**For my students**

# Contents

## Chapter 6. Transportation Problems



## Chapter 7. Matrix Games



## Chapter 8. Linear Approximation



## Appendix. Guide to Mathematical Programming

# Preface

**Why This Book Was Written**

This text has evolved from an advanced undergraduate math course serving students with different mathematical backgrounds and various majors, including mathematics, computer science, statistics, engineering, secondary education, actuarial science, computer engineering, science, and business. Some students are in the integrated five-year science/business program leading to a Master's in Business Administration, and some are Ph.D. students.

Since this course does not require deep mathematical theories such as calculus, differential equations, abstract algebra, topology or number theory, it offers an opportunity for students with modest mathematical backgrounds to learn some useful and important mathematics. With this in mind, I tried to avoid, whenever possible, deep or complicated mathematical concepts such as vector spaces, determinants, echelon forms, limits, and derivatives.

Many students enroll because linear programming is widely used in business and other areas. They need to learn how to formulate real-life problems, how to adapt the formulation for specific computer software, and how to interpret and apply the results of computations back to real-life problems. In the case when a computer does not produce any result or produces nonsense, they should be able to adjust the problem or choose an appropriate software.

There are many excellent textbooks on linear programming, but most of them require a strong mathematical background and are accessible only to mathematics majors or written for advanced students and contain much more than can be covered in a one-semester course.

It is a real challenge to keep both advanced students and beginners in one class! Although linear algebra is a prerequisite for linear programming at Penn State, some students in class have difficulty solving systems of linear equations. On the other hand, some students in class are strong in mathematics or computer science.

So I tried to avoid texts that are a bit like the porridge that Goldilocks rejected—they either presented material that was too "cold" in that it was trivialized or they presented material that was too "hot" in that it was predicated on a rigorous mathematical background. In the former case, many students were bored; in the latter case, many students found the material to be inaccessible.

The text starts from the beginning, assuming very little mathematical background. Thus I offer the reader the opportunity to learn

linear programming integrally by first considering relevant tools from linear algebra and logic before seeing the simplex method. The section on logic is an important, albeit often ignored, component of linear programming. Throughout the book, I present a rich palette of examples and applications and I ask students to attempt exercises of differing degrees of difficulty. Students like this approach to linear programming, as evidenced by the enrollment and remarks they have made on their end-of-semester evaluation forms.

The widespread availability of computers has not eliminated the need for computational skills, but it has increased the relative importance of logical skills. It is a curiosity rather than an important asset, nowadays, if you can compute 100 digits of $\pi$ by hand. Today computers can compute the first $10^{10}$ digits. But is it logically possible to compute the $10^{100}$-th digit of $\pi$?

**How to Use This Book**

The text is written at three levels. Most of it is accessible to students who even do not know linear algebra or calculus. Remarks and some exercises are addressed to more sophisticated students. The appendix at the end of the book gives a general idea of further developments in linear and mathematical programming. It is intended as a guide for further studies. Also it gives details on topics mentioned in Chapters 1 through 8 requiring a stronger mathematical background and a level of experience and sophistication on the part of students that is beyond what is typically achieved by undergraduate students in the United States.

Many examples with solutions are given in the text, so I feel that it is not necessary to give students solutions for numerous exercises. However, at the end of book, answers are given to some exercises including those that could be tricky. Exercises are of different difficulties, but all of them can be solved by hand. Computers are allowed but not required in class. I do not give problems where computer skills give a big advantage. The exercises in the first section of Chapter 1, besides checking understanding of definitions, also test the mathematical background of students.

**Acknowledgments and References**

My class notes evolved over several years, and many students and graders contributed to their improvement by pointing out misprints and errors and asking questions. Reviewers and editors of Prentice Hall made numerous corrections and improvements.

I deliberately chose not to tie this book to a particular software package, because I believe that if students learn the material in this text, they can intelligently apply this knowledge when using any one of the excellent software packages now available. Another reason is that any particular package will be soon outdated with the appearance of new packages and advances in computers and operating systems.

However, the students in class are allowed to use any hardware and software they like, even during tests. Software that can do linear programming includes *Mathematica*, *Maple*, and *Excel*.

More software for linear programming is available via the Internet either to be downloaded for free or to be used online. There is a lot of useful information about linear programming on the Internet. I list some URLs, but keep in mind that the Web changes rapidly:

- http://carbon.cudenver.edu/ hgreenbe/glossary/
  (Mathematical Programming Glossary)
- http://www.mathprog.org/
  (Mathematical Programming Society)
- http://iris.gmu.edu/ asofer/siagopt.html
  (SIAM Activity Group on Optimization)
- http://solon.cma.univie.ac.at/ neum/glopt.html
  (Global Optimization, Wien)
- http://www.informs.org/Resources/
  (INFORMS)

Searching the Web with the keyword "linear programming" will yield many other Web sites. There are many books on linear programming. The Web site

http://www.amazon.com

listed 771 matches for "linear programming" on August 16, 2002.

There are also many journals that publish papers on linear and nonlinear programming. The Web site

http://www.informs.org/Resources/

lists 36 hardcopy journals and 14 online journals on operations research (as of August 16, 2002). It lists also 35 societies related to operations research.

Leon Vaserstein
vstein@math.psu.edu