^{(s)}$ in any stage s) and non-increasing, so we can use Lemma 6.6 to get

$$\sum_{s=1}^{\lceil q \rceil} \mathcal{P}^{\mathrm{OPT}}(b^{(s)}) \leq (1 + \ln(\alpha)) \cdot \mathcal{P}^{\mathrm{OPT}}(\ell^{(1)}) \qquad \text{where } q = \frac{\alpha - 1}{\alpha} \cdot N .$$

The profit of any solution x to $(1,k)$-Pack can be expressed in terms of Ax by $\frac{(Ax)_1+(Ax)_2}{k+1}$. So the budgets $b^{(s)}$ in each stage s will always be exhausted by an optimal solution as otherwise there would be a solution with strictly greater profit in this stage. Also, since $k=\alpha$, the profit $\mathcal{P}^{\mathrm{OPT}}(\ell^{(1)})$ is bounded by $2/(\alpha+1)$.

As the algorithm guarantees monotonicity of the budget vectors we know $b_1^{(s)} \geq b_1^{(1)} = 1/N$ for all $1 \leq s \leq N$. Then after $\lceil q \rceil$ many stages, the slack $c_1 - (Ax)_1$ on the first constraint has dropped below $1/\alpha$, limiting the overall additional profit from stage $\lceil q \rceil + 1$ on to at most this value. By putting this all together we get

$$\begin{aligned}\mathcal{P}^{\mathrm{ALG}} &= \sum_{s=1}^{\lceil q \rceil} \mathcal{P}^{\mathrm{OPT}}(b^{(s)}) + \sum_{s=\lceil q \rceil+1}^{t} \mathcal{P}^{\mathrm{OPT}}(b^{(s)}) \\ &\leq (1+\ln(\alpha)) \cdot \frac{2}{\alpha+1} + \frac{1}{\alpha} \\ &\leq \frac{1}{\alpha} \cdot (3 + 2 \cdot \ln(\alpha))\end{aligned}$$

while an optimal offline solution has profit 1. Hence EQUITABLE is $\Theta(\frac{\ln(\alpha)}{\alpha})$-competitive. ◀

6.3 The ENVISAGE-Algorithm

An algorithm similar to EQUITABLE but with a more aggressive usage of the residual capacities is considered in the current section. The algorithm aims to exhaust exactly $\frac{1}{N}$-th of the true capacities in each stage. When the algorithm recognizes that by the worst-case estimation of the lower bounds not the complete envisaged budget was used, it immediately allows to exhaust the surplus in the subsequent stage.

Algorithm ENVISAGE

Input: Online packing LP $\mathcal{P}$ of the form $\max p^\top x$ subject to $Ax \leq c$, $x \geq 0$, with unknown c.

1. Fix the number N of stages to be large enough.
2. Set $x^{(0)} = y^{(0)} := 0 \in \mathbb{R}^n_{\geq 0}$ and $L^{(0)} = 0 \in \mathbb{R}^m_{\geq 0}$.
3. For $s := 1$ to N do
   a. Probe the lower bounds $\ell^{(s)} = (\ell_1^{(s)}, \ldots, \ell_m^{(s)})$ on the remaining capacities.
   b. For each constraint j set $L_j^{(s)} := \max\{L_j^{(s-1)}, \ell_j^{(s)} + (Ax^{(s-1)})_j\}$ to be the best known lower bound on c_j.
   c. Define the current capacity budget for stage s to be

$$b^{(s)} := \frac{s}{N} \cdot L^{(s)} - Ax^{(s-1)} \ .$$

   d. Compute an optimal solution $y^{(s)}$ for the (offline) linear program $\max p^\top x$ subject to $Ax \leq b^{(s)}$, $x \geq 0$, and set $x^{(s)} := x^{(s-1)} + y^{(s)}$.
4. Output solution $x^{(N)}$ (if wanted use the remaining slack with GREEDY).

As for EQUITABLE we can bound the marginal profit made by ENVISAGE in each step. While the bound given below is the same as for the EQUITABLE algorithm the proof is different from the one of Lemma 6.4 as the budgets do not behave monotonically. However, as we are only interested in the sum of the already used budgets, it suffices to ensure monotonicity in average.

▶ **Lemma 6.8.** *We can bound the marginal profit computed by ENVISAGE in stage $s+1$ by*

$$\mathcal{P}^{OPT}(b^{(s+1)}) \geq \frac{\mathcal{P}^{OPT}}{s + \alpha \cdot (N - s)} .$$

Proof. By definition of $L^{(s)}$ we have

$$c \leq \alpha \ell^{(s)} + Ax^{(s-1)} \leq \alpha(L^{(s)} - Ax^{(s-1)}) + Ax^{(s-1)}$$

where the first inequality follows from the α-sharpness of any feasible lower bound.

We exploit the fact that the lower bounds L are monotonically increasing. Obviously for any stage s we have

$$L^{(s)} \leq N \cdot b^{(s)}$$

since we can bound $Ax^{(s-1)}$ in $b^{(s)} = \frac{s}{N} L^{(s)} - Ax^{(s-1)}$ by

$$Ax^{(s-1)} \leq Ax^{(s-2)} + b^{(s-1)} = \frac{s-1}{N} L^{(s-1)} \leq \frac{s-1}{N} L^{(s)} .$$

This directly translates into an upper bound on $Ax^{(s-1)}$ in terms of $b^{(s)}$ as

$$Ax^{(s-1)} \leq \frac{s-1}{N} L^{(s)} \leq (s-1) \cdot b^{(s)} .$$

In order to replace $L^{(s)}$ and $Ax^{(s-1)}$ by their respective upper bounds we have to get rid of the negative occurrence of $Ax^{(s-1)}$ by substituting $-Ax^{(s-1)}$ by $b^{(s)} - \frac{s}{N} L^{(s)}$. With this we can derive the bound

$$\begin{aligned} c \leq \alpha(L^{(s)} - Ax^{(s-1)}) + Ax^{(s-1)} &= \alpha(b^{(s)} + \frac{N-s}{N} L^{(s)}) + Ax^{(s-1)} \\ &\leq \alpha(b^{(s)} + (N-s)b^{(s)}) + (s-1)b^{(s)} \end{aligned}$$

thus proving

$$\mathcal{P}^{\mathrm{OPT}}(c) \leq \mathcal{P}^{\mathrm{OPT}}((\alpha \cdot (N - (s-1)) + (s-1)) \cdot b^{(s)})$$

for any $2 \leq s \leq N$ (for $s = 1$ this holds trivially). ◀

This directly implies the same lower bound on the competitiveness as shown for algorithm EQUITABLE.

▶ **Theorem 6.9.** *The profit achieved by algorithm ENVISAGE with a total number of $N \geq \frac{\alpha-1}{\alpha^{\epsilon}-1}$ stages is at least*

$$\mathcal{P}^{ALG} \geq (1-\epsilon) \cdot \frac{\ln(\alpha)}{\alpha - 1} \cdot \mathcal{P}^{OPT} .$$

Proof. Same as in Theorem 6.5. ◀

Thus, we can conclude that ENVISAGE is asymptotically at least as good as EQUITABLE as the competitiveness of EQUITABLE is (by Theorem 6.7) limited to $\Theta(\frac{\ln(\alpha)}{\alpha})$. It may, however, be possible that ENVISAGE achieves a better competitive ratio. Up to now, no worst-case instance for ENVISAGE with a competitive ratio smaller than the general bound of $O(\frac{1}{\sqrt{\alpha}})$ for all deterministic online algorithms could be given. Note that Lemma 6.6 is not applicable to ENVISAGE as the budgets can increase much faster.

6.4 An $O(1/\sqrt{\alpha})$ Upper Bound for Deterministic Online Algorithms

So far, the competitive ratios shown for the proposed online algorithms might seem very poor and one could hope to reach much better competitive ratios by the construction of more sophisticated online algorithms. However, as we will see in the current section, any online algorithm suffers from a non-constant loss in the competitive ratio even on the very simple problem class of $(1, k)$-Pack.

▶ **Theorem 6.10.** *Any deterministic online algorithm is at most $\frac{3}{\sqrt{\alpha}}$-competitive.*

Proof. Consider the $(1, k)$-Pack problem for $k = \sqrt{\alpha}$ and lower bound functions

$$\ell'(\lambda) = \begin{cases} \sqrt{\alpha} - \lambda & \text{for } 0 \leq \lambda < \sqrt{\alpha} - 1 \\ \alpha - \lambda & \text{for } \sqrt{\alpha} - 1 \leq \lambda \leq \alpha \end{cases}$$

and

$$\bar{\ell}(\lambda) = \sqrt{\alpha} - \lambda \quad \text{for } 0 \leq \lambda \leq \sqrt{\alpha}.$$

for $c' = \alpha$ and $\bar{c} = \sqrt{\alpha}$.

The lower bounds on c' and $\bar{c}$ behave identical up to a load of $\sqrt{\alpha} - 1$. Hence, an online algorithm can not distinguish between $(1, k)$-Pack with $c_1 = c'$ and $c_2 = \bar{c}$ or the one with $c_1 = \bar{c}$ and $c_2 = c'$ until the load on at least one of the constraints exceeds $\sqrt{\alpha} - 1$. The first constraint to exceed this threshold is then set to capacity $\bar{c}$ with lower bound function $\bar{\ell}$.

Since the profit of any solution x to $(1, k)$-Pack can be expressed in terms of Ax by $\frac{(Ax)_1 + (Ax)_2}{k+1}$ and up to this moment the load on both constraints is not more than $\sqrt{\alpha}$, the intermediate profit is bounded by 2.

After the lower bound functions have been determined, the additional profit is bounded by the remaining slack on the constraint with capacity $\bar{c}$ which is not more than 1 at this point.

So the achievable online profit is bounded by 3 but the optimal offline solution achieves a profit of $\sqrt{\alpha}$. ◀

Nevertheless, there is still a huge gap between the general upper bound of $O(1/\sqrt{\alpha})$ and the previously given lower bounds of $\Omega(\ln(\alpha)/\alpha)$. At least for the $(1, k)$-Pack problem we will close this gap in Section 6.5 and give a special tailored algorithm achieving a tight competitive factor of $1/\sqrt{\alpha}$ on instances of those form.

Note that for the simple structured $(1, k)$-Pack problem with only two variables randomization can indeed help to achieve a better (even constant) competitive factor in expectation. However, as we will see in Section 7.1.1 this is not the case for more complex problem classes like packing spanning trees.

6.5 The BALANCING-Algorithm

We now introduce an online algorithm which is asymptotically optimal for the $(1, k)$-Pack problem. The algorithm continuously monitors the lower bounds of the remaining capacities and always increments the variable with the actual largest bottleneck capacity. This continuous incrementation can easily be discretized by executing a sufficiently large number of infinitesimal small stages.

While the BALANCING algorithm achieves good results on the $(1, k)$-Pack problem, this approach performs poorly on more general problems even when given the true offline capacities.

This is intuitively clear from the applied greedy allocation rule. We will give unbounded worst-case instances for the tree packing and maximum flow problems in Sections 7.1.2.3 and 7.2.2.3.

However, the algorithm provides evidence that for better upper bounds on the general competitive ratio more complex problem instances than the easy $(1,k)$-Pack problem have to be considered.

Algorithm BALANCING

Input: Online packing LP $\mathcal{P}$ of the form $\max p^\top x$ subject to $Ax \leq c$, $x \geq 0$, with unknown c.

