SA305 – Linear Programming Spring 2014
Asst. Prof. Nelson Uhan

Lesson 17. Improving Search: Finding Better Solutions

1 Overview

- Last time: a general optimization model with only continuous variables
  - Decision variables: $\mathbf{x} = (x_1, \dots, x_n)$
  - Multivariable functions in $\mathbf{x}$: $f(\mathbf{x})$ and $g_i(\mathbf{x})$ for $i \in \{1, \dots, m\}$
  - Constant scalars: b_i for $i \in \{1, \dots, m\}$

$$\begin{aligned} \text{minimize / maximize} \quad & f(\mathbf{x}) \\ \text{subject to} \quad & g_i(\mathbf{x}) \left\{ \begin{matrix} \leq \\ \geq \\ = \end{matrix} \right\} b_i \quad \text{for } i \in \{1, \dots, m\} \end{aligned} \tag{$*$}$$

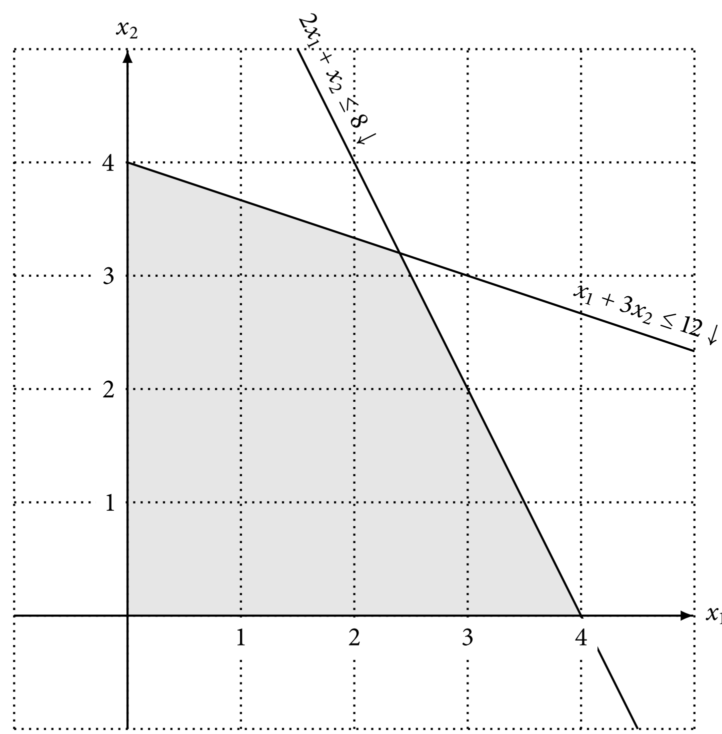
  - <u>Linear programs fit into this framework</u>

- Also last time: preview of the improving search algorithm
  - Start at a feasible solution
  - Repeatedly move to a "close" feasible solution with better objective function value

- Today: let's start formalizing these ideas behind improving search

Example 1.

$$\begin{aligned} \text{maximize} \quad & 4x_1 + 2x_2 \\ \text{subject to} \quad & x_1 + 3x_2 \leq 12 \quad && (1) \\ & 2x_1 + x_2 \leq 8 \quad && (2) \\ & x_1 \geq 0 \quad && (3) \\ & x_2 \geq 0 \quad && (4) \end{aligned}$$



2 Local and global optimal solutions

- ε-**neighborhood** $N_\varepsilon(\mathbf{x})$ of a solution $\mathbf{x} = (x_1, \ldots, x_n) \in \mathbb{R}^n$ (where $\varepsilon > 0$):

$$N_\varepsilon(\mathbf{x}) = \{\mathbf{y} \in \mathbb{R}^n : \text{distance}(\mathbf{x}, \mathbf{y}) \le \varepsilon\} \qquad \text{where} \qquad \text{distance}(\mathbf{x}, \mathbf{y}) = \sqrt{\sum_{i=1}^{n} (x_i - y_i)^2}$$

- A feasible solution $\mathbf{x}$ to optimization model $(*)$ is **locally optimal** if for some value of $\varepsilon > 0$:

$$f(\mathbf{x}) \quad \text{is better than} \quad f(\mathbf{y}) \qquad \text{for all feasible solutions } \mathbf{y} \in N_\varepsilon(\mathbf{x})$$

