MSBA 403:

Optimization

Lecture 1

January 3rd, 2022

UCLA Anderson School of Management

Optimization and decision-making

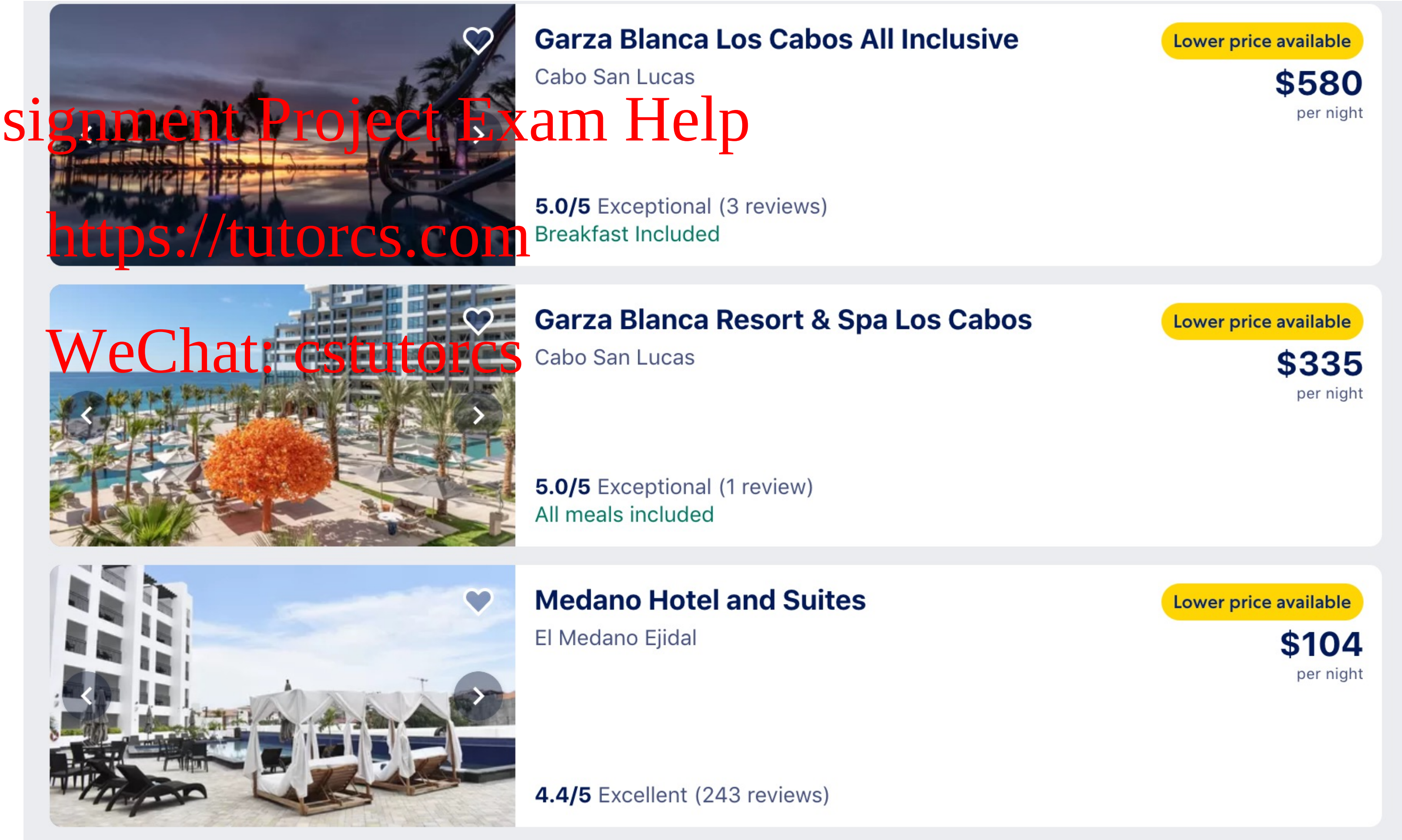
- We live in a world that is resource-constrained:
  - Money
  - Time
  - Labor
  - Physical space

- Optimization is fundamentally about making **decisions** that lead to favorable outcomes in the face of **constraints**.

- Building optimization models is both an art and science
  - Science because we rely on math and algorithms
  - Art because we often want to translate **real-world** problems into a **model.** The best way to do this is not always obvious!

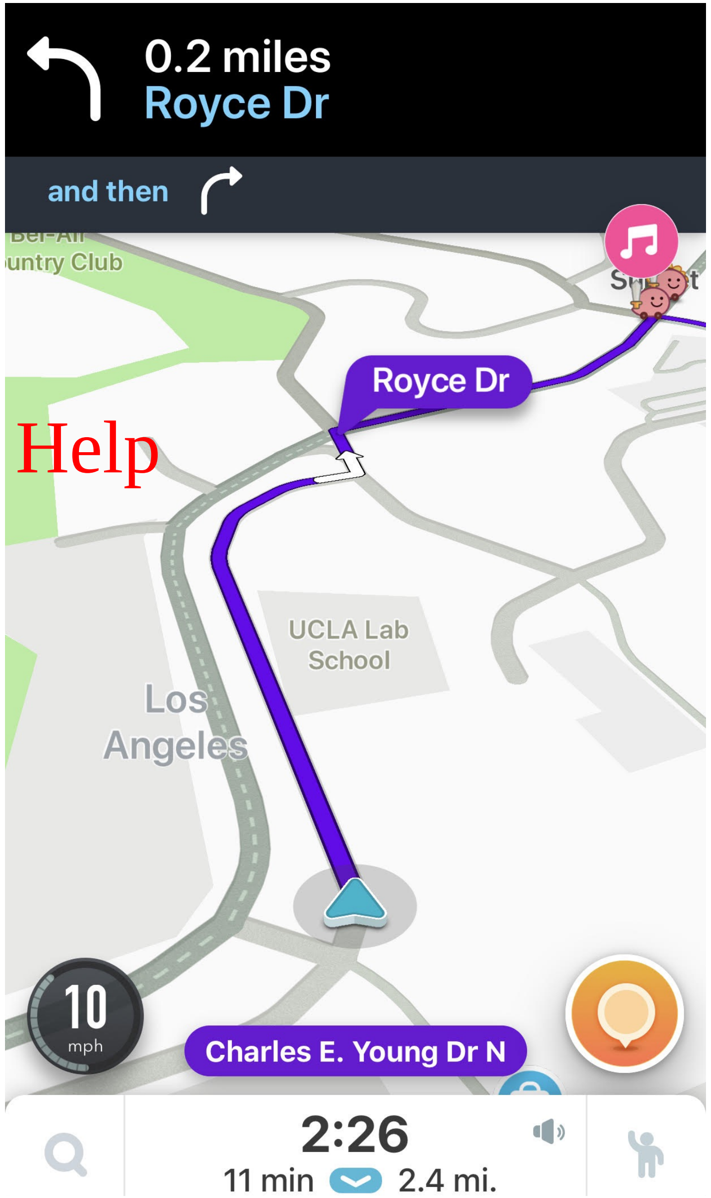
Example: Pricing and revenue management

How should Expedia set prices and manage capacity to maximize revenue?



Example: GPS routing

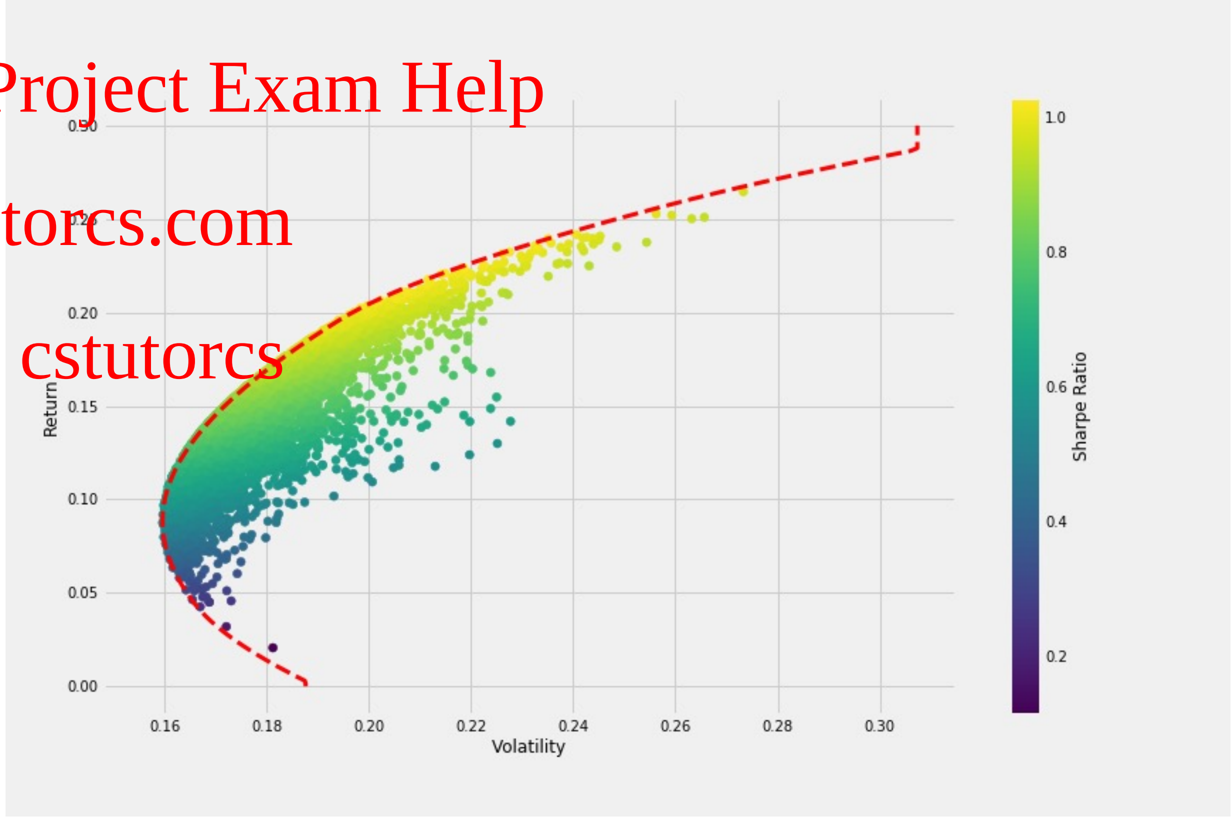
What is the optimal route to get home the fastest?