1. Normalize the constraint matrix with respect to the profit vector, i.e. set $a_{ij} := a_{ij}/p_i$ for all $1 \leq i \leq n$ and $1 \leq j \leq m$ after removing variables with zero profit. Let all columns have at least one non-zero entry (otherwise the solution is unbounded).
2. Set $x = 0 \in \mathbb{R}^n_{\geq 0}$.
3. For each variable x_i continuously compute its bottleneck capacity

$$b(x_i) = \min_j \{\ell_j/a_{ij} \mid a_{ij} > 0\}$$

with respect to the actual lower bounds on the remaining capacities and adaptively increment the variable with the actual largest bottleneck capacity until all bottleneck capacities have become zero.
4. Output solution x.

We will see that this simple online algorithm is asymptotically optimal with respect to the general upper bound on deterministic online algorithms.

▶ **Theorem 6.11.** *BALANCING is $\frac{1}{\sqrt{\alpha}}$-competitive on $(1,k)$-Pack.*

Proof. Consider an instance of $(1,k)$-Pack for arbitrary k.

Due to the form of this linear program it is not difficult to see that the optimal value is $\min\{c_1, c_2, \frac{c_1+c_2}{k+1}\}$. The first argument corresponds to the case when $c_1 \geq k \cdot c_2$ (then only x_1 is positive), the second one is symmetric for $c_2 \geq k \cdot c_1$ and the third corresponds to $\frac{1}{k} \cdot c_2 \leq c_1 \leq k \cdot c_2$ (then we have mixed solutions).

Note that the BALANCING algorithm works in a continuous fashion and tries to keep the slacks $\ell_1((Ax)_1)$ and $\ell_2((Ax)_2)$ always on the same level. (If at any time $\ell_1((Ax)_1) > \ell_2((Ax)_2)$ then the bottleneck capacity of x_1 is bigger than the one of x_2 and by increasing x_1 the load on the first constraint increases stronger than the load on the second constraint. The algorithm behaves vice versa in the opposite case.)

If at the end of the algorithm both slacks are zero then an optimal solution is found. So let's assume the algorithm ends with $\ell_1((Ax)_1) > 0$ and $\ell_2((Ax)_2) = 0$. Fix the last moment in time when the two lower bounds were identical and denote the corresponding x-vector by $\hat{x}$. Furthermore, we denote the slack values of this particular time by s_1 and s_2. From then on we always have $\ell_1((Ax)_1) > \ell_2((Ax)_2)$ and although the algorithm hence only increases x_1 the second constraint becomes tight first. So the algorithm increases x_1 by s_2 and stops. This implies $s_1 > k \cdot s_2$ and we can write $s_1 = \gamma \cdot k \cdot s_2$ for a $\gamma \geq 1$. Note that if the lower bounds did never coincide then the algorithm always increased the same variable and obtained an optimal solution.

As $\frac{1}{k} \cdot (A\hat{x})_2 \leq (A\hat{x})_1 \leq k \cdot (A\hat{x})_2$ holds, $\hat{x}$ is an optimal solution of value $\frac{(A\hat{x})_1+(A\hat{x})_2}{k+1}$ to the instance $\mathcal{P}((A\hat{x})_1, (A\hat{x})_2)$ which implies $\mathcal{P}^{ALG}(c_1,c_2) = \mathcal{P}^{OPT}((A\hat{x})_1, (A\hat{x})_2) + s_2$.

Putting all this together we get:

$$\mathcal{P}^{OPT}(c_1, c_2) \leq \frac{c_1 + c_2}{k+1} \leq \frac{(A\hat{x})_1 + (A\hat{x})_2}{k+1} + \frac{s_1 + s_2}{k+1}$$
$$= \mathcal{P}^{OPT}((A\hat{x})_1, (A\hat{x})_2) + \frac{s_1 + s_2}{k+1}\,.$$

However, by

$$\frac{s_1 + s_2}{k+1} = \frac{\gamma \cdot k \cdot s_2 + s_2}{k+1} \leq \gamma \cdot s_2$$

we get

$$\frac{\mathcal{P}^{ALG}(c_1, c_2)}{\mathcal{P}^{OPT}(c_1, c_2)} \geq \frac{\mathcal{P}^{OPT}((A\hat{x})_1, (A\hat{x})_2) + s_2}{\mathcal{P}^{OPT}((A\hat{x})_1, (A\hat{x})_2) + \gamma \cdot s_2} \geq \frac{1}{\gamma}. \tag{6.1}$$

Since the lower bounds were identical when defining s_1 and s_2 and since the bounds approximate those values by a factor of α we know $\frac{1}{\alpha} \leq \frac{s_1}{s_2} \leq \alpha$ and hence

$$\frac{1}{\alpha} \leq \gamma \cdot k \leq \alpha. \tag{6.2}$$

Suppose for the sake of contradiction that $\mathcal{P}^{ALG}(c_1, c_2) < \frac{1}{\sqrt{\alpha}} \cdot \mathcal{P}^{OPT}(c_1, c_2)$. By 6.1 this implies $\gamma > \sqrt{\alpha}$ and which together with 6.2 leads to $k < \sqrt{\alpha}$.

For the output x of the algorithm we can always estimate the profit by

$$\mathcal{P}^{ALG}(c_1, c_2) = \frac{(Ax)_1 + (Ax)_2}{k+1} \geq \frac{(1 + 1/k) \cdot \max\{(Ax)_1, (Ax)_2\}}{k+1}$$
$$\geq \frac{1}{k} \cdot \min\{c_1, c_2\} \geq \frac{1}{k} \cdot \mathcal{P}^{OPT}(c_1, c_2).$$

But this implies $\mathcal{P}^{ALG}(c_1, c_2) > \frac{1}{\sqrt{\alpha}} \cdot \mathcal{P}^{OPT}(c_1, c_2)$, a contradiction. ◀

6.6 Applicability to Network Lifetime Maximization

Up to now our study of the online model and the proposed algorithms was of a more abstract nature. We will discuss the computationally efficient applicability of the developed techniques to the context of combinatorial packing and network lifetime maximization in the current section.

In the definition of our online model and the description of the given online algorithms it was assumed that the problem is specified in terms of a linear packing program. Since in classic competitive analysis complexity considerations are not an issue, we could theoretically just transform each given combinatorial packing problem into its inflated representation and work on the corresponding (possibly exponentially sized) linear program.

However, in practice this is not an option. Especially in the context of wireless ad-hoc sensor networks we have to deal with limited computational resources. Luckily, the developed online algorithms (with exception of the artificial BALANCING algorithm) do not depend on an explicit description of the linear packing program. They just partition the online problem into a (polynomial sized) sequence of corresponding offline problems each with updated capacities depending on the current lower bounds.

The algorithms GREEDY, EQUITABLE and ENVISAGE provide an algorithmic framework that allows to plug in any optimal offline optimization algorithm for a combinatorial packing problem as a black box in order to achieve provable worst-case competitiveness for the corresponding online problem. The only explicit information the online framework has to keep track of are the current lower bounds on the remaining capacities and the already incurred capacity loads.

6.6.1 Approximation Algorithms

One obstacle in network lifetime maximization already seen in Chapter 5 is that most variants of lifetime optimization problems are NP-hard and can —if at all— only be approximated.

Nevertheless, also under these conditions our online algorithms are applicable. It is easy to see that the proof of Theorem 6.5 by linearity of the underlying problem also adapts to the situation where the online algorithm can only approximate an optimal solution of the packing problem $\mathcal{P}(b^{(s)})$.

▶ **Corollary 6.12.** *If for $r \leq 1$ in every stage s of the algorithm EQUITABLE or ENVISAGE, a feasible r-approximate solution for $\mathcal{P}(b^{(s)})$ is computed, the overall solution is $((1-\epsilon)\cdot\frac{\ln(\alpha)}{\alpha-1}\cdot r)$-competitive after a total number of $N \geq \frac{\alpha-1}{\alpha^\epsilon-1}$ many stages.*

Trivially, the analogue result with a competitive ratio of r/α for $r \leq 1$ is true for GREEDY.

The property of approximation preservation can also be useful when integral solutions are favored over fractional ones. If the capacities are large enough, randomized rounding can be used to get an integral $(1-\epsilon)$-approximation to the optimal fractional packing in every step without violating the capacity budgets.

6.6.2 Randomized Rounding

In many cases integral solutions are favored over fractional ones. This can easily be accomplished by a randomized rounding of the fractional solutions in the stages of EQUITABLE or ENVISAGE. In typical applications the total battery capacities are large compared to the energy needed for establishing a communication structure. Our algorithm guarantees that the number of stages is very limited, hence the budget vectors are still large in comparison to the entries of the constraint matrix. The following analysis is in analogy to [Sri99] but we aim to get a stronger result allowing only to lose an ϵ-fraction of the objective value.

▶ **Lemma 6.13.** *Let $0 < \epsilon \leq 1$ and assume $b_i^{(s)} \geq 6\log(em)/\epsilon^4$ for all $i = 1, \ldots, m$ in the constraint-normalized packing LP. Then the expected number of randomized rounding steps to obtain a feasible integral solution that is within a factor of $1-\epsilon$ of the optimum is at most 4.*

Proof. Given an optimal solution x^* to the linear program $\max p^\top x$ subject to $Ax \leq b^{(s)}$, $x \geq 0$, with normalized packing constraints as in stage s of EQUITABLE or ENVISAGE, we can do the following. For $0 < \epsilon \leq 1$ we define $y := x^*/(1+\epsilon)$ and for each positive entry y_j we round it up to $\lfloor y_j \rfloor + 1$ with probability $y_j - \lfloor y_j \rfloor$. Otherwise we round it down. Note that although the LP might have an exponential number of variables, only a polynomial number of them is positive in a basic solution to common network lifetime problems.

For ease of analysis we define random variables $Y_j^1, \ldots, Y_j^{\lfloor y_j \rfloor + 1} \in \{0,1\}$ for each positive entry y_j with $\mathbf{Pr}\left[Y_j^k = 1\right] := 1$ for $k = 1, \ldots, \lfloor y_j \rfloor$ and $\mathbf{Pr}\left[Y_j^{\lfloor y_j \rfloor + 1} = 1\right] := y_j - \lfloor y_j \rfloor$.

To estimate the obtained objective value $Y := \sum_j p_j \sum_k Y_j^k$ we observe that

$$\mathbf{E}[Y] = p^\top y = \frac{1}{1+\epsilon} \cdot p^\top x^* \ .$$

For normalized constraints (implying $\max_j a_{i,j} = 1$) with $b_i^{(s)} \geq t := 6\log(em)/\epsilon^4$ for all $1 \leq i \leq m$ we can lower bound $p^\top x^*$ by $t \cdot \max_j(p_j)$. Then a Chernoff bound shows

$$\mathbf{Pr}\left[Y \leq (1-\epsilon)p^\top x^*\right] = \mathbf{Pr}\left[Y \leq (1-\epsilon^2)E[Y]\right] \leq e^{-\frac{\epsilon^4 \cdot p^\top x}{2(1+\epsilon)\cdot \max_j(p_j)}} \leq e^{-\frac{\epsilon^4 t}{6}} \ .$$

For any row i of the LP we set $Y' := \sum_j a_{i,j} \sum_k Y_j^k$ and see that

$$\mathbf{E}[Y'] = (Ay)_i \leq \frac{1}{1+\epsilon} \cdot b_i^{(s)}$$

which by applying a Chernoff bound leads to

$$\mathbf{Pr}\left[Y' \geq b_i^{(s)}\right] \leq e^{-\frac{\epsilon^2}{3(1+\epsilon)} b_i^{(s)}} \leq e^{-\frac{\epsilon^4 t}{6}} .$$

For $b_i^{(s)} \geq t := 6\log(em)/\epsilon^4$ for all $i \in [m]$, the above probabilities are less than $1/(em)$.

