Department of Industrial and Systems Engineering
Spring 2003

Stochastic Programming

(IE 495)

Meeting: Monday and Wednesday 4:10–5:30PM 453 Mohler Lab

Jeff Linderoth
Office: 325 Mohler
Office hours: Monday and Wednesday 5:30–6:30PM, Thursday 2–4PM(also by appt.)
Phone: 610-758-4879
E-mail: `jtl3@lehigh.edu`
Web: `http://www.lehigh.edu/~jtl3/teaching/slp`

Many optimization problems involve uncertainty, where the eventual outcome depends on a future random event. Stochastic programming is concerned with using mathematical optimization to help make decisions in the presence of uncertainty.

Required Text

John R. Birge and Fran cois Louveaux. *Introduction to Stochastic Programming* (Springer-Verlag, 1997).

Overview

The aim of stochastic programming is to find optimal decisions in problems which involve uncertain data. This field is currently developing rapidly with contributions from many disciplines including operations research, mathematics, and probability. Conversely, it is being applied in a wide variety of subjects ranging from agriculture to financial planning and from industrial engineering to computer networks. This is a first course in stochastic programming suitable for students with a basic knowledge of linear programming, elementary analysis, and probability. We will make a broad overview of the main themes and methods of the subject. Since stochastic programs are computationally very challenging, there will be a particular emphasis in this course on implementation and tools for solving difficult stochastic programming instances.

Requirements and Grading

This course is (hopefully) *not* about getting a good grade. Instead, it should be about challenging yourself and learning about stochastic programming—one of the "hottest" topics in mathematical optimization. It won't be easy!

Problem Sets

There will be weekly or bi-weekly problem sets. I am planning for seven problem sets in all. You may work in groups of at most two students. On each assignment, there will a benefit to working alone. (Either there will be less problems to do or there will be a a bonus for doing all of the problems). There is to be *no* collaboration on the problem sets with students other than your teammate.

For the problem sets, I encourage the use of reference materials, but you *must* site references.

Exam

There will be one take-home final exam. No references (other than the textbook and class notes) are allowed, and there is to be no collaboration.

Projects

The project portion will be worth 25% of the grade. It is designed to be a semester-long, ongoing project. I am encouraging students to work on an implementation-based project for the course. This will require some programming skill (probably in C++). For interested students, I am willing to work with you, and some of the projects have the possibility of leading to funded research or archival journal publications. Besides the actual implementation, such projects will consist of a short summary of the technique and computational results. Here is a smattering of ideas for implementation-based projects.

- Work on (state-of-the-art) code for two-stage stochastic linear programming. Items include
  - Implementing and testing trust-region variants:
    - Scaling and parameter tuning issues,
    - 2-norm trust region.
  - Implementation and testing of multi-cut methods
    - Static
    - Dynamic
  - Implementation and testing of "bunching" concept
- Sampling Techniques
  - Implementation and testing of variance reduction techniques
- Stochastic integer programming
  - Investigate and implement a column-generation approach
  - Collect and solve test instances
  - Investigate and implement a IP-value function approximation solution method.
- Multi-stage stochastic programming
  - Sampling methods
  - Work on (parallel) nested decomposition scheme
- Create and collect test problems in SMPS format
- Many of the above methods can be combined with high-performance (or commodity) computing to solve large scale instances!
- **Define your own!**
  - If you have a particular approach or application area with which you are interested, *please* discuss it with me, and I am sure we can develop a suitable project.

As an alternative to doing an implementation-based project[1], the student can read and report on three research papers in stochastic programming. The report will consist of a short written summary of each paper and a 20–30 minute presentation of the results of the paper to the class. I will develop a bibliography of acceptable papers as the class progresses. You will be asked to select your research project early in the semester (probably before the end of January).

Grading Scheme

The course grade will be based on a weighted average of three components:

- 25% Project
- 50% Problem Sets
- 25% Final Exam

For the problem sets, in general, I will grade some of the problems in detail, and I will scan the solutions to other problems. The problems I grade in detail will be worth six points, and the other problems will be worth three points. I will provide detailed solutions to all of the problems.

There is a penalty of 10% for each day that an assignment is late, without exception. Once I have graded and returned an assignment, that homework will no longer be accepted. Each student's lowest score on a problem set will be dropped before computing the average at the end of the semester.

(Tentative) Chronological Syllabus

Part I: **Modeling Background** — *Jan. 13–Jan. 22*

Basic stochastic programming modeling concepts. Formulating and solving the deterministic equivalent of stochastic programming problems using algebraic modeling languages.

Readings: Chapters 1 and 2

Problem Set #1: Due: Jan. 27

Part II: **Two-Stage SLP – Theory** — *Jan. 27–Feb 12.*

Basic theory and properties of two-stage stochastic linear programming with fixed recourse. Extensions of this theory to problems with probabilistic constraints, stochastic integer programs, and multi-stage stochastic programs.

Readings: Chap 3.1, 3.2, 3.3, 3.5

[1] Again, I am trying to put an emphasis on implementation, so please think about doing an implementation-based project

Problem Set #2: Due: Feb. 10

Problem Set #3: Due: Feb. 17

Part III: **Bounds in Stochastic Programming** *Feb 17–Feb. 19*

Expected Value of Perfect Information, Value of Stochastic Solution, Wait-and-See bounds, Jensen's Inequality, Edmunson-Madansky Inequality. Brief discussion of solution approaches based on bounds.

Readings: Chapter 4, Chapter 9.1, 9.2

Problem Set #4: Due Feb. 24

Part IV: **Computational Methods** *Feb. 24 – March 19*

The L-Shaped method. Multicut methods. Bunching and other algorithmic techniques. Stabilizing the L-Shaped method. Regularized Decomposition, Progressive Hedging. Student paper presentations.

Readings: Chapters 5 and 6

Problem Set #5: Due: Mar. 17

Problem Set #6: Due: Mar. 24

Part V: **Sampling Methods** *Mar.24 – Apr. 2*

Interior and exterior sampling methods for large scale problems. Stochastic Decomposition. Variance Reduction Techniques.

Readings: Chapter 10.1, 10.2 Selected Papers

Problem Set #7: Due: Apr. 9

Part VI: **Applications and Presentations** *Apr. 7 – Apr. 23*

Presentation of new research papers and applications that use the modeling and solution techniques we have learned. Student presentations.

Readings: Selected Papers

Final Exam: Due: Apr. 30