Lecture 18: Linear Programming and Optimization

Sam McCauley

November 14, 2025

Williams College

Admin

- Assignment Resubmissions open! Will give more detail next slide
- Colloquium after class today: “Programming for Everyone: Developers, Scientists, and Creators of Tomorrow”
    - Faculty candidate! Come and (optionally) give us anonymous feedback
- Questions?

Assignment Resubmission

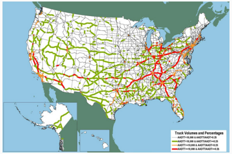
- Can bring your grade up 10 points out of 100
- Reminder: cap at 95 unless extra credit
- Submit your new version under (e.g.) "Assignment 1 Resubmission" on gradescope
- Please *say where you fixed it* by typing RESUBMIT at the beginning of any problem/code you want me to look at
- For code, just need to git push the new version (and write RESUBMIT on the relevant parts of the pdf)
- Reminder that honor code/etc. is still in effect; no LLMs allowed

Linear Programming and Optimization

What is an algorithmic problem?

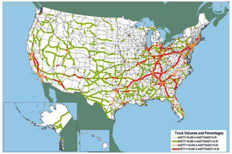
- Constraints
- Objective
- What if we had a *single tool* that could solve *any* problem with certain kinds of constraints and objectives?

Next section of the course



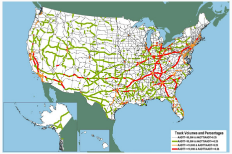
- Framework to phrase large varieties of algorithmic problems

Next section of the course



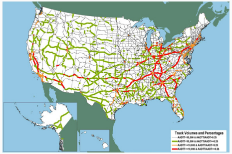
- Framework to phrase large varieties of algorithmic problems
- Allow practical solutions for a wide variety of otherwise-intractable problems

Next section of the course



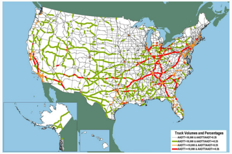
- Framework to phrase large varieties of algorithmic problems
- Allow practical solutions for a wide variety of otherwise-intractable problems
- “Optimization” problems that come up frequently in practice

Next section of the course



- Framework to phrase large varieties of algorithmic problems
- Allow practical solutions for a wide variety of otherwise-intractable problems
- “Optimization” problems that come up frequently in practice
- This topic is much older and much much broader than anything else we’ve covered

Next section of the course



- Framework to phrase large varieties of algorithmic problems
- Allow practical solutions for a wide variety of otherwise-intractable problems
- “Optimization” problems that come up frequently in practice
- This topic is much older and much much broader than anything else we’ve covered
- Focus for this class: using linear programming and integer linear programming (and their solvers) to obtain optimal solutions to difficult problems. (Won’t be focusing on structure, mathematical properties.)

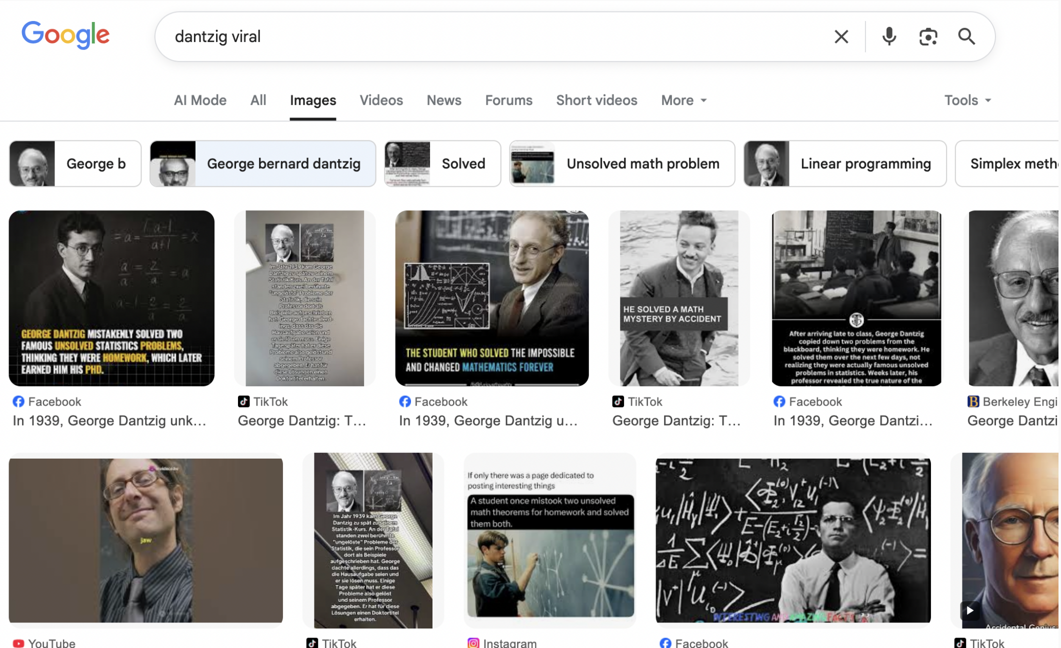
Context

> *I have a strong interest in the question of where mathematical ideas come from, and a strong conviction that they always result from a fairly systematic process—and that the opposite impression, that some ideas are incredible bolts from the blue that require "genius" or "sudden inspiration" to find, is an illusion.*
>
> *Timothy Gowers*