By a union bound we see that the probability for non of the above cases to hold is at least

$$1 - (m+1) \cdot \frac{1}{em} \geq 1 - 2e^{-1} \geq \frac{1}{4}$$

so in expectation at most four randomized rounding steps are needed to obtain a feasible near-optimal integral solution. ◀

Chapter 7

Competitiveness for Classic Combinatorial Packing Problems

In Chapter 6 we analyzed our novel online model for packing problems with gradually improving capacity estimations in the context of general linear programming. Hence, the given approximation guarantees for the algorithms GREEDY, EQUITABLE, and ENVISAGE hold on any problem that can be formulated in terms of a general linear packing problem. However, as only general properties were exploited, one could hope that for specific combinatorial packing problems better upper and lower bounds on the competitiveness may be achieved.

For the problem of packing spanning trees —which is a relevant subproblem in a broad class of network lifetime problems— indeed the result from Section 6.4 can be strengthened: we show a general bound of $O(1/\sqrt{\alpha})$ on the competitiveness of any *randomized* online algorithm in the following section.

Furthermore, we will see that the analysis of algorithms GREEDY and EQUITABLE stays tight for the problems of packing spanning trees and single-commodity maximum flow, while BALANCING is non-competitive in both cases. Thus, the special substructure of these combinatorial packing problems can not be exploited to derive better results for the proposed online algorithms. Also the gap between the bounds on algorithm ENVISAGE remains, as no stronger bounds on the competitiveness could be derived for the specific combinatorial packing problems considered in this chapter.

In Section 7.2 the upper bound of $O(1/\sqrt{\alpha})$ on any deterministic online algorithm given in Chapter 6 will be translated to instances of the maximum flow problem. As the specific construction of the corresponding worst-case instance requires at least two commodities, it remains open if single-commodity flow problems exhibit the same bound.

7.1 Packing Spanning Trees

We analyze the impact of our online model on the classic maximum spanning tree packing problem according to Definition 5.9. This problem often arises in the context of network lifetime maximization where problem instances are reduced to appropriate tree/arborescence packing problems in order to be efficiently solved (see, e.g., [OY05]).

Note that the problem of packing spanning trees can be easily reduced to the maximum lifetime convergecast or broadcast problem (both for omnidirectional and unidirectional antennas) by replacing each capacitated edge by an intermediate capacitated transceiver node and allow the original (uncapacitated) nodes only to communicate through these proxies.

Thus, by showing upper bounds on the competitiveness of the maximum spanning tree problem for general and specific online algorithms those bounds immediately translate to a broad class of lifetime maximization problems.

7.1.1 A General $O(1/\sqrt{\alpha})$ Upper Bound for Randomized Online Algorithms

In this section we will see that no online algorithm (even randomized) for packing spanning trees can achieve a competitive ratio better than $O(1/\sqrt{\alpha})$. This improves the corresponding

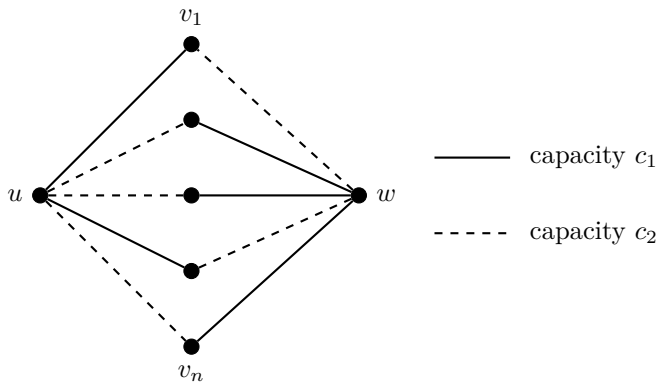


Figure 7.1 Diamond Gadget $D(n, \vec{a}, c_1, c_2)$ for $n = 5$ and $\vec{a} = 01101$.

deterministic upper bound given in Chapter 6. Similar to Theorem 6.10 we will first prove an upper bound on deterministic online algorithms for the problem of packing spanning trees, which we then can extend to randomized algorithms by applying Yao's principle (for details on Yao's principle see, e.g., [BEY98]).

In order to bound the competitiveness of any deterministic online algorithm we define a class of worst-case instances.

▶ **Definition 7.1** (Diamond Gadget $D(n, \vec{a}, c_1, c_2)$)**.** For any $n \in \mathbb{N}$ and $\vec{a} \in \{0,1\}^n$ let $D(n, \vec{a}, c_1, c_2)$ be the complete (undirected) bipartite graph $K_{2,n}$ defined by $V = A \cup B = \{u, w\} \cup \{v_1, \ldots, v_n\}$ and $E = A \times B$. For each node $v_i \in B$ the capacities of the incident edges $\delta(v_i)$ are defined by setting edge $\{u, v_i\}$ to capacity c_1 and edge $\{w, v_i\}$ to capacity c_2 if $\vec{a}_i = 0$ and vice versa for $\vec{a}_i = 1$.

Obviously any offline solution to the spanning tree packing problem on graphs of this class is independent of the choice of the parameter $\vec{\alpha}$. As we will show, however, it allows us to limit the quality of any online solution.

▶ **Lemma 7.2.** *For any diamond gadget $D(n, \vec{a}, c_1, c_2)$ with $c_1 = \alpha$ and $n \geq c_2 = \sqrt{\alpha}$ there is an optimal (offline) algorithm achieving a spanning tree packing with total multiplicity α.*

Proof. Let $\bar{E}$ be the set of edges of capacity $\sqrt{\alpha}$ and $E' = E - \bar{E}$. For each of the n many edges $\bar{e} \in \bar{E}$ just pack the spanning tree defined by $\{\bar{e}\} \cup E'$ with a multiplicity of $\alpha/n \leq c_2$. ◀

▶ **Theorem 7.3.** *For any deterministic online algorithm $\mathcal{A}$ there exists an $\vec{a} \in \{0,1\}^n$ such that $\mathcal{A}$ is not able to obtain a spanning tree packing of total multiplicity more than $4\sqrt{\alpha}$ on the diamond gadget $D(n, \vec{a}, \alpha, \sqrt{\alpha})$ with $n = \lceil \sqrt{\alpha} \rceil$.*

Proof. We define the lower bound functions for the edges of capacities $c_1 = \alpha$ and $c_2 = \sqrt{\alpha}$ to be

$$\ell_1(\lambda) = \begin{cases} \sqrt{\alpha} - \lambda & \text{for } 0 \leq \lambda < \sqrt{\alpha} - 1 \\ \alpha - \lambda & \text{for } \sqrt{\alpha} - 1 \leq \lambda \leq \alpha \end{cases} \text{ and } \qquad \ell_2(\lambda) = \sqrt{\alpha} - \lambda \quad \text{for } 0 \leq \lambda \leq \sqrt{\alpha},$$

respectively.

The lower bounds on all edges behave identical up to a load of $\sqrt{\alpha}-1$. Hence, an online algorithm can not distinguish the different types of edges at the beginning. For every vertex $v_i \in B$ consider the one of its two incident edges where the load exceeds the threshold of $\sqrt{\alpha}-1$ first. If this edge e is $\{u, v_i\}$ we fix $\vec{a}_i = 1$. Otherwise $e = \{w, v_i\}$ and we set $\vec{a}_i = 0$. At this point the offline adversary reveals the corresponding true capacities for the two edges incident to v_i.

When true capacities are finally determined for all edges the additional spanning tree multiplicity that can be packed is at most $n \leq 2\sqrt{\alpha}$. This is easy to see, as n of the edges have a remaining capacity of less than $\ell_2(\sqrt{\alpha}-1) = 1$, however at least one of them is needed in any spanning tree.

How many spanning trees can be packed by an online algorithm before all the capacities are determined, then? For each set $\delta(z) = \{\bar{e}, e'\}$ the sum of the edge loads $\lambda_{\bar{e}} + \lambda_{e'}$ gets increased by at least one for every spanning tree in the packing solution. Thus, if the number of packed spanning trees exceeds $2\sqrt{\alpha}$, at least one of the edges of every set has blocked capacity of more than $\sqrt{\alpha}$. This renders the solution infeasible if the last lower bound probing derived from blocked capacity not more than $\sqrt{\alpha}-1$ for both edges of at least one of the sets $\delta(v)$. So the best possible overall online solution is bounded by $4\sqrt{\alpha}$. ◀

▶ **Corollary 7.4.** *No deterministic online algorithm for packing spanning trees is better than* $4/\sqrt{\alpha}$*-competitive.*

The proof of our randomized upper bound is based upon Yao's minimax principle. We use the more general variant of this property after Stougie and Vestjens [SV02].

▶ **Lemma 7.5** (Yao's principle [SV02])**.** *Given an online maximization problem with set $\mathcal{I}$ of possible input instances and set $\mathcal{D}$ of possible deterministic online algorithms (both possibly infinite). Then for any random instance I_p (defined as a probability distribution p over $\mathcal{I}$) and any randomized online algorithm B the corresponding competitive ratio c_B can be bounded by*

$$c_B \leq \max_{A \in \mathcal{D}} \frac{\mathbf{E}_p\left[\mathcal{P}^A(I_p)\right]}{\mathbf{E}_p\left[\mathcal{P}^{OPT}(I_p)\right]}$$

where $\mathcal{P}^A(I)$ is the profit achieved by algorithm A and $\mathcal{P}^{OPT}(I)$ denotes the optimal profit on instance $I \in \mathcal{I}$.

Proof. We follow the very nice and simple proof of [SV02]. Suppose there exists a randomized online algorithm $B = A_q$ with competitive ratio c_B, which can be seen as a probability distribution q over deterministic algorithms $A \in \mathcal{D}$. By definition of the competitive ratio for randomized algorithms we have

$$c_B = \min_{I \in \mathcal{I}} \frac{\mathbf{E}_q\left[\mathcal{P}^{A_q}(I)\right]}{\mathcal{P}^{\mathrm{OPT}}(I)}$$

implying

$$c_B \cdot \mathcal{P}^{\mathrm{OPT}}(I) \leq \mathbf{E}_q\left[\mathcal{P}^{A_q}(I)\right] \qquad \forall I \in \mathcal{I}\,.$$

Thus

$$c_B \cdot \mathbf{E}_p\left[\mathcal{P}^{\mathrm{OPT}}(I_p)\right] \leq \mathbf{E}_p\left[\mathbf{E}_q\left[\mathcal{P}^{A_q}(I_p)\right]\right] = \mathbf{E}_q\left[\mathbf{E}_p\left[\mathcal{P}^{A_q}(I_p)\right]\right] \leq \max_{A \in \mathcal{D}} \mathbf{E}_p\left[\mathcal{P}^A(I_p)\right]$$

follows for any random instance I_p, which proves the claim. ◀

By Yao's principle it suffices to define a probability distribution over instances such that for any fixed deterministic online algorithm the expected profit is bounded by some value $\mathcal{P}_1$ whereas the expected optimal profit on this random instance is at least $\mathcal{P}_2$, in order to show an upper bound of $\mathcal{P}_1/\mathcal{P}_2$ on the competitive ratio of any randomized online algorithm.

In our case, we construct a distribution of instances where an optimal spanning tree packing has multiplicity α (independent of the outcome of the random experiment) but every deterministic online algorithm has an expected multiplicity of at most $5\sqrt{\alpha}$.

▶ **Definition 7.6** (Random Graph $\hat{R}$)**.** The random graph $\hat{R}$ consists of M many diamond gadgets $D(n, \vec{a}, \alpha, \sqrt{\alpha})$ according to Definition 7.1 with $n = \lceil \sqrt{\alpha} \rceil$, where $\vec{a} \in \{0,1\}^n$ is chosen independently and uniformly at random for each of the gadgets. Let $\{u_i, w_i\}$ be the nodes of the partition of size two in the i-th diamond gadget. We connect all gadgets by the path $P = (u_1, u_2, \ldots, u_M)$ with edge capacities α.

