Integer Programming
IE418

Lecture 1

Dr. Ted Ralphs
Reading for This Lecture

• N&W Sections I.1.1-I.1.4
• Wolsey Chapter 1
Mathematical Programming Models

• What does *mathematical programming* mean?

• Programming here means “planning.”

• Literally, these are “mathematical models for planning.”

• Also called *optimization models*.

• The essential element is the existence of an *objective*.

• Some classifications of mathematical programs (see the NEOS Guide).
  
  – Linear/nonlinear
  – Convex/nonconvex
  – Discrete/continuous
  – Stochastic/deterministic
Forming a Mathematical Programming Model

- Decision variables
- Constraints
- Objective Function
- Parameters and Data

The general form of a *mathematical programming model* is:

$$\begin{align*}
\text{min or max } & \quad f(x) \\
\text{s.t.} & \quad g_i(x) \begin{cases} 
\leq & b_i \\
= & \\
\geq & \end{cases} \\
& \quad x \in X
\end{align*}$$

where $X$ might be a discrete set. **Question**: What is a discrete set?
Solutions

• A *solution* is an assignment of values to variables.
• A solution can hence be thought of as an $n$-dimensional vector.
• A *feasible solution* is an assignment of values to variables such that all the constraints are satisfied.
• The *objective function value* of a solution is obtained by evaluating the objective function at the given point.
• An *optimal solution* (assuming maximization) is one whose objective function value is greater than or equal to that of all other feasible solutions.
• Note that a math program may not have a feasible solution.
• **Question**: What are the different ways in which this can happen?
Possible Outcomes

• When we say we are going to “solve” mathematical program, we mean to determine
  – whether it is feasible, and
  – whether it has an optimal solution.

• We may also want to know some other things, such as the status of its “dual” or about sensitivity.
Types of Math Programs

• The type of a math program is determined primarily by
  – The form of the objective and the constraints.
  – The form of the set $X$.

• In 406, you learned about linear models.
  – The objective function is linear.
  – The constraints are linear.

• The most important determinants of whether a mathematical program is “tractable” are the convexity of
  – The objective function.
  – The feasible region.
The General Setting for This Course

• In this class, we will consider linear models in which we additionally impose that $X = \mathbb{Z}_+^p \times \mathbb{R}_+^{n-p}$.

• The general form of a MIP we will consider is

$$\max\{c^\top x \mid Ax \leq b, x \in \mathbb{Z}_+^p \times \mathbb{R}_+^{n-p}\},$$

where $A \in \mathbb{Q}^{m \times n}, b \in \mathbb{R}^m, c \in \mathbb{R}^n$.

• This type of model is called a mixed integer linear programming model, or simply a mixed integer program (MIP).

• If $p = n$, then we have a pure integer linear programming model, or integer program (IP).

• The first $p$ components of $x$ are the discrete or integer variables and the remaining components consist of the continuous variables.
The Bigger Picture: Modeling and Formulation

- Why are we interested in solving mathematical programs in the first place?
- Mathematical programs are used to *model* real-world systems.
- The system we are modeling is typically (but not always) one we are seeking to control by determining its “operating state.”
- The (independent) variables in our model represent aspects of the system we have control over.
- The values that these variables take in the model tell us how to set the operating state of the system in the real world.
- Modeling and formulation is the process of constructing a mathematical program whose solution tell the optimal state for the system.
- This is far from an exact science.
The Modeling Process

• The modeling process consists generally of the following steps.
  – Determine the “real-world” state variables, system constraints, and goal(s) or objective(s) for operating the system.
  – Translate these variables and constraints into the form of a mathematical program (the “formulation”).
  – Solve the mathematical program.
  – Interpret the solution in terms of the real-world system.

• This process presents many challenges.
  – Simplifications may be required in order to ensure the eventual mathematical program is “tractable”.
  – The mappings from the real-world system to the model and back are sometimes not very obvious.
  – There may be more than one valid “formulation”.

• All in all, an intimate knowledge of mathematical programming definitely helps during the modeling process.
Some Notes

• The text book considers maximization problems by default.

