LECTURE 4

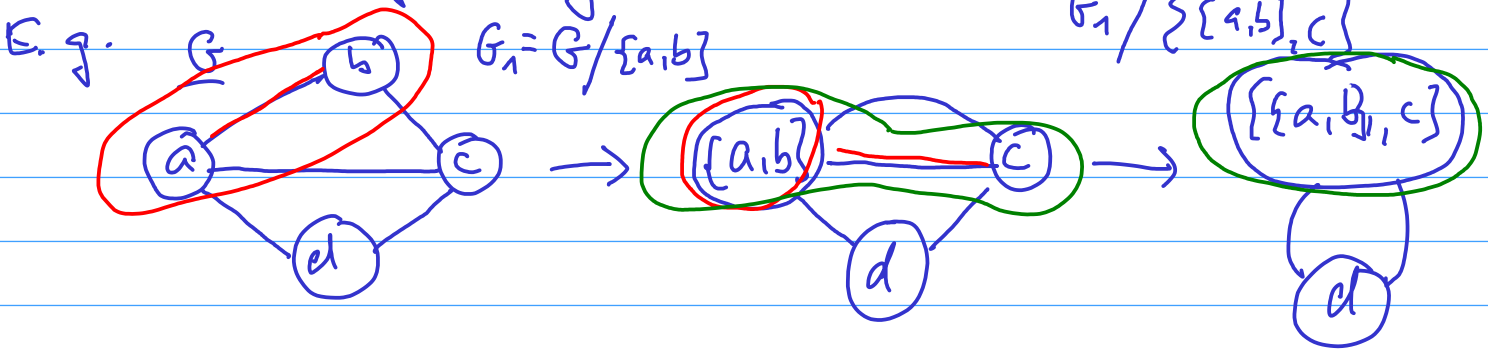
22/10/2020

GLOBAL MINIMUM CUT

HW?

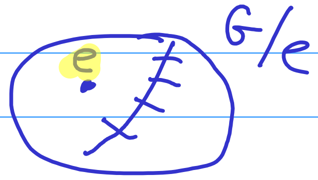
contraction of an edge

E.g.

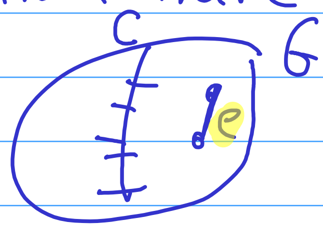


global cut $S=\{a,b,c\}$

- Every cut in G/e corresponds to a cut in G of the same size.
  $\Rightarrow$ the $\text{Mincut}(G) \le \text{Mincut}(G/e)$



- If C is a cut in G and $e \notin C$, then there is a cut of size $|C|$ in G/e.
  $\Rightarrow$ For a mincut C and $e \notin C$
  $$\boxed{\text{Mincut}(G/e) = \text{Mincut}(G)}$$



Fix a mincut C in G.

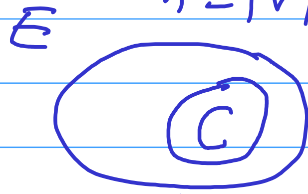
- If the algorithm never chooses $e \in C$, then it will output a cut of size C.

ALGORITHM RAND MINCUT(G)
while G has more than 2 vertices
choose uniformly at random $e \in G$
$G \leftarrow G/e$
output the cut corresponding to the 2 vertices

What is the probability that we get the right answer?

let $k = |C|$ where C is the fixed global mincut

Note: (1) $|E| \geq \frac{n \cdot k}{2}$ $\quad n = |V|$



$|C| = k \Rightarrow dg(v) \geq k$

(2) $Pr[e \in C \text{ chosen in the first iteration}] = \frac{k}{|E|} \leq \frac{2}{n}$

(3) there are $n-2$ iterations.
the number of vertices decreases by 1 in each iteration

let $n_i = n - i + 1$ ... # of vertices at the begining of iter. i

(4) Assuming no $e \in C$ contracted in the first $i-1$ iterations: $Pr[e \in C \text{ chosen in iter. } i] \leq \frac{k}{\frac{n_i \cdot k}{2}} = \frac{2}{n_i}$

edges ... $\geq \frac{n_i \cdot k}{2}$

let E_i be the event that no $e \in C$ contracted in iteration i

$$Pr[E_1] \geq 1 - \frac{2}{n} = \frac{n-2}{n}$$

$$Pr[E_i \mid E_1 \cap E_2 \cap \dots \cap E_{i-1}] \geq 1 - \frac{2}{n_i} = \frac{n_i - 2}{n_i} = \frac{n-i-1}{n-i+1}$$

Lemma: $Pr[\underbrace{E_1 \cap E_2 \cap \dots}_{A} \cap \underbrace{E_k}_{B}] = Pr[E_1] Pr[E_2 | E_1] Pr[E_3 | E_2 \cap E_1] \dots Pr[E_k | E_{k-1} \cap \dots E_2 \cap E_1]$

Proof: $Pr[A \cap B] = P[A] \cdot P[B|A]$

$= Pr[E_k | \underbrace{E_{k-1} \cap E_{k-2} \cap \dots E_2 \cap E_1}_{A}] Pr[\underbrace{E_1 \cap E_2 \cap \dots \cap E_{k-1}}_{A}]$ ∎

$$P[E_1 \cap E_2 \cap \dots \cap E_{n-2}] = Pr[E_1] Pr[E_2 | E_1] \dots Pr[E_{n-2} | E_{n-3} \cap \dots \cap E_2 \cap E_1] \geq$$

$$\geq \frac{\cancel{n-2}}{n} \cdot \frac{\cancel{n-3}}{n-1} \cdot \frac{\cancel{n-4}}{\cancel{n-2}} \cdots \frac{2}{\cancel{4}} \cdot \frac{1}{\cancel{3}} = \frac{2}{n(n-1)}$$

Claim: The algorithm returns a global mincut with probability at least $\frac{2}{n(n-1)}$.