Note that each of the 2^n possible diamond gadgets is chosen with equal probability. By estimating the probability that any deterministic algorithm performs bad on at least one of the M many gadgets and thereby limiting the expected total multiplicity we get the following result.

▶ **Theorem 7.7.** *The spanning tree packing of every deterministic online algorithm has an expected multiplicity of at most $5\sqrt{\alpha}$ on the random graph $\hat{R}$ with $M \geq 2^{n-1}\ln(\alpha)$. So by Yao's principle the competitive ratio of any randomized online algorithm is at most $\frac{5}{\sqrt{\alpha}}$.*

Proof. Let T denote the multiplicity obtained by the deterministic algorithm on the whole graph $\hat{R}$ and denote the restriction to one of the M gadgets by T_i. It holds

$$\mathbf{Pr}\left[T \leq 4\sqrt{\alpha}\right] = \mathbf{Pr}\left[\exists_i T_i \leq 4\sqrt{\alpha}\right] = 1 - \mathbf{Pr}\left[\forall_i T_i > 4\sqrt{\alpha}\right] \ .$$

By independence of the gadgets, it follows

$$\mathbf{Pr}\left[T \leq 4\sqrt{\alpha}\right] = 1 - \prod_{i=1}^{M} \mathbf{Pr}\left[T_i > 4\sqrt{\alpha}\right] = 1 - \prod_{i=1}^{M} \left(1 - \mathbf{Pr}\left[T_i \leq 4\sqrt{\alpha}\right]\right) \ .$$

Now, for $1 \leq i \leq M$, since each gadget is chosen uniformly at random from 2^n possibilities, and by Theorem 7.3 for any deterministic online algorithm, at least one of the diamond gadgets results in a packing of at most $4\sqrt{\alpha}$ many trees, we get

$$\mathbf{Pr}\left[T_i \leq 4\sqrt{\alpha}\right] \geq 2^{-n} \ .$$

As a consequence,

$$\mathbf{Pr}\left[T \leq 4\sqrt{\alpha}\right] \quad \geq \quad 1 - \left(1 - 2^{-n}\right)^M \quad \geq \quad 1 - \exp\left(-2^{-n}M\right)$$

as $1 - x \leq e^{-x}$ for any real number x. Setting $M = 2^{n-1}\ln(\alpha)$ we obtain

$$\mathbf{Pr}\left[T \leq 4\sqrt{\alpha}\right] \geq 1 - 1/\sqrt{\alpha} \ .$$

Since the optimal solution has multiplicity α, we get

$$\mathbf{E}\left[T\right] \leq 4\sqrt{\alpha} + \alpha \cdot \mathbf{Pr}\left[T > 4\sqrt{\alpha}\right] \leq 4\sqrt{\alpha} + \alpha \cdot 1/\sqrt{\alpha} \leq 5\sqrt{\alpha} \ .$$

That is, we have presented a probability distribution over instances with optimal multiplicity α for which every deterministic online algorithm has an expected multiplicity of at most $5\sqrt{\alpha}$. Thus, by Yao's principle, this gives an upper bound of $\frac{5}{\sqrt{\alpha}}$ on the competitive ratio of any randomized online algorithm. ◀

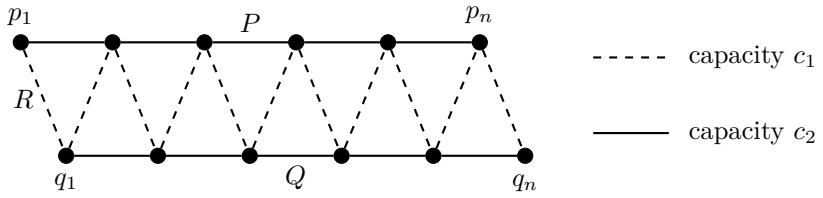


Figure 7.2 Snake Gadget $S(n, c_1, c_2)$ for $n = 6$.

7.1.2 Worst-Case Instances for Specific Algorithms

The negative bounds shown in Chapter 6 on the competitiveness of the proposed online algorithms do not only hold for the artificial $(1, k)$-Pack problem but can also be translated to the problem of packing spanning trees. For this we define a graph instance which captures the essential properties of $(1, k)$-Pack.

▶ **Definition 7.8** (Snake Gadget $S(n, c_1, c_2)$)**.** We define the snake gadget $S(n, c_1, c_2)$ of dimension n and capacities c_1 and c_2 to be an undirected graph $G = (V, E)$ built of $2n$ many nodes in the following way: two disjoint paths $P = (p_1, \ldots, p_n)$ and $Q = (q_1, \ldots, q_n)$, are connected by additional edges induced by the path $R = (p_1, q_1, p_2, q_2, \ldots, p_n, q_n)$. The capacities of the edges $E_1 = E(R)$ belonging to R are c_1 while the capacities of the edges of $E_2 = E(P) \cup E(Q)$ are c_2 (see Figure 7.2 for a graphical representation).

▶ **Lemma 7.9.** *Any optimal spanning tree packing to a snake gadget $S(n, c_1, c_2)$ with capacities $c_1 \geq \frac{c_2}{2n-1}$ has total multiplicity $\frac{1}{|V|-1} \sum_{e \in E} c_e = c_1 + (1 - \frac{1}{2n-1}) \cdot c_2$ and completely exhausts all edge capacities.*

Proof. Obviously $\frac{1}{|V|-1} \sum_{e \in E} c_e$ is an upper bound on spanning tree packings for any graph, since each spanning tree charges $|V| - 1$ many edges with its corresponding multiplicity.

In order to see tightness of this bound on instances of the specified form consider the packing consisting of the $(2n - 1)$ many spanning trees containing only a single edge of E_1 each with equally shared multiplicity $\frac{c_2}{2n-1}$ together with the tree E_1 using the remaining capacity $c_1 - \frac{c_2}{2n-1}$.

Thus, any optimal solution to instances of this form has value $c_1 + (1 - \frac{1}{2n-1}) \cdot c_2$ and does completely exhaust all edge capacities. ◀

7.1.2.1 A $2/\alpha$ Upper Bound on GREEDY

▶ **Lemma 7.10.** *GREEDY is less than $2/\alpha$-competitive for the spanning tree packing problem.*

Proof. We consider the snake gadget of dimension $n \geq \lceil \alpha \rceil$ and capacities $c_1 = 1$ and $c_2 = \alpha$ with initial lower bounds $\ell^{(1)} = (1; 1)$.

GREEDY computes an optimal solution to the snake gadget $S(n, 1, 1)$ which by Lemma 7.9 has value $2 - \frac{1}{2n-1}$ and exhausts from all edges the assumed uniform capacity 1. After that, the remaining real capacities of the edges are $c_1 = 0$ and $c_2 = \alpha - 1$. Since each spanning tree has to contain at least one edge of E_1 no further spanning trees can be added in the second iteration of GREEDY. However, since $1 \geq \frac{\alpha}{2n-1}$ an optimal offline solution on the original capacities $(1; \alpha)$ would achieve a total multiplicity of $1 + \alpha - \frac{\alpha}{2n-1} \geq \alpha$. ◀

7.1.2.2 An $O(\ln(\alpha)/\alpha)$ Upper Bound on EQUITABLE

▶ **Lemma 7.11.** *EQUITABLE is at most $O(\ln(\alpha)/\alpha)$-competitive for the problem of packing spanning trees.*

Proof. Consider the snake gadget of dimension $n = \lceil \alpha \rceil$ and capacities $c_1 = 1$ and $c_2 = \alpha$. Let the corresponding lower bounds be defined as functions in the constraint usage $\lambda_j = (Ax)_j$.

$$\ell_1(\lambda) = 1 - \lambda \quad \text{for } \lambda \leq 1 \qquad\qquad \ell_2(\lambda) = 1 - \frac{\lambda}{\alpha} \quad \text{for } \lambda \leq \alpha$$

These bounds are α-sharp and non-increasing and thereby fulfill the premise of Lemma 6.6. So the achieved profit up to stage $\lceil q \rceil$ with $q = \frac{\alpha-1}{\alpha} \cdot N$ (where N denotes the total number of stages made by EQUITABLE) can be bounded by

$$\sum_{s=1}^{\lceil q \rceil} \mathcal{P}^{\mathrm{OPT}}(b^{(s)}) \leq (1 + \ln(\alpha)) \cdot \frac{1}{|V| - 1} \sum_{e \in E} \ell_e^{(1)} < 2 \cdot (1 + \ln(\alpha))$$

on this instance.

As long as $b_1^{(s)} \geq \frac{b_2^{(s)}}{2n-1}$ holds, we know by Lemma 7.9 that the budgets of all edges of the graph will be completely exhausted by EQUITABLE, thereby guaranteeing that all edges from E_1 (as well as all edges from E_2) are uniformly decreased by the corresponding budget values $b_1^{(s)}$ and $b_2^{(s)}$, respectively.

Hence, the situation that the capacities of the edges in E_1 are not uniform anymore can occur only after c_1 (resp. the corresponding residual capacity) has dropped below $\frac{\ell_2(0)}{2n-1} \leq \frac{1}{n} \leq \frac{1}{\alpha}$. However, from that point on, the additionally achievable total multiplicity is limited by $\sum_{e \in E_1} \frac{1}{\alpha} \leq \frac{2\alpha+1}{\alpha} \leq 3$.

As the algorithm guarantees monotonicity of the budget vectors we have $b_1^{(s)} \geq b_1^{(1)} = \frac{1}{N}$ for all $s \in [N]$. Thereby the number of stages until the residual capacity of all edges contained in E_1 drops below $\frac{1}{\alpha}$ is at most $\lceil q \rceil$.

Thus, we can bound the total multiplicity achieved by EQUITABLE by $\mathcal{P}^{\mathrm{ALG}} < 5 + 2\ln(\alpha)$ while the optimal offline solution yields a packing of multiplicity greater than α. ◀

7.1.2.3 Non-Competitiveness of BALANCING

▶ **Lemma 7.12.** *BALANCING is not competitive for any fixed value (independent of the input size) for the spanning tree packing problem.*

Proof. Consider the snake gadget $S(n, c_1, c_2)$ for arbitrary $n \in \mathbb{N}$ where each edge of $E_2 = E(P) \cup E(Q)$ is replaced by a copy of the diamond gadget $D(4n^2, \vec{a}, c_3, c_4)$ with arbitrary $\vec{a}$ and orientation. We set $c_1 = 1$, $c_3 = \frac{1}{2n}$ and $c_4 = 2n$ (all edges of capacity c_2 have been replaced) and allow BALANCING to work on the true offline capacities.

Note that for the chosen capacities each diamond gadgets admits a sub-packing of spanning trees on the induced subgraph of total multiplicity $2n$ without using any additional edges.

