MERRIMACK COLLEGE

CSC6023 - Advanced Algorithms

Integer and Linear Programming

Patrick Neff


Insertion
Selection
Bubble
Shell
Merge
Heap
Quick
Quick3
Random
Nearly Sorted
Reversed
Few Unique

Linear Programmin g

After this sixth week topic and we really are close to the course ending. So, let's see the Integer and Linear Programming, a problem with an approximated solution.

MERRIMACK COLLEGE

Agenda Of The Presentation

Linear Programming

- Optimization problems
- Linear programming
    a. Maximization of a linear objective function
    b. Two variables example
    c. Three variables example

Integer (linear) Programming

- Linear Programming Relaxation
    a. An approximation
- Exhaustive solution and variations
- The simplex algorithm
    a. A heuristic

MERRIMACK COLLEGE

Big Idea

Maximizing the value of an objective function, subject to constraints represented by linear relationships.

Extreme values occur at vertices of a concave, bounded feasible region.

Please note: there are no economies of scale in this lecture.

Linear Programming

Optimization Problems

How to maximize or minimize a complex system

- A store has requested a manufacturer to produce pants and sports jackets
  - The manufacturer has 750 m^2 of cotton textile and 1000 m^2 of polyester
  - Every pants (1 unit) needs 1 m^2 of cotton and 2 m^2 of polyester
  - Every jacket (1 unit) needs 1.5 m^2 of cotton and 1 m^2 of polyester
  - The price of the pants is fixed at $50 and the jacket at $40
- How many pants and jackets should be produced to maximize the sale?

Source: www.superprof.co.uk

Linear Programming

Optimization Problems

How to maximize or minimize a complex system

- The manufacturer has 750 m^2 of cotton textile and 1000 m^2 of polyester
- Every pants (1 unit) needs 1 m^2 of cotton and 2 m^2 of polyester
- Every jacket (1 unit) needs 1.5 m^2 of cotton and 1 m^2 of polyester
- The price of the pants is fixed at $50 and the jacket at $40

- You can make 500 jackets and make $20,000
  - 500 m polyester left over

- You can make 500 pants and make $25,000
  - 250 m cotton left over

- There is probably a better way… can you guess?

MERRIMACK COLLEGE

What happens if you make 498 pants?

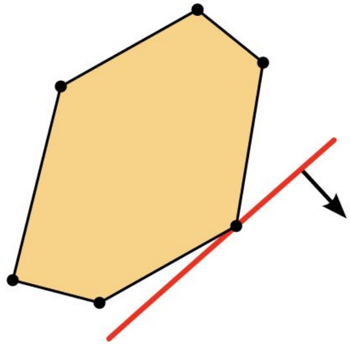
- 498*50 = $24,900
- 252 cotton left over
- 4 polyester left over
- Now we have enough to make four jackets so:
  - 498 pants -> $24,900
  - 4 jacket -> $160
  - $24,900 + $160 = $25,060!

Optimization Problems

This problem is linear because:

- Every pants (1 unit) needs 1 m^2 of cotton and 2 m^2 of polyester
  - 5 pants need 5 m^2 of cotton and 10 m^2 of polyester
- Every jacket (1 unit) needs 1.5 m^2 of cotton and 1 m^2 of polyester
  - 5 jackets need 7.5 m^2 of cotton and 5 m^2 of polyester
- The price of the pants is fixed at $50 and the jacket at $40
  - 5 pants pays $250 and 5 jackets pay $200
- Therefore, you can use linear programming

The LP solution



Linear Programming (LP), a.k.a. Linear Optimization

- Find the best outcome in a mathematical model with linear relationships
- A linear objective function subject to linear equality and linear inequality constraints
- Find a vector *x*
  - $\boldsymbol{x}$
- that maximizes the product of vector c transposed times *x*
  - $\boldsymbol{c^T x}$
- subject to constraints expressed by the equations expressed by matrix *A* times *x* smaller or equal to vector *b*
  - $\boldsymbol{Ax \leq b}$
- and all elements of *x* are greater or equal to 0
  - $\boldsymbol{x \geq 0}$

