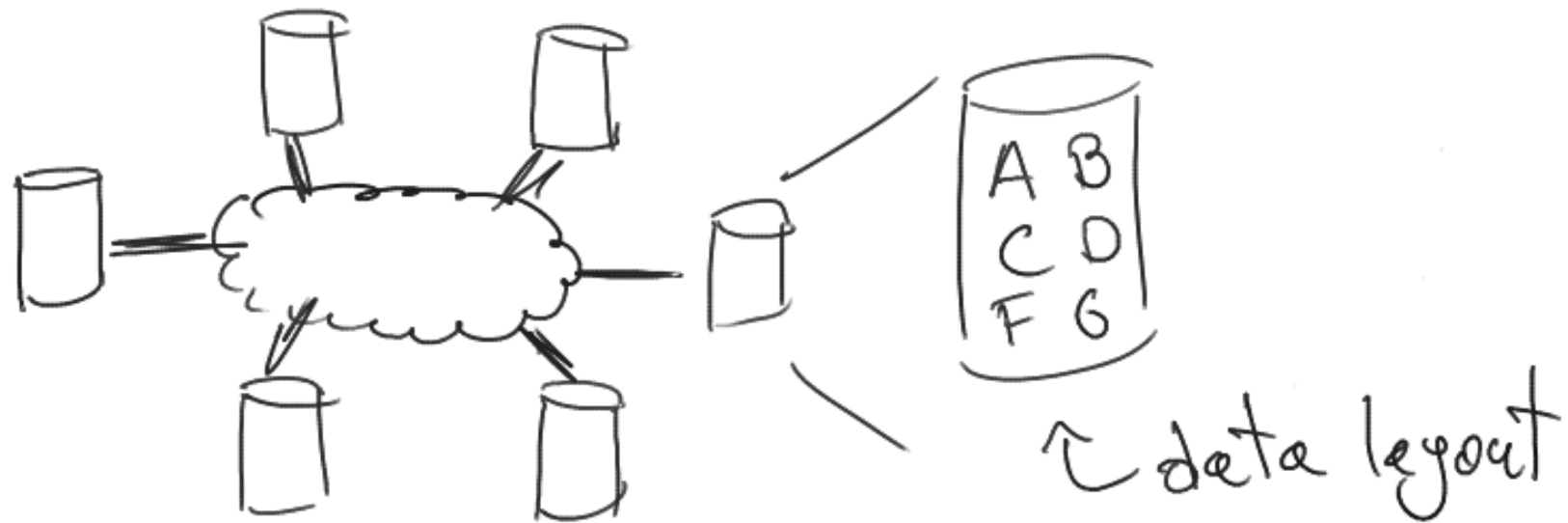


Combinatorial Algorithms for Data Migration

Rajiv Gandhi & Julian Mestre

The Data Migration Problem



- # of requests per item
- disk load

may change over time

More formally...

Given $G = (V, E) \begin{cases} \rightarrow V : \text{disks} \\ \searrow E : \text{transfers} \end{cases}$

We want to schedule E

\equiv break into matchings M_1, \dots, M_k

Minimize

→ $\max_e \phi(e)$ [EDGE COLORING]

→ $\sum_e \phi(e)$ [SUM EDGE/VERTEX]

→ $\sum_v \max_{e \in \delta(v)} \phi(e)$ [COMP. TIME]

where $\phi: E \rightarrow \mathbb{N}$ • $e \in M_{\phi(e)}$

Previous Results

Edge coloring: $\min \max_e \phi(e)$

- NP-hard
- can get $\Delta+1$ (gen. graphs) [V 64]
- can get Δ (for bipartite graphs)

Previous Results

$$\min \sum_e \phi(e)$$

- APX-hard in bipartite graphs [M]
- 2-approx [BHKSS 00]
- 1.79-approx (for bipartite) [HKS 03]

↳ Our result: $\sqrt{2}$ -approx

Previous Results

$$\min_v \sum_{e \in \delta(v)} \max \phi(e)$$

- NP-hard (for general graphs)
- 3-approx (LP rounding) [K03]

↳ Our result: Primal-Dual
3-approx

Algorithm for $\min \sum \phi(e)$

Find matchings M_1, \dots, M_Δ s.t.