(My drive home)

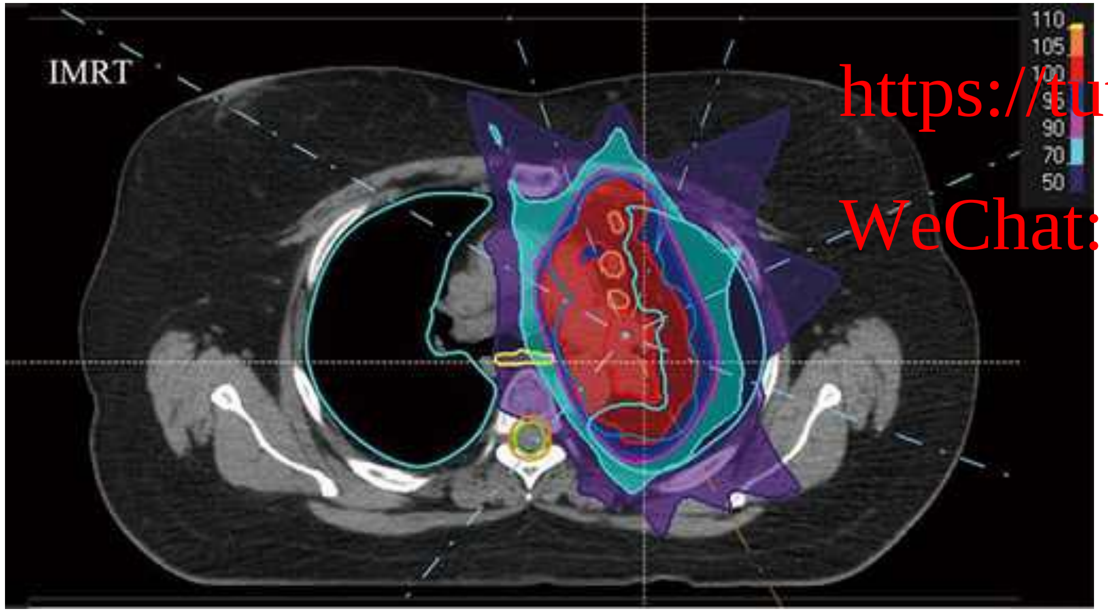
Example: Portfolio optimization

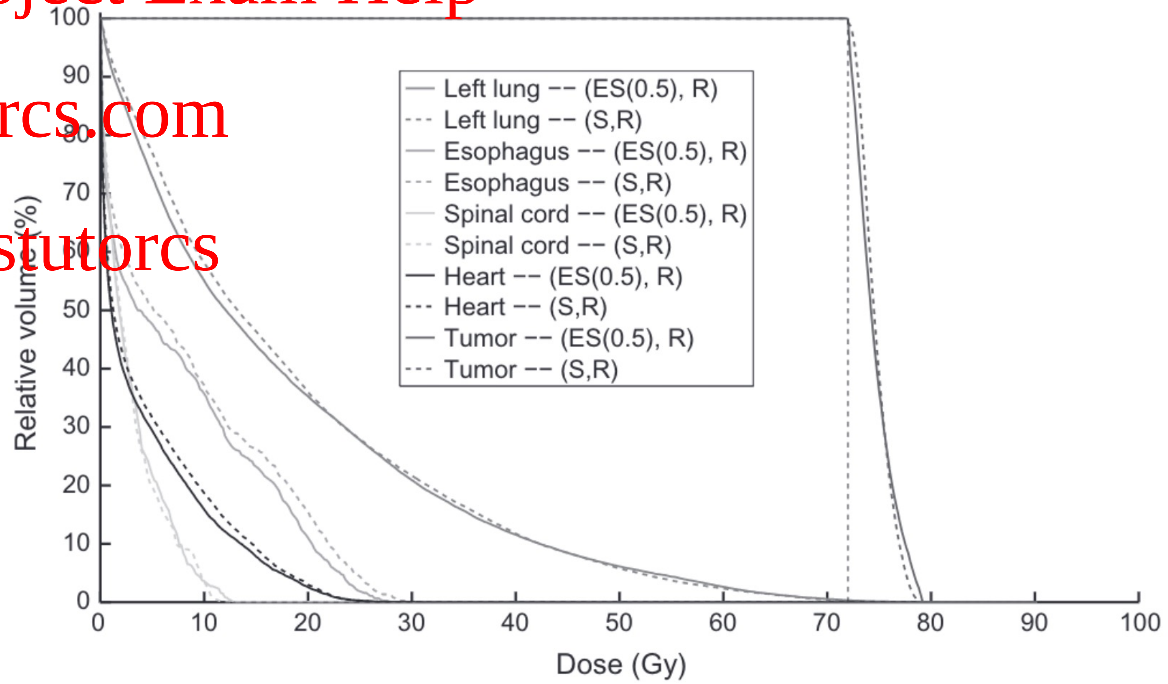
Which financial assets should we invest in to maximize returns?



Example: Radiation therapy

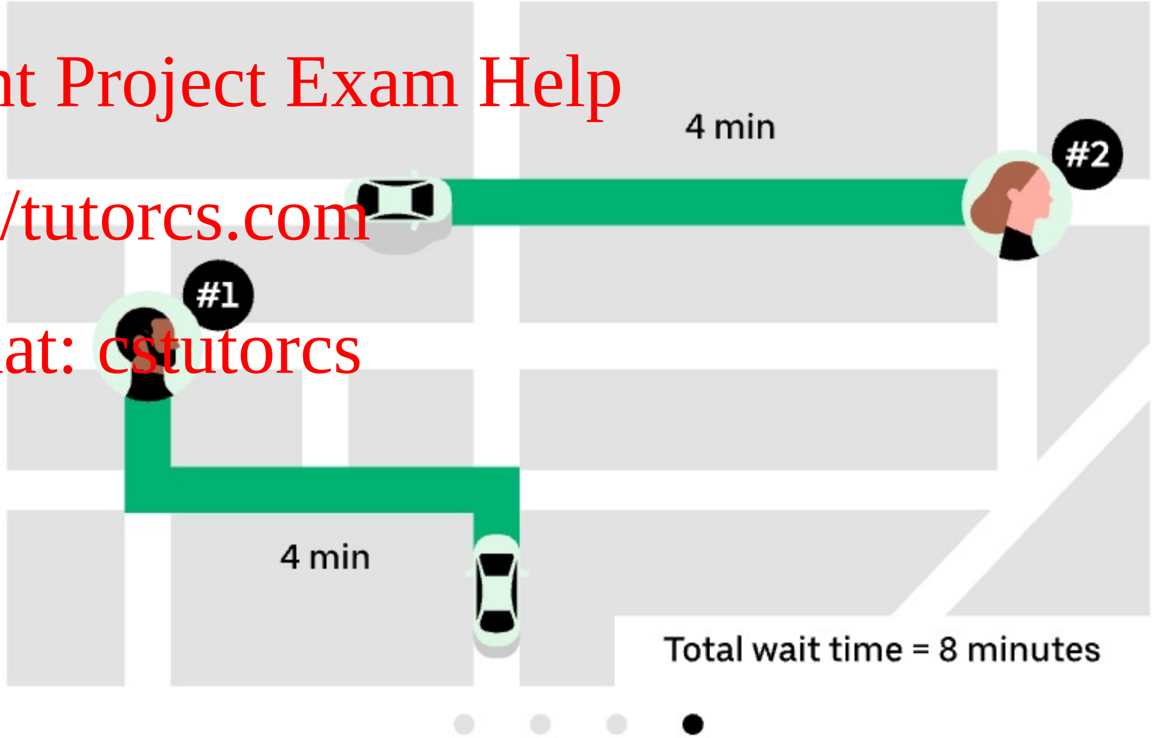
How can we maximize radiation to a tumor while minimizing damage to surrounding tissues?





Example: Ride-hailing matching

How should Uber match riders and drivers to minimize wait times?



Course objectives

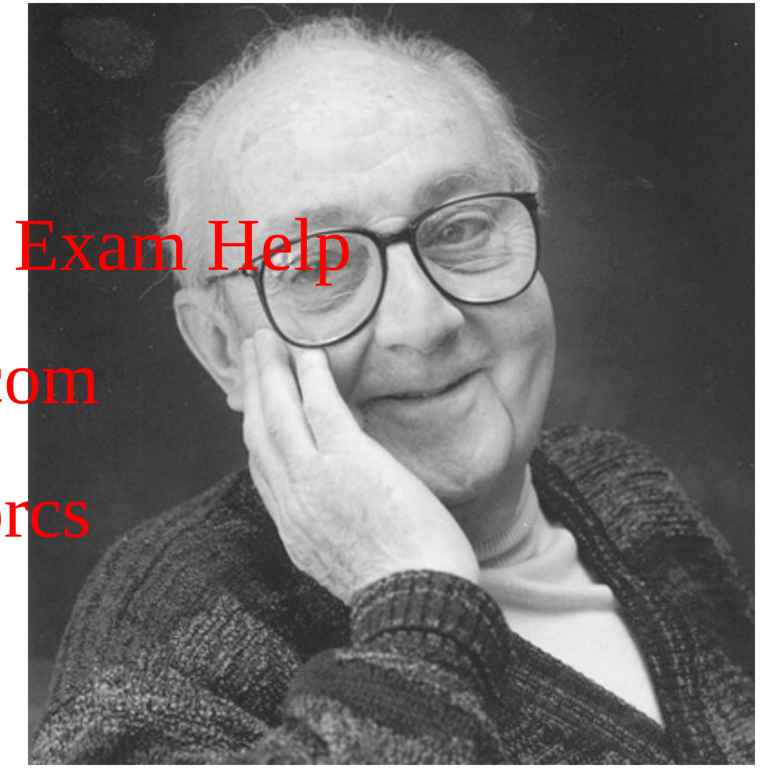
- Understand the fundamentals of optimization
- Express various optimization problems as mathematical models
- Implement and solve optimization models computationally
- Gain an appreciation for the value of optimization modeling **in practice**

Syllabus

Models vs reality

"All models are wrong, but some are useful."

George Box



- Mathematical models are just an *approximation* of reality
- Goal is to capture the most important features of the problem

Model structure: Decision variables

- **Decision variables** are denoted by letters, for example: $x_1, x_2, \ldots, x_n$
- Decision variables are at the heart of optimization
  - What prices should we set to maximize profit?
  - Which streets should we take?
  - How much should we invest in each stock?
  - How much radiation should be delivered from each beam?
  - Which rider should be matched to which Uber driver?
- The value of these variables are to be set by the decision-maker
- Often interpretable, but not always the case

Model structure: Constraints

- **Constraints** are denoted by functions of decision variables:

$$
\begin{aligned}
f_1(x_1, x_2, \ldots, x_n) &\leq 0 \\
f_2(x_1, x_2, \ldots, x_n) &\leq 0 \\
&\vdots \\
f_m(x_1, x_2, \ldots, x_n) &\leq 0
\end{aligned}
$$

- Can have sign ≤, ≥ or =

- Constraints capture limitations on functions of the decision variables, e.g.
  - Prices cannot be negative
  - Geographic constraints in GPS routing
  - Maximum acceptable risk or variance in portfolio
  - Maximum acceptable dose to healthy organs
  - Maximum time or distance constraints in ride-hailing matching

Model structure: Objective function

- The **objective function** is the quantity we would like to either **maximize** or **minimize:**

$$\max \quad f_0(x_1, x_2, \ldots, x_n)$$

or

$$\min \quad f_0(x_1, x_2, \ldots, x_n)$$

- Examples of objective functions:
  - Revenue from bookings
  - Total travel time by car
  - Return on investment
  - Radiation dosage to tumor
  - Total ride-hailing passenger wait time

Model structure: Summary

- The general form of an optimization model is given by

$$
\begin{aligned}
\max \quad & f_0(x_1, x_2, \ldots, x_n) \\
\text{s.t.} \quad & f_1(x_1, x_2, \ldots, x_n) \leq 0 \\
& f_2(x_1, x_2, \ldots, x_n) \leq 0 \\
& \vdots \\
& f_m(x_1, x_2, \ldots, x_n) \leq 0
\end{aligned}
$$

- Optimization problems written in this form are called **mathematical programs**
- Development of algorithms to solve mathematical programs is a significant area of research (although we won't focus on algorithms too much in this course)
- The format above is very general: We will first focus on an important class known as **linear programming**

Linear programming (LP)

- A linear program is an optimization model in which the objective function f_0 and constraints $f_1, f_2, \ldots, f_m$ are all linear functions of the decision variables $x_1, x_2, \ldots, x_n$

- Sounds restrictive, but this is an **extremely useful** class of models

- In practice, LPs can be solved very efficiently for thousands (or sometimes millions) of decision variables, which makes them a powerful optimization framework

- We will focus on LPs for the remainder of today's class

Example: Bag production

- Consider a company that produces UCLA-branded bags to be sold in the UCLA Store.
- Two types of bags can be made: Standard and Deluxe. Production consumes 4 resources: **cutting**, **sewing**, **finishing**, **inspecting**
- Due to machine and labor restrictions, the company has 630 hours of cutting, 600 hours of sewing, 708 hours of finishing, and 135 hours of inspecting
- Each Standard bag requires 7/10 hours of cutting, ½ hours of sewing, 1 hour of finishing, and 1/10 hour of inspecting
- Each Deluxe bag requires 1 hour of cutting, 5/6 hours of sewing, 2/3 hours of finishing, and 1/4 hours of inspecting
- **How many of each bag should be produced to maximize profit?**

Let's formulate the production problem as a linear program.

