Tight Approximation Algorithms for Ordered Covering

Jatin Batra, Syamantak Das and **A. Jha**

TIFR Mumbai, IIIT-Delhi and EPFL

2023

Minimum Vertex Cover

Definition

Given a graph $G = (V, E)$, find the smallest cardinality subset $S \subseteq V$ such that for all edges e in E, $e \cap S \neq \emptyset$.

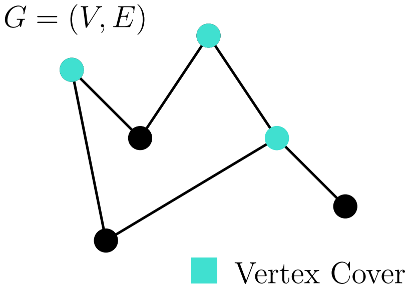


Figure: Vertex Cover in G

Alternate View (Scheduling)

1. One Machine.
2. The vertices are the jobs.

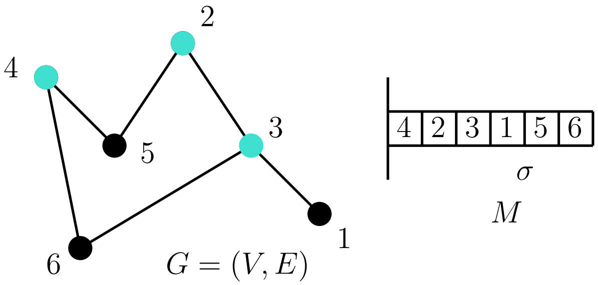


Figure: Scheduling Vertices on a Machine

Objective Function?

Alternate View (Scheduling): Cover Times

Definition

Given a graph $G = (V, E)$ and a schedule $\sigma : V \to [n]$ of its vertices, the cover time of an edge is the first moment a vertex in the edge is scheduled.

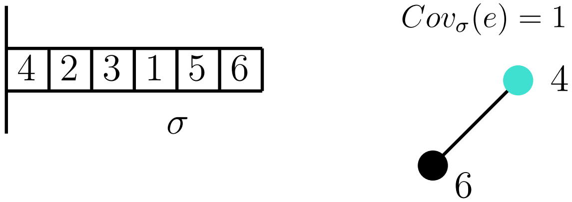


Figure: Cover Time of an Edge

1. Vertex Cover: $\min_{\sigma} \max_{e \in E} Cov_{\sigma}(e)$.
2. Min-Sum Vertex Cover: $\min_{\sigma} \sum_{e \in E} Cov_{\sigma}(e)$ [FLT04, BBFT21, Sta22]

Can we unify?

Ordered Optimization [AS, CSa, BSS18, CSb, CS19]

Given a schedule σ, arrange e in descending order as:
$Cov_\sigma(e_1) \geq Cov_\sigma(e_2) \geq \ldots Cov_\sigma(e_m)$.

$\min_\sigma Cov_\sigma(e_1)$ $\qquad\qquad$ $\min_\sigma \sum_{j=1}^{m} Cov_\sigma(e_j)$

?

Figure: Generalized Objective Functions

Ordered Optimization [AS, CSa, BSS18, CSb, CS19]

Given a schedule σ, arrange e in descending order as: $Cov_\sigma(e_1) \geq Cov_\sigma(e_2) \geq \ldots Cov_\sigma(e_m)$.

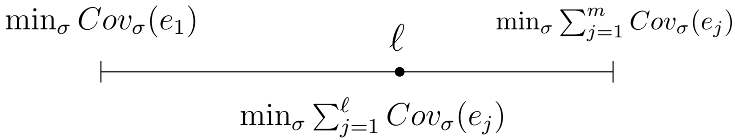


Figure: Top-ℓ Objective Functions

Ordered Objective Functions:

$$\min_\sigma \sum_{j=1}^{m} w_j Cov_\sigma(e_j)$$

for $w_1 \geq w_2 \geq \ldots w_m \geq 0$. This framework has gained a lot of attention for clustering (Ordered k-Median) and Load Balancing problems.

Our Contribution

Figure: One Ring to Rule Them All

Current State

Current State

- Vertex Cover

Current State

- Vertex Cover
  - Upper Bound: 2-approx (Folklore)
  - Hardness of Approximation: $(2 - \varepsilