Load Balancing Problem

(Approximation Example)

Input:

m resources with equal processing power

n jobs, where job j takes t_j to process

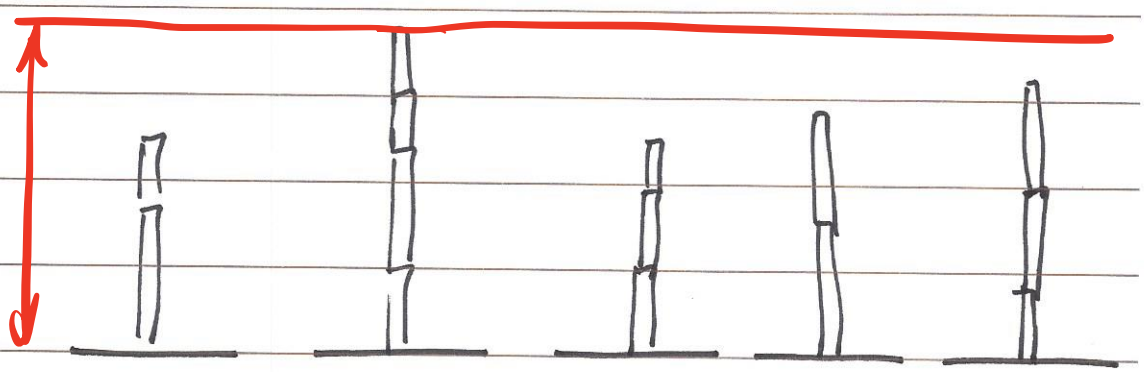
objective:

Assignment of jobs to resources such that the maximum load on any machine is minimized.

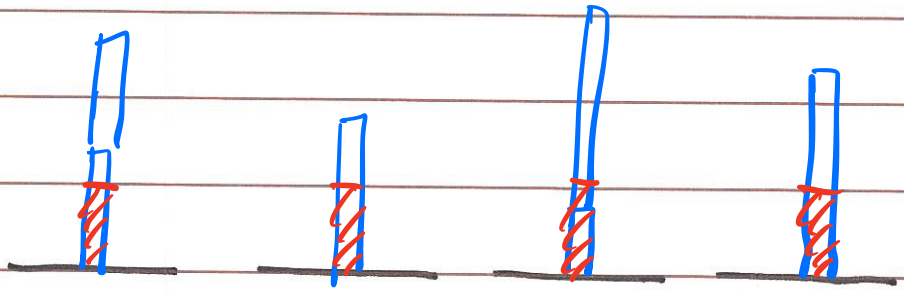
Notation:

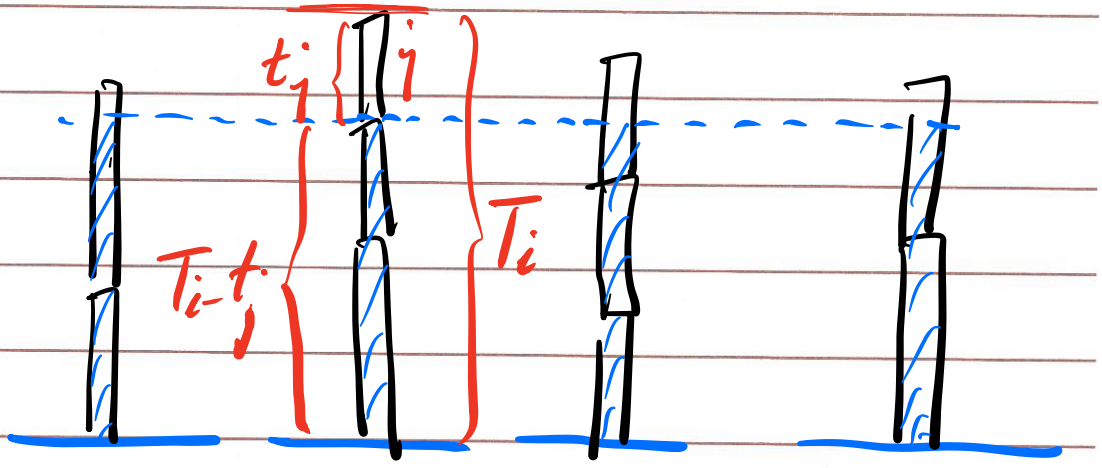
T_i : load on machine/resource i

T^* : value of the optimal solution



Greedy Balancing





$t_j \leq T^*$ (j: i)