$\bigcup_{i \leq b} M_i$ is a maximal b -matching

Thm: this is a $\sqrt{2}$ -approx

How do we find M_1, \dots, M_Δ ?

1) Find matching M_Δ hitting all degree Δ vertices

2) Remove M_Δ , repeat for $M_{\Delta-1} \dots M_1$

we can always do ①

for bipartite graphs, but... 

Algorithm for $\min \sum \phi(e)$

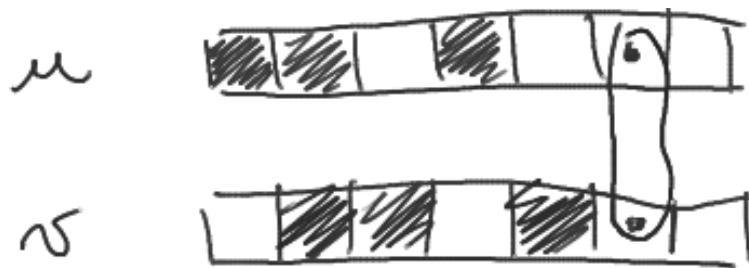
Find matchings M_1, \dots, M_Δ s.t.

$\bigcup_{i \leq b} M_i$ is a maximal b -matching

Thm: this is a $\sqrt{2}$ -approx

Warm-up

Thm: any greedy schedule
is a 2-approx [BHKSS 00]



scheduling (u, v)
at $\phi(u, v) = 6$

Lowerbound



$$S \subseteq \delta(u)$$

Any solution must spend on S

$$\geq 1 + 2 + \dots + |S| = \frac{|S|(|S|+1)}{2}$$

Warm-up

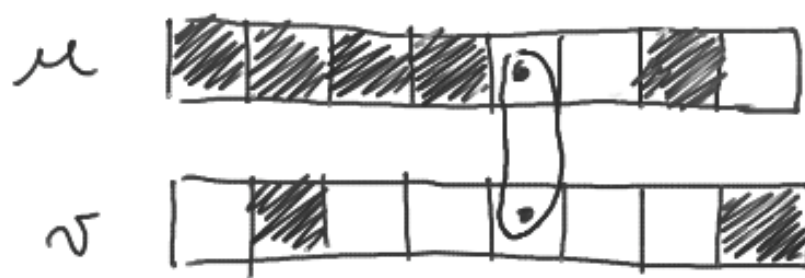


$$LB(u) = 1 + 2 + \dots + d(u)$$

$$\text{Cost greedy} \leq \sum_u LB(u) \leq 2 \text{ OPT}$$

Analysis

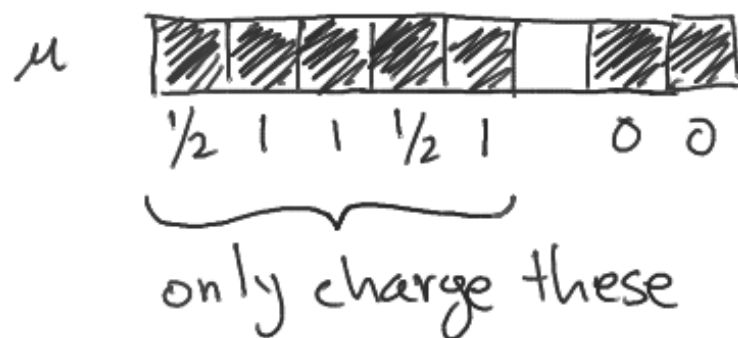
$\forall (u, v)$ • either u or v is full