MERRIMACK COLLEGE

Linear Programming

The LP solution

	pants	jackets	available
cotton	1	1,5	750
polyester	2	1	1,000

$$\begin{bmatrix} 2 & 3 \\ 2 & 1 \end{bmatrix} \begin{bmatrix} x_1 \\ x_2 \end{bmatrix} \leq \begin{bmatrix} 1500 \\ 1000 \end{bmatrix}$$

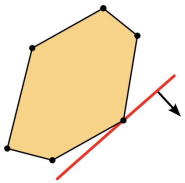
MERRIMACK COLLEGE

Linear Programming applied to pants and jackets

- x_1 = number of pants, x_2 = number of jackets
  - $\mathbf{x = [\, x_1 , x_2 \,]}$
- Objective function
  - $c = (50\ ,\ 40)$ *c is another vector.*
  - $\mathbf{f(x) = 50\, x_1 + 40\, x_2}$ $\mathbf{c^T x}$
- Constraints for cotton: 1 for pants, 1.5 for jackets, only 750 available
  - $1\, x_1 + 1.5\, x_2 \leq 750$
  - $\mathbf{2\, x_1 + 3\, x_2 \leq 1500}$
- Constraints for polyester: 2 for pants, 1 for jacket, only 1000 available
  - $\mathbf{2\, x_1 + 1\, x_2 \leq 1000}$
- $A = [\ [2,3]\ ,\ [2,1]\ ]$ $b = [1500,1000]$ $\mathbf{Ax \leq b}$
- All elements of *x* are greater or equal to 0
  - $\mathbf{x_1 \geq 0\ ,\ x_2 \geq 0}$ $\mathbf{x \geq 0}$

Linear Programming

The LP solution

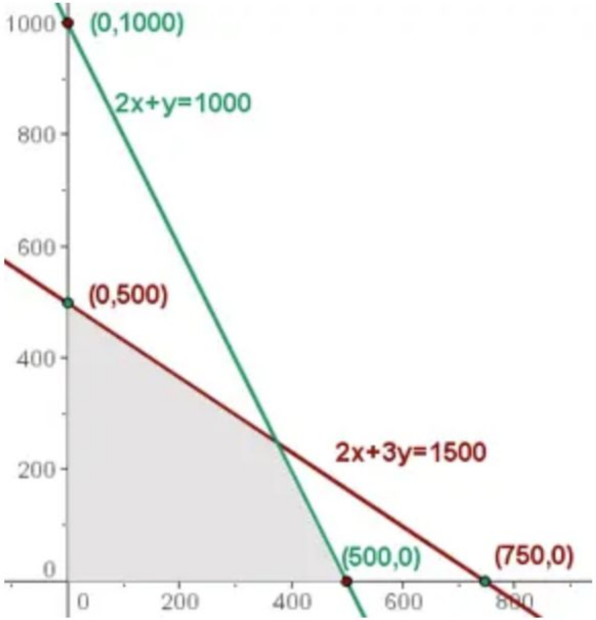


Linear Programming applied to pants and jackets

Green - polyester; Red - cotton

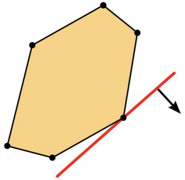
Horizontal axis - pants; Vertical axis - jackets

- $f(x) = 50\,x_1 + 40\,x_2$
- $2\,x_1 + 3\,x_2 \leq 1500$
- $2\,x_1 + 1\,x_2 \leq 1000$
- $x_1 \geq 0\,,\ x_2 \geq 0$
- Drawing the constraints
- Identifying the feasible region
  - $x_1 = 0\,,\ x_2 = 0$
    - $0 \leq 1500$
    - $0 \leq 1000$