History

- Starts with a legend

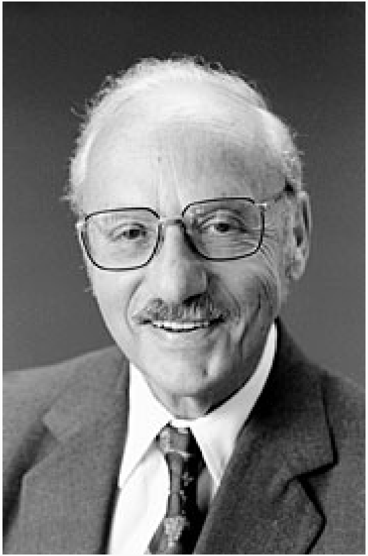
(You may have seen heard this story before)



History

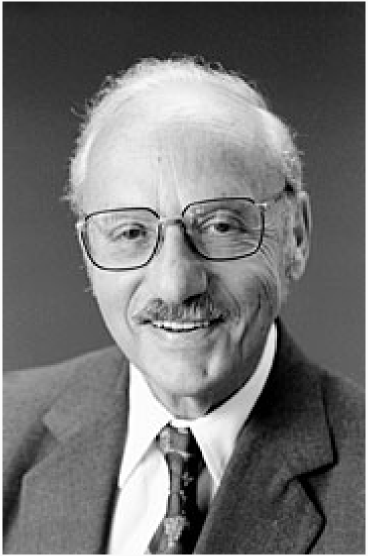
- Starts with a legend

George Dantzig



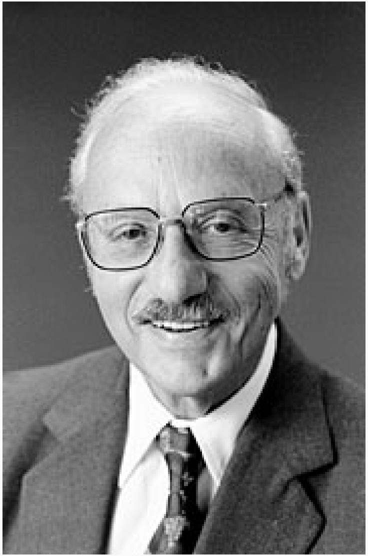
- Father of Linear Programming

George Dantzig



- Father of Linear Programming
- Worked for military during World War 2

George Dantzig

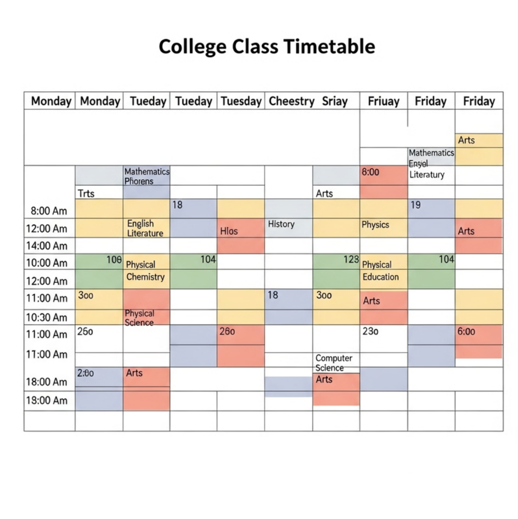


- Father of Linear Programming
- Worked for military during World War 2
- Invented the simplex algorithm

Where we’re headed

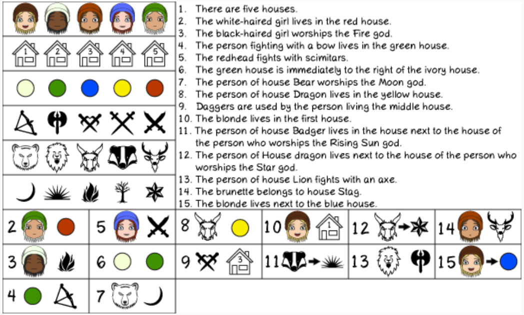
- A “big hammer” to solve an extremely wide variety of algorithmic problems

Where we’re headed



- A “big hammer” to solve an extremely wide variety of algorithmic problems

Where we're headed



- A "big hammer" to solve an extremely wide variety of algorithmic problems

Linear Programming

A *linear program* consists of:

- a linear objective function, and

Linear Programming

A *linear program* consists of:

- a linear objective function, and
- a set of linear constraints.

Linear Programming

A *linear program* consists of:

- a linear objective function, and
- a set of linear constraints.
- (We'll discuss what we mean by linear in a moment.)