Since BALANCING continuously adds the spanning tree with the largest bottleneck capacity to the packing, edges with capacity c_3 are avoided until the bottleneck capacity of the path R drops below c_3. Any spanning tree without edges with capacity c_3 contains all edges from R, so up to the time when $c_1 \leq c_3$ for the first time, the algorithm has achieved a total multiplicity of less than $c_1 = 1$. After that, the additional multiplicity that can be achieved is bounded by $(2n-1)/(2n) \leq 1$ since at least one of the edges from R is needed in any spanning tree. So while BALANCING only has total multiplicity less than 2 the offline optimum is more than $2n - 1$. As we may chose n arbitrarily this proves the claim. ◀

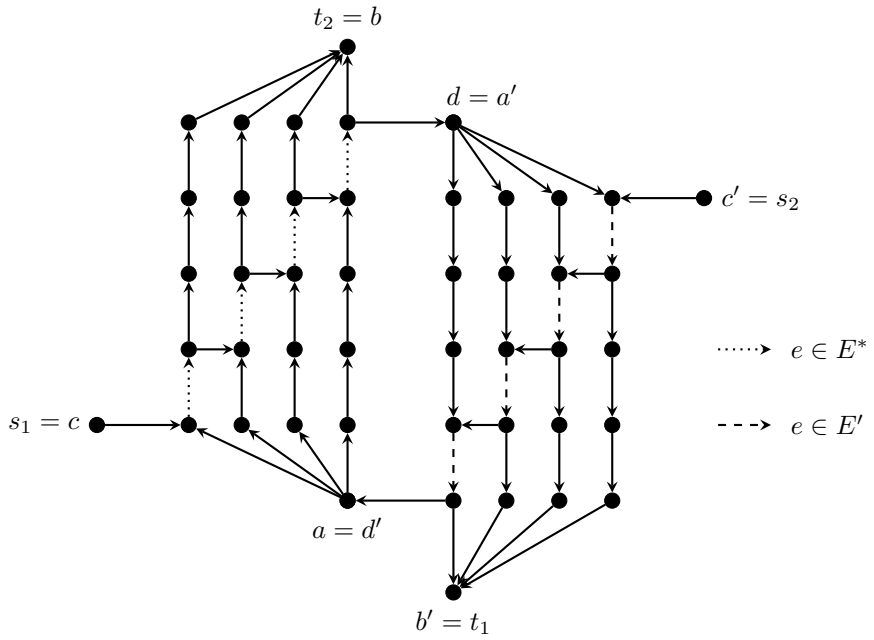


Figure 7.3 Two interweaved waterfall gadgets of dimension 4.

7.2 Maximum Flow

In the current chapter we analyze our online model in the context of the maximum flow problem according to Definition 5.8.

7.2.1 A Deterministic $O(1/\sqrt{\alpha})$ Upper Bound

To derive a general upper bound for the maximum flow problem we define a (multi-commodity) flow network where any (deterministic) online algorithm can not be better than $O(1/\sqrt{\alpha})$-competitive.

▶ **Definition 7.13** (Waterfall Gadget $W(n, E^*, a, b, c, d)$)**.** We define the waterfall gadget $W(n, E^*, a, b, c, d)$ of dimension n with designated nodes a, b, c, d and critical edge-subset E^* to be a directed graph constructed in the following way: Begin with n directed internally node disjoint paths $P^1, \ldots, P^n$ of length $n+2$ sharing a as a start and b as an end node, that is each P^i having the form $P^i = (a, v_1^i, \ldots, v_{n+1}^i, b)$. The edges in $E^* = \{(v_i^i, v_{i+1}^i) \mid 1 \leq i \leq n\}$ are called critical edges. For each $2 \leq i \leq n$ we add an edge leading from P^{i-1} to P^i at the node pair (v_i^{i-1}, v_i^i). The construction is completed by adding nodes c and d with edges (c, v_1^1) and (v_{n+1}^n, d).

▶ **Lemma 7.14.** *Any deterministic online algorithm is less than $\frac{4}{\sqrt{\alpha}}$-competitive for the maximum flow problem in the worst case.*

Proof. Consider two waterfall gadgets $W(n, E^*, a, b, c, d)$ and $W(n, E', a', b', c', d')$ both of dimension $n = \lceil \sqrt{\alpha} \rceil$ identified at the nodes $d = a'$ and $d' = a$ (see Figure 7.3). To the first commodity we assign the source $s_1 = c$ and sink $t_1 = b'$ whereas for the second commodity we have $s_2 = c'$ and $t_2 = b$.

For any edge e not in E^* or E' the true capacities of $c(e) = \alpha$ are directly revealed by reporting the lower bounds $\ell_e(\lambda) = \alpha - \lambda$ for $\lambda \leq \alpha$.

The lower bounds for the critical edges in $E^* \cup E'$ will be defined by threshold functions

$$\vartheta_t(\lambda) = \begin{cases} 1 & \text{for } \lambda < t \\ 1 - (\lambda - t) & \text{for } t \leq \lambda \leq t + 1 \end{cases}$$

for a threshold $t \leq \alpha - 1$. Until defined otherwise we assume threshold $t = \alpha - 1$ on all edges $e \in E^* \cup E'$.

Consider the point in time when a flow of at least $\sqrt{\alpha} - 1$ is reached by one of the commodities (w.l.o.g. let commodity 1 be the first commodity satisfying this condition). For any edge $e \in E^*$ let $f(e)$ be its flow at that time. Note that the total flow achieved so far by the algorithm is limited to $2\sqrt{\alpha}$ which thereby is also an upper bound on $f(e)$ for each edge e. Set $\ell_e = \vartheta_{f(e)}$ for each $e \in E^*$, thus limiting the residual capacity on those edges to exactly 1.

Now the total additional flow that can be routed by the algorithm is at most $n \cdot 1 < 2\sqrt{\alpha}$, since any source-sink path has to contain at least one of the edges in E^*. But the optimal offline solution would have achieved a total flow of α by routing flow along appropriate (s_2, t_2)-paths (note that at the time of fixing the threshold on each $e \in E^*$ we have $f(e) \geq \sqrt{\alpha} - 1$ for all $e \in E^*$ thereby implying a capacity of at least $\sqrt{\alpha}$ on those edges whereas all other edges have a capacity of α) thus resulting in a competitive ratio of less than $\frac{4}{\sqrt{\alpha}}$. ◀

7.2.2 Worst-Case Instances for Specific Algorithms

Though the waterfall gadget above provides a good upper bound for general online algorithms it is not suited to show tight results for the competitiveness of the specific algorithms proposed in Chapter 6 as even GREEDY is able to achieve a competitive ratio of $1/\sqrt{\alpha}$ on instances of this form. In order to show better upper bounds we define another class of worst case instances.

▶ **Definition 7.15** (Meander Gadget $M(n, c_1, c_2)$)**.** We define the meander gadget $M(n, c_1, c_2)$ of dimension n and capacities c_1 and c_2 to be a directed graph $G = (V, E)$ with source s and sink t built of $2n + 2$ many nodes in the following way: one path $R = (s, v_1, v_2, \ldots, v_{2n}, t)$ with additional edges $E' = \{(s, v_{2i+1}) \mid 1 \leq i \leq n - 1\}$ and $E'' = \{(v_{2i}, t) \mid 1 \leq i \leq n - 1\}$.

The capacities of the edges $E_1 = E(R)$ belonging to R are c_1 while the capacities of the edges of $E_2 = E' \cup E''$ are c_2 (see Figure 7.4 for a graphical representation).

▶ **Lemma 7.16.** *Any maximum flow in the meander gadget $M(n, c_1, c_2)$ with $c_1 \geq c_2$ has value $(n-1)c_2 + c_1$ and completely exhausts all edges, whereas in the case of $c_1 < c_2$ any maximum flow has value nc_1 and exhausts all edges of E_1.*

Proof. Consider the cuts

$$S = \{s\}, \qquad T = V \setminus S \qquad \text{and}$$
$$S' = \{s\} \cup \{v_{2i+1} \mid 0 \leq i \leq n - 1\}, \qquad T' = V \setminus S' .$$

Then the capacities of (S, T) and (S', T') are $(n-1)c_2 + c_1$ and nc_1 respectively, thus bounding the value on the maximum flow. However, a flow of value nc_1 can be easily constructed when $c_2 \geq c_1$ by packing the path-flows (s, v_{2i-1}, v_{2i}, t) with their bottleneck capacity. If

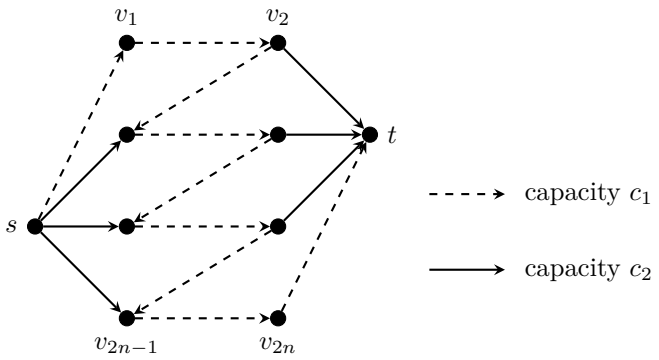


Figure 7.4 Meander Gadget $M(n, c_1, c_2)$ for $n = 4$.

$c_2 < c_1$ those path flows with an additional flow on R of value $c_1 - c_2$ yields a total flow of $(n-1)c_2 + c_1$ where the remaining capacity of any edge is 0.

In any maximum flow the edges of R have to be completely used up as otherwise flow could be augmented along that path. Now consider an edge $e \in E_2$ with flow $f(e) < c(e)$ in the case of $c_1 \geq c_2$. Then e crosses either the cut $(\{s\}, V \setminus \{s\})$ or $(V \setminus \{t\}, \{t\})$ bounding the flow across the cut to be at most $(n-2)c_2 + c_1 + c(e)$ which is less than the maximum flow value for that case shown before. ◀

7.2.2.1 A $1/\alpha$ Upper Bound on GREEDY

▶ **Lemma 7.17.** *GREEDY is not better than $\frac{1}{\alpha}$-competitive for the maximum flow problem in the worst-case.*

Proof. Let us assume GREEDY could achieve a competitive ratio of $\frac{1+\epsilon}{\alpha}$ for some $\epsilon > 0$ on any input instance.

We consider the meander gadget of dimension $n \geq \alpha/\epsilon$ with uniform capacities $c_1 = c_2 = 1$ but with initial lower bounds $\ell_1^{(1)} = 1$ and $\ell_2^{(1)} = 1/\alpha$.

According to Lemma 7.16 algorithm GREEDY has to exhaust the capacities of edges in E_1 by their assumed capacity 1 in order to achieve a maximum flow of value $1 + (n-1)/\alpha$ on the initial lower bounds. By the complete exhaustion of the edges in E_1 no further augmentation is possible in subsequent stages.

An optimal offline solution, however, yields a flow of value n which results in a competitive ratio of $\frac{1}{\alpha} + \frac{1}{n} - \frac{1}{\alpha n} < \frac{1+\epsilon}{\alpha}$. ◀

The lemma above shows that in general the analysis for GREEDY is not only asymptotically tight but that in worst-case the competitiveness of GREEDY exactly equals the guaranteed overall competitive ratio of $1/\alpha$.

7.2.2.2 An $O(\ln(\alpha)/\alpha)$ Upper Bound on EQUITABLE

▶ **Lemma 7.18.** *EQUITABLE is at most $O(\frac{\ln \alpha}{\alpha})$-competitive for the maximum flow problem in the worst-case.*

Proof. We look at the meander gadget of dimension $n = \lceil \alpha \rceil$ with capacities $c_1 = c_2 = 1$ but with lower bound functions

$$\ell_1(\lambda) = 1 - \lambda \quad \text{for } \lambda \leq 1 \qquad\qquad \ell_2(\lambda) = \frac{1-\lambda}{\alpha} \quad \text{for } \lambda \leq 1\,.$$

Using Lemma 6.6 we bound the total flow achieved up to stage $\lceil q \rceil$ by

$$\sum_{s=1}^{\lceil q \rceil} \mathcal{P}^{\mathrm{OPT}}(b^{(s)}) \leq (1+\ln(\alpha)) \cdot \mathcal{P}^{\mathrm{OPT}}(\ell^{(1)})$$
$$\leq (1+\ln(\alpha)) \cdot (1+\frac{n-1}{\alpha}) \leq 2 \cdot (1+\ln(\alpha)) \,.$$

From Lemma 7.16 it follows that each stage with $\ell_1^{(s)} \geq \ell_2^{(s)}$ the budgets assigned by the algorithm are completely exhausted. So until $\ell_1^{(s)} < \ell_2^{(s)} \leq \ell_2^{(1)} = 1/\alpha$ all edges from E_1 are uniformly decreased by the capacity $b_1^{(s)} \geq b_1^{(s)} = 1/N$.