← blame the full endpoint

If both endpoints are full
 \Rightarrow they share the cost

Analysis



Let α be s -full $\rightarrow \alpha s : \text{pay } 1$
 $\rightarrow (1-\alpha)s : \text{pay } \frac{1}{2}$

we need $\leq \frac{1}{2} \sum_{i=1}^s i + \frac{1}{2} \sum_{i=(1-\alpha)s+1}^s i$

BUDGET = $\rho \sum_{i=1}^s i + \rho \sum_{i=1}^{\alpha s} i \Rightarrow \sqrt{2}$ -approx

A few comments

- Almost tight : 1.375 example
- Downside : only unweighted
- Integrality gap of $\frac{10}{9}$

Integrality Gap?

$$\max \sum_S \frac{|S|(|S|+1)}{2} y_S$$

$$\sum_{e \in S} y_S \leq 1 \quad \forall e$$

$$y_S \geq 0 \quad \forall S \subseteq \delta(u)$$



Algorithm for $\min_u \sum_{e \in \delta(u)} \max \phi(e)$

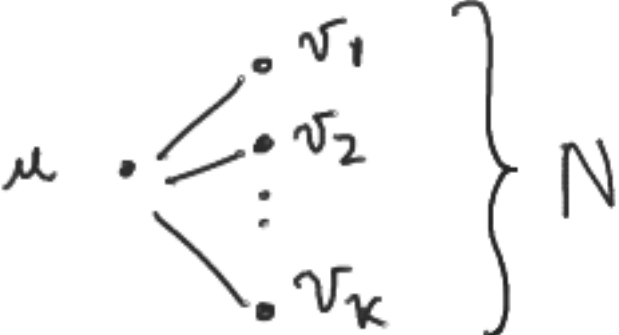
I) Every vertex is unlabeled

[Find u with most unlabeled neighbors N
for all $v \in N$ set $l(v) = |N|$

II) Sort edges: $\min(l(u), l(v)), \max(l(u), l(v))$

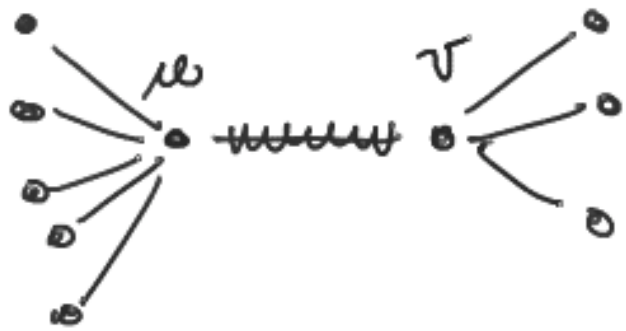
III) Greedy Schedule

Lowerbound


$$\sum_i \text{COMP TIME } v_i \geq \frac{|N|(|N|+1)}{2} \geq \frac{1}{2} \sum l(v_i)$$

Lemma: $\sum_v l(v) \leq 2 \text{OPT}$

Analysis



How many edges were scheduled before (u, v) ?

Lemma: $\forall u$ finishes by $\ell(u) + \text{deg}(u)$

Analysis

Lemma: $\sum_v l(v) \leq 2 \text{OPT}$

Lemma: $\forall u$ finishes by $l(u) + \text{deg}(u)$

Thm: Alg is a 3-approx

A few comments

- Greedy Schedule
- Dual update takes $O(m)$ time
- Can handle weights
- Integrality Gap $4/3$
- Extends to Arb. proc times

Future Work

- For min sum edge coloring Greedy is the best for general graphs
- Almost b -maximal
- Bipartite edge coloring in $O(m \log n)$
- PD for scheduling problems

Thanks for
your attention !

Combinatorial Algorithms
for the
Data Migration Problem

Julian Mestre
Rajiv Gandhi

Other Generalizations

→ arbitrary proc. times

→ weights

→ sum vertex coloring