Goal: achieve the best possible objective function value while satisfying the constraints

Why linear programming

- Black-box tools to solve important optimization problems that would be otherwise intractable

Why linear programming

- Black-box tools to solve important optimization problems that would be otherwise intractable
- Probably the most powerful tool you'll learn about to solve difficult algorithmic problems

Why linear programming

- Black-box tools to solve important optimization problems that would be otherwise intractable
- Probably the most powerful tool you'll learn about to solve difficult algorithmic problems
    - More powerful (in a sense) than dynamic programming

Why linear programming

- Black-box tools to solve important optimization problems that would be otherwise intractable
- Probably the most powerful tool you'll learn about to solve difficult algorithmic problems
    - More powerful (in a sense) than dynamic programming
    - Strictly *generalizes* network flows

Why linear programming

- Black-box tools to solve important optimization problems that would be otherwise intractable
- Probably the most powerful tool you'll learn about to solve difficult algorithmic problems
  - More powerful (in a sense) than dynamic programming
  - Strictly *generalizes* network flows
  - Essentially gives a free method to solve continuous optimization problems—as well as some others

Why linear programming

- Black-box tools to solve important optimization problems that would be otherwise intractable
- Probably the most powerful tool you'll learn about to solve difficult algorithmic problems
  - More powerful (in a sense) than dynamic programming
  - Strictly *generalizes* network flows
  - Essentially gives a free method to solve continuous optimization problems—as well as some others
- 2004 survey: 85% of fortune 500 companies report using linear programming

What do I mean by “linear”?

- Let’s say our variables are $x_1, \ldots x_n$.

What do I mean by "linear"?

- Let's say our variables are $x_1, \ldots x_n$.
- A linear function is the sum of a subset of these variables, each (possibly) multiplied by a constant.

What do I mean by "linear"?

- Let's say our variables are $x_1, \ldots x_n$.
- A linear function is the sum of a subset of these variables, each (possibly) multiplied by a constant.
- Linear inequality: this can be set $\geq, \leq$, or $=$ a final constant.

What do I mean by “linear”?

- Let’s say our variables are $x_1, \ldots x_n$.
- A linear function is the sum of a subset of these variables, each (possibly) multiplied by a constant.
- Linear inequality: this can be set $\geq, \leq$, or $=$ a final constant.
- Example: $4x_1 - 3x_2 \leq 7$ is linear

What do I mean by “linear”?

- Let’s say our variables are $x_1, \ldots x_n$.
- A linear function is the sum of a subset of these variables, each (possibly) multiplied by a constant.
- Linear inequality: this can be set $\geq, \leq$, or $=$ a final constant.
- Example: $4x_1 - 3x_2 \leq 7$ is linear
- Example 2: $4x_1x_2 + x_1 = 3$ is not

What do I mean by “linear”?

- Let’s say our variables are $x_1, \ldots x_n$.
- A linear function is the sum of a subset of these variables, each (possibly) multiplied by a constant.
- Linear inequality: this can be set $\geq$, $\leq$, or $=$ a final constant.
- Example: $4x_1 - 3x_2 \leq 7$ is linear
- Example 2: $4x_1x_2 + x_1 = 3$ is not
- Example 3: $|\sqrt{x_3} - x_7| \geq 5$ is not

Linear Programming

A linear program consists of:

- a linear objective function, (min or max) and
- a set of constraints, which are linear inequalities.

Linear Programming

A linear program consists of:

- a linear objective function, (min or max) and
- a set of constraints, which are linear inequalities.
- Goal: achieve the best possible objective function value while satisfying *all of* the constraints

Linear Programming

A linear program consists of:

- a linear objective function, (min or max) and
- a set of constraints, which are linear inequalities.
- Goal: achieve the best possible objective function value while satisfying *all of* the constraints
- Note that variables need not be integer or positive

Example of a Linear Program

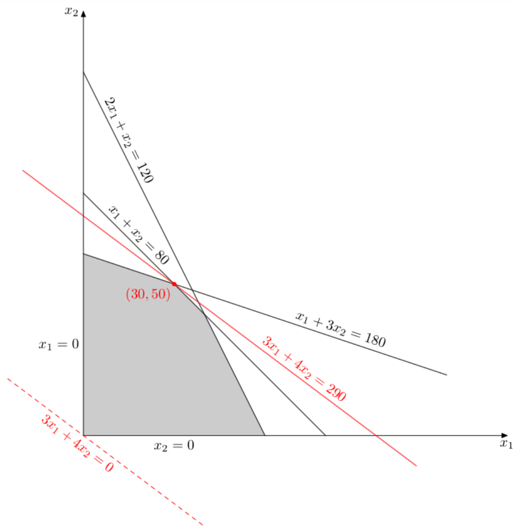
Objective:

$$\max \quad 3x_1 + 4x_2$$

Subject to:

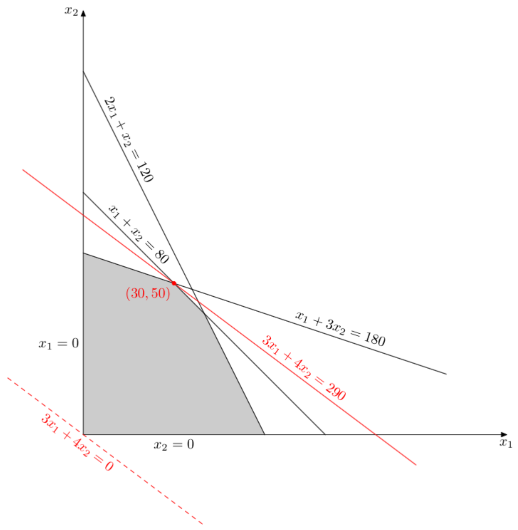
$$\begin{aligned}
2x_1 + x_2 &\leq 120 \\
x_1 + 3x_2 &\leq 180 \\
x_1 + x_2 &\leq 80 \\
x_1 &\geq 0 \\
x_2 &\geq 0
\end{aligned}$$

Feasibility



- An LP is *feasible* if there exists an assignment of variables that satisfies the constraints

Feasibility



- An LP is *feasible* if there exists an assignment of variables that satisfies the constraints
- Nontrivial result: feasibility is not trivial to determine. In the worst case, it is as difficult as solving the entire LP.

Matrix Representation

Objective:

$$\begin{bmatrix} 3 & 4 \end{bmatrix}$$

Subject to:

$$\begin{bmatrix} 2 & 1 \\ 0 & 3 \\ 0 & 1 \\ -1 & 0 \\ 0 & -1 \end{bmatrix} \begin{bmatrix} x_1 \\ x_2 \end{bmatrix} \leq \begin{bmatrix} 120 \\ 180 \\ 80 \\ 0 \\ 0 \end{bmatrix}$$

- Can represent with a matrix and vector
- Useful! Can use linear algebra tehniques and optimizations for these problems.
- I don't plan to use this representation again in this class

Visual representation

- We can plot these inequalities
- Works best for instances with 2 or 3 variables
- We'll use extensively as it gives good intuition

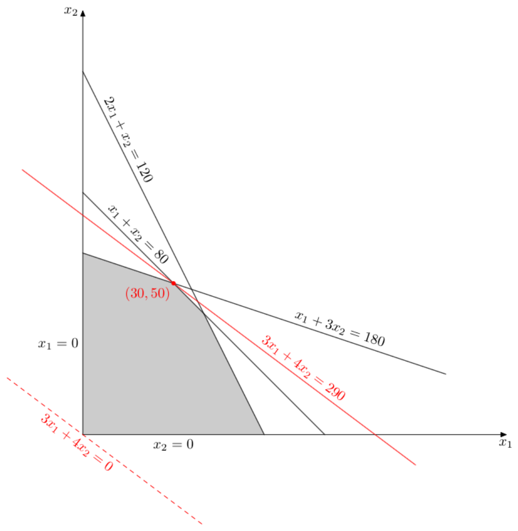
Plotting an LP

Objective:

$$\max \quad 3x_1 + 4x_2$$

Subject to:

$$\begin{aligned} 2x_1 &+ x_2 &\leq 120 \\ x_1 &+ 3x_2 &\leq 180 \\ x_1 &+ x_2 &\leq 80 \\ x_1 & &\geq 0 \\ & x_2 &\geq 0 \end{aligned}$$



Why are we looking at this?

- Let's say I gave you a tool that could solve *any* linear program

Why are we looking at this?

- Let's say I gave you a tool that could solve *any* linear program
    - I will! It's called CPLEX. (It's free and open source so I'm not really "giving" it.)

Why are we looking at this?

- Let's say I gave you a tool that could solve *any* linear program
  - I will! It's called CPLEX. (It's free and open source so I'm not really "giving" it.)
- Guarantees correct, optimal solutions!

Why are we looking at this?

- Let’s say I gave you a tool that could solve *any* linear program
    - I will! It’s called CPLEX. (It’s free and open source so I’m not really “giving” it.)
- Guarantees correct, optimal solutions!
- Frequently very fast in practice (though these tools are often exponential in the worst case)

Why are we looking at this?

- Let's say I gave you a tool that could solve *any* linear program
    - I will! It's called CPLEX. (It's free and open source so I'm not really "giving" it.)
- Guarantees correct, optimal solutions!
- Frequently very fast in practice (though these tools are often exponential in the worst case)
- Our goal: use this tool to solve computational problems

Why are we looking at this?

- Many problems can be phrased as a linear program
  - We'll start with some slightly contrived problems to build intuition, and eventually get to problems with more real-world importance.

Why are we looking at this?

- Many problems can be phrased as a linear program
    - We'll start with some slightly contrived problems to build intuition, and eventually get to problems with more real-world importance.

- Linear programs can be solved efficiently

Why are we looking at this?

- Many problems can be phrased as a linear program
  - We'll start with some slightly contrived problems to build intuition, and eventually get to problems with more real-world importance.
- Linear programs can be solved efficiently
- For today: take as a given that efficient solving is possible. How can we use linear programming to solve these problems?