- $x_1 = 0\,,\ x_2 = 0$ is within the feasible region

The LP solution



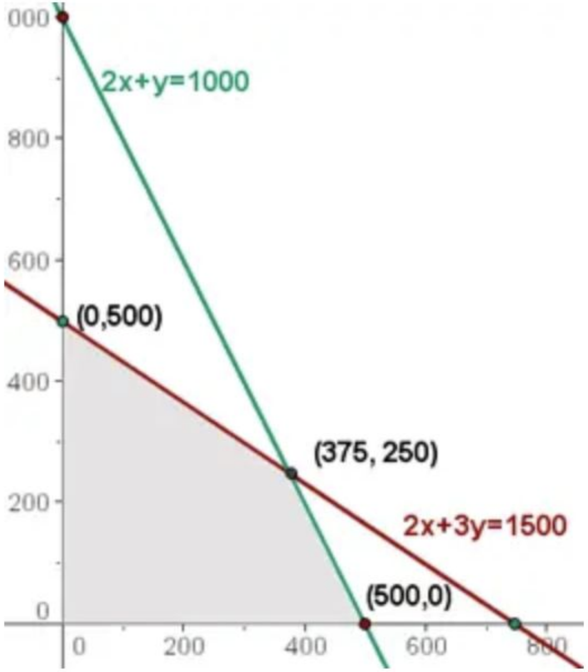
Linear Programming applied to pants and jackets

- $f(x) = 50\,x_1 + 40\,x_2$
- $2\,x_1 + 3\,x_2 \leq 1500$
- $2\,x_1 + 1\,x_2 \leq 1000$
- $x_1 \geq 0\,,\ x_2 \geq 0$

- Solving the systems
- $x_1 = 0$
- $2\,x_1 + 3\,x_2 \leq 1500$
    - $x_2 = 500$

- $x_2 = 0$
- $2\,x_1 + 1\,x_2 \leq 1000$
    - $x_1 = 500$

- $2\,x_1 + 3\,x_2 \leq 1500$
- $2\,x_1 + 1\,x_2 \leq 1000$
    - $x_1 = 375\,,\ x_2 = 250$



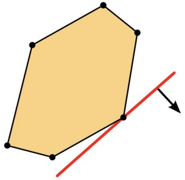
The LP solution

(just a parenthesis … solving this linear equation system)

- Initially:
  - $2\,x_1 + 3\,x_2 = 1500$
  - $2\,x_1 + 1\,x_2 = 1000$
- Isolating x_2 in the second equation:
  - $1\,x_2 = 1000 - 2\,x_1$
- Replacing x_2 in the first equation:
  - $2\,x_1 + 3\,(1000 - 2\,x_1) = 1500$
- Multiplying 3 times the x_2 replacement:
  - $2\,x_1 + 3000 - 6\,x_1 = 1500$
- Isolating x_1:
  - $-4\,x_1 = -1500$
  - $x_1 = 375$
- Replacing x_1 in the second equation:
  - $2\,(375) + 1\,x_2 = 1000$
  - $x_2 = 250$

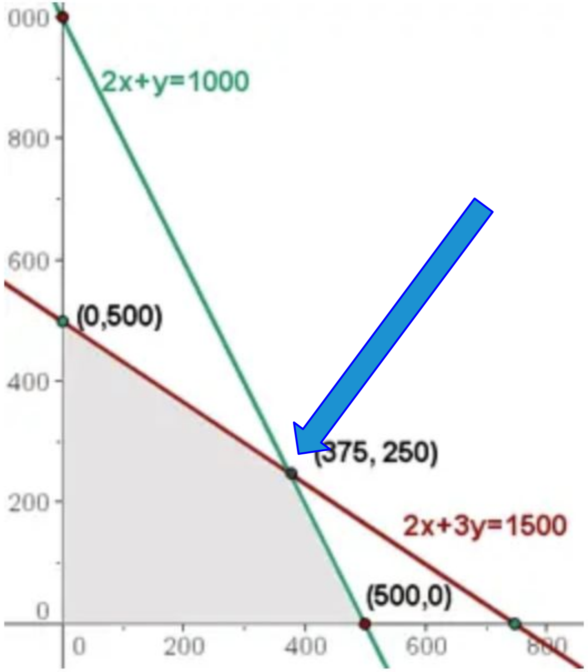
Linear Programming

The LP solution



Linear Programming applied to pants and jackets

- ***$f(x) = 50\,x_1 + 40\,x_2$***
- ***$2\,x_1 + 3\,x_2 \leq 1500$***
- ***$2\,x_1 + 1\,x_2 \leq 1000$***
- ***$x_1 \geq 0\,,\ x_2 \geq 0$***
- Verify the objective function
- ***$x_1 = 0\,,\ x_2 = 500$***
    - $20,000
- ***$x_1 = 500\,,\ x_2 = 0$***
    - $25,000
- ***$x_1 = 375\,,\ x_2 = 250$***
    - $28,750