Decision variables

x_S = number of standard bags produced

x_D = number of deluxe bags produced

Objective function

Each standard bag sold generates a profit of \$10; each Deluxe bag sold generates a profit of \$9. So total profit is:

$$10x_S + 9x_D$$

Constraints

There is one constraint for each of the four resources: cutting, sewing, finishing, and inspecting.

Cutting: S requires 7/10 hour, D requires 1 hour. 630 hours available: $\frac{7}{10}x_S + x_D \leq 630$

Sewing: S requires 1/2 hour, D requires 5/6 hour. 600 hours available: $\frac{1}{2}x_S + \frac{5}{6}x_D \leq 600$

Finishing: S requires 1 hour, D requires 2/3 hour. 708 hours available: $x_S + \frac{2}{3}x_D \leq 708$

Inspecting: S requires 1/10 hour, D requires 1/4 hour. 135 hours available: $\frac{1}{10}x_S + \frac{1}{4}x_D \leq 135$

Non-negativity: Cannot produce negative quantities: $x_S, x_D \geq 0$

Example: Bag production

Putting it all together:

$$
\begin{aligned}
\max_{x_S, x_D} \quad & 10x_S + 9x_D && \text{profit function} \\
\text{s.t.} \quad & \frac{7}{10}x_S + x_D \leq 630 && \text{hours of cutting used} \leq \text{hours of cutting available} \\
& \frac{1}{2}x_S + \frac{5}{6}x_D \leq 600 && \text{hours of sewing used} \leq \text{hours of sewing available} \\
& x_S + \frac{2}{3}x_D \leq 708 && \text{hours of finishing used} \leq \text{hours of finishing time available} \\
& \frac{1}{10}x_S + \frac{1}{4}x_D \leq 135 && \text{hours of inspection used} \leq \text{hours of inspection time available} \\
& x_S, x_D \geq 0. && \text{non-negativity constraints}
\end{aligned}
$$

Optimal solution: **$(x_S, x_D) = (540, 252)$**

(This is obtained computationally; we will discuss the solution algorithm for linear programs shortly)

Example: Cargo revenue management

You are the manager of a cargo plane that ships goods overseas. You are deciding how much cargo to accept for an upcoming flight. The cargo plane has three compartments: front, center and back. Each of these compartments have capacities on both weight and volume:

Compartment	Weight Capacity (tons)	Volume Capacity (cubic feet)
Front	12	7,000
Center	18	9,000
Back	10	5,000

To ensure that the weight distribution on the plane remains balanced, the *ratio of cargo weights* in the respective compartments must be equal to the *ratio of weight capacities*. The following four cargoes have been offered for shipment for the flight:

Cargo	Weight (tons)	Volume(cubic feet /ton)	Profit ($/ton)
1	20	500	320
2	16	700	400
3	25	600	360
4	13	400	290

Any portion of each cargo can be accepted. Your decision problem is to determine how many tons of cargo to accept, and how to distribute them among the compartments of the plane to maximize profit. Formulate a linear program for this problem.

HINT: Let x_{ij} be tons of cargo i accepted into compartment j, where $j = 1$ is the front compartment, $j = 2$ is the center, and $j = 3$ is the back.
E.g.: x_{11} = tons of cargo 1 accepted into front compartment

Example: Cargo revenue management

Let x_{ij} be tons of cargo i accepted into compartment j, where $j = 1$ is the front compartment, $j = 2$ is the center, and $j = 3$ is the back.

Objective function:

$$\text{maximize} \;\; 320(x_{11}+x_{12}+x_{13}) + 400(x_{21}+x_{22}+x_{23}) + 360(x_{31}+x_{32}+x_{33}) + 290(x_{41}+x_{42}+x_{43})$$

Weight capacity constraints:

$$x_{11}+x_{21}+x_{31}+x_{41} \leq 12$$
$$x_{12}+x_{22}+x_{32}+x_{42} \leq 18$$
$$x_{13}+x_{23}+x_{33}+x_{43} \leq 10$$

Volume capacity constraints:

$$500x_{11}+700x_{21}+600x_{31}+400x_{41} \leq 7000$$
$$500x_{12}+700x_{22}+600x_{32}+400x_{42} \leq 9000$$
$$500x_{13}+700x_{23}+600x_{33}+400x_{43} \leq 5000$$

Example: Cargo revenue management

Maximum available cargo:

$$x_{11} + x_{12} + x_{13} \leq 20$$
$$x_{21} + x_{22} + x_{23} \leq 16$$
$$x_{31} + x_{32} + x_{33} \leq 25$$
$$x_{41} + x_{42} + x_{43} \leq 13$$

Balance constraints:

$$18(x_{11} + x_{21} + x_{31} + x_{41}) = 12(x_{12} + x_{22} + x_{32} + x_{42})$$
$$10(x_{11} + x_{21} + x_{31} + x_{41}) = 12(x_{13} + x_{23} + x_{33} + x_{43})$$

Non-negativity constraints:

$$x_{ij} \geq 0,\ \ i = 1, 2, 3, 4,\ \ j = 1, 2, 3.$$

Solving linear programs

- How do we solve a linear program?
- Let's focus on the bag production problem again

$$
\begin{aligned}
\max_{x_S, x_D} \quad & 10x_S + 9x_D \\
\text{s.t.} \quad & \frac{7}{10}x_S + x_D \leq 630 \\
& \frac{1}{2}x_S + \frac{5}{6}x_D \leq 600 \\
& x_S + \frac{2}{3}x_D \leq 708 \\
& \frac{1}{10}x_S + \frac{1}{4}x_D \leq 135 \\
& x_S, x_D \geq 0.
\end{aligned}
$$

Feasible solutions and optimal solutions

- A **feasible solution** to a linear program is a set of values for the decision variables that does not violate any constraints
  - E.g. $(x_S, x_D) = (100, 200)$ is *feasible*, because it satisfies all constraints
  - E.g. $(x_S, x_D) = (50, 550)$ is *infeasible*, because it violates the inspection time constraint

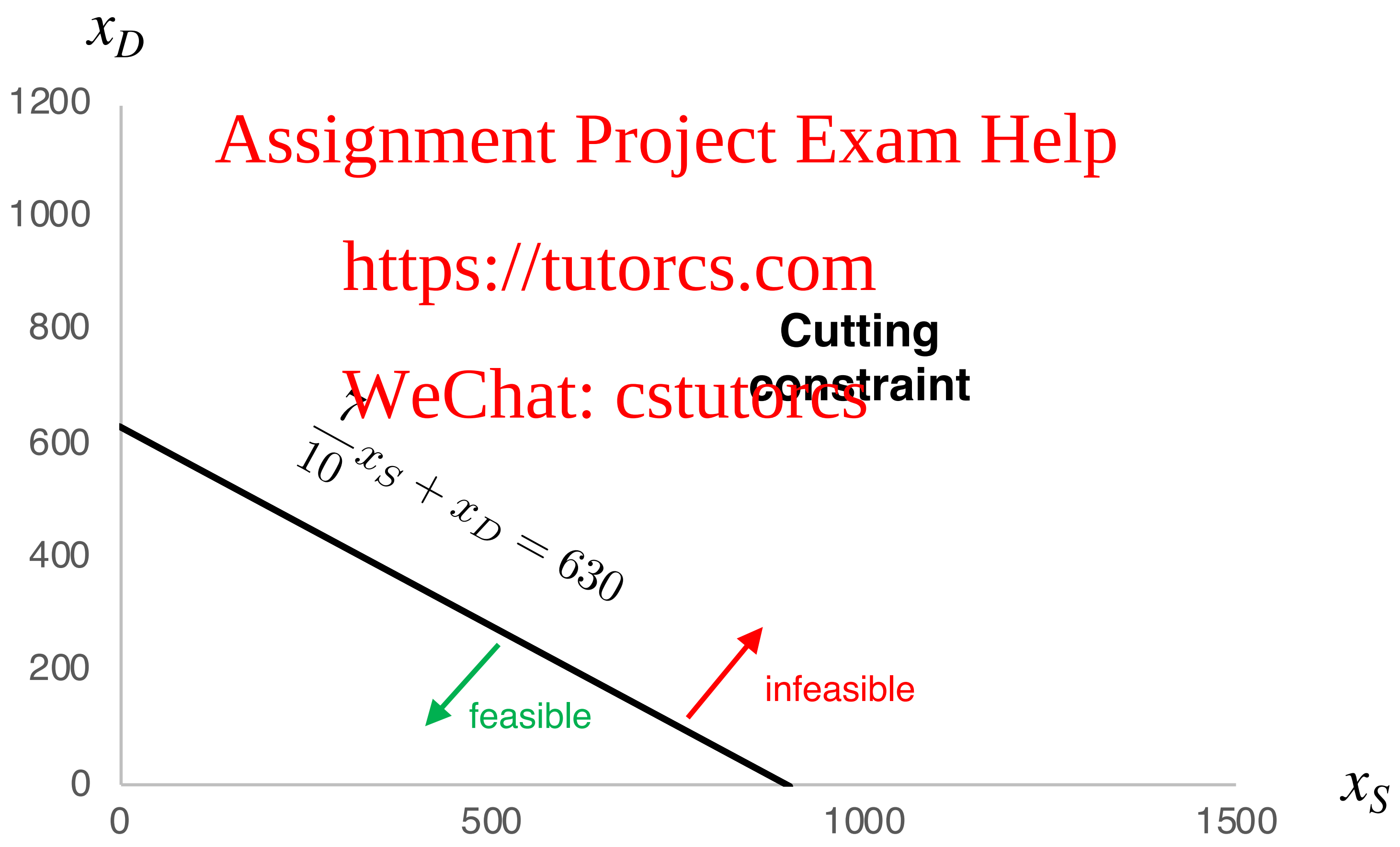
- An **optimal solution** is a feasible solution that produces *at least* as good of an objective function value as every other possible feasible solution

- We will refer to the best possible objective function value as the **optimal value**

- There can be more than one optimal solution! In other words, two different solutions can attain the optimal value