Why are we looking at this?

- Many problems can be phrased as a linear program
    - We'll start with some slightly contrived problems to build intuition, and eventually get to problems with more real-world importance.

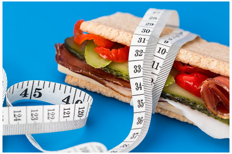
- Linear programs can be solved efficiently

- For today: take as a given that efficient solving is possible. How can we use linear programming to solve these problems?

- Essentially a reduction: similar to using Network Flow to solve problems

Solving Problems with Linear Programming

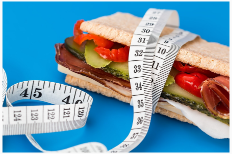
Optimization Problems



Example 1: Diet

- You need to eat 46 grams of protein and 130 grams of carbs every day

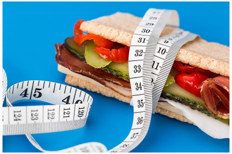
Optimization Problems



Example 1: Diet

- You need to eat 46 grams of protein and 130 grams of carbs every day
- Available foods:

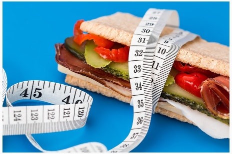
Optimization Problems



Example 1: Diet

- You need to eat 46 grams of protein and 130 grams of carbs every day
- Available foods:
  - 100g Peanuts: 25.8g of protein, 16.1g carbs, $1.61

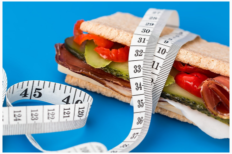
Optimization Problems



Example 1: Diet

- You need to eat 46 grams of protein and 130 grams of carbs every day
- Available foods:
  - 100g Peanuts: 25.8g of protein, 16.1g carbs, $1.61
  - 100g Rice: 2.5g protein, 28.7g carbs, $.79

Optimization Problems

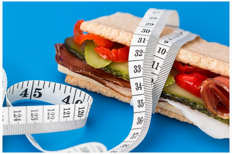


Example 1: Diet

- You need to eat 46 grams of protein and 130 grams of carbs every day
- Available foods:
    - 100g Peanuts: 25.8g of protein, 16.1g carbs, $1.61
    - 100g Rice: 2.5g protein, 28.7g carbs, $.79
    - 100g Chicken: 13.5g protein, 0g carbs, $.70

What is the cheapest way you can hit your diet goals?

Optimization Problems



Example 1: Diet

- You need to eat 46 grams of protein and 130 grams of carbs every day
- Available foods:
  - 100g Peanuts: 25.8g of protein, 16.1g carbs, $1.61
  - 100g Rice: 2.5g protein, 28.7g carbs, $.79
  - 100g Chicken: 13.5g protein, 0g carbs, $.70

What is the cheapest way you can hit your diet goals?

In Pairs: try to phrase this problem as a linear program. What variables do we want? What is the objective function? What are the constraints?

Diet Problem

How can we phrase this as a linear program? (I'll write the problem on the board; you should think about how you would do it.)

- Let p be the amount of peanuts, r be the amount of rice, and c be the amount of chicken you buy.

Diet Problem

How can we phrase this as a linear program? (I'll write the problem on the board; you should think about how you would do it.)

- Let p be the amount of peanuts, r be the amount of rice, and c be the amount of chicken you buy.
- Then what is our *objective function*?

Diet Problem

How can we phrase this as a linear program? (I'll write the problem on the board; you should think about how you would do it.)

- Let p be the amount of peanuts, r be the amount of rice, and c be the amount of chicken you buy.
- Then what is our *objective function*?
- Answer: The price is $1.61p + .79r + .7c$

Diet Problem

How can we phrase this as a linear program? (I'll write the problem on the board; you should think about how you would do it.)

- Let p be the amount of peanuts, r be the amount of rice, and c be the amount of chicken you buy.
- Then what is our *objective function*?
- Answer: The price is $1.61p + .79r + .7c$
- Do we want to maximize or minimize this?

Diet Problem

How can we phrase this as a linear program? (I'll write the problem on the board; you should think about how you would do it.)

- Let p be the amount of peanuts, r be the amount of rice, and c be the amount of chicken you buy.
- Then what is our *objective function*?
- Answer: The price is $1.61p + .79r + .7c$
- Do we want to maximize or minimize this?
- $\min 1.61p + .79r + .7c$

Diet Problem Constraints

$\min 1.61p + .79r + .7c$

Reminder:

- You need to eat 46 grams of protein and 130 grams of carbs every day
- 100g Peanuts: 25.8g of protein, 16.1g carbs, $1.61
- 100g Rice: 2.5g protein, 28.7g carbs, $.79
- 100g Chicken: 13.5g protein, 0g carbs, $.70

Diet Problem Constraints

$\min 1.61p + .79r + .7c$

- Protein: $25.8p + 2.5r + 13.5c \geq 46$

Reminder:

- You need to eat 46 grams of protein and 130 grams of carbs every day
- 100g Peanuts: 25.8g of protein, 16.1g carbs, $1.61
- 100g Rice: 2.5g protein, 28.7g carbs, $.79
- 100g Chicken: 13.5g protein, 0g carbs, $.70

Diet Problem Constraints

$\min 1.61p + .79r + .7c$

- Protein: $25.8p + 2.5r + 13.5c \geq 46$
- Carbs: $16.1p + 28.7r \geq 130$
- Anything else?