MERRIMACK COLLEGE

Linear Programming

Find the Optimum Balance

Task #1 for this week's In-class exercises

- A fancy food truck makes pizzas and sandwiches:
  - A pizza is sold for $50 and it costs $25 to make
  - A sandwich is sold for $20 and it costs $5 to make
  - It takes 8 minutes to prepare a pizza, and it takes 3 minutes to prepare a sandwich
  - Assuming that the food truck has only one hour (sixty minutes) to prepare food and a total of 10 pizzas and sandwiches alike can be made at most
  - How many pizzas and sandwiches should be made to maximize the profit?

Write down your solution and your computations to arrive to it.
Save it in any textual file and submit it in the appropriate delivery room

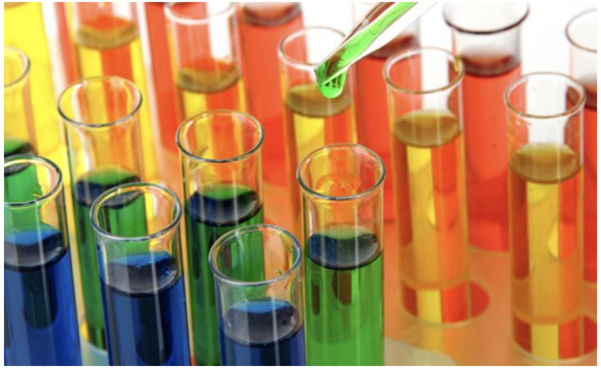
Deadline: This Friday 11:59 PM EST

Maximize the Profit

Linear Programming

Another Example

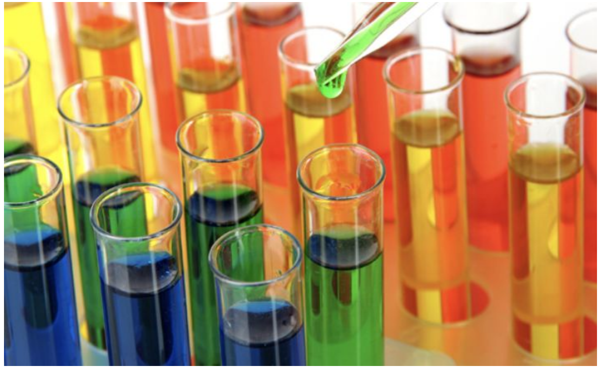
- A chemical factory produces 3 kinds of formulas (a, b, and c) using three kinds of supplies (x, y, and z):
- The need for supplies, their availability, and the profit for the formulas are:



supply	x	y	z	profit
a (1 unit)	2 gal.	1 gal.	8 gal.	$3K
b (1 unit)	4 gal.	2 gal.	0 gal.	$2K
c (1 unit)	5 gal.	4 gal.	3 gal.	$2K
availability	300 gal.	200 gal.	300 gal.	

- How much of each formula should be produced to maximize the profit?

Another Example



Maximize the Profit

- *f(a,b,c) = 3a + 2b + 2c*
  - $2a + 4b + 5c \le 300$
  - $a + 2b + 4c \le 200$
  - $8a + 3c \le 300$

Solver: WolframAlpha
Or this one, which is free:
https://matrix.reshish.com/gauss-jordanElimination.php