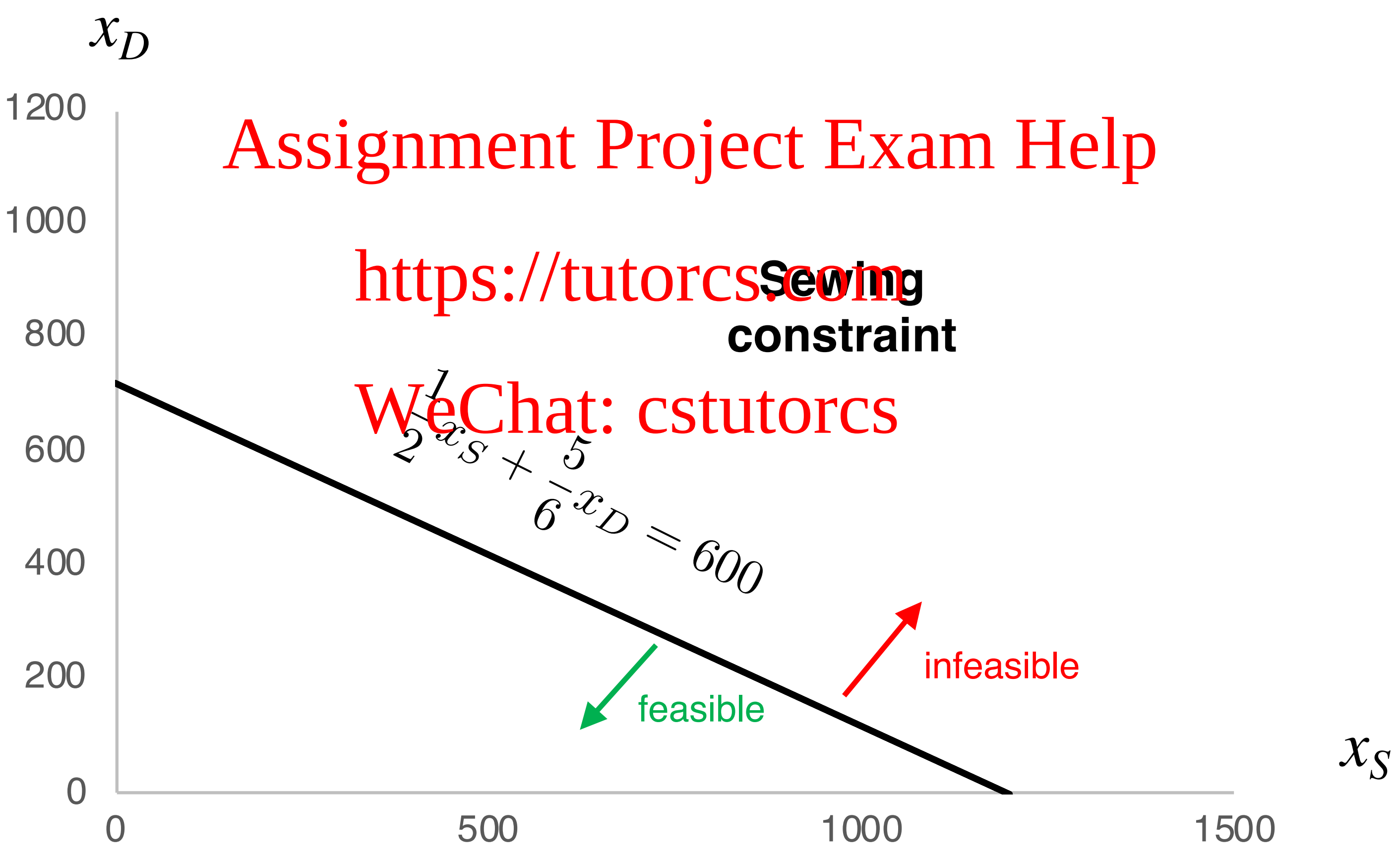
Graphical representation

- Because we have two decision variables in the bag production problem, we can represent the feasible solution of this LP graphically



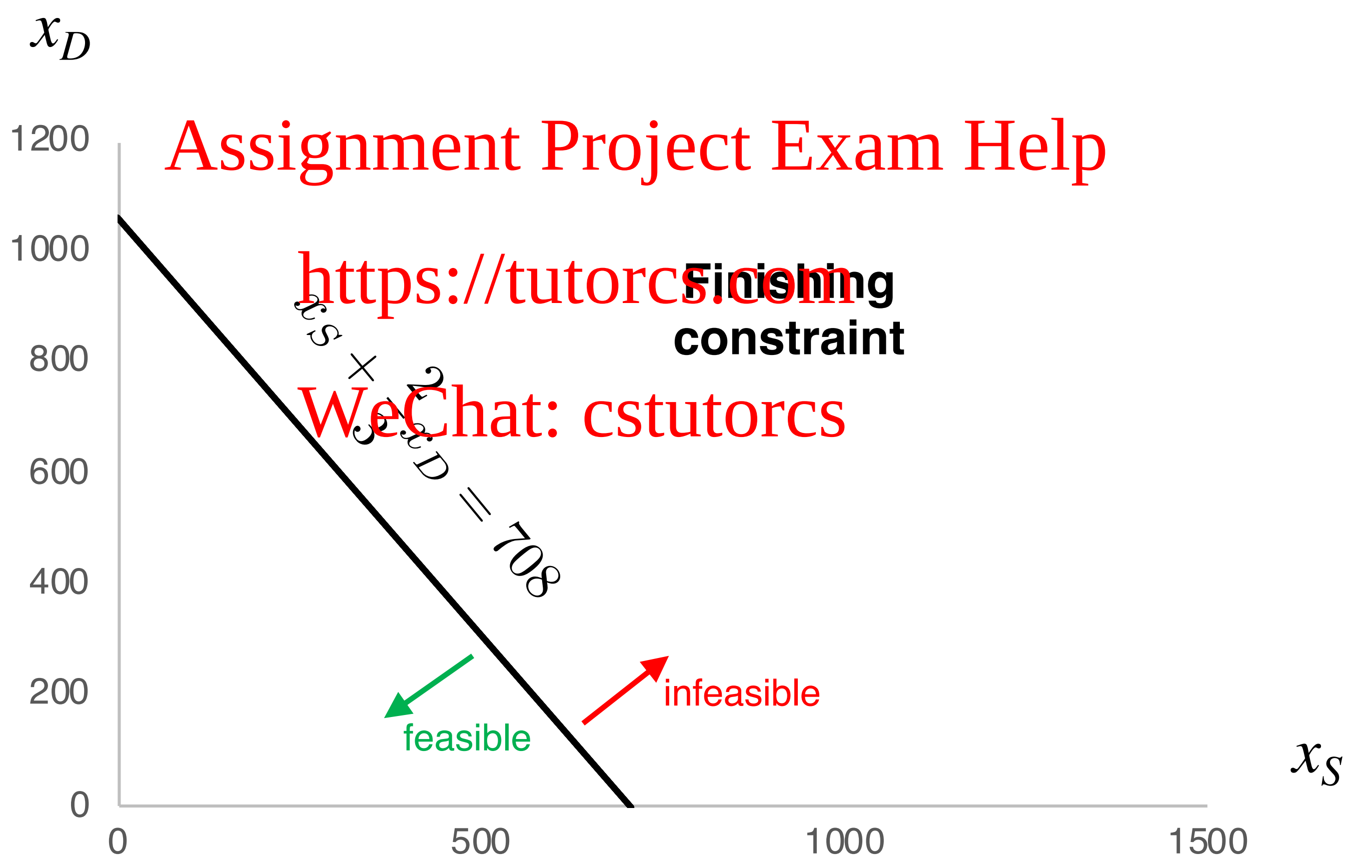
Graphical representation

- Because we have two decision variables in the bag production problem, we can represent the feasible solution of this LP graphically



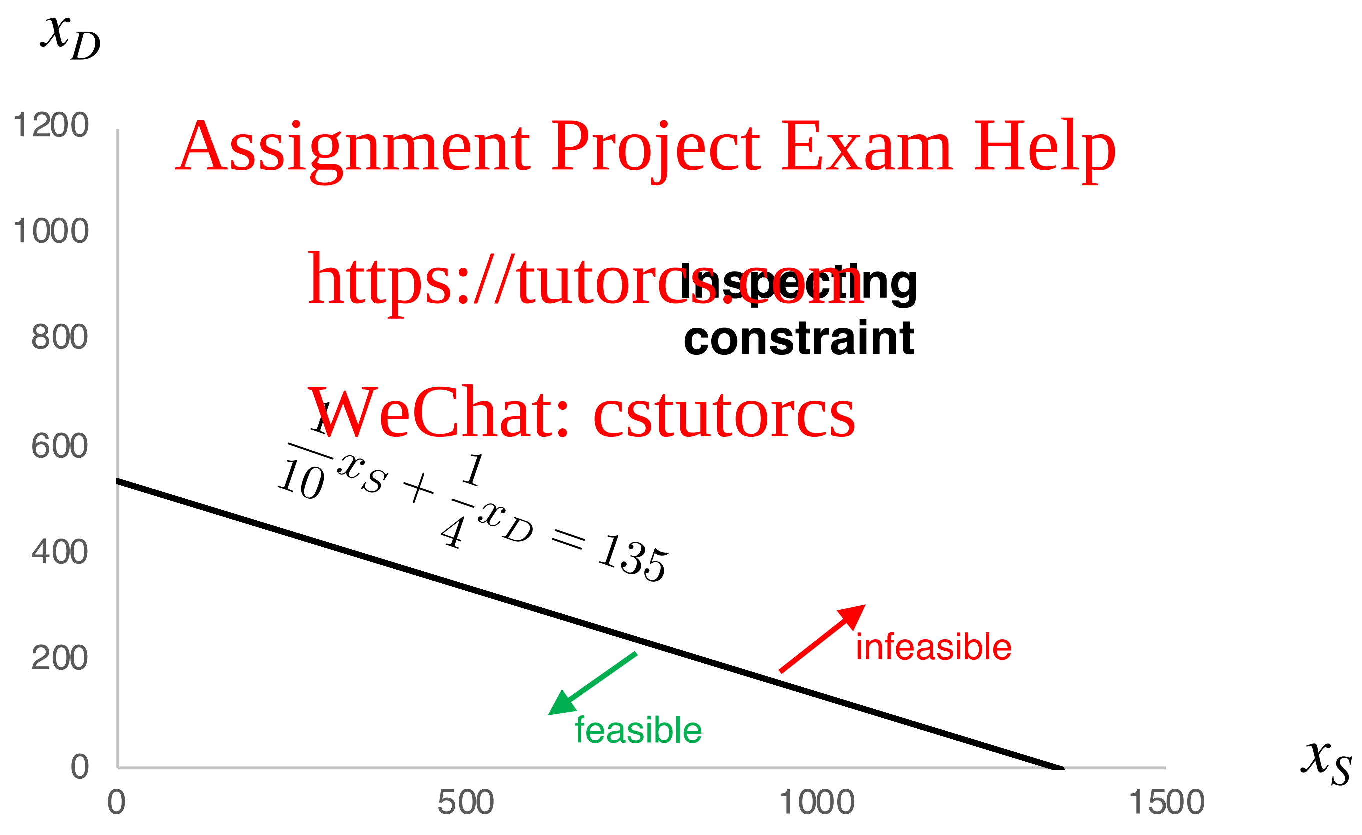
Graphical representation

- Because we have two decision variables in the bag production problem, we can represent the feasible solution of this LP graphically