• I normally consider minimization by default, so please be aware, this may cause some confusion.

• Also note that all variables are assumed to be nonnegative.

• One further assumption we will make is that the constraint matrix is rational. Why?
Some Notes

• The textbook considers maximization problems by default.

• I normally consider minimization by default, so please be aware, this may cause some confusion.

• Also note that all variables are assumed to be nonnegative.

• One further assumption we will make is that the constraint matrix is rational. Why?
  – This is an important assumption since with irrational data, certain “intuitive” results no longer hold (such as what?)
  – A computer can only understand rational data anyway, so this is not an unreasonable assumption.
Special Case: Binary Integer Programs

• In many cases, the variables of an IP represent yes/no decisions or logical relationships.
• These variables naturally take on values of 0 or 1.
• Such variables are called *binary*.
• Integer programs involving only binary variables are called *binary integer programs* (BIPs).
Special Case: Combinatorial Optimization Problems

• A combinatorial optimization problem \( CP = (N, \mathcal{F}) \) consists of
  – A finite ground set \( N \),
  – A set \( \mathcal{F} \subseteq 2^N \) of feasible solutions, and
  – A cost function \( c \in \mathbb{Z}^n \).

• The cost of \( F \in \mathcal{F} \) is \( c(F) = \sum_{j \in F} c_j \).

• The combinatorial optimization problem is then

\[
\max \{ c(F) \mid F \in \mathcal{F} \}
\]

• There is a natural association with a 0-1 math program.

• Many COPs can be written as BIPs or MIPs.
How Hard is Integer Programming?

- Solving general integer programs can be much more difficult than solving linear programs.
- There is no known \textit{polynomial-time} algorithm for solving general MIPs.
- Solving the associated \textit{linear programming relaxation} results in an upper bound on the optimal solution to the MIP.
- In general, an optimal solution to the LP relaxation does not tell us much about an optimal solution to the MIP.
  - \textit{Rounding} to a feasible integer solution may be difficult.
  - The optimal solution to the LP relaxation can be arbitrarily far away from the optimal solution to the MIP.
  - Rounding may result in a solution far from optimal.
  - We can bound the difference between the optimal solution to the LP and the optimal solution to the MIP (\textit{how}?).
**The Geometry of Integer Programming**

- Let's consider again an integer linear program
  \[
  \begin{align*}
  \text{max} & \quad c^\top x \\
  \text{s.t.} & \quad Ax \leq b \\
  & \quad x \in \mathbb{Z}_+^n
  \end{align*}
  \]

- The feasible region is the integer points inside a polyhedron.

- Why does solving the LP relaxation not necessarily yield a good solution?
Integer Programming and Convexity

- The feasible region of an integer program is nonconvex.
- The nonconvexity is of a rather special form, though all forms of nonconvexity are in some sense equivalent.
- Although the feasible set is nonconvex, there is a convex set over which we can optimize in order to get a solution (why?).
- The challenge is that we do not know how to describe that set.
- Even if we knew the description, it would in general be too large to write down explicitly.
- Integer variables can be used to replace other forms of nonconvexity, as we will see later on.
Integer Programming and Logic

• Integer programming can be studied from the point of view of a number of fundamental mathematical disciplines:
  
  – Algebra
  – Geometry
  – Topology
  – Combinatorics
    * Matroid theory
    * Graph theory
  – Logic
    * Set theory
    * Proof theory
    * Computability/complexity theory

• There are also a number of other related disciplines
  
  – Constraint programming
  – Satisfiability
  – Artificial intelligence
Basic Themes

Our goal will be to expose the geometrical structure of the feasible region (at least near the optimal solution). We can do this by

- Outer approximation
- Inner approximation
- Division

An important component of the algorithms we consider will be mechanisms for computing bounds by either

- Relaxation
- Duality

When all else fails, we will employ a basic principle: divide large, difficult problems into smaller ones.

- Logic (conjunction/disjunction)
- Implicit enumeration
- Decomposition