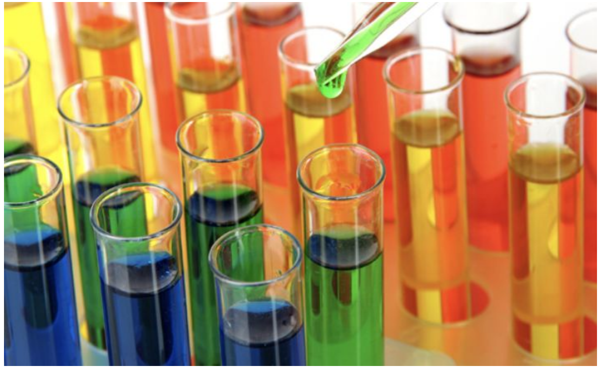
- Only a: a = 37.5 units
- Only b: b = 75.0 units
- Only c: c = 50 units
- Balanced:
  - a = 25 units
  - b = 20.8333 units
  - c = 33.3333 units

supply	x	y	z	profit
a (1 unit)	2 gal.	1 gal.	8 gal.	$3K
b (1 unit)	4 gal.	2 gal.	0 gal.	$2K
c (1 unit)	5 gal.	4 gal.	3 gal.	$2K
availability	300 gal.	200 gal.	300 gal.	

MERRIMACK COLLEGE

Linear Programming

Another Example



Maximize the Profit

- ***$f(a,b,c) = 3a + 2b + 2c$***
  - *$2a + 4b + 5c \leq 300$*
  - *$a + 2b + 4c \leq 200$*
  - *$8a + 3c \leq 300$*

- Only a: a = 37.5 units
- Only b: b = 75.0 units
- Only c: c = 50 units
- Balanced:
  - **a = 25 units**
  - **b = 20.8333 units**
  - **c = 33.3333 units**

- Only a: **$112.5K**
- Only b: **$150K**
- Only c: **$100K**

- **Balanced: $183.333K**

It is an optimal solution, but it is not an exact number of units

MERRIMACK COLLEGE

Break

Linear Programming

Sixth Assignment

Project - this week's Assignment

- Implement a Linear Programming solver as described in the slides.
- Basically, your program will receive the necessary information for an LP problem namely:
    - The number of variables (that will be equal to the number of constraint coefficients)
        - ***x***
    - The coefficient of each variable for the objective function (the column profit for slide 16 table)
        - ***f(x)***
    - The data for the square matrix stating the constraint (the inner elements of slide 16 table)
        - ***A***
    - The constraint limits (the last row of slide 16 table)
        - ***b***

MERRIMACK COLLEGE

Sixth Assignment

Project - this week's Assignment

- Having all input parameters, you will need to compute the possible solution for each variable alone and the solution of the linear equation system ***Ax = b***
  - To solve the linear equation system you should use the *numpy* package (further explanation here)
    - Roughly you will need to import *numpy* and use the functions:
      - *array(...)* to create a numpy array
      - *linalg.inv(...)* to perform a matrix inversion
      - *linalg.dot(...)* to perform a dot product
  - For example, the input for slide 17 example can be:
  - 3 variables/constraints, [3,2,2] as weights for the objective function, A = [[2,4,5],[1,2,4],[8,0,3]], and b = [300,200,300]

Linear Programming

Sixth Assignment

Project - this week's Assignment

- For example, the Input for slide 17 example would be:
    - 3 variables/constraints, [3,2,2] as weights for the objective function, A = [[2,4,5],[1,2,4],[8,0,3]], and b = [300,200,300]
- For this example, the call of linear equation system solution can be:

```
import numpy as np

A = np.array([[2,4,5],[1,2,4],[8,0,3]])
inv_A = np.linalg.inv(A)
b = np.array([300,200,300])
x = np.linalg.inv(A).dot(b)
```

- The output should be the Real number of unit to each of the variables

Let's run some code!

