Preface

The goal of this textbook is twofold. First, the book serves as an introduction to the field of parameterized algorithms and complexity accessible to graduate students and advanced undergraduate students. Second, it contains a clean and coherent account of some of the most recent tools and techniques in the area.

Parameterized algorithmics analyzes running time in finer detail than classical complexity theory: instead of expressing the running time as a function of the input size only, dependence on one or more parameters of the input instance is taken into account. While there were examples of nontrivial parameterized algorithms in the literature, such as Lenstra's algorithm for integer linear programming [319] or the disjoint paths algorithm of Robertson and Seymour [402], it was only in the late 1980s that Downey and Fellows [149], building on joint work with Langston [180, 182, 183], proposed the systematic exploration of parameterized algorithms. Downey and Fellows laid the foundations of a fruitful and deep theory, suitable for reasoning about the complexity of parameterized algorithms. Their early work demonstrated that fixed-parameter tractability is a ubiquitous phenomenon, naturally arising in various contexts and applications. The parameterized view on algorithms has led to a theory that is both mathematically beautiful and practically applicable. During the 30 years of its existence, the area has transformed into a mainstream topic of theoretical computer science. A great number of new results have been achieved, a wide array of techniques have been created, and several open problems have been solved. At the time of writing, Google Scholar gives more than 4000 papers containing the term "fixed-parameter tractable". While a full overview of the field in a single volume is no longer possible, our goal is to present a selection of topics at the core of the field, providing a key for understanding the developments in the area.

Why This Book?

The idea of writing this book arose after we decided to organize a summer school on parameterized algorithms and complexity in Będlewo in August 2014. While planning the school, we realized that there is no textbook that contains the material that we wanted to cover. The classical book of Downey and Fellows [153] summarizes the state of the field as of 1999. This book was the starting point of a new wave of research in the area, which is obviously not covered by this classical text. The area has been developing at such a fast rate that even the two books that appeared in 2006, by Flum and Grohe [189] and Niedermeier [376], do not contain some of the new tools and techniques that we feel need to be taught in a modern introductory course. Examples include the lower bound techniques developed for kernelization in 2008, methods introduced for faster dynamic programming on tree decompositions (starting with *Cut & Count* in 2011), and the use of algebraic tools for problems such as Longest Path. The book of Flum and Grohe [189] focuses to a large extent on complexity aspects of parameterized algorithmics from the viewpoint of logic, while the material we wanted to cover in the school is primarily algorithmic, viewing complexity as a tool for proving that certain kinds of algorithms do not exist. The book of Niedermeier [376] gives a gentle introduction to the field and some of the basic algorithmic techniques. In 2013, Downey and Fellows [154] published the second edition of their classical text, capturing the development of the field from its nascent stages to the most recent results. However, the book does not treat in detail many of the algorithmic results we wanted to teach, such as how one can apply important separators for Edge Multiway Cut and Directed Feedback Vertex Set, linear programming for Almost 2-SAT, *Cut & Count* and its deterministic counterparts to obtain faster algorithms on tree decompositions, algorithms based on representative families of matroids, kernels for Feedback Vertex Set, and some of the reductions related to the use of the Strong Exponential Time Hypothesis.

Our initial idea was to prepare a collection of lecture notes for the school, but we realized soon that a coherent textbook covering all basic topics in equal depth would better serve our purposes, as well as the purposes of those colleagues who would teach a semester course in the future. We have organized the material into chapters according to techniques. Each chapter discusses a certain algorithmic paradigm or lower bound methodology. This means that the same algorithmic problem may be revisited in more than one chapter, demonstrating how different techniques can be applied to it. Thanks to the rapid growth of the field, it is now nearly impossible to cover every relevant result in a single textbook. Therefore, we had to carefully select what to present at the school and include in the book. Our goal was to include a self-contained and teachable exposition of what we believe are the basic techniques of the field, at the expense of giving a complete survey of the area. A consequence of this is that we do not always present the strongest result for a particular problem. Nevertheless, we would like to point out that for many problems the book actually contains the state of the art and brings the reader to the frontiers of research. We made an effort to present full proofs for most of the results, where this was feasible within the textbook format. We used the opportunity of writing this textbook to revisit some of the results in the literature and, using the benefit of hindsight, to present them in a modern and didactic way.

