Math 482: Topics Covered in Exam 1

Misha Lavrov

February 19, 2020

The first exam will, broadly speaking, cover all the material covered in class through Monday, February 17^{th}. This includes general intro material to linear programs, and all topics related to the simplex method.

Below I try to summarize the important things we've covered that will be on the exam. It's possible that I've missed a few topics, and it's hard to summarize three weeks of content in two pages anyway; if you're not sure about something, ask me.

1 Things you should know

Here are the definitions you should know:

- The **objective value**, **constraints**, **feasible solutions**, **optimal solutions**, and **feasible region** of a linear program.
- **Convex sets**, **vertices**, and **extreme points**.
- The **basic variables**, **nonbasic variables**, **basic feasible solution**, **reduced costs** of a simplex method tableau.
- **Slack variables** and (for the two-phase simplex method) **artificial variables**.
- **Unbounded** and **inconsistent** linear programs.
- **Degenerate pivots**, **cycling**, and **pivot rules**.
- Specific pivot rules: the **largest/most negative reduced cost rule**, **Bland's rule**, **lexicographic pivoting**.

You should know and understand the following results about linear programs:

- If the feasible region of a linear program is nonempty and bounded, then the linear program has an optimal solution, and at least one optimal solution is a basic feasible solution.
- The vertices/extreme points of the feasible region are precisely the basic feasible solutions of the linear program.
- In non-degenerate cases, the simplex method is guaranteed to produce better and better solutions, and when it stops, it finds an optimal solution.
- Both Bland's rule and the lexicographic pivoting rule prevent cycling.

- Both Bland's rule and the lexicographic pivoting rule can be tricked into checking exponentially many corner points, in the worst case.
- Formulas for the pieces of a simplex tableau: $\mathbf{p}$, Q, $\mathbf{r}^\mathsf{T}$, and z_0 (in the notation of Lecture 9).

2 Things you should be able to do

Represent a word problem as a linear program.

Convert between different forms of a linear program:

- Turn inequalities into equations and equations into inequalities.
- Convert a program with unconstrained variables to one with nonnegative variables.
- Convert a minimization problem into a maximization problem.

Skills related to the simplex method. (You can probably summarize most of these as "be able to use the simplex method", but maybe you'd appreciate specifics.)

- Read off the basic feasible solution and objective value in a tableau.
- Move between valid tableaux by pivoting on variables.
- Choose entering variables (based on the objective function) and leaving variables (based on the nonnegativity constraints).
- Notice when a tableau represents an optimal solution.
- Notice when a linear program is unbounded, and find a ray of feasible solutions along which the objective value improves without bound.
- For linear programs with constraints $\{A\mathbf{x} \le \mathbf{b}, \mathbf{x} \ge \mathbf{0}\}$ in which $\mathbf{x} = \mathbf{0}$ is feasible, use the slack variables to write down an initial feasible solution.
- For general linear programs, use the two-phase simplex method to find an initial tableau.
- When using the two-phase simplex method, notice when the linear program is infeasible.
- Use the most negative reduced cost rule (when minimizing) to choose entering variables.
- Use Bland's rule to choose entering and leaving variables.
- Use the lexicographic pivoting rule (introducing $\epsilon_1, \dots, \epsilon_m$) to choose leaving variables.

You should be able to use the revised simplex method instead of a tableau, computing and updating the matrix $A_\mathcal{B}^{-1}$ and the vector $A_\mathcal{B}^{-1}\mathbf{b}$), and finding reduced costs on the fly.

In general, unless I specify a pivoting rule I want you to use, you can make any (valid) choices of entering and leaving variables you like when solving a linear program.