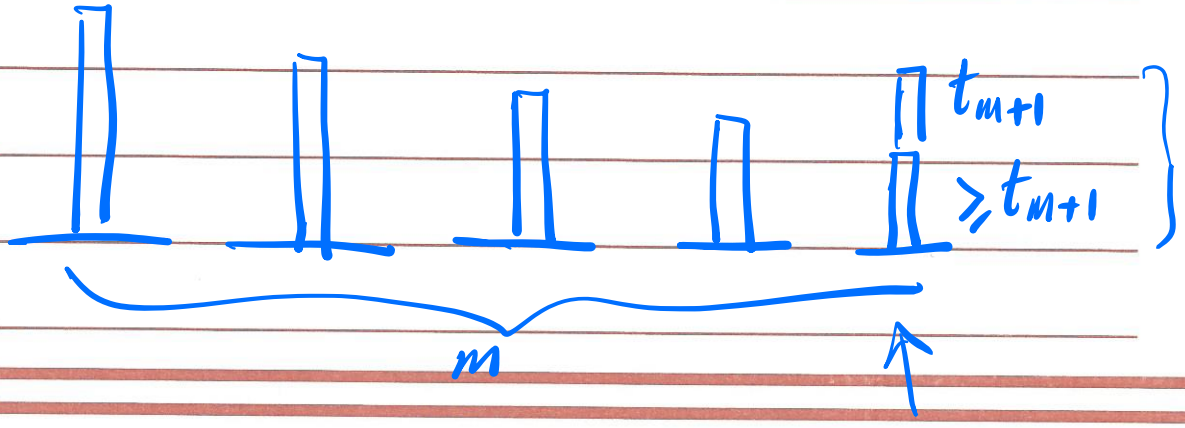
$T_i - t_j \leq T^*$

$T_i \leq 2 \cdot T^*$

a 2-approximation

Improved approximation to greedy balancing

Initially sort jobs in decreasing order of length then use same greedy balancing



$T^* \geq 2 * t_{m+1}$

$t_j \leq t_{m+1}$

$\rightarrow T^* \geq 2 t_j$ $\quad \frac{1}{2} T^* \geq t_j$

$T^* \geq T_i - t_j$

$1.5 T^* \geq T_i$

$\Rightarrow$ 1.5- approximation

Vertex Cover Problem
(Approximation Example)

Problem Statement: Find the smallest vertex cover in graph G.

A 2-approximation algorithm to the vertex Cover problem:

Start with S = Null

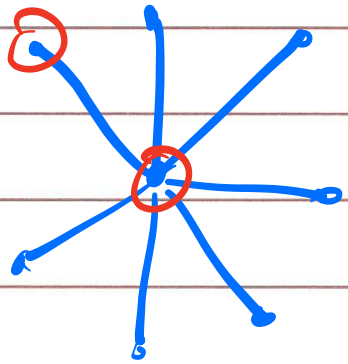
While S is not a vertex cover

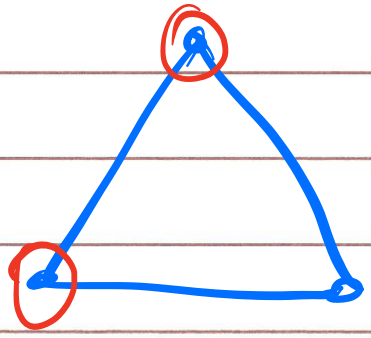
Select an edge e not covered by S

Add both ends of e to S

Endwhile

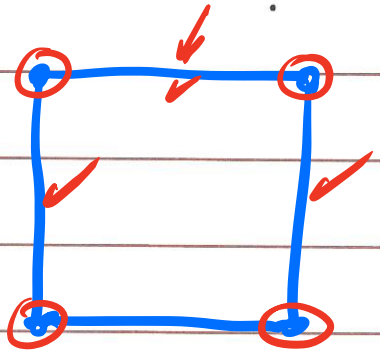
Why is this a 2-approximation?