At the end of each chapter we provide sections with exercises, hints to exercises and bibliographical notes. Many of the exercises complement the main narrative and cover important results which have to be omitted due to space constraints. We use (✎) and (☠) to identify easy and challenging exercises. Following the common practice for textbooks, we try to minimize the occurrence of bibliographical and historical references in the main text by moving them to bibliographic notes. These notes can also guide the reader on to further reading.

Organization of the Book

The book is organized into three parts. The first seven chapters give the basic toolbox of parameterized algorithms, which, in our opinion, every course on the subject should cover. The second part, consisting of Chapters 8-12, covers more advanced algorithmic techniques that are featured prominently in current research, such as important separators and algebraic methods. The third part introduces the reader to the theory of lower bounds: the intractability theory of parameterized complexity, lower bounds based on the Exponential Time Hypothesis, and lower bounds on kernels. We adopt a very pragmatic viewpoint in these chapters: our goal is to help the algorithm designer by providing evidence that certain algorithms are unlikely to exist, without entering into complexity theory in deeper detail. Every chapter is accompanied by exercises, with hints for most of them. Bibliographic notes point to the original publications, as well as to related work.

- Chapter 1 motivates parameterized algorithms and the notion of fixed-parameter tractability with some simple examples. Formal definitions of the main concepts are introduced.
- Kernelization is the first algorithmic paradigm for fixed-parameter tractability that we discuss. Chapter 2 gives an introduction to this technique.
- Branching and bounded-depth search trees are the topic of Chapter 3. We discuss both basic examples and more advanced applications based on linear programming relaxations, showing the fixed-parameter tractability of, e.g., Odd Cycle Transversal and Almost 2-SAT.
- Iterative compression is a very useful technique for deletion problems. Chapter 4 introduces the technique through three examples, including Feedback Vertex Set and Odd Cycle Transversal.

- Chapter 5 discusses techniques for parameterized algorithms that use randomization. The classic color coding technique for Longest Path will serve as an illustrative example.
- Chapter 6 presents a collection of techniques that belong to the basic toolbox of parameterized algorithms: dynamic programming over subsets, integer linear programming (ILP), and the use of well-quasi-ordering results from graph minors theory.
- Chapter 7 introduces treewidth, which is a graph measure that has important applications for parameterized algorithms. We discuss how to use dynamic programming and Courcelle's theorem to solve problems on graphs of bounded treewidth and how these algorithms are used more generally, for example, in the context of bidimensionality for planar graphs.
- Chapter 8 presents results that are based on a combinatorial bound on the number of so-called "important separators". We use this bound to show the fixed-parameter tractability of problems such as Edge Multicut and Directed Feedback Vertex Set. We also discuss randomized sampling of important cuts.
- The kernels presented in Chapter 9 form a representative sample of more advanced kernelization techniques. They demonstrate how the use of min-max results from graph theory, the probabilistic method, and the properties of planar graphs can be exploited in kernelization.
- Two different types of algebraic techniques are discussed in Chapter 10: algorithms based on the inclusion–exclusion principle and on polynomial identity testing. We use these techniques to present the fastest known parameterized algorithms for Steiner Tree and Longest Path.
- In Chapter 11, we return to dynamic programming algorithms on graphs of bounded treewidth. This chapter presents three methods (subset convolution, *Cut & Count,* and a rank-based approach) for speeding up dynamic programming on tree decompositions.
- The notion of matroids is a fundamental concept in combinatorics and optimization. Recently, matroids have also been used for kernelization and parameterized algorithms. Chapter 12 gives a gentle introduction to some of these developments.
- Chapter 13 presents tools that allow us to give evidence that certain problems are not fixed-parameter tractable. The chapter introduces parameterized reductions and the W-hierarchy, and gives a sample of hardness results for various concrete problems.
- Chapter 14 uses the (Strong) Exponential Time Hypothesis to give running time lower bounds that are more refined than the bounds in Chapter 13. In many cases, these stronger complexity assumptions allow us to obtain lower bounds essentially matching the best known algorithms.
- Chapter 15 gives the tools for showing lower bounds for kernelization algorithms. We use methods of composition and polynomial-parameter transformations to show that certain problem, such as Longest Path, do not admit polynomial kernels.