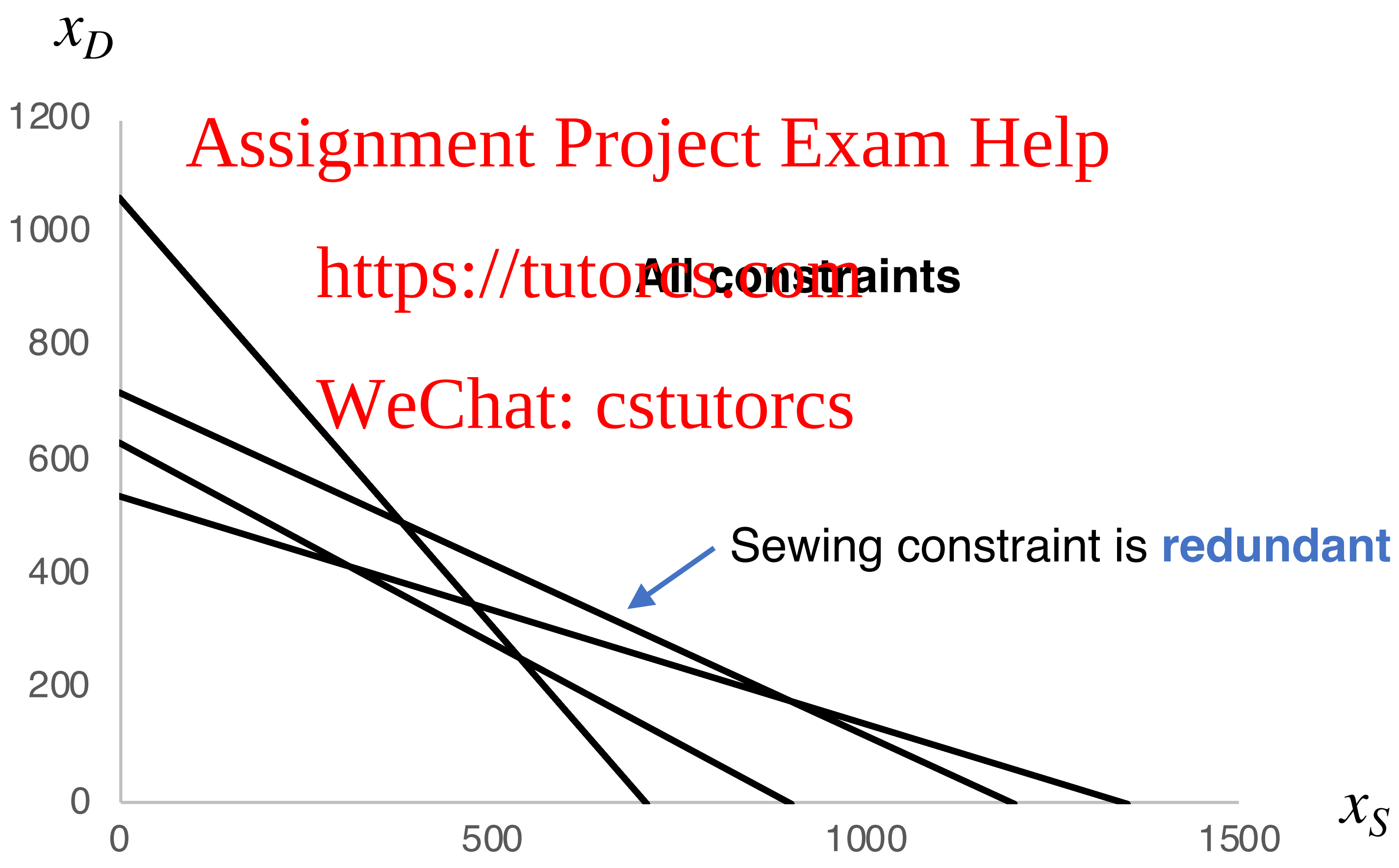
Graphical representation

- Because we have two decision variables in the bag production problem, we can represent the feasible solution of this LP graphically



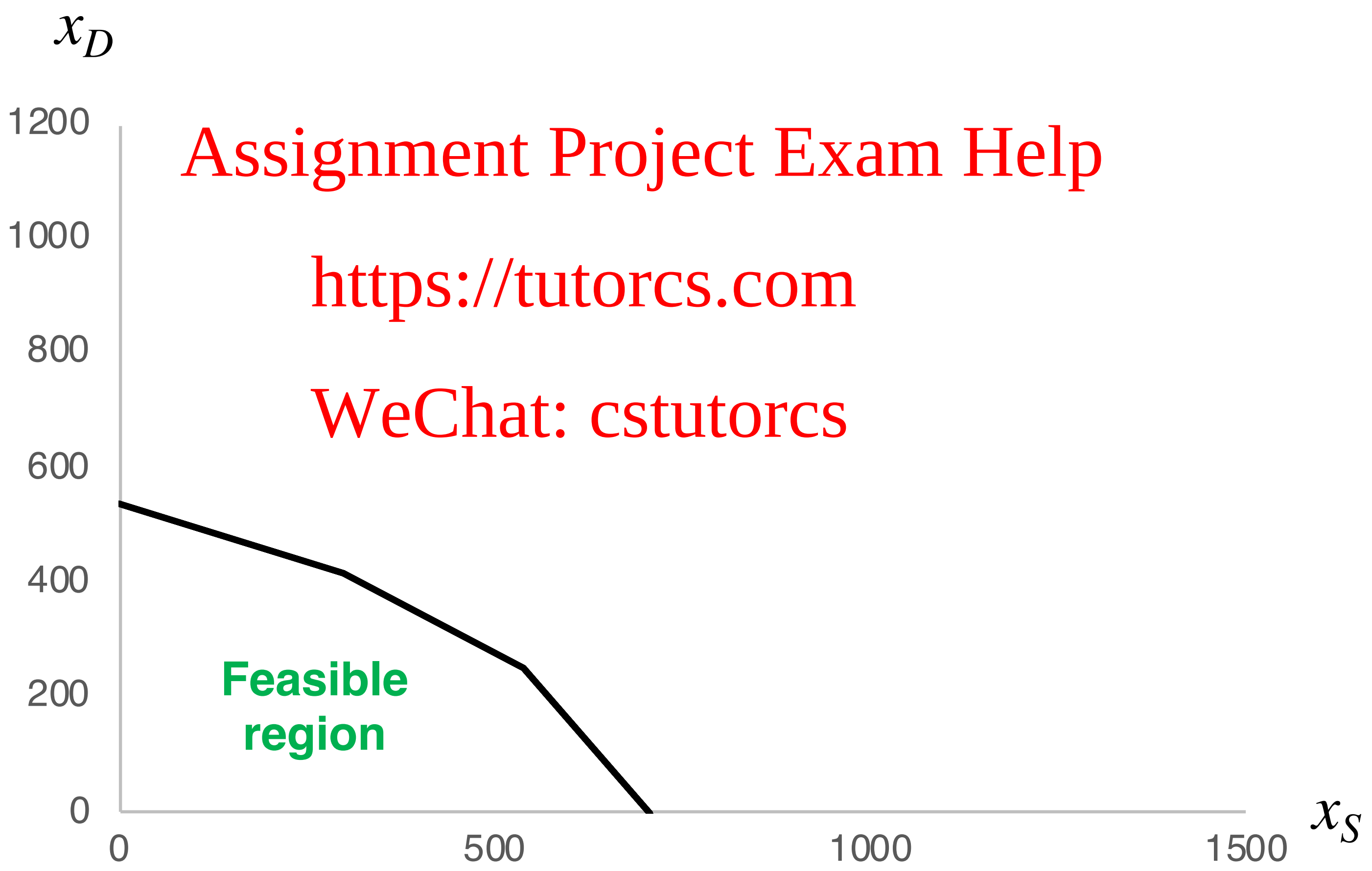
Graphical representation

- Because we have two decision variables in the bag production problem, we can represent the feasible solution of this LP graphically



