# Preface

In the first $N$ math classes of your career, you can be misled as to what the world is truly like. How? You're given exact problems and you find exact solutions. The real world is far more complicated, though this leads to interesting mathematics to handle these issues. Often, we cannot solve problems exactly; moreover, the problems we try to solve are themselves merely approximations to the world! We are forced to develop techniques to approximate not just the solution, but even the statement of the problem. Additionally, we often need the solutions quickly; it does us little to no good to have a method that requires years to reach a solution, as businesses cannot afford to wait that long.

This book is meant to serve as an introduction to Optimization Theory, which has a large overlap with Operations Research (OR). It has been used for a senior capstone class, tying together much of the math seen in the undergraduate curriculum, as well as for a transition course on the issues in applying mathematics to the real world. The choice depends on the interests of instructor and students, and leads to different topics being emphasized; we highlight some possibilities for various semester classes in the next section.

While this book touches on some standard problems in OR, our subject matter is deliberately wider, as one of our goals is to showcase commonalities between very different areas of mathematics. Since many problems in OR are related to finding optimal choices, there is a natural connection between these themes, which gives us the freedom to revisit much of the mathematics you may have seen with a different perspective. Thus, large parts of this book are concerned with developing mathematical theory, while other parts deal with implementing it. These two themes build naturally on each other and show how each influences the development of the other.

Operations Research was born (or perhaps it's better to say came of age) as a discipline during the tumultuous events of World War II as a way of efficiently finding optimal or close to optimal solutions. Throughout the chapters below, we constantly re-examine problems you have seen, but now keep in mind the computational cost of our approach. For example, primes play a critical role in modern encryption and e-commerce. While it's trivial to determine if $n$ is prime (simply check to see if any number from 2 to $\sqrt{n}$ divides it), for industrial applications we often deal with numbers as large as or larger than $10^{300}$, making such an approach unusable. If we're slightly more clever, we can significantly reduce the run-time from $n$ to on the order of $\sqrt{n}$, since if $n$ factors as $xy$, then at least one of the two factors is at most $\sqrt{n}$. Sadly, this is still far too slow to be of use, as it requires checking at least $10^{150}$ numbers. To put that in perspective, the universe is a little less than $5 \cdot 10^{17}$ seconds old, and there are less than $10^{100}$ particles and photons in it. Thus, if every such object were a supercomputer capable of checking $10^{20}$ numbers per second (far better than our current computers can do!) and if they had been running since the dawn of time, we would have examined fewer than $10^{140}$ numbers!

The simple example above highlights the predicament we face. The approach we take is to discuss these issues and develop a good amount of the theory to resolve them. As this is a first course, we are not able to fully prove the efficiency of many modern algorithms; however, we are frequently able to motivate and justify the material by looking at related problems which are easier to analyze and which showcase many of the techniques. We refer the reader to the literature for fuller explanations.

While much of this book concerns linear programming, a powerful method to solve or approximately solve numerous optimization problems, our purpose is *not* to write a linear programming textbook. Those exist and do the job better than we can as they are devoted primarily to that topic. See for example [**Ar, BC, BP, Fr, GKT, Me, Ra, Va**] for a subset of excellent books covering linear programming and its generalizations (among other applied topics in operations research and optimization theory). Our goal is to showcase connections between lots of mathematical areas, while at the same time introducing the issues and difficulties of applying math to problems in the real world. Thus this book is a mix of theory and applications, with numerous options for the instructor to design the course appropriate for their class. In order not to get carried away in digressions, a lot of good, related subjects are relegated to exercises or supplemental material, though many topics that would not typically be mentioned in an operations course will be discussed, as they serve as excellent springboards.

As mentioned above, this book has been used to teach both senior capstone classes as well as introductory and advanced applied mathematics courses. To assist the instructor and students, all the lectures, homeworks and additional comments from the different versions of this course taught at Williams are available online at

http://web.williams.edu/Mathematics/sjmiller/public_html/377Fa16/
https://web.williams.edu/Mathematics/sjmiller/public_html/416/index.htm
https://web.williams.edu/Mathematics/sjmiller/public_html/317/index.htm

(exam questions are also available from the author). Our purposes are to introduce the reader to a variety of important problems and techniques, while at the same time showcasing how material from earlier courses is in fact important! Thus our approach is a bit non-standard, as normally one would not find number theory mixed with linear programming. The advantage of such an approach is that instead of exploring in great detail many algorithms whose analysis requires significant technical machinery, we can gain insights from subjects that are more readily accessible to an undergraduate; for instructors or students with less familiarity in these areas, the online lectures should provide a detailed supplement.

We hope that after reading this book you'll have a new appreciation for the challenges in applying mathematics, as well as the power of theory and computers. The following quote, from Jamie Lesser (one of my shared thesis students at Williams, where she was a Math/CS double major), illustrates this nicely. A key part of her thesis (Optimal Precision Monte Carlo Integration and Application in Computer Graphics) was figuring out how to precisely formulate the problem, which we emphasize consistently in the pages below.

> One of the most important parts of research is learning how to frame a question in a good way, a way that highlights the key features of the problem and allows us to then apply our tools to solve it. This is not easy, and this is often the step that requires the most creativity.

Opportunities for efficiency surround us, and frequently there are easy approaches. In the very first chapter we'll see faster ways to multiply numbers and matrices then you've probably seen before, and their implementation is not significantly harder than what you're used to. Figure 1 gives a wonderful example of how frequently a very simple idea can solve a complicated problem well. The picture is of my two kids, right before we went on the Peter Pan ride at Disney World. People visit to go on rides, not to wait in line. Guests want to plan accordingly but need the right data; in particular, they want to know how long a wait will be so they can determine if it's worth staying in line for a ride or if it's better to move on to something else. Disney has a terrific, low-budget, low-tech solution to constantly check and see if their wait time estimates are accurate. Visitors are periodically given a red card as they enter a line, which is scanned as it's distributed. When they get to the end of the line, the red card is collected and scanned again. This allows the company to collect and compare waiting times, providing simple, almost real-time updates to themselves and visitors.

We end with one final warning, returning to a point we mentioned above. There are two aspects to efficiency: the first is how fast the algorithm is to run or implement, while the second is how close it is to the solution. Sometimes it's better to have a quick algorithm that gets us closer to the solution instead of a slow algorithm that is more accurate but takes too much time to run. Linear programming has great examples of this – frequently the Simplex Method runs fast and we can bound how close we are to the optimal solution. Since the parameters of the problem are often approximations or guesses, the danger in only reaching an approximate optimal solution might be minimal. Thus, one of the goals of this book is to change your perspective; it's usually enough to get close to the answer quickly. I'll leave you with one of my favorite television quotes from my childhood. It's from Disney's *Zorro and Son* television series from 1983. Zorro is old and not

**Figure 1.** Cameron and Kayla helping management assess wait times at Disney World.

able to defend the people as he did in his youth, and sends for his son to help. He arrives, and the eager father wants to see what the son has learned. The son is happy to oblige, and takes out his sword. His father always drew a $\mathcal{Z}$ in his battles (on enemy soldiers or items on the scene). Holding the blade carefully, and steadying it by using his other hand for support, he slowly and beautifully carves a $\mathcal{Z}$, which he proudly shows to his father. Zorro shakes his head in disgust:

> `In this business, you usually don't have time for anything fancy. You've got to get in, make your` $\mathcal{Z}$`, and get out!`