Approximation Algorithms

An approximation algorithm to a problem is an algorithm that returns a nearly correct result.

Load Balancing Problem

Have m machines, a set of n jobs. Each job j has a processing time t_j.

Let $A(i)$ be the jobs assigned to M_i, then M_i needs to do

$$L_i = \sum_{j \in A_i} t_j$$

processing time.

The load balancing problem is to find the assignments $A(i)$ s.t.
makespan $\longrightarrow \max L_i$ is as small as possible.

Load Balancing Problem

The load-balancing problem is NP-complete.

Want to find an assignment that is "close" to the best one.

We don't know what the value is for the best assignment!

Try a Greedy Algorithm

Start with job 1, and iteratively assign each job to the machine with smallest load.

Example: $m = 3$, $n = 6$ $\{t_i\} = \overset{t_1}{2}, \overset{t_2}{3}, \overset{\cdots}{4}, 6, 2, 2$

8	5	6
=	=	=
6	2	2
2	3	4
M_1	M_2	M_3

7		6
=		=
3		4
2		
2	6	2
M_1	M_2	M_3

Lower Bounds for Optimal

Observation: The optimal makespan L^* is at least

$$\frac{1}{m}\sum_{i=1}^{n} t_i$$

If we could show that our greedy algorithm always returned an answer less than $2 \cdot \frac{1}{m}\sum t_i$, we'd be done

$$\left(\frac{1}{m}\sum t_i \leq L^* \leq G \leq 2\,\frac{1}{m}\sum t_i.\right)$$

Doesn't actually work

$m = 3,\ n = 6,\ \{t_i\} = \{1, 1, 1, 1, 1, 100\}$

		1
	1	1
100	1	1
M_1	M_2	M_3

$\frac{1}{m} \sum t_i = \frac{105}{3} = 35.$

The problem is that greedy (and even OPT) is more than $2 \frac{1}{m} \sum t_i$.

Observation 2: L^* is at least $\max_j t_j$.

The makespan L returned by greedy satisfies

$$L \le 2L^*.$$

Let M_i be the machine with highest load, and let j be the last job assigned to M_i.

Before j is assigned, M_i had load $L_i - t_j$.

Because of the greedy selection, we know $L_i - t_j \le L_k$ for

Summing over k: $m(L_i - t_j) \le \sum L_k = \sum_{j=1}^{n} t_j$

$$L_i - t_j \le \frac{1}{m} \sum t_j \le L^*$$

The makespan L returned by greedy satisfies
$$L \leq 2L^*.$$

Let M_i be the machine with highest load, and let j be the last job assigned to M_i.

Before j is assigned, M_i had load $L_i - t_j$.

Because of the greedy selection, we know $L_i - t_j \leq L_k$ for all k

Summing over k: $m(L_i - t_j) \leq \sum L_k = \sum_{j=1}^{n} t_j$

$$L_i - t_j \leq \frac{1}{m} \sum t_j \leq L^*$$

$$t_j \leq \max_k t_k \leq L^*$$

$$L_i = L_i - t_j + t_j \leq L^* + L^* = 2L^*$$

Center Selection Problem

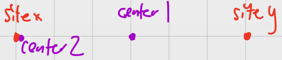
Set S of n sites in $\mathbb{R}^2$.

Want to output k centers C such that the maximum distance from a site to its nearest center is as small as possible.

Let $r(C)$ be the smallest radius such that the circles of radius $r(C)$ cover S. (covering radius).

Try a Greedy Algorithm

Pick centers to minimize the covering radius at each step.



covering radius of $|x-y|$

optimal is 0.

BAD approximation!

What if we know the optimal covering radius?

The exists a set C^* where every site is within $r(C^*)$ of some center.

Idea: Take a random site s. s is close to some center $c^* \in C^*$. Choose s as a center in C, and show that everything "close" to s must also be "close" to things close to c^*.