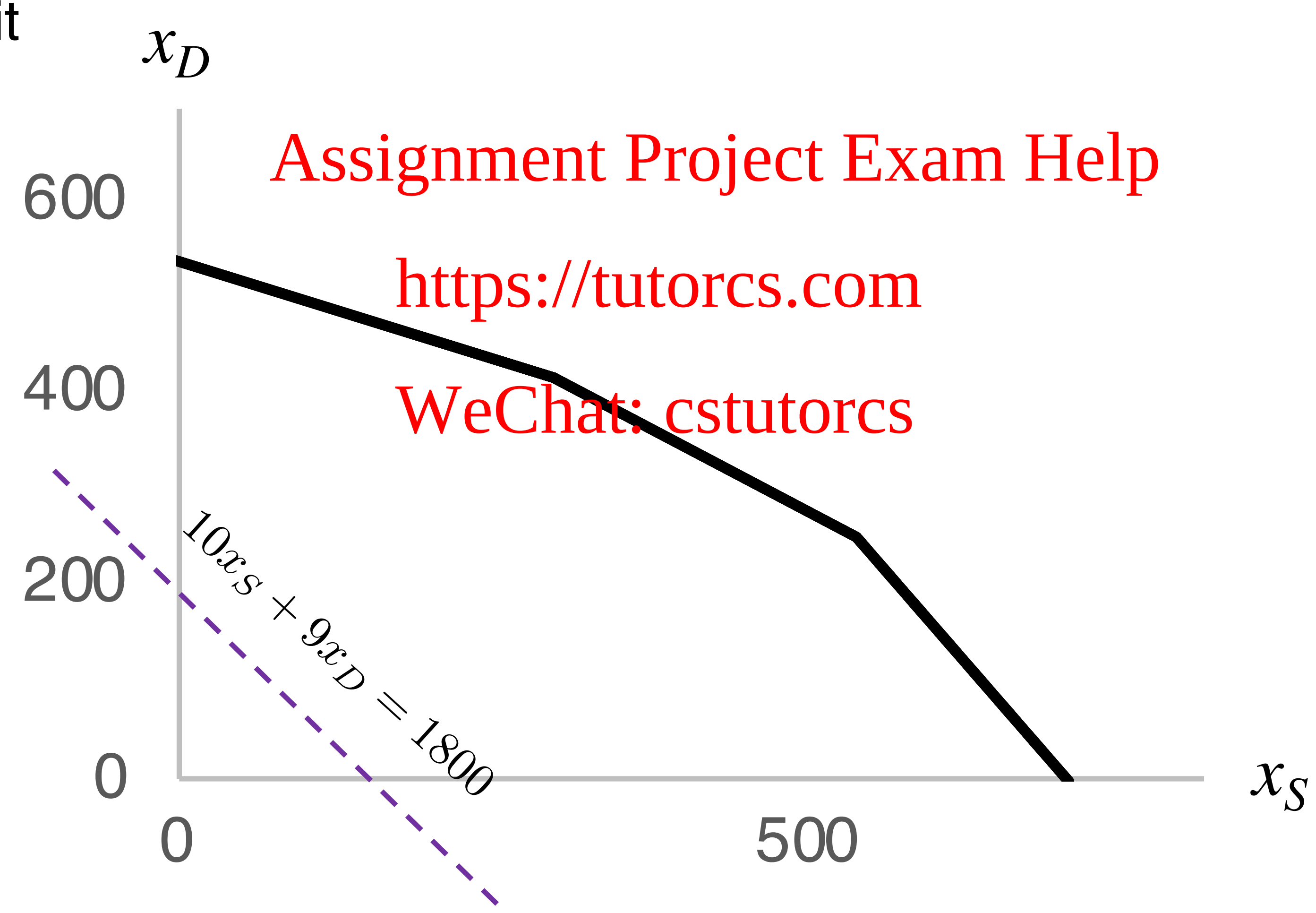
Graphical representation

- The **feasible region** is the set of all feasible solutions



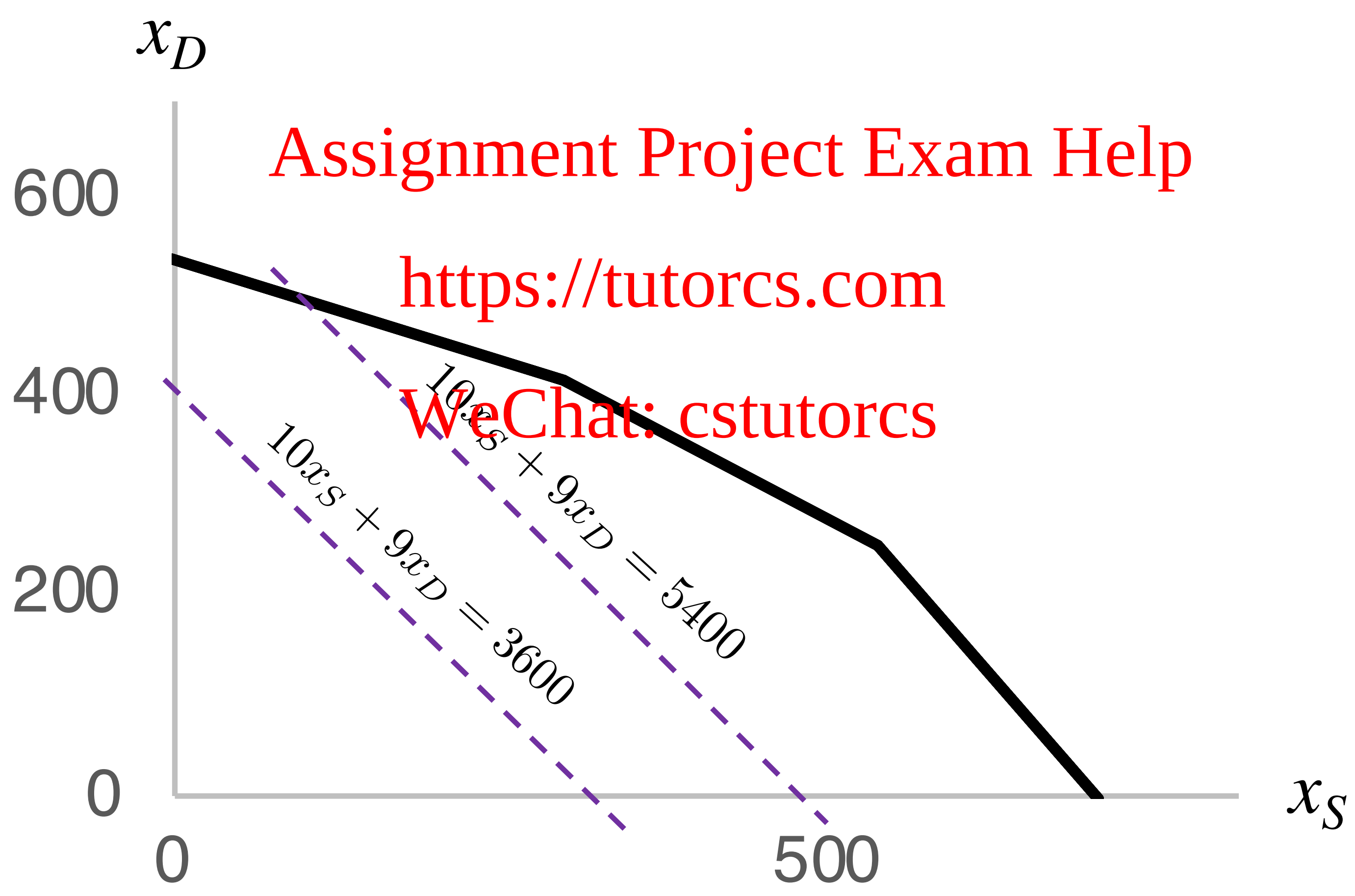
Visualizing the objective function

- What is the objective function value (i.e. profit) associated with each solution?
- We can visualize the objective by using profit line: every point along this line yields the same profit



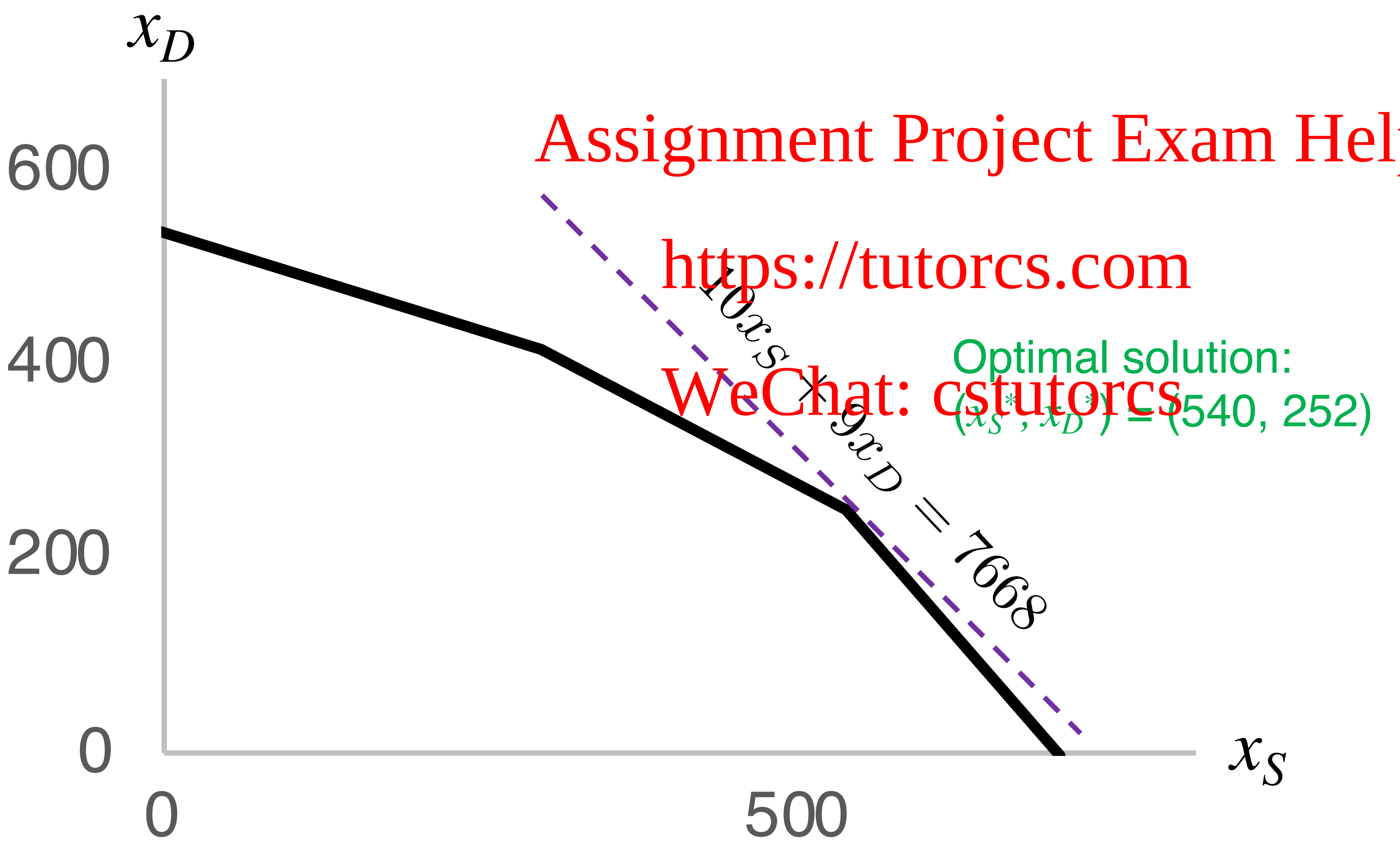
Visualizing the objective function

- Shifting the profit line yields a higher profit



Visualizing the objective function

- Optimal solution is attained at an **extreme point** of the feasible region
- Optimal solution is to make **540** Standard bags and **252** Deluxe bags, which gives profit of **$7668**



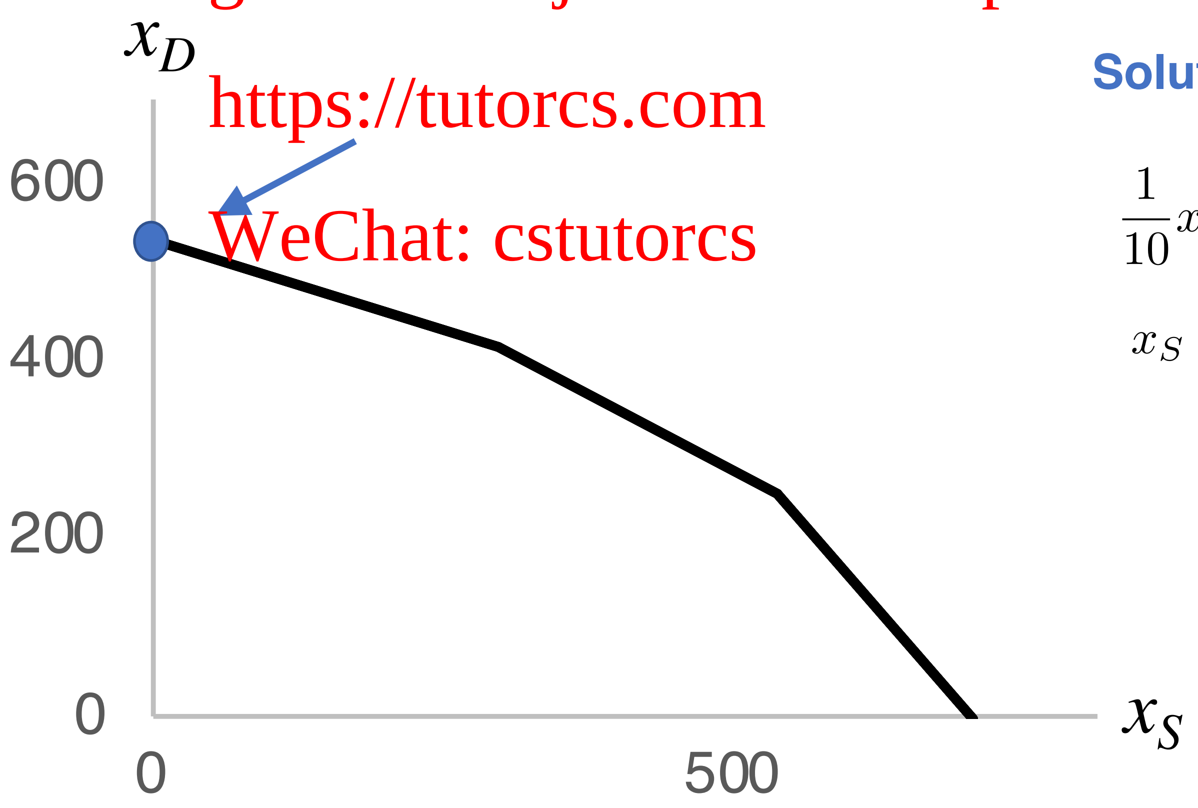
Is an optimal solution to a linear program *always* attained at an extreme point of the feasible region?

Yes!

Solving linear programs

- Property of LP that a solution is always obtained at an extreme point is very useful – *it means we only need to search extreme points to find an optimal solution*

- Note that each extreme point is a solution to a set of linear equations



Solution to:

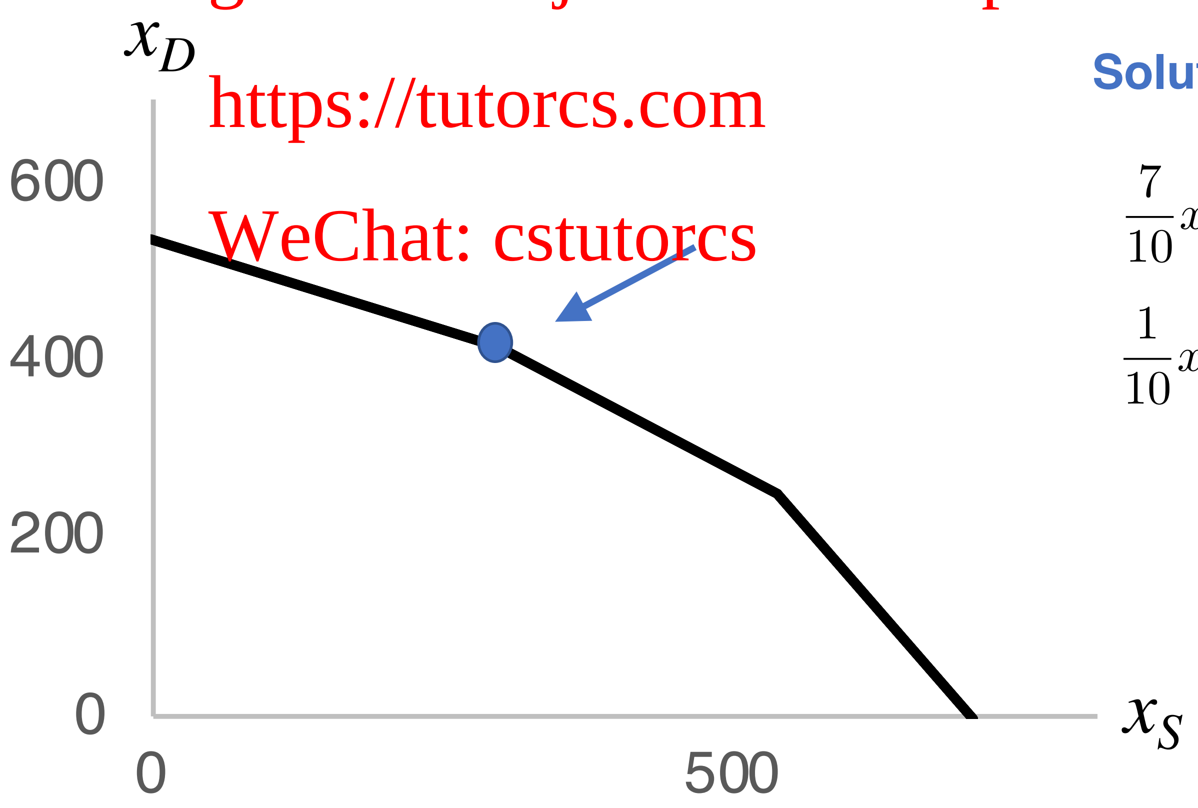
$$\frac{1}{10}x_S + \frac{1}{4}x_D = 135$$

$$x_S = 0$$

Solving linear programs

- Property of LP that a solution is always obtained at an extreme point is very useful – *it means we only need to search extreme points to find an optimal solution*