Thus, after at most $\lceil q \rceil$ many stages we have $\ell_1^{(s)} \leq 1/\alpha$. But then any optimal solution with the remaining capacities has value at most $n \cdot \ell_1^{(s)} \leq \frac{\alpha+1}{\alpha} \leq 2$. In total we get $\mathcal{P}^{\mathrm{ALG}} \leq 4 + 2\ln(\alpha)$ while an optimal offline solution has value $\lceil \alpha \rceil$. ◀

7.2.2.3 Non-Competitiveness of BALANCING

▶ **Lemma 7.19.** *BALANCING is not competitive for the maximum flow problem.*

Proof. Consider the BALANCING algorithm on the meander gadget $M(n, c_1, c_2)$ for arbitrary $n \in \mathbb{N}$ where each edge $e = (u, v)$ of capacity c_2 is replaced by n internally node disjoint paths of length two with start node u and end node v. The edge capacities on these paths are set to $c_3 = 1/n$.

Let $c_1 = 1$. Since BALANCING continuously augments flow along the path with the largest bottleneck capacity, it uses the unique path R until the remaining bottleneck capacity of R drops below c_3. (any other path from s to t has bottleneck capacity c_3).

Up to that point the algorithm has achieved a total flow of less than $c_1 = 1$. And afterwards the additional flow that can be achieved is bounded by $n/n = 1$. So while BALANCING only has total flow less than 2 the optimal offline solution is n for arbitrary n. ◀

7.3 Summary

In the current chapter we proved tightness of the analysis of the proposed online algorithms GREEDY, EQUITABLE, and BALANCING on selected classic combinatorial packing problems, showing that they indeed do not achieve a better competitive ratio than in the case of general packing programs considered in Chapter 6. Competitiveness for the artificial BALANCING algorithm only holds for instances of $(1, k)$-Pack. As we have seen in the current chapter it is not suited to achieve any competitive ratio in general.

While the general upper bound of $O(1/\sqrt{\alpha})$ on the competitiveness of deterministic online algorithms was confirmed to still hold in the considered special problem classes it could further be extended to the case of arbitrary randomized online algorithms for the problem of packing spanning trees which is a relevant subproblem in a broad class of network lifetime problems.

Chapter 8

Conclusions and Open Problems

In the first part of this thesis we conducted a theoretical study on the solvability and approximability of the OFDMA scheduling problem.

While proving NP-hardness for most relevant subclasses of OFDMA scheduling we also identified several interesting instance classes that are polynomial time solvable, namely the cases of linear rate-power functions and binary channel gains. Especially the binary channel gain model could turn out to be useful in practice as it allows to generalize existing heuristics based on channel assignment routines assuming purely uniform channel gains.

An interesting open problem in this context would be the question whether Min-Power OFDMA scheduling is still efficiently solvable if channel gains are allowed to be drawn from a set $\{p, q\}$ for arbitrary gains $p, q \in \mathbb{R}_{\geq 0}$ instead of just $\{0, p\}$ covered by the binary channel gain model. As the given hardness proof requires at least three distinct channel gain values this gap remains open.

Switching to the analysis of approximability we saw that Max-Rate OFDMA is hard to approximate up to any factor better than $1/2$ for a very restricted problem class analogue to the big goods/small goods case of Santa Claus scheduling (even in the case of uniform demands). As an immediate consequence of the given reduction we obtained that Min-Power OFDMA could not be approximated up to any polynomial time computable factor with respect to the number of terminals and channels in this class. Even worse, Min-Power OFDMA could be proven to be inapproximable by any factor computable in polynomial time on the class of uniformly related channel costs (which is neither contained nor contains the class analyzed before).

For Max-Rate OFDMA, however, we were able to give simple constant factor approximation algorithms for instances with uniformly related channel costs as well as constructing a polynomial time approximation scheme for this case. The derivation of the PTAS for non-uniform demands settled an open question from [OV09] showing that the whole class of uniformly related channel costs could be approximated arbitrarily close to the optimum. Yet, it is still open if the construction of an FPTAS would be possible.

Considering general unrelated channel costs, finding good approximate solutions is much more difficult. Though the best known upper bound on the approximability of the general Max-Rate OFDMA problem is defined by the $1/2$-inapproximability result discussed above, the derivation of a constant factor approximation algorithm seems to be too ambitious as even for the (in comparison) simpler structured Santa Claus problem no sub-polynomial factor approximation algorithm is known today.

However, by adapting some techniques from the context of Santa Claus scheduling we were able to construct a simple matching-based $1/m$-approximation algorithm for the general Max-Rate OFDMA problem.

By deriving an appropriate fractionally relaxed version of the OFDMA scheduling problem we could show that a special rounding of the obtained convex programming solution allows the computation of a half-integral $1/2$-approximation in the general model. Additionally for the restricted assignment model with uniform demands a $1/n$-approximate integral solution could be achieved. Allowing half-integral solutions may at first seem like a trick from a

theoretical point of view but it could turn out to be very relevant in practical settings. While conventional relaxations of the integrality constraint might result in arbitrarily small fractional assignment values imposing infinitesimal short symbol durations or the application of subsequent heuristic rounding steps, the restriction to half-integral schedules requires only a reduction of the time-frames by a factor of $1/2$ in order to achieve an objective value with a provable good performance guarantee in the worst-case.

An obvious open problem, of course, is closing the huge gap between the best given approximation guarantee of $1/m$ and the $1/2$-inapproximability result for the computation of integral OFDMA schedules in the general unrelated cost model. The search for better approximation algorithms for interesting subclasses like the restricted assignment model could also provide valuable insights.

Furthermore, restricting the range of considered rate-power functions might be promising when searching for good approximation algorithms. As the inapproximability results mainly exploit the steepness of the fundamental Shannon rate-power function, parameterizing this attribute (e.g. by considering only functions with bounded elasticity) could lead to the design of approximation algorithms with corresponding sharp bounds on their approximation factors.

A whole field of open problems can be opened up when considering problem variants which are structurally different from the downlink OFDMA scheduling problem considered in this thesis. For instance, the uplink model with fixed individual energy budgets replacing the global energy constraint should be much more accessible to approximation techniques from the context of machine scheduling since the originally complex terminal dependencies become separable in that case.

A slightly modified version of this energy-restricted rate-allocation problem could also be analyzed under the online aspect considered in the second part of this dissertation by modeling it as a combinatorial packing problem.

Using competitive online analysis we studied the impact of gradually improving capacity estimations arising in the context of wireless ad-hoc networks. We introduced an adversarial online model for general linear packing problems capturing that capacities in network lifetime optimization problems are typically not known a priori but estimates become more and more exact when using the corresponding resources.

We showed a general upper bound of $O(1/\sqrt{\alpha})$ on the competitive ratio of any randomized online algorithm. In order to prove that no randomized algorithm with a better competitiveness can exist in our model we used Yao's principle on an appropriate worst-case instance for the problem of packing spanning trees, which directly translates into a corresponding upper bound for many network lifetime problems. Equivalent deterministic bounds were given for the multi-commodity maximum flow problem as well as for the simple $(1, k)$-Pack problem.

Several distinct online algorithms for the defined model have been proposed. Both the naive one-stage algorithm and its natural extension GREEDY were tightly shown to be $1/\alpha$-competitive. However, the more restrained EQUITABLE algorithm could be proven to asymptotically approach a competitive ratio of $\ln(\alpha)/\alpha-1$, hence significantly improving on the trivial performance guarantee.

As we have seen, already for a small number of stages (e.g. $N = \alpha^2/\ln(\alpha)$) EQUITABLE returns a solution that is at least $\ln(\alpha)/\alpha$-competitive. Together with the property that the algorithm may use approximative offline algorithms as a black box while preserving the approximation factor in terms of a multiplicative coefficient this makes the given algorithmic framework very suitable for solving network lifetime problems. Also (under reasonable assumptions on the capacities) we have seen that fractional solutions could be easily translated

into integral ones by randomized rounding techniques. Another proposed algorithm which we called ENVISAGE could be shown to exhibit the same positive properties as EQUITABLE in terms of competitiveness and complexity.

In order to complement the general $O(1/\sqrt{\alpha})$ upper bound we constructed the artificial BALANCING algorithm for solving $(1, k)$-Pack instances, which actually has a competitive ratio of $1/\sqrt{\alpha}$ on this special problem class. However, in general this algorithm is not competitive by any factor.

An interesting open problem is closing the gap between the general upper bound of $O(1/\sqrt{\alpha})$ and the lower bound of $\Omega(\ln(\alpha)/\alpha)$ on the competitiveness of deterministic and randomized online algorithms for general linear packing programs or common combinatorial packing problems like spanning tree packing or maximum flow (for the special case of single-commodity flows even no upper bound on the competitiveness is known at all). A promising candidate for the improvement of lower bounds might be the ENVISAGE algorithm which seems very resistant to the construction of worst-case instances. We conjecture that the lower bound of $\ln(\alpha)/\alpha-1$ is not tight for this algorithm.

Bibliography

[AE98] Yossi Azar and Leah Epstein. Approximation schemes for covering and scheduling in related machines. In Klaus Jansen and José Rolim, editors, *Approximation Algorithms for Combinatiorial Optimization*, volume 1444 of *Lecture Notes in Computer Science*, pages 39–47. Springer Berlin / Heidelberg, 1998. doi:10.1007/BFb0053962.

[AFS08] Arash Asadpour, Uriel Feige, and Amin Saberi. Santa Claus meets hypergraph matchings. In Ashish Goel, Klaus Jansen, JoséD.P. Rolim, and Ronitt Rubinfeld, editors, *Approximation, Randomization and Combinatorial Optimization. Algorithms and Techniques*, volume 5171 of *Lecture Notes in Computer Science*, pages 10–20. Springer Berlin Heidelberg, 2008. doi:10.1007/978-3-540-85363-3_2.

[Alb03] Susanne Albers. Online algorithms: a survey. *Mathematical Programming*, 97:3–26, 2003. doi:10.1007/s10107-003-0436-0.

[Amb05] Christoph Ambühl. An optimal bound for the MST algorithm to compute energy efficient broadcast trees in wireless networks. In Luís Caires, Giuseppe Italiano, Luís Monteiro, Catuscia Palamidessi, and Moti Yung, editors, *Automata, Languages and Programming*, volume 3580 of *Lecture Notes in Computer Science*, pages 100–100. Springer Berlin / Heidelberg, 2005. doi:10.1007/11523468_92.

[AMO93] Ravindra K. Ahuja, Thomas L. Magnanti, and James B. Orlin. *Network flows: theory, algorithms, and applications*. Prentice Hall, 1993.

[AS07] Arash Asadpour and Amin Saberi. An approximation algorithm for max-min fair allocation of indivisible goods. In *Proceedings of the thirty-ninth annual ACM symposium on Theory of computing*, STOC '07, pages 114–121, New York, NY, USA, 2007. ACM. doi:10.1145/1250790.1250808.

[BCG09] MohammadHossein Bateni, Moses Charikar, and Venkatesan Guruswami. Maxmin allocation via degree lower-bounded arborescences. In *Proceedings of the 41st annual ACM symposium on Theory of computing*, STOC '09, pages 543–552, New York, NY, USA, 2009. ACM. doi:10.1145/1536414.1536488.

[BD05] Ivona Bezáková and Varsha Dani. Allocating indivisible goods. *SIGecom Exch.*, 5(3):11–18, April 2005. doi:10.1145/1120680.1120683.