Reminder:

- You need to eat 46 grams of protein and 130 grams of carbs every day
- 100g Peanuts: 25.8g of protein, 16.1g carbs, $1.61
- 100g Rice: 2.5g protein, 28.7g carbs, $.79
- 100g Chicken: 13.5g protein, 0g carbs, $.70

Diet Problem Constraints

$\min 1.61p + .79r + .7c$

- Protein: $25.8p + 2.5r + 13.5c \geq 46$
- Carbs: $16.1p + 28.7r \geq 130$
- Anything else?
- $p \geq 0, r \geq 0, c \geq 0$

Reminder:

- You need to eat 46 grams of protein and 130 grams of carbs every day
- 100g Peanuts: 25.8g of protein, 16.1g carbs, $1.61
- 100g Rice: 2.5g protein, 28.7g carbs, $.79
- 100g Chicken: 13.5g protein, 0g carbs, $.70

Diet Problem Solution

$$\min 1.61p + .79r + .7c$$

- Protein: $25.8p + 2.5r + 13.5c \geq 46$
- Carbs: $16.1p + 28.7r \geq 130$
- $p \geq 0, r \geq 0, c \geq 0$

Solution: $p = 0, r = 2.9216..., c = 2.86636...$

So we want to buy about 293g of rice, and 287g of chicken, for total cost $4.32

Diet Problem Solution

A mathematically optimal meal

$\min 1.61p + .79r + .7c$

- Protein: $25.8p + 2.5r + 13.5c \geq 46$
- Carbs: $16.1p + 28.7r \geq 130$
- $p \geq 0, r \geq 0, c \geq 0$

Solution: $p = 0$, $r = 2.9216...$, $c = 2.86636...$

So we want to buy about 293g of rice, and 287g of chicken, for total cost $4.32

Example 2: Extending the Diet

- What if I wanted to limit the amount of rice I eat to 100g?

Example 2: Extending the Diet

- What if I wanted to limit the amount of rice I eat to 100g?
  - Add a constraint: $r \leq 1$

Example 2: Extending the Diet

- What if I wanted to limit the amount of rice I eat to 100g?
  - Add a constraint: $r \leq 1$
- What if I wanted a *balanced* diet—the amount of any pair of foods is within 50 grams of each other?

Example 2: Extending the Diet

- What if I wanted to limit the amount of rice I eat to 100g?
  - Add a constraint: $r \leq 1$
- What if I wanted a *balanced* diet—the amount of any pair of foods is within 50 grams of each other?
  - First: how would we write these constraints if we don't require that they are linear?

Example 2: Extending the Diet

- What if I wanted to limit the amount of rice I eat to 100g?
  - Add a constraint: $r \leq 1$
- What if I wanted a *balanced* diet—the amount of any pair of foods is within 50 grams of each other?
  - First: how would we write these constraints if we don't require that they are linear?
  - $|r - c| \leq .5, |c - p| \leq .5, |r - p| \leq .5$

Example 2: Extending the Diet

- What if I wanted to limit the amount of rice I eat to 100g?
  - Add a constraint: $r \leq 1$
- What if I wanted a *balanced* diet—the amount of any pair of foods is within 50 grams of each other?
  - First: how would we write these constraints if we don't require that they are linear?
  - $|r - c| \leq .5$, $|c - p| \leq .5$, $|r - p| \leq .5$
  - Then: how can we use a sequence of constraints to achieve this?

Example 2: Extending the Diet

- What if I wanted to limit the amount of rice I eat to 100g?
  - Add a constraint: $r \leq 1$
- What if I wanted a *balanced* diet—the amount of any pair of foods is within 50 grams of each other?
  - First: how would we write these constraints if we don't require that they are linear?
  - $|r - c| \leq .5$, $|c - p| \leq .5$, $|r - p| \leq .5$
  - Then: how can we use a sequence of constraints to achieve this?
  - We have $|x - y| \leq c$ when both $x - y \leq c$ and $y - x \leq c$. So:

Example 2: Extending the Diet

- What if I wanted to limit the amount of rice I eat to 100g?
  - Add a constraint: $r \leq 1$
- What if I wanted a *balanced* diet—the amount of any pair of foods is within 50 grams of each other?
  - First: how would we write these constraints if we don't require that they are linear?
  - $|r - c| \leq .5$, $|c - p| \leq .5$, $|r - p| \leq .5$
  - Then: how can we use a sequence of constraints to achieve this?
  - We have $|x - y| \leq c$ when both $x - y \leq c$ and $y - x \leq c$. So:

$$\begin{array}{ccc} r - c \leq .5 & c - p \leq .5 & r - p \leq .5 \\ c - r \leq .5 & p - c \leq .5 & p - r \leq .5 \end{array}$$