MERRIMACK COLLEGE

Programming Assignment

Project - this week's Assignment

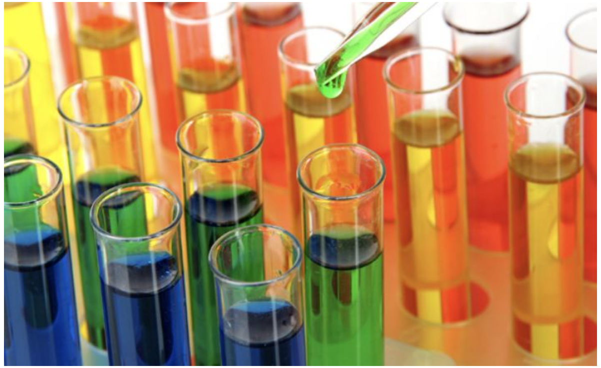
- This program must be your own, do not use someone else's code
- Any specific questions about it, please bring to the Office hours meeting this Friday or contact me by email
- This is a challenging program to make sure you are mastering your Python programming skills
- Don't be shy with your questions

Go to IDLE and try to program it
Save your program in a .py file and submit it in the appropriate delivery room

Deadline: Next Monday 11:59 PM EST

Linear Programming

Another Example



Let's run some code!

MERRIMACK COLLEGE

Maximize the Profit

- $f(a,b,c) = 3a + 2b + 2c$
  - $2a + 4b + 5c \le 300$
  - $a + 2b + 4c \le 200$
  - $8a + 3c \le 300$

Solver: WolframAlpha

- Only a: a = 37.5 units
- Only b: b = 75.0 units
- Only c: c = 50 units
- Balanced:
  - a = 25 units
  - b = 20.8333 units
  - c = 33.3333 units

supply	x	y	z	profit
a (1 unit)	2 gal.	1 gal.	8 gal.	$3K
b (1 unit)	4 gal.	2 gal.	0 gal.	$2K
c (1 unit)	5 gal.	4 gal.	3 gal.	$2K
availability	300 gal.	200 gal.	300 gal.	

Linear Programming applied to pants and jackets

Linear Programming

	pants	jackets	available
cotton	1	1,5	750
polyester	2	1	1,000

^From earlier slide, but here it is in better format (consistent with chemical example:

Supply	Cotton	Polyester	profit
Pants	1	2	50
Jackets	1.5	1	40
	750	1000	

Let's run some code!

MERRIMACK COLLEGE

- x_1 = number of pants, x_2 = number of jackets
  - $\boldsymbol{x = [\, x_1 , x_2 \,]}$
- Objective function
  - $c = (50 , 40)$
  - $\boldsymbol{f(x) = 50\, x_1 + 40\, x_2}$ $\quad\quad \boldsymbol{c^T x}$
- Constraints for cotton: 1 for pants, 1.5 for jackets, only 750 available
  - $1\, x_1 + 1.5\, x_2 \le 750$
  - $\boldsymbol{2\, x_1 + 3\, x_2 \le 1500}$
- Constraints for polyester: 2 for pants, 1 for jacket, only 1000 available
  - $\boldsymbol{2\, x_1 + 1\, x_2 \le 1000}$
- $A = [\, [2,3] , [2,1] \,]$ $\; b = [1500,1000]$ $\quad\quad \boldsymbol{Ax \le b}$
- All elements of x are greater or equal to 0
  - $\boldsymbol{x_1 \ge 0 , x_2 \ge 0}$ $\quad\quad \boldsymbol{x \ge 0}$

Integer Programming

Linear Programming Relaxation

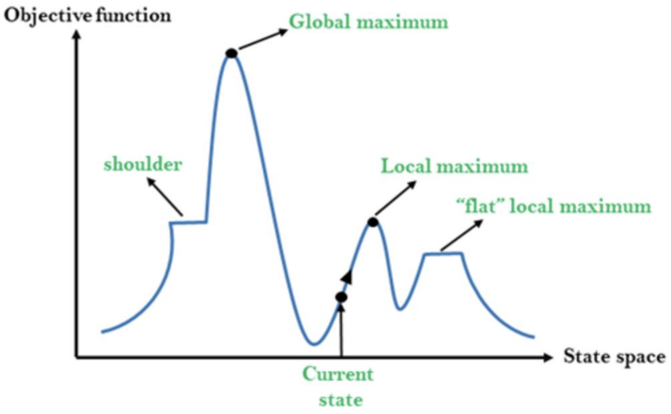
When your solution has to be Integer values