Question: Since Independent Set $\leq_p$ Vertex Cover,

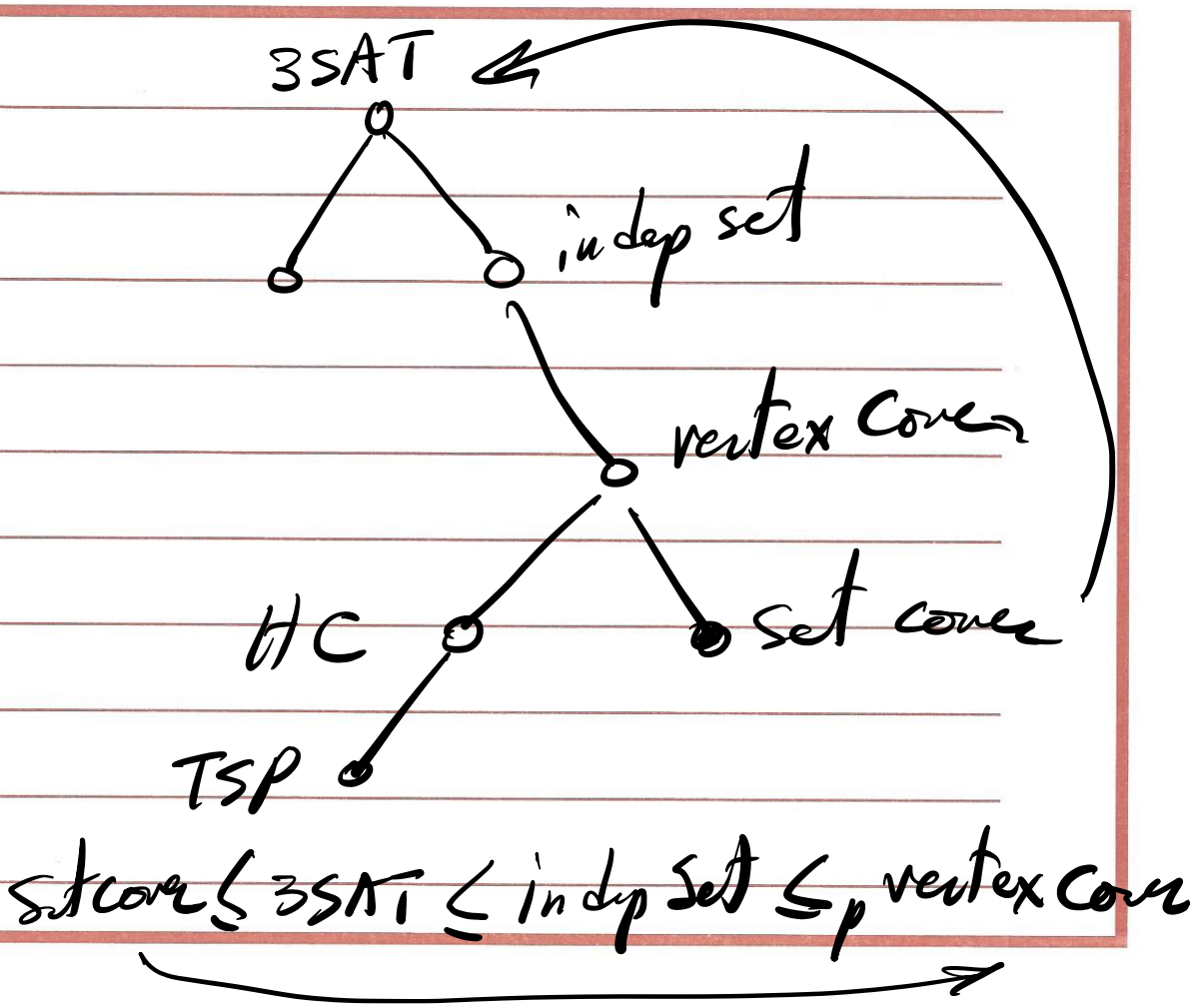
Can I use our 2-approximation algorithm for vertex cover to find a .5-approximation to independent set?