Example 3: Facility Location (Harder Problem)

- Given coordinates for n roommates $(x_1, y_1), (x_2, y_2), \ldots, (x_n, y_n)$

Example 3: Facility Location (Harder Problem)

- Given coordinates for n roommates $(x_1, y_1), (x_2, y_2), \ldots, (x_n, y_n)$
- Goal: find location for a router that minimizes the average distance to each roommate

Example 3: Facility Location (Harder Problem)

- Given coordinates for n roommates $(x_1, y_1), (x_2, y_2), \ldots, (x_n, y_n)$
- Goal: find location for a router that minimizes the average distance to each roommate
- Distance from (x, y) to (x_i, y_i) is $|x - x_i| + |y - y_i|$

Example 3: Facility Location (Harder Problem)

- Given coordinates for n roommates $(x_1, y_1), (x_2, y_2), \ldots, (x_n, y_n)$
- Goal: find location for a router that minimizes the average distance to each roommate
- Distance from (x, y) to (x_i, y_i) is $|x - x_i| + |y - y_i|$
- Cannot have distance more than 10 from any roommate

Example 3: Facility Location

Objective:

Constraints:

- Given roommates at $(3, 4)$ and $(13, 5)$
- Goal: find location for a router that minimizes the average distance to each roommate
- Distance from (x, y) to (x_i, y_i) is $|x - x_i| + |y - y_i|$
- Cannot have distance > 10 from any roommate

Example 3: Facility Location

Objective:

Constraints:

$$(x-3)+(y-4) \leq 10$$
$$(-x+3)+(y-4) \leq 10$$
$$(-x+3)+(-y+4) \leq 10$$
$$(x-3)+(-y+4) \leq 10$$

- Given roommates at $(3,4)$ and $(13,5)$
- Goal: find location for a router that minimizes the average distance to each roommate
- Distance from (x,y) to (x_i,y_i) is $|x-x_i|+|y-y_i|$
- Cannot have distance > 10 from any roommate

Example 3: Facility Location

Objective:

Constraints:

$$(x-3)+(y-4)\leq 10$$
$$(-x+3)+(y-4)\leq 10$$
$$(-x+3)+(-y+4)\leq 10$$
$$(x-3)+(-y+4)\leq 10$$
$$(x-13)+(y-5)\leq 10$$
$$(-x+13)+(y-5)\leq 10$$
$$(-x+13)+(-y+5)\leq 10$$
$$(x-13)+(-y+5)\leq 10$$

- Given roommates at $(3,4)$ and $(13,5)$
- Goal: find location for a router that minimizes the average distance to each roommate
- Distance from (x,y) to (x_i,y_i) is $|x-x_i|+|y-y_i|$
- Cannot have distance > 10 from any roommate

Example 3: Facility Location

Objective:

Constraints:

$$(x-3)+(y-4)\le 10$$
$$(-x+3)+(y-4)\le 10$$
$$(-x+3)+(-y+4)\le 10$$
$$(x-3)+(-y+4)\le 10$$
$$(x-13)+(y-5)\le 10$$
$$(-x+13)+(y-5)\le 10$$
$$(-x+13)+(-y+5)\le 10$$
$$(x-13)+(-y+5)\le 10$$

Can't make objective function. Idea: add new variables!

- Given roommates at $(3,4)$ and $(13,5)$
- Goal: find location for a router that minimizes the average distance to each roommate
- Distance from (x,y) to (x_i,y_i) is $|x-x_i|+|y-y_i|$
- Cannot have distance > 10 from any roommate

Example 3: Facility Location

Objective: $\min d_1 + d_2$

Constraints:

- Given roommates at $(3, 4)$ and $(13, 5)$
- Goal: find location for a router that minimizes the average distance to each roommate
- Distance from (x, y) to (x_i, y_i) is $|x - x_i| + |y - y_i|$
- Cannot have distance > 10 from any roommate

Example 3: Facility Location

Objective: $\min d_1 + d_2$

Constraints:

$$
\begin{aligned}
(x-3)+(y-4) &\leq d_1 \\
(-x+3)+(y-4) &\leq d_1 \\
(-x+3)+(-y+4) &\leq d_1 \\
(x-3)+(-y+4) &\leq d_1 \\
(x-13)+(y-5) &\leq d_2 \\
(-x+13)+(y-5) &\leq d_2 \\
(-x+13)+(-y+5) &\leq d_2 \\
(x-13)+(-y+5) &\leq d_2 \\
d_1 &\leq 10 \\
d_2 &\leq 10
\end{aligned}
$$

- Given roommates at $(3, 4)$ and $(13, 5)$
- Goal: find location for a router that minimizes the average distance to each roommate
- Distance from (x, y) to (x_i, y_i) is $|x - x_i| + |y - y_i|$
- Cannot have distance > 10 from any roommate

