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 $G = (V, E)$
- $V$: disks
- $E$: transfers

We want to schedule $E$

$\equiv$ break into matchings $M_1, \ldots, M_k$
Minimize

\[ \rightarrow \max_e \phi(e) \quad \text{[EDGE COLORING]} \]

\[ \rightarrow \sum_e \phi(e) \quad \text{[SUM EDGE/VERTEX]} \]

\[ \rightarrow \sum_v \max_e \phi(e) \quad \text{[COMP. TIME]} \]

where \( \phi: E \rightarrow \mathbb{N} \) for each edge e.
Previous Results

Edge coloring: \[ \min \max \varphi(e) \]

- \textit{NP-hard}
- can get \( \Delta + 1 \) (gen. graphs) [V64]
- can get \( \Delta \) (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]

\[ \Rightarrow \text{Our result} \doteq \sqrt{2} \text{-approx} \]
Previous Results

\[
\min \sum_{v} \max_{e \in S(v)} \phi(e)
\]

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

Our result: Primal-Dual 3-approx
Algorithm for \( \min \sum \phi(e) \)

Find matchings \( M_1, \ldots, M_a \) s.t.
\[ \bigcup_{i \leq b} M_i \text{ is a maximal } b\text{-matching} \]

Thm: this is a \( \sqrt{2} \)-approx
How do we find $M_1, \ldots, M_\Delta$?

1) Find matching $M_\Delta$ hitting all degree $\Delta$ vertices

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

we can always do 1

for bipartite graphs, but...
Algorithm for $\min \sum \psi(e)$

Find matchings $M_1, \ldots, M_\Delta$ s.t.

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

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

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

\[ \mu = \begin{array}{cccc}
1 & \frac{1}{2} & 1 & \frac{1}{2} \\
\end{array} \]

\[ \nu = \begin{array}{cccc}
\frac{1}{2} & 1 & 1 & \frac{1}{2} \\
\end{array} \]

Scheduling \((\mu, \nu)\) at \(\phi((\mu, \nu)) = 6\)
Warm-up

\[ \mu = \overline{1 2 3 4 5} \]

\[ \text{LB}(\mu) = 1 + 2 + \ldots + d(\mu) \]

Cost greedily \( \leq \sum_{\mu} \text{LB}(\mu) \leq 2 \text{ OPT} \)
Lower bound

\[ \mu \leq S \subseteq S(\mu) \]

Any solution must spend on \( S \)

\[ \geq 1 + 2 + \cdots + |S| = \frac{|S|(|S|+1)}{2} \]
Analysis

\[ \forall (u,v) \cdot \text{either } u \text{ or } v \text{ is full} \]

If both endpoints are full

\[ \Rightarrow \text{they share the cost} \]
Analysis

\[ \mu \]
\[ \frac{1}{2} 1 1 \frac{1}{2} 1 0 0 \]
\[ \text{only change these} \]

Let \( \mu \) be s-fall \( \Rightarrow (1-a)s = \text{pay } \frac{1}{2} \).

we need \[ \leq \frac{1}{2} \sum_{i=1}^{s} i + \frac{1}{2} \sum_{i=(1-a)s+1}^{s} i \]

\[ \text{BUDGET} = p \sum_{i=1}^{s} i + p \sum_{i=1}^{\frac{\Delta S}{s}} i \Rightarrow \sqrt{2} - \text{approx} \]
A few comments

- Almost tight: 1.375 example
- Downside: only unweighted
- Integrality gap of $10\%$
Integrality gap?

\[
\max \sum_{S} \frac{|S|(|S|+1)}{2} y_S
\]

\[
\sum_{S \in \mathcal{S}(u)} y_S \leq 1 \quad \forall e
\]

\[
y_S \geq 0 \quad \forall S \in \mathcal{S}(u)
\]
Other Generalizations

- arbitrary proc. times
- weights
- sum vertex coloring
Algorithm for \( \min \sum_{u} \max_{e \in S(u)} \phi(e) \)

1) Every vertex is unlabeled
   
   \[
   \text{Find } u \text{ with most unlabeled neighbors } N
   \]
   
   for all \( v \in N \) set \( l(v) = |N| \)

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

3) Greedy Schedule
Lower bound

\[ \sum_{i} \text{COMP TIME } v_i \geq \frac{1}{2} \frac{|N| (|N| + 1)}{2} \geq \frac{1}{2} \sum \lambda(v_i) \]

Lemma: \[ \sum_{v} \lambda(v) \leq 2 \text{OPT} \]
Analysis

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

Lemma: The finishes by \(l(u) + \text{deg}(u)\)
Analysis

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

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

Thm: Aly is a 3-approx
A few comments

→ Greedy Schedule
→ Dual update takes $O(m)$ time
→ Can handle weights
→ Integrality Gap $\frac{4}{3}$
→ Extends to Arb. proc times
Thanks for your attention!
Combinatorial Algorithms for the Data Migration Problem

Julian Mestre
Rajiv Gandhi
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