- For Linear Programming (LP) $x \in \mathbb{R}^n$
  - A vector of reals (positive)
- For Integer Linear Programming (ILP) $x \in \mathbb{N}^n$
  - A vector of positive integers
- A naive approximated approach is to solve with Reals and round it
  - It is called Linear Programming Relaxation
  - Not bad, but inexact (approximated)
- Sometimes the best solution is missed

MERRIMACK COLLEGE

Integer Programming

Exhaustive Solution



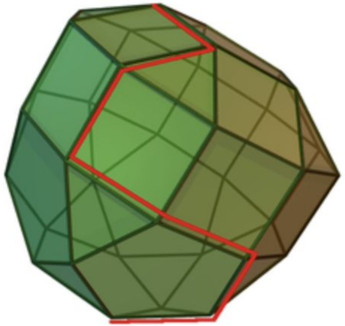
MERRIMACK COLLEGE

When your solution has to be Integer values

- For all possible combinations of (discrete) values for the variables compute the linear objective function
  - It may be incredibly slow for large range of options and large number of variables
  - Some optimization techniques can be applied as:
    - **Hill climbing** - choose an initial combination of values and start changing towards a direction
      - Effective with low dimension (few variables)
      - Misleading for large problems
      - Suboptimal solution

Integer Programming

The Simplex Algorithm



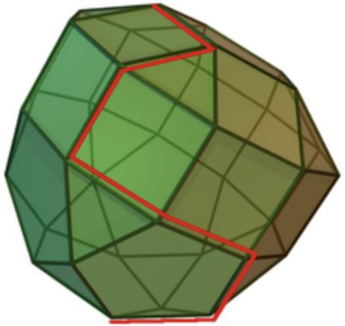
MERRIMACK COLLEGE

An old, popular, and reasonable solution

- Proposed by George Dantzig in 1947
- Mostly effective, but the solution may not be possible as ILP is a NP-hard problem
  - Simplex looks for the solution at the boundaries of the polyhedron that represents the feasible region
  - Simplex may deliver suboptimal results since the optimal solution may not be in one of the boundaries of the polyhedron
  - Simplex may also stall and not compute a solution

Integer Programming

The Simplex Algorithm



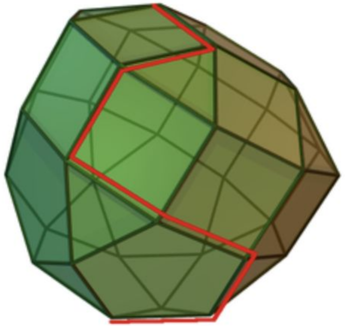
An old, popular, and reasonable solution

- Simplex basic technique is to replace inequalities by adding variables
    - For example the inequality $x_1 \geq 8$ may be replaced by the equation:
        - $x_1 = y_1 + 8$ where the variable y_1 was introduced ($y_1 \geq 0$)
- Another important procedure of Simplex are the pivot operations, when the algorithm moves from a basic feasible solution to an adjacent feasible solution
    - These operations are usually costly

MERRIMACK COLLEGE

Integer Programming

The Simplex Algorithm



MERRIMACK COLLEGE

An old, popular, and reasonable solution

- There are many implementations of the Simplex algorithm available in the Internet, as for example:
  - Online Calculator: Simplex Method
- The worst-case complexity of Simplex is exponential
  - However, in practice, many cases are considerably more efficient than the worst-case scenario
- A considerable number of problems of ILP have only approximate solutions that do not guarantee optimal solution or even a timely solution

Simplex Video

If you are interested in learning more about the simplex algorithm, the following video is recommended: https://www.youtube.com/watch?v=rzRZLGD_aeE

That’s all for today folks!

This week’s tasks

- Discussion: First response due Friday by midnight; two replies due by next Monday.
- Task #1 for the In-class exercises
  - Deadline: Friday 11:59 PM EST
- Quiz #6 to be available this Friday
  - Deadline: Next Monday 11:59 PM EST
- Project #6 assignment
  - Deadline: Next Monday 11:59 PM EST

- Try all exercises seen in class and consult the reference sources, as the more you practice, the easier it gets

Next week

- Randomized Algorithms

- Don’t let work pile up!
- Don’t be shy about your questions

MERRIMACK COLLEGE

Have a Great Week!