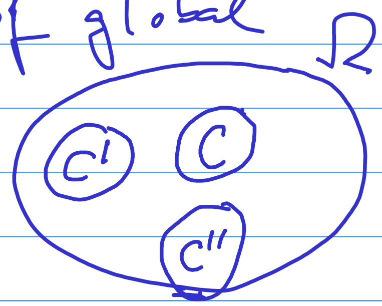
- what if we repeat the algorithm
- $\frac{n(n-1)}{2}$ times?

$\underline{\text{failure probability}} \le \left(1 - \frac{2}{n(n-1)}\right)^{\frac{n(n-1)}{2}} \le \frac{1}{e}$

- $\frac{n(n-1)}{2} \cdot \ln n$ times? $\dots \le \left(\frac{1}{e}\right)^{\ln n} = \frac{1}{n}$

Lemma: If we repeat the algorithm $\frac{n(n-1)}{2} \cdot \ln n$ times, we get a global mincut with probability $\ge 1 - \frac{1}{n}$.

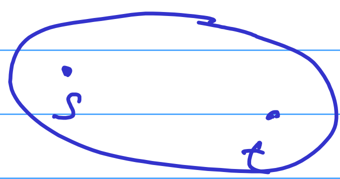
Question: what is the number of global min. cuts?



$$\Rightarrow \le \frac{1}{\frac{2}{n(n-1)}} = \frac{n(n-1)}{2}$$

Corollary: There are at most $\frac{n(n-1)}{2}$ global mincuts in G.

HW: # of min s-t cuts?

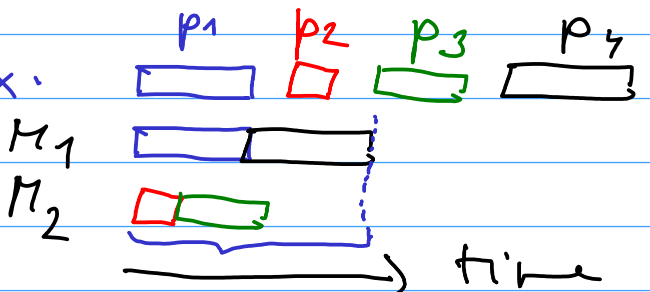


GREEDY ALGORITHMS

SCHEDULING JOBS ON IDENTICAL MACHINES & LOCAL SEARCH

PROBLEM: n jobs, p_i .. processing time of job i
m identical machines running in parallel
each job must be processed on one of the machines
each machine can process ≤ 1 job at a time
GOAL: complete all jobs as soon as possible

Ex.



$I_1 = \{1, 4\}$
$I_2 = \{2, 3\}$

i.e. find a partition of $\{1, 2, \dots, n\}$ into sets $I_1, I_2, \dots, I_m$ s.t.

$$\max_{i \in \{1,\dots,m\}} \sum_{j \in I_i} p_j \quad \text{is minimized.}$$

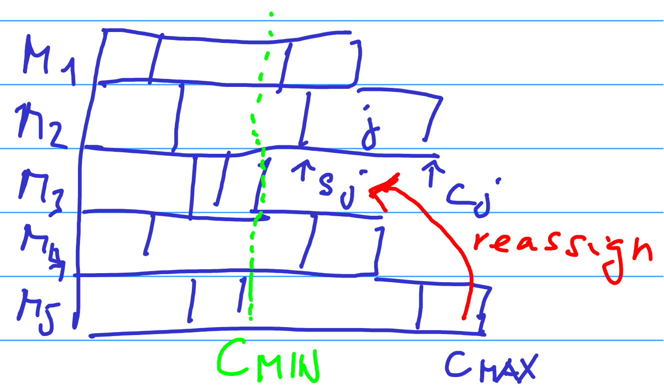
Notation: for a given schedule and job $j \in I_i$

start up time of j: $S_j = \sum_{k \in I_i,\, k < j} p_k$

completion time of j: $C_j = S_j + p_j$

$C_{MAX} = \max_j C_j$

$C_{MIN} = \min_i \max_{j \in I_i} C_j$



Question: given a schedule how can you improve it easily.

ALGORITHM LOCAL SEARCH FOR SCHEDULING

1. Start with any schedule
2. if there is a job j s.t. $\boxed{\begin{array}{l} C_j = C_{MAX} \\ \text{and } C_{MIN} < S_j. \end{array}}$
   reassign it on a machine with minimum completion time.
   else OUTPUT the current schedule. STOP
3. repeat STEP 2.

THM: the approximation ratio of the algorithm is 2 (resp. $2-\frac{1}{m}$, for fixed m).

Proof: • poly time ✓
• Quality

i) C_{MIN} never decreases.

ii) Every job is reassigned at most once.

$\Rightarrow$ poly time.