Theorem: Unless P=NP, there is no $\frac{1}{n^{1-\epsilon}}$ approximation for the Maximum Independent Set problem for any $\epsilon > 0$ where n is the no. of nodes in the graph.

Question: Since Vertex Cover $\leq_p$ Set Cover

Can I use a 2-approximation algorithm for set cover to find a 2-approximation to vertex cover?



MAX-3SAT

Given a set of clauses of length 3, find a truth assignment that satisfies the largest number of clauses.

A .5-approximation to the MAX-3SAT problem:

- Set everything to true
  If less than 50% of clauses evaluate to true, then
- Set everything to False

More sophisticated approximation methods can get to within a factor of 8/9 of the optimal sol.

System of Linear Equations

$$[A][x] = [B]$$

RHS vector

Linear Programming

$$[A][x] \leq [B]$$

Objective function: $[C^T][x]$

Goal: Maximize the objective function subject to the above constraints

Linear Programming Standard Form

- All constraints are of the form $\leq$
- All variables are non-negative
- Objective function is maximized.

Ex.: $x_1 - x_2 \geq 0$

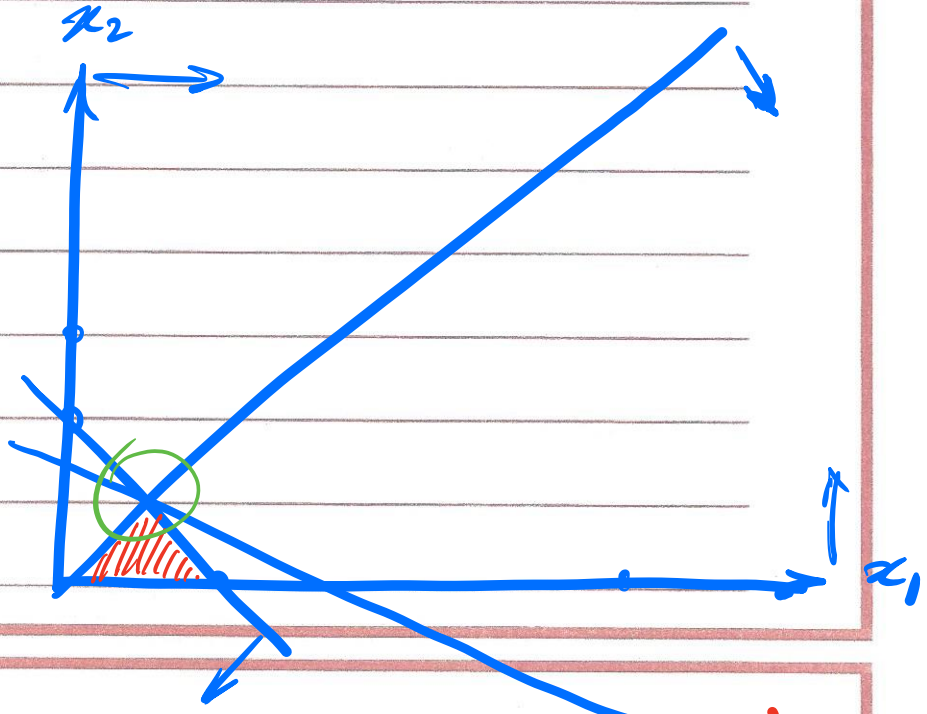
$x_1 \geq 0$

$x_2 \geq 0$

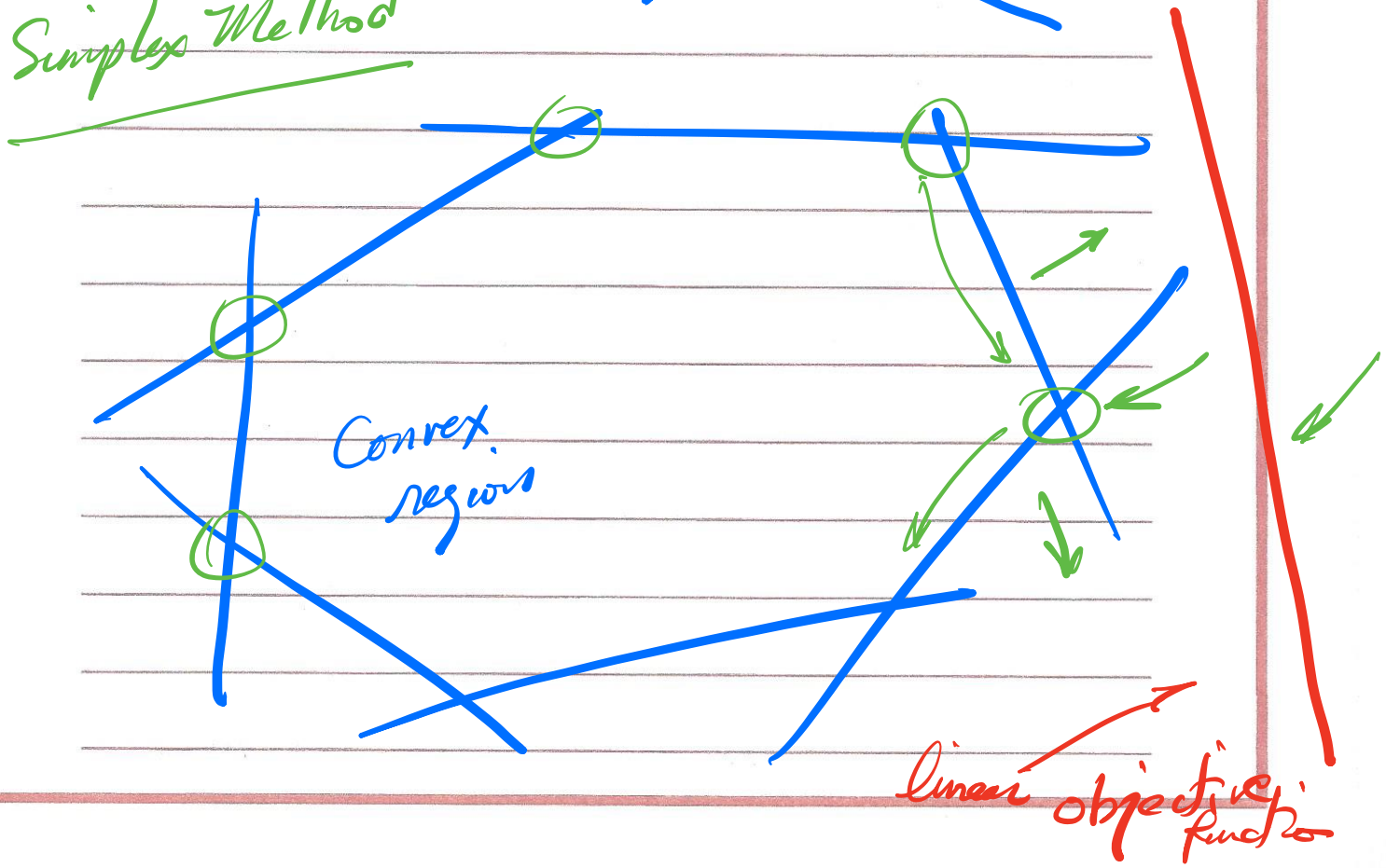
$x_1 + x_2 \leq 4$

Maximize

$x_1 + 2x_2$



Simplex Method



Weighted Vertex Cover Problem

For $G=(V,E)$, $S \subseteq V$ is a vertex cover set such that each edge has at least one end in S

Also, $w_i \geq 0$ for each $i \in V$

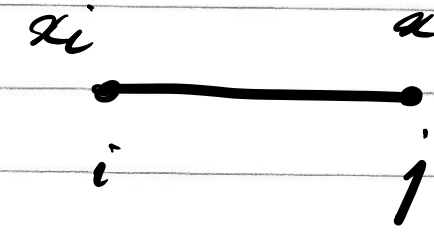
So, the total weight of the set $= W(S) = \sum_{i \in S} w_i$

Objective: Minimize $W(S)$

x_i is a decision variable for each $i \in V$

$$\begin{cases} x_i = 0 & i \notin S \\ x_i = 1 & i \in S \end{cases}$$

$x_i + x_j \geq 1$



Objective function:

(1)

$$\text{Minimize} \quad \sum w_i x_i$$

(2)