- Note that each extreme point is a solution to a set of linear equations



Solution to:

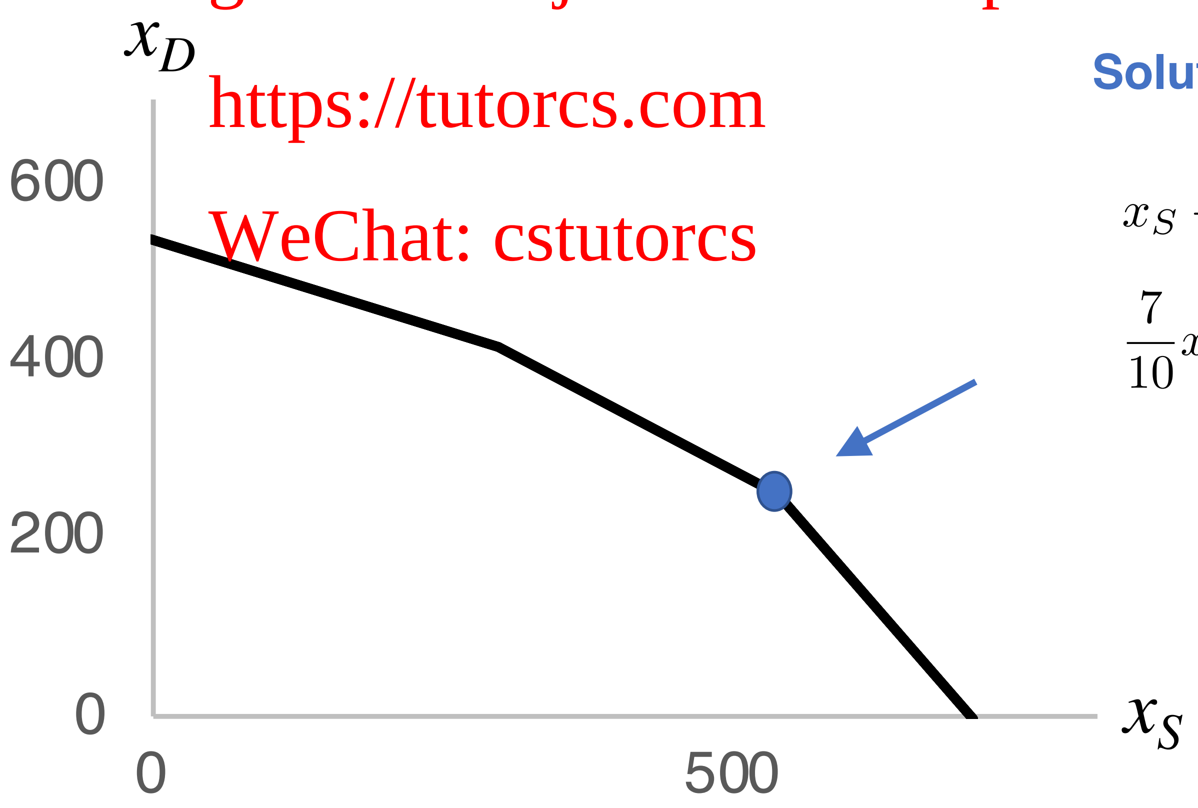
$$\frac{7}{10}x_S + x_D = 630$$

$$\frac{1}{10}x_S + \frac{1}{4}x_D = 135$$

Solving linear programs

- Property of LP that a solution is always obtained at an extreme point is very useful – *it means we only need to search extreme points to find an optimal solution*

- Note that each extreme point is a solution to a set of linear equations



Solution to:

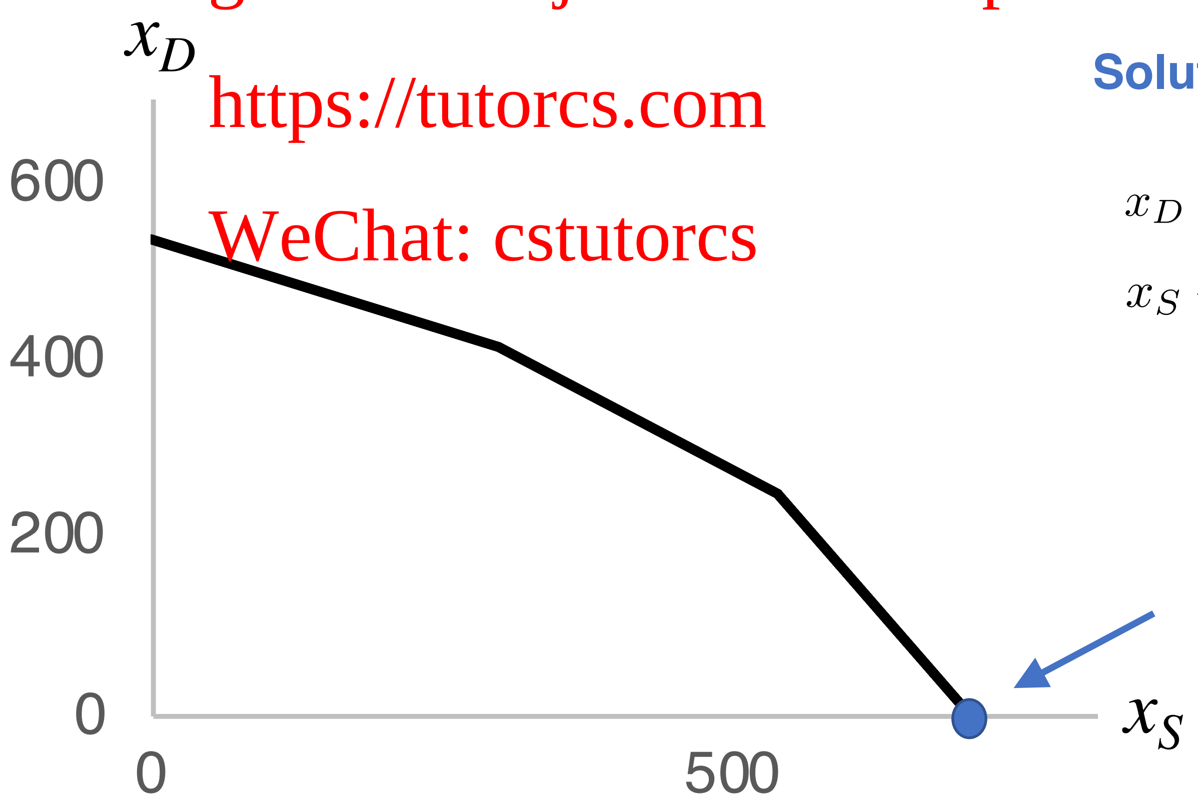
$$x_S + \frac{2}{3}x_D = 708$$

$$\frac{7}{10}x_S + x_D = 630$$

Solving linear programs

- Property of LP that a solution is always obtained at an extreme point is very useful – *it means we only need to search extreme points to find an optimal solution*

- Note that each extreme point is a solution to a set of linear equations



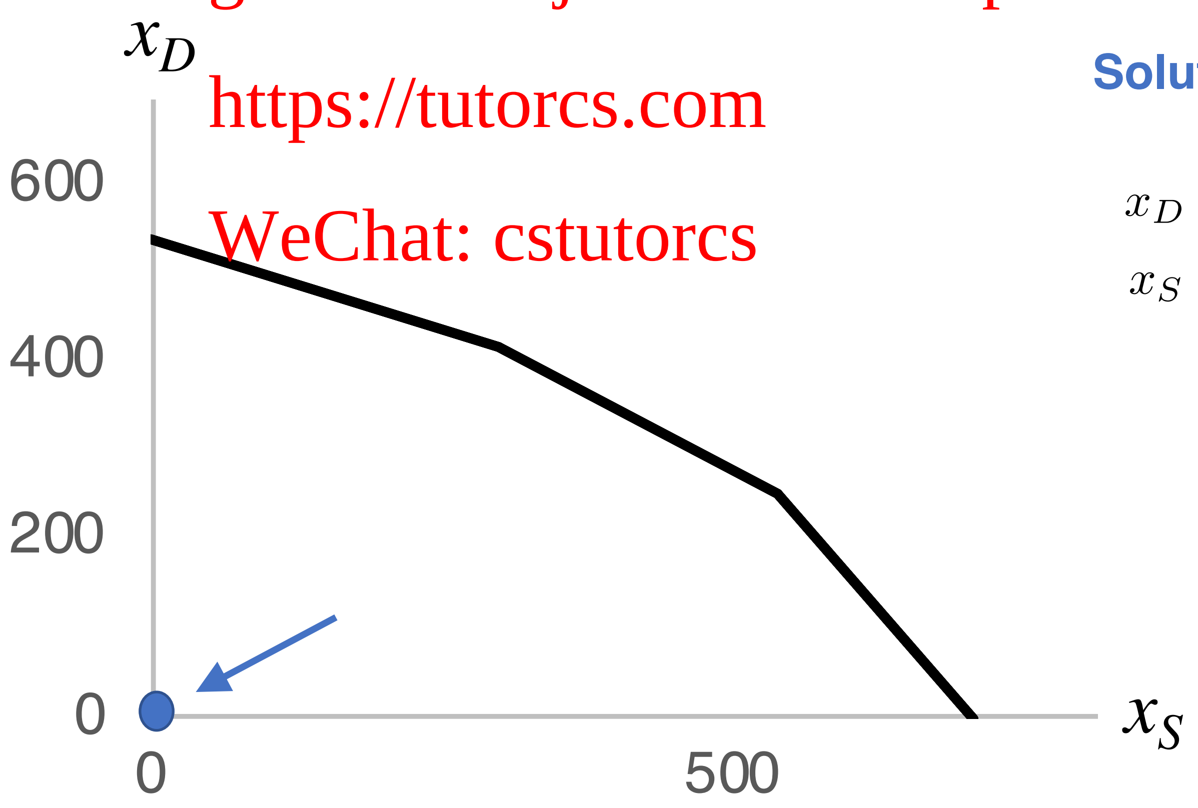
Solution to:

$$x_D = 0$$
$$x_S + \frac{2}{3}x_D = 708$$

Solving linear programs

- Property of LP that a solution is always obtained at an extreme point is very useful – *it means we only need to search extreme points to find an optimal solution*

- Note that each extreme point is a solution to a set of linear equations



Solution to:

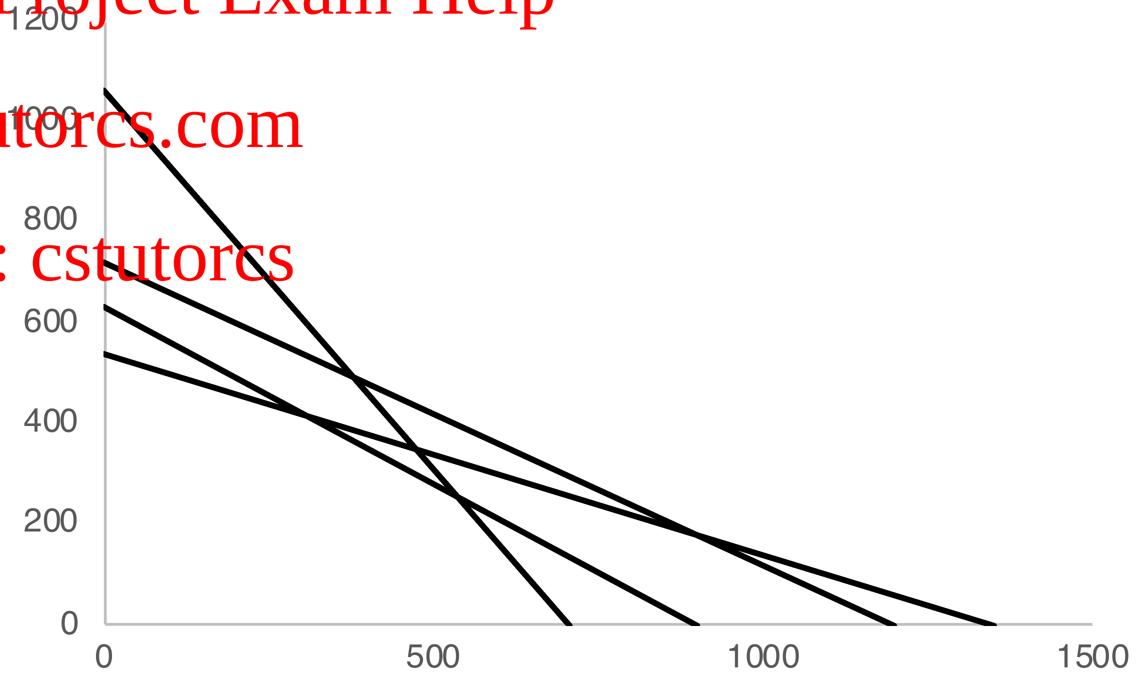
$$x_D = 0$$
$$x_S = 0$$

Solving linear programs

- What if we just solved for every possible extreme point? How many are there?
- An upper bound on the number of extreme points of the feasible region is the number of intersections between constraint lines
- Because we have 6 constraints (including non-negativity) and 2 variables, there are a total of

$$\binom{6}{2} = \frac{6!}{2!(6-2)!} = 15$$

intersection points.



- In general, with m constraints and n variables, the number of intersection points is **combinatorial:**

$$\binom{m}{n} = \frac{m!}{n!(m-n)!}$$

Enumerating all possible solutions is not practical!

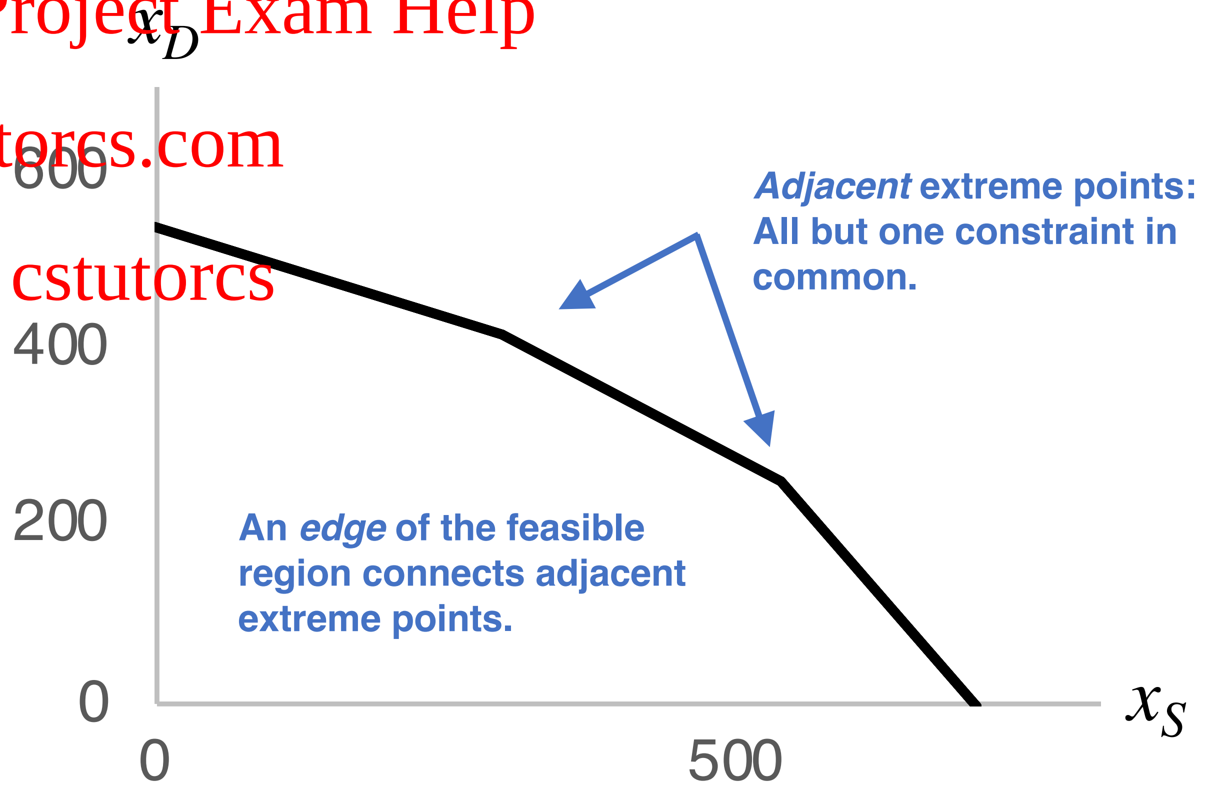
The Simplex Method

- LPs can be solved using an extremely efficient algorithm known as the simplex method (also known as “simplex algorithm”)

- The simplex algorithm is the reason why we can solve large scale linear programs with thousands or millions of variables

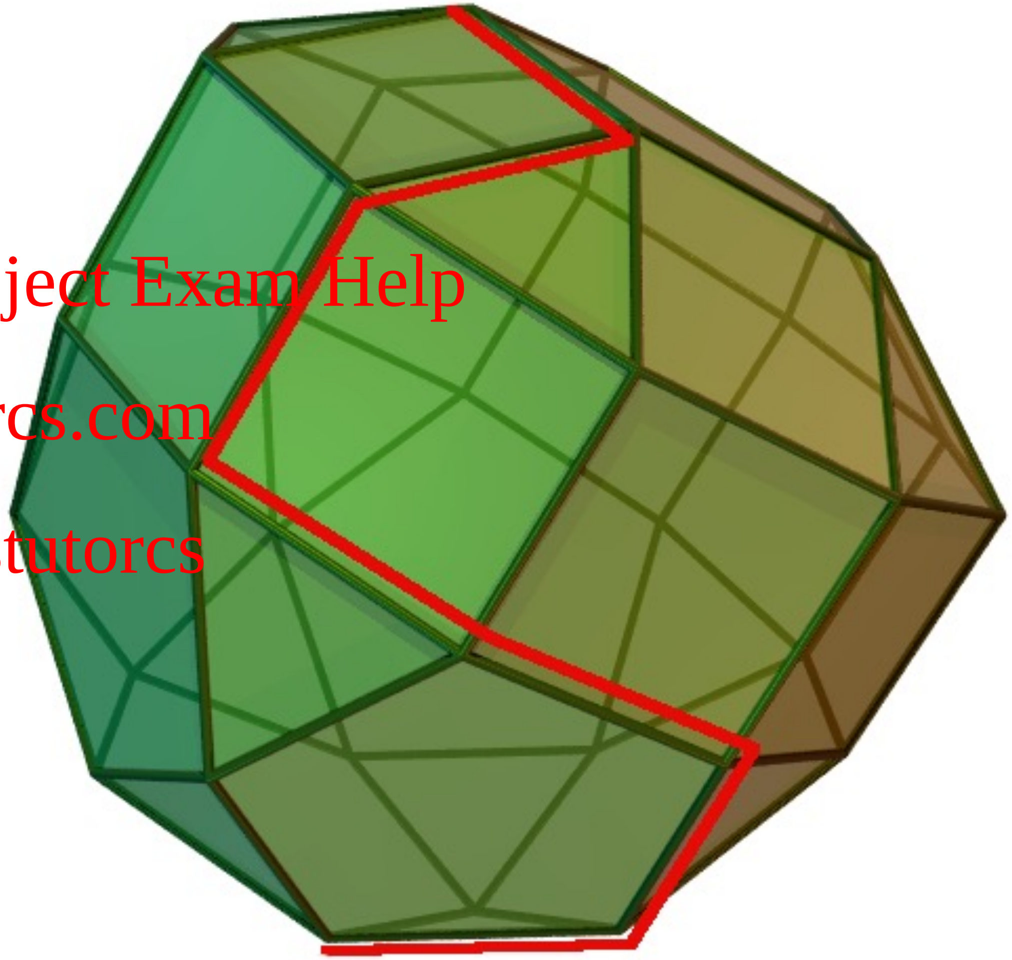
(Rough) Simplex Method Steps

1. Start at an extreme point.
2. Check to see if objective function can be improved by moving to an *adjacent* extreme point.
3. If objective can be improved, move to adjacent extreme point, and go back to step 2.
4. If objective cannot be improved, current extreme point is optimal.



The Simplex Method

The simplex method moves from one extreme point to another, by traversing edges of the feasible region, until an optimal solution is found.



Linear programs: General notation

- For larger problems, it may not be realistic to write out all of the variables and constraints explicitly

- We can write linear programs in a concise way by using *set* and *index notation*

Example: Product delivery

Consider a company that wishes to delivery a product to a set of customers in different locations in the most cost-efficient way possible.

- The available supply of the product is s_i at warehouses $i = 1,2,\ldots,m$.
- The company must satisfy demand d_j at locations $j = 1,2,\ldots,n$.
- The cost of delivering the product warehouse i to location j is given by c_{ij}.

Formulate a linear program that provides a cost-minimizing delivery plan.

Example: Product delivery

The linear program for the product delivery problem is given by:

$$\min_{\mathbf{x}} \quad \sum_{i=1}^{m} \sum_{j=1}^{n} c_{ij} x_{ij}$$

Minimize total delivery cost

$$\text{s.t.} \quad \sum_{i=1}^{m} x_{ij} \geq d_j, \quad j = 1, 2, \ldots, n,$$

Total delivery to each location must satisfy demand

$$\sum_{j=1}^{n} x_{ij} \leq s_i, \quad i = 1, 2, \ldots, m,$$

Each warehouse cannot deliver more than available supply

$$x_{ij} \geq 0, \quad i = 1, 2, \ldots, m, \; j = 1, 2, \ldots, n.$$

Non-negativity constraints

Solving LPs with Python and Gurobi

- We will use Python to build and solve optimization models in this course
- In particular, we will use an **optimization solver** called Gurobi, which implements various solution algorithms (including the Simplex Method)
- Let's look at a couple of examples in Python

Python and Gurobi installation

1. Install Python using the Anaconda distribution.
https://www.anaconda.com/distribution/

2. Request a free academic license for Gurobi using your UCLA email address. Follow the steps in the page below to retrieve and install your license key.
https://www.gurobi.com/academia/academic-program-and-licenses/

3. Install Gurobi for Python. See the following link for assistance:

https://support.gurobi.com/hc/en-us/articles/360044290292-How-do-I-install-Gurobi-for-Python-

4. We will use a user-friendly interface for Python called Jupyter Notebook. To launch Jupyter Notebook, open terminal or cmd and type

```
jupyter notebook
```

This will launch Jupyter Notebook in your browser. We can now open and run Jupyter Notebook files (extension .ipynb) from within the browser.

Do this before lecture 2!