- A feasible solution $\mathbf{x}$ to optimization model $(*)$ is **globally optimal** if:

$$f(\mathbf{x}) \quad \text{is better than} \quad f(\mathbf{y}) \qquad \text{for all feasible solutions } \mathbf{y}$$

  - Also known simply as an **optimal solution**

- Global optimal solutions are locally optimal, but not vice versa
- In general: harder to check for global optimality, easier to check for local optimality

3 The improving search algorithm

1 Find an initial feasible solution $\mathbf{x}^0$
2 Set $k = 0$
3 **while** $\mathbf{x}^k$ is <u>not</u> locally optimal **do**
4   Determine a new <u>feasible</u> solution $\mathbf{x}^{k+1}$ that <u>improves</u> the objective value at $\mathbf{x}^k$
5   Set $k = k + 1$
6 **end while**

- Generates sequence of feasible solutions $\mathbf{x}^0, \mathbf{x}^1, \mathbf{x}^2, \ldots$
- In general, improving search converges to a local optimal solution, not a global optimal solution
- Today: concentrate on Step 4 – finding better feasible solutions

4 Moving between solutions

- How do we move from one solution to the next?

$$\mathbf{x}^{k+1} = \mathbf{x}^k + \lambda \mathbf{d}$$

- For example:

5 Improving directions

- We want to choose $\mathbf{d}$ so that $\mathbf{x}^{k+1}$ has a better value than $\mathbf{x}^k$
- $\mathbf{d}$ is an **improving direction** at solution $\mathbf{x}^k$ if

$$f(\mathbf{x}^k + \lambda\mathbf{d}) \quad \text{is better than} \quad f(\mathbf{x}^k) \qquad \text{for all positive } \lambda \text{ ``close'' to } 0$$

- How do we find an improving direction?
- The **directional derivative** of f in the direction $\mathbf{d}$ is

- Maximizing f: $\mathbf{d}$ is an improving direction at $\mathbf{x}^k$ if
- Minimizing f: $\mathbf{d}$ is an improving direction at $\mathbf{x}^k$ if
- In Example 1:

- For linear programs in general: if $\mathbf{d}$ is an improving direction at $\mathbf{x}^k$, then $f(\mathbf{x}^k + \lambda\mathbf{d})$ improves as $\lambda \to \infty$

6 Step size

- We have an improving direction $\mathbf{d}$ – now how far do we go?
- One idea: find maximum value of λ so that $\mathbf{x}^k + \lambda\mathbf{d}$ is still feasible
- Graphically, we can eyeball this
- Algebraically, we can compute this – in Example 1:

7 Feasible directions

- Some improving directions don't lead to any new feasible solutions
- **d** is a **feasible direction** at feasible solution $\mathbf{x}^k$ if $\mathbf{x}^k + \lambda \mathbf{d}$ is feasible for all positive λ "close" to 0
- Again, graphically, we can eyeball this
- For linear programs:
  - We have constraints of the form:

$$a_1 x_1 + a_2 x_2 + \cdots + a_n x_n \begin{Bmatrix} \leq \\ \geq \\ = \end{Bmatrix} b \quad \Leftrightarrow$$

  - **d** is a feasible direction at **x** if

$$\mathbf{a}^\top \mathbf{d} \begin{Bmatrix} \leq \\ \geq \\ = \end{Bmatrix} 0 \quad \text{for each active constraint of the form} \quad \mathbf{a}^\top \mathbf{x} \begin{Bmatrix} \leq \\ \geq \\ = \end{Bmatrix} b$$

    - A constraint is **active** at feasible solution **x** if it is satisfied with equality
- In Example 1:

8 Detecting unboundedness

- Suppose **d** is an improving direction at feasible solution $\mathbf{x}^k$ to a linear program
- Also, suppose $\mathbf{x}^k + \lambda \mathbf{d}$ is feasible for all $\lambda \geq 0$
- What can you conclude?

9 Summary

- Step 4 boils down to finding an improving and feasible direction **d** and an accompanying step size λ
- We have conditions on whether a direction is improving and feasible
- We don't know how to systematically find such directions... yet