[BEY98] Allan Borodin and Ran El-Yaniv. *Online Computation and Competitive Analysis*, volume 2. Cambridge University Press, 1998.

[Bin90] John A.C. Bingham. Multicarrier modulation for data transmission: an idea whose time has come. *Communications Magazine, IEEE*, 28(5):5 –14, may 1990. doi:10.1109/35.54342.

[BL97] John R. Birge and François Louveaux. *Introduction to Stochastic Programming*. Springer Series in Operations Research Series. Springer London, Limited, 1997.

[BN05] Niv Buchbinder and Joseph (Seffi) Naor. Online primal-dual algorithms for covering and packing problems. In Gerth Brodal and Stefano Leonardi, editors, *Algorithms – ESA 2005*, volume 3669 of *Lecture Notes in Computer Science*, pages 689–701. Springer Berlin / Heidelberg, 2005. doi:10.1007/11561071_61.

[BN09] Niv Buchbinder and Joseph (Seffi) Naor. Online primal-dual algorithms for covering and packing. *Math. Oper. Res.*, 34(2):270–286, May 2009. doi:10.1287/moor.1080.0363.

[BS06] Nikhil Bansal and Maxim Sviridenko. The Santa Claus problem. In *Proceedings of the thirty-eighth annual ACM symposium on Theory of computing*, STOC '06, pages 31–40, New York, NY, USA, 2006. ACM. doi:10.1145/1132516.1132522.

[BSE04] Ahmad R.S. Bahai, Burton R. Saltzberg, and Mustafa Ergen. *Multi-carrier Digital Communications: Theory And Applications Of OFDM*. Information technology–transmission, processing, and storage. Springer, 2004.

[BV04] Stephen Boyd and Lieven Vandenberghe. *Convex Optimization*. Cambridge University Press, 2004.

[Cam98] Jorge Campello. Optimal discrete bit loading for multicarrier modulation systems. In *Information Theory, 1998. Proceedings. 1998 IEEE International Symposium on*, page 193, aug 1998. doi:10.1109/ISIT.1998.708791.

[CCC+98] Moses Charikar, Chandra Chekuri, To-yat Cheung, Zuo Dai, Ashish Goel, Sudipto Guha, and Ming Li. Approximation algorithms for directed steiner problems. In *Proceedings of the ninth annual ACM-SIAM symposium on Discrete algorithms*, SODA '98, pages 192–200, Philadelphia, PA, USA, 1998. Society for Industrial and Applied Mathematics.

[CCK09] Deeparnab Chakrabarty, Julia Chuzhoy, and Sanjeev Khanna. On allocating goods to maximize fairness. In *Foundations of Computer Science, 2009. FOCS '09. 50th Annual IEEE Symposium on*, pages 107 –116, oct. 2009. doi:10.1109/FOCS.2009.51.

[CCP+01] Andrea Clementi, Pilu Crescenzi, Paolo Penna, Gianluca Rossi, and Paola Vocca. On the complexity of computing minimum energy consumption broadcast subgraphs. In Afonso Ferreira and Horst Reichel, editors, *STACS 2001*, volume 2010 of *Lecture Notes in Computer Science*, pages 121–131. Springer Berlin / Heidelberg, 2001. doi:10.1007/3-540-44693-1_11.

[CFM07] Ioannis Caragiannis, Michele Flammini, and Luca Moscardelli. An exponential improvement on the MST heuristic for minimum energy broadcasting in ad hoc wireless networks. In Lars Arge, Christian Cachin, Tomasz Jurdzinski, and Andrzej Tarlecki, editors, *Automata, Languages and Programming*, volume 4596 of *Lecture Notes in Computer Science*, pages 447–458. Springer Berlin / Heidelberg, 2007. doi:10.1007/978-3-540-73420-8_40.

[CH89] Wen-Tsuen Chen and Nen-Fu Huang. The strongly connecting problem on multihop packet radio networks. *Communications, IEEE Transactions on*, 37(3):293 –295, mar 1989. doi:10.1109/26.20105.

[ČHE02] Mario Čagalj, Jean-Pierre Hubaux, and Christian Enz. Minimum-energy broadcast in all-wireless networks: NP-completeness and distribution issues. In *Proceedings of the 8th annual international conference on Mobile computing and networking*, MobiCom '02, pages 172–182, New York, NY, USA, 2002. ACM. doi:10.1145/570645.570667.

[ČHE05] Mario Čagalj, Jean-Pierre Hubaux, and Christian Enz. Energy-efficient broadcasting in all-wireless networks. *Wirel. Netw.*, 11(1-2):177–188, January 2005. doi:10.1007/s11276-004-4754-9.

[CHP+02] Andrea E.F. Clementi, Gurvan Huiban, Paolo Penna, Gianluca Rossi, and Yann C. Verhoeven. Some recent theoretical advances and open questions on energy consumption in ad-hoc wireless networks. In *ARACNE'02, 3rd workshop on Approximation and Randomization Algorithms in Communication Networks*, pages 23–38. Springer-Verlag, 2002.

[CKOZ03] Gruia Calinescu, Sanjiv Kapoor, Alexander Olshevsky, and Alexander Zelikovsky. Network lifetime and power assignment in ad hoc wireless networks. In Giuseppe Di Battista and Uri Zwick, editors, *Algorithms - ESA 2003*, volume 2832 of *Lecture Notes in Computer Science*, pages 114–126. Springer Berlin / Heidelberg, 2003. doi:10.1007/978-3-540-39658-1_13.

[CT91] Thomas M. Cover and Joy A. Thomas. *Elements of information theory*. Wiley series in telecommunications. Wiley, 1991.

[CT99] Jae-Hwan Chang and Leandros Tassiulas. Routing for maximum system lifetime in wireless ad-hoc networks. In *in 37-th Annual Allerton Conference on Communication, Control, and Computing*, 1999.

[CT00] Jae-Hwan Chang and Leandros Tassiulas. Energy conserving routing in wireless ad-hoc networks. In *INFOCOM 2000. Nineteenth Annual Joint Conference of the IEEE Computer and Communications Societies. Proceedings. IEEE*, volume 1, pages 22 –31 vol.1, 2000. doi:10.1109/INFCOM.2000.832170.

[CV02] Robert Carr and Santosh Vempala. Randomized metarounding. *Random Structures & Algorithms*, 20(3):343–352, 2002. doi:10.1002/rsa.10033.

[DM08] Bernard Dacorogna and Pierre Maréchal. The role of perspective functions in convexity, polyconvexity, rank one convexity and separate convexity. *Journal of Convex Analysis*, 15:271–284, 2008.

[EK72] Jack Edmonds and Richard M. Karp. Theoretical improvements in algorithmic efficiency for network flow problems. *J. ACM*, 19(2):248–264, April 1972. doi:10.1145/321694.321699.

[ELN+11] Michael Elkin, Yuval Lando, Zeev Nutov, Michael Segal, and Hanan Shpungin. Novel algorithms for the network lifetime problem in wireless settings. *Wireless Networks*, 17:397–410, 2011. doi:10.1007/s11276-010-0287-6.

[ES04] Leah Epstein and Jiri Sgall. Approximation schemes for scheduling on uniformly related and identical parallel machines. *Algorithmica*, 39:43–57, 2004. doi:10.1007/s00453-003-1077-7.

[Fei08] Uriel Feige. On allocations that maximize fairness. In *Proceedings of the nineteenth annual ACM-SIAM symposium on Discrete algorithms*, SODA '08, pages 287–293, Philadelphia, PA, USA, 2008. Society for Industrial and Applied Mathematics.

[FMR08] Anke Feiten, Rudolf Mathar, and Michael Reyer. Rate and power allocation for multiuser OFDM: An effective heuristic verified by branch-and-bound. *Wireless Communications, IEEE Transactions on*, 7(1):60 –64, jan. 2008. doi:10.1109/TWC.2008.060559.

[FT87] Michael L. Fredman and Robert Endre Tarjan. Fibonacci heaps and their uses in improved network optimization algorithms. *J. ACM*, 34(3):596–615, July 1987. doi:10.1145/28869.28874.

[GGLS08] Naveen Garg, Anupam Gupta, Stefano Leonardi, and Piotr Sankowski. Stochastic analyses for online combinatorial optimization problems. In *Proceedings of the nineteenth annual ACM-SIAM symposium on Discrete algorithms*, SODA '08, pages 942–951, Philadelphia, PA, USA, 2008. Society for Industrial and Applied Mathematics.

[GK98] Naveen Garg and Jochen Könemann. Faster and simpler algorithms for multicommodity flow and other fractional packing problems. In *Foundations of Computer Science, 1998. Proceedings.39th Annual Symposium on*, pages 300 –309, nov 1998. doi:10.1109/SFCS.1998.743463.

[GK07] Naveen Garg and Jochen Könemann. Faster and simpler algorithms for multicommodity flow and other fractional packing problems. *SIAM Journal on Computing*, 37(2):630–652, 2007. doi:10.1137/S0097539704446232.

[Gol05] Daniel Golovin. Max-min fair allocation of indivisible goods. Technical Report CMU-CS-05-144, School of Computer Science, Carnegie Mellon University, June 2005.

[HS87] Dorit S. Hochbaum and David B. Shmoys. Using dual approximation algorithms for scheduling problems theoretical and practical results. *J. ACM*, 34(1):144–162, January 1987. doi:10.1145/7531.7535.

[HS88] Dorit S. Hochbaum and David B. Shmoys. A polynomial approximation scheme for scheduling on uniform processors: Using the dual approximation approach. *SIAM Journal on Computing*, 17(3):539–551, 1988. doi:10.1137/0217033.

[HSS11] Bernhard Haeupler, Barna Saha, and Aravind Srinivasan. New constructive aspects of the Lovász Local Lemma. *J. ACM*, 58(6):28:1–28:28, December 2011. doi:10.1145/2049697.2049702.

[IKMM04] Nicole Immorlica, David Karger, Maria Minkoff, and Vahab S. Mirrokni. On the costs and benefits of procrastination: approximation algorithms for stochastic combinatorial optimization problems. In *Proceedings of the fifteenth annual ACM-SIAM symposium on Discrete algorithms*, SODA '04, pages 691–700, Philadelphia, PA, USA, 2004. Society for Industrial and Applied Mathematics.

[JL03] Jiho Jang and Kwang Bok Lee. Transmit power adaptation for multiuser OFDM systems. *Selected Areas in Communications, IEEE Journal on*, 21(2):171 – 178, feb 2003. doi:10.1109/JSAC.2002.807348.

[JMS03] Kamal Jain, Mohammad Mahdian, and Mohammad R. Salavatipour. Packing Steiner trees. In *Proceedings of the fourteenth annual ACM-SIAM symposium on Discrete algorithms*, SODA '03, pages 266–274, Philadelphia, PA, USA, 2003. Society for Industrial and Applied Mathematics.

[JVG04] Nihar Jindal, Sriram Vishwanath, and Andrea Goldsmith. On the duality of Gaussian multiple-access and broadcast channels. *Information Theory, IEEE Transactions on*, 50(5):768 – 783, may 2004. doi:10.1109/TIT.2004.826646.

[Kal89] Irving Kalet. The multitone channel. *Communications, IEEE Transactions on*, 37(2):119 –124, feb 1989. doi:10.1109/26.20079.

[KDN03] Konstantinos Kalpakis, Koustuv Dasgupta, and Parag Namjoshi. Efficient algorithms for maximum lifetime data gathering and aggregation in wireless sensor networks. *Computer Networks*, 42(6):697 – 716, 2003. doi:10.1016/S1389-1286(03)00212-3.