Example 3: Facility Location (Equations Simplified)

Objective: $\min d_1 + d_2$

Constraints:

$$x + y - d_1 \leq 7$$
$$-x + y - d_1 \leq 1$$
$$-x - y - d_1 \leq -7$$
$$x - y - d_1 \leq -1$$
$$x + y - d_2 \leq 18$$
$$-x + y - d_2 \leq -8$$
$$-x - y - d_2 \leq -18$$
$$x - y - d_2 \leq 8$$

- Given roommates at $(3, 4)$ and $(13, 5)$
- Goal: find location for a router that minimizes the average distance to each roommate
- Distance from (x, y) to (x_i, y_i) is $|x - x_i| + |y - y_i|$
- Cannot have distance > 10 from any roommate

Proving Correctness

- How can we show that the above LP works?

Proving Correctness

- How can we show that the above LP works?
- Idea: an LP is feasible if and only if it corresponds to a correct router placement

Proving Correctness

- How can we show that the above LP works?
- Idea: an LP is feasible if and only if it corresponds to a correct router placement
- 1st: if there exists a feasible LP solution has values d_1, d_2, x, y then there exists a router placement at (x, y) with distance *at most* d_1 and d_2 from roommates 1 and 2, with $d_1 \leq 10$ and $d_2 \leq 10$

Proving Correctness

- How can we show that the above LP works?
- Idea: an LP is feasible if and only if it corresponds to a correct router placement
- 1st: if there exists a feasible LP solution has values d_1, d_2, x, y then there exists a router placement at (x, y) with distance *at most* d_1 and d_2 from roommates 1 and 2, with $d_1 \leq 10$ and $d_2 \leq 10$
- 2nd: any placement of a router at location (x, y), with distance $d_1 \leq 10$ and $d_2 \leq 10$ from the first and second roommate respectively corresponds to a feasible LP solution with variables d_1, d_2, x, y

Proving Correctness

- How can we show that the above LP works?
- Idea: an LP is feasible if and only if it corresponds to a correct router placement
- 1st: if there exists a feasible LP solution has values d_1, d_2, x, y then there exists a router placement at (x, y) with distance *at most* d_1 and d_2 from roommates 1 and 2, with $d_1 \leq 10$ and $d_2 \leq 10$
- 2nd: any placement of a router at location (x, y), with distance $d_1 \leq 10$ and $d_2 \leq 10$ from the first and second roommate respectively corresponds to a feasible LP solution with variables d_1, d_2, x, y
- If we can prove these claims then solving this LP solves the router placement problem: we get the min total distance placement

Proving Correctness

Lemma

If there exists a feasible LP solution with variables d_1, d_2, x, y then a router at (x, y) has distance at most d_1 and d_2 from roommates 1 and 2, with $d_1 \leq 10$ and $d_2 \leq 10$

Proof: Router at (x, y) has distance $\hat{d}_1 = |x - 3| + |y - 4|$ from roommate 1. Because the LP soln is feasible, we have:

$$(x - 3) + (y - 4) \leq d_1 \qquad (-x + 3) + (y - 4) \leq d_1$$

$$(-x + 3) + (-y + 4) \leq d_1 \qquad (x - 3) + (-y + 4) \leq d_1$$

Since $\hat{d}_1$ is equal to the left side of one of these equations, $\hat{d}_1 \leq d_1$. Furthermore, since the LP solution is feasible, $d_1 \leq 10$, so $\hat{d}_1 \leq 10$.

Same argument works for roommate 2

Proving Correctness

Lemma

Any placement of a router at location (x, y), with distance $d_1 \leq 10$ and $d_2 \leq 10$ from the first and second roommate respectively corresponds to a feasible LP solution with variables d_1, d_2, x, y

Proof summary: We have $d_1, d_2 \leq 10$ by definition. We need to show the roommate constraints are satisfied. Let's focus on d_1. We have $d_1 = |x - 3| + |y - 4|$.

For any x, y we have:

$$x - 3 \leq |x - 3| \qquad -x + 3 \leq |x - 3|$$

$$y - 4 \leq |y - 4| \qquad -y + 4 \leq |y - 4|$$

Substituting, all equations for d_1 are satisfied.

Proving Correctness

- Therefore, the best LP solution gives the best router placement!

Proving Correctness

- Therefore, the best LP solution gives the best router placement!
- So we can solve this problem by solving an LP

Router Example Discussion

- Can we add new roommates?

Router Example Discussion

- Can we add new roommates?
  - Yes!

Router Example Discussion

- Can we add new roommates?
    - Yes!
- New constraints? (E.g. can’t have the router in a certain portion of the house, or can’t be too close to one of the roommates)

Router Example Discussion

- Can we add new roommates?
  - Yes!
- New constraints? (E.g. can’t have the router in a certain portion of the house, or can’t be too close to one of the roommates)
  - Yes—if they’re linear