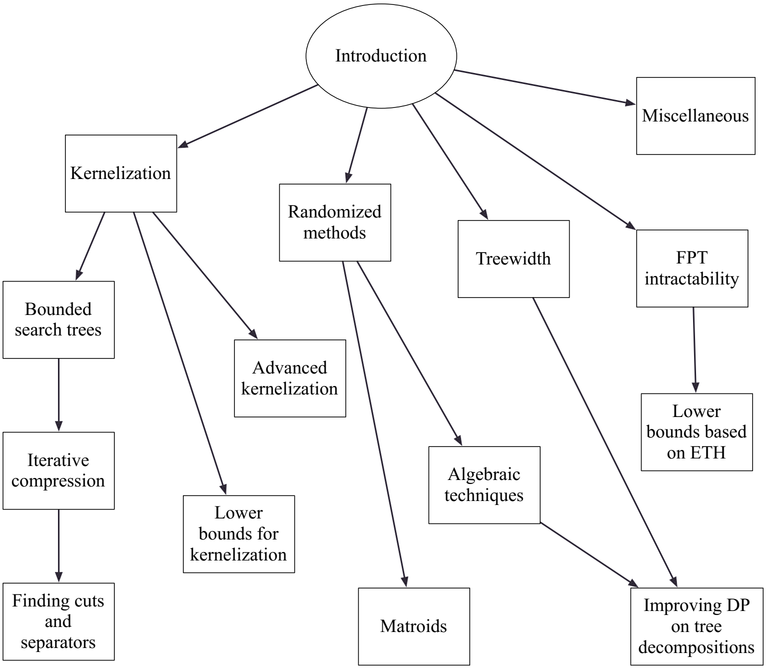

Fig. 0.1: Dependencies between the chapters

As in any textbook we will assume that the reader is familiar with the content of one chapter before moving to the next. On the other hand, for most chapters it is not necessary for the reader to have read *all* preceeding chapters. Thus the book does not have to be read linearly from beginning to end. Figure 0.1 depicts the dependencies between the different chapters. For example, the chapters on Iterative Compression and Bounded Search Trees are considered necessary prerequisites to understand the chapter on finding cuts and separators.

Using the Book for Teaching

A course on parameterized algorithms should cover most of the material in Part I, except perhaps the more advanced sections marked with an asterisk. In Part II, the instructor may choose which chapters and which sections to teach based on his or her preferences. Our suggestion for a coherent set of topics from Part II is the following:

- All of Chapter 8, as it is relatively easily teachable. The sections of this chapter are based on each other and hence should be taught in this order, except that perhaps Section 8.4 and Sections 8.5–8.6 are interchangeable.
- Chapter 9 contains four independent sections. One could select Section 9.1 (Feedback Vertex Set) and Section 9.3 (Connected Vertex Cover on planar graphs) in a first course.
- From Chapter 10, we suggest presenting Section 10.1 (inclusion–exclusion principle), and Section 10.4.1 (Longest Path in time $2^k \cdot n^{\mathcal{O}(1)}$).
- From Chapter 11, we recommend teaching Sections 11.2.1 and *11.2.2, as they are most illustrative for the recent developments on algorithms on tree decompositions.
- From Chapter 12 we recommend teaching Section 12.3. If the students are unfamiliar with matroids, Section 12.1 provides a brief introduction to the topic.

Part III gives a self-contained exposition of the lower bound machinery. In this part, the sections not marked with an asterisk give a set of topics that can form the complexity part of a course on parameterized algorithms. In some cases, we have presented multiple reductions showcasing the same kind of lower bounds; the instructor can choose from these examples according to the needs of the course. Section 14.4.1 contains some more involved proofs, but one can give a coherent overview of this section even while omitting most of the proofs.

Bergen, Budapest, Chennai, Warsaw
June 2015
Marek Cygan, Fedor V. Fomin, Łukasz Kowalik, Daniel Lokshtanov, Dániel Marx, Marcin Pilipczuk, Michał Pilipczuk and Saket Saurabh

http://www.springer.com/978-3-319-21274-6

Parameterized Algorithms
Cygan, M.; Fomin, F.V.; Kowalik, Ł.; Lokshtanov, D.; Marx, D.; Pilipczuk, M.; Pilipczuk, M.; Saurabh, S.
2015, XVII, 613 p. 84 illus., 25 illus. in color., Hardcover
ISBN: 978-3-319-21274-6