[KKKP97] Lefteris Kirousis, Evangelos Kranakis, Danny Krizanc, and Andrzej Pelc. Power consumption in packet radio networks. In Rüdiger Reischuk and Michel Morvan, editors, *STACS 97*, volume 1200 of *Lecture Notes in Computer Science*, pages 363–374. Springer Berlin / Heidelberg, 1997. doi:10.1007/BFb0023473.

[KLL03] Didem Kivanc, Guoqing Li, and Hui Liu. Computationally efficient bandwidth allocation and power control for OFDMA. *Wireless Communications, IEEE Transactions on*, 2(6):1150 – 1158, nov. 2003. doi:10.1109/TWC.2003.819016.

[KM97] Alexander V. Karzanov and S. Thomas McCormick. Polynomial methods for separable convex optimization in unimodular linear spaces with applications. *SIAM J. Comput.*, 26(4):1245–1275, August 1997. doi:10.1137/S0097539794263695.

[KP83] Rolf G. Karlsson and Patricio V. Poblete. An O(mlog logD) algorithm for

shortest paths. *Discrete Applied Mathematics*, 6(1):91 – 93, 1983. doi:10.1016/0166-218X(83)90104-X.

[KP00] Bala Kalyanasundaram and Kirk Pruhs. Speed is as powerful as clairvoyance. *J. ACM*, 47(4):617–643, July 2000. doi:10.1145/347476.347479.

[KP03] Intae Kang and Radha Poovendran. Maximizing static network lifetime of wireless broadcast ad hoc networks. In *Communications, 2003. ICC '03. IEEE International Conference on*, volume 3, pages 2256 – 2261 vol.3, may 2003. doi:10.1109/ICC.2003.1204068.

[KP05] Intae Kang and Radha Poovendran. Maximizing network lifetime of broadcasting over wireless stationary ad hoc networks. *Mobile Networks and Applications*, 10:879–896, 2005. doi:10.1007/s11036-005-4445-5.

[KPL06] Inhyoung Kim, In-Soon Park, and Yong H. Lee. Use of linear programming for dynamic subcarrier and bit allocation in multiuser OFDM. *Vehicular Technology, IEEE Transactions on*, 55(4):1195 –1207, july 2006. doi:10.1109/TVT.2006.877490.

[KV06] Bernhard Korte and Jens Vygen. *Combinatorial optimization: theory and algorithms*. Algorithms and combinatorics. Springer, 2006.

[Lia02] Weifa Liang. Constructing minimum-energy broadcast trees in wireless ad hoc networks. In *Proceedings of the 3rd ACM international symposium on Mobile ad hoc networking & computing*, MobiHoc '02, pages 112–122, New York, NY, USA, 2002. ACM. doi:10.1145/513800.513815.

[LN01] Fulu Li and Ioanis Nikolaidis. On minimum-energy broadcasting in all-wireless networks. In *Local Computer Networks, 2001. Proceedings. LCN 2001. 26th Annual IEEE Conference on*, pages 193 –202, 2001. doi:10.1109/LCN.2001.990787.

[LST90] Jan Karel Lenstra, David B. Shmoys, and Éva Tardos. Approximation algorithms for scheduling unrelated parallel machines. *Mathematical Programming*, 46:259–271, 1990. doi:10.1007/BF01585745.

[MPF96] Kia Makki, Niki Pissinou, and Ophir Frieder. Efficient solutions to multicast routing in communication networks. *Mobile Networks and Applications*, 1:221–232, 1996. doi:10.1007/BF01193338.

[MS00] Todd K. Moon and Wynn C. Stirling. *Mathematical methods and algorithms for signal processing*. Prentice Hall, 2000.

[ORV12] Marcel Ochel, Klaus Radke, and Berthold Vöcking. Online packing with gradually improving capacity estimations and applications to network lifetime maximization. In Artur Czumaj, Kurt Mehlhorn, Andrew Pitts, and Roger Wattenhofer, editors, *Automata, Languages, and Programming*, volume 7392 of *Lecture Notes in Computer Science*, pages 648–659. Springer Berlin / Heidelberg, 2012. doi:10.1007/978-3-642-31585-5_57.

[OV09] Marcel Ochel and Berthold Vöcking. Approximability of OFDMA scheduling. In Amos Fiat and Peter Sanders, editors, *Algorithms - ESA 2009*, volume 5757 of *Lecture Notes in Computer Science*, pages 385–396. Springer Berlin / Heidelberg, 2009. doi:10.1007/978-3-642-04128-0_35.

[OY04] Ariel Orda and Ben-Ami Yassour. Maximum-lifetime routing algorithms for wireless networks. Technical Report CCIT Pub. No. 493, Dept. of Electrical Engineering, Technion, June 2004.

[OY05] Ariel Orda and Ben-Ami Yassour. Maximum-lifetime routing algorithms for networks with omnidirectional and directional antennas. In *Proceedings of the*

6th ACM international symposium on Mobile ad hoc networking and computing, MobiHoc '05, pages 426–437, New York, NY, USA, 2005. ACM. doi:10.1145/1062689.1062742.

[PS82] Christos H. Papadimitriou and Kenneth Steiglitz. *Combinatorial optimization: algorithms and complexity*. Prentice-Hall, Inc., Upper Saddle River, NJ, USA, 1982.

[PS05] Joongseok Park and Sartaj Sahni. Maximum lifetime broadcasting in wireless networks. *Computers, IEEE Transactions on*, 54(9):1081 – 1090, sept. 2005. doi:10.1109/TC.2005.148.

[RC00] Wonjong Rhee and John M. Cioffi. Increase in capacity of multiuser OFDM system using dynamic subchannel allocation. In *Vehicular Technology Conference Proceedings, 2000. VTC 2000-Spring Tokyo. 2000 IEEE 51st*, volume 2, pages 1085 –1089 vol.2, 2000. doi:10.1109/VETECS.2000.851292.

[Rey08] Michael Reyer. *Optimierungsalgorithmen zur Ressourcenallokation in OFDMA-Systemen*. PhD thesis, RWTH Aachen, 2008.

[RM99] Volkan Rodoplu and Teresa H. Meng. Minimum energy mobile wireless networks. *Selected Areas in Communications, IEEE Journal on*, 17(8):1333 –1344, aug 1999. doi:10.1109/49.779917.

[RVR03] Ravishankar Rao, Sarma Vrudhula, and Daler N. Rakhmatov. Battery modeling for energy aware system design. *Computer*, 36(12):77 – 87, dec. 2003. doi:10.1109/MC.2003.1250886.

[SAE05] Zukang Shen, J.G. Andrews, and B.L. Evans. Adaptive resource allocation in multiuser OFDM systems with proportional rate constraints. *Wireless Communications, IEEE Transactions on*, 4(6):2726 – 2737, nov. 2005. doi:10.1109/TWC.2005.858010.

[Sch03] Alexander Schrijver. *Combinatorial Optimization. Polyhedra and Efficiency*. Springer, 2003.

[Seg08] Michael Segal. Fast algorithm for multicast and data gathering in wireless networks. *Information Processing Letters*, 107(1):29 – 33, 2008. doi:10.1016/j.ipl.2007.12.007.

[Sri99] Aravind Srinivasan. Approximation algorithms via randomized rounding: A survey. In *Series in Advanced Topics in Mathematics, Polish Scientific Publishers PWN*, pages 9–71, 1999.

[SS04] David B. Shmoys and Chaitanya Swamy. Stochastic optimization is (almost) as easy as deterministic optimization. In *Foundations of Computer Science, 2004. Proceedings. 45th Annual IEEE Symposium on*, pages 228 – 237, oct. 2004. doi:10.1109/FOCS.2004.62.

[ST93] David B. Shmoys and Éva Tardos. An approximation algorithm for the generalized assignment problem. *Mathematical Programming*, 62:461–474, 1993. doi:10.1007/BF01585178.

[SV02] Leen Stougie and Arjen P.A. Vestjens. Randomized algorithms for on-line scheduling problems: how low can't you go? *Operations Research Letters*, 30(2):89 – 96, 2002. doi:10.1016/S0167-6377(01)00115-8.

[SYKC06] Kibeom Seong, David D. Yu, Youngjae Kim, and John M. Cioffi. Cth03-5: Optimal resource allocation via geometric programming for OFDM broadcast and multiple access channels. In *Global Telecommunications Conference, 2006. GLOBECOM '06. IEEE*, pages 1 –5, 27 2006-dec. 1 2006. doi:10.1109/GLOCOM.2006.50.

[Tom71] N. Tomizawa. On some techniques useful for solution of transportation network problems. *Networks*, 1(2):173–194, 1971. doi:10.1002/net.3230010206.

[VD06] Manish Vemulapalli and Soura Dasgupta. NP-hardness of bit allocation in multiuser multicarrier communications. In *Proceedings of EUSIPCO-2006*, 2006.

[VW10] José Verschae and Andreas Wiese. On the configuration-LP for scheduling on unrelated machines. Technical Report 025-2010, Technische Universität Berlin, November 2010.

[VW11] José Verschae and Andreas Wiese. On the configuration-LP for scheduling on unrelated machines. In Camil Demetrescu and Magnús Halldórsson, editors, *Algorithms – ESA 2011*, volume 6942 of *Lecture Notes in Computer Science*, pages 530–542. Springer Berlin / Heidelberg, 2011. doi:10.1007/978-3-642-23719-5_45.

[WCLF01] Peng-Jun Wan, Gruia Calinescu, X.-Y. Li, and Ophir Frieder. Minimum-energy broadcast routing in static ad hoc wireless networks. In *INFOCOM 2001. Twentieth Annual Joint Conference of the IEEE Computer and Communications Societies. Proceedings. IEEE*, volume 2, pages 1162 –1171 vol.2, 2001. doi:10.1109/INFCOM.2001.916310.

[WCLM99] Cheong Yui Wong, Roger S. Cheng, Khaled Ben Letaief, and Ross D. Murch. Multiuser OFDM with adaptive subcarrier, bit, and power allocation. *Selected Areas in Communications, IEEE Journal on*, 17(10):1747 –1758, oct 1999. doi:10.1109/49.793310.

[WM64] David W. Walkup and M. Donald MacLaren. A multiple-assignment problem. AD0602672, BOEING Scientific Research Labs SEATTLEWA, April 1964. Mathematical note No. 347.

[WNE00] Jeffrey E. Wieselthier, Gam D. Nguyen, and Anthony Ephremides. On the construction of energy-efficient broadcast and multicast trees in wireless networks. In *INFOCOM 2000. Nineteenth Annual Joint Conference of the IEEE Computer and Communications Societies. Proceedings. IEEE*, volume 2, pages 585 –594 vol.2, 2000. doi:10.1109/INFCOM.2000.832232.

[WNE01] Jeffrey E. Wieselthier, Gam D. Nguyen, and Anthony Ephremides. Algorithms for energy-efficient multicasting in static ad hoc wireless networks. *Mob. Netw. Appl.*, 6(3):251–263, June 2001. doi:10.1023/A:1011478717164.

[Woe97] Gerhard J. Woeginger. A polynomial-time approximation scheme for maximizing the minimum machine completion time. *Operations Research Letters*, 20(4):149 – 154, 1997. doi:10.1016/S0167-6377(96)00055-7.

[WSEA04] Ian C. Wong, Zukang Shen, Brian L. Evans, and Jeffrey G. Andrews. A low complexity algorithm for proportional resource allocation in OFDMA systems. In *Signal Processing Systems, 2004. SIPS 2004. IEEE Workshop on*, pages 1 – 6, oct. 2004. doi:10.1109/SIPS.2004.1363015.

[YC00] Wei Yu and John M. Cioffi. FDMA capacity of the Gaussian multiple access channel with ISI. In *Communications, 2000. ICC 2000. 2000 IEEE International Conference on*, volume 3, pages 1365 –1369 vol.3, 2000. doi:10.1109/ICC.2000.853720.