Subject to:

$$x_i + x_j \geq 1 \quad \text{for } (i,j) \in E$$

(3)

$$x_i \in \{0,1\}$$

Integer Linear Program

- Linear Programming
  Continuous Variables
- Integer Programming
  Discrete variables
- Mixed Integer Programming
  Both Cont. & Discrete
- Non-Linear Programming
  Non Linear Constraints or Objective Function

To find an approximate solution using LP, drop the requirement that $x_i \in \{0,1\}$ and solve the LP in polynomial time to find $\{x_i^*\}$ between 0 & 1.

$$w_{LP} = \sum_i w_i x_i^*$$

Assume S' is the optimal vertex cover set and $w(S')$ is the weight of the opt. solution

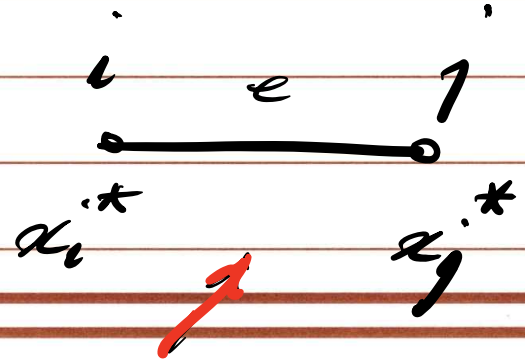
$$w_{LP} \leq w(S')$$

$x_i^* = 0 \implies i \notin S$

$x_i^* = 1 \implies i \in S$

Say $S_{\geq 1/2} = \{ i \in V : x_i^* \geq 1/2 \}$

$x_i^* + x_j^* \geq 1$



$w_{LP} \leq w(S')$

$w(S) \leq 2 * w_{LP}$

$w(S) \leq 2 * w(S')$

↳ This is a 2 approximation

Maxflow Problem

Variables are flows over edges

objective function: Maximize $\sum_{e \text{ out of } S} f(e)$

Subject to

$l_e \leq f(e) \leq c_e$ for each edge $e \in E$

$\sum_{e \text{ in to } v} f(e) - \sum_{e \text{ out of } v} f(e) = 0$ for $v \in V$ except for s & t

$A = B$

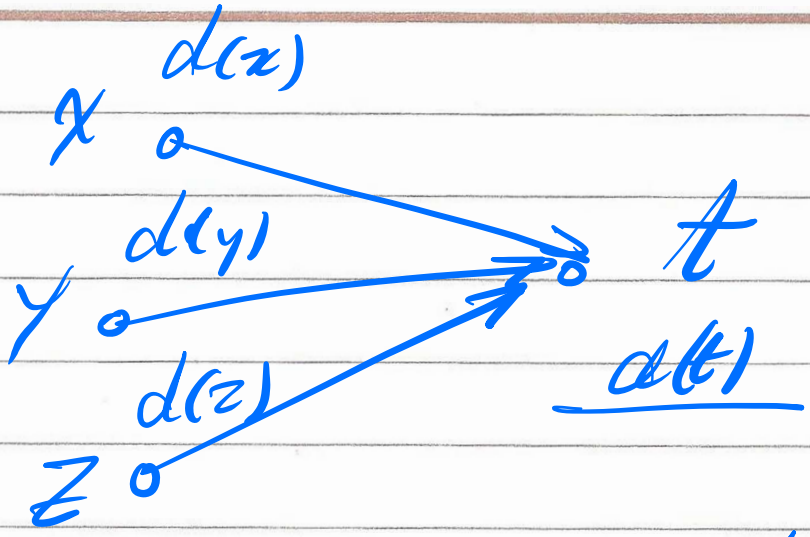
$A \leq B$

$B \leq A$

Shortest Path using LP

Problem Statement:
Find shortest path from s to t.

shortest distance from s to v is $d(v)$ for each $v \in V$.



$$\begin{cases} d(t) \leq d(x) + c_{xt} \\ d(t) \leq d(y) + c_{yt} \\ d(t) \leq d(z) + c_{zt} \end{cases}$$

$$\begin{cases} d(v) \leq d(u) + w(u,v) \quad \text{for each edge } (u,v) \in E \\ d(s) = 0 \end{cases}$$

Objective function:

~~Maximize~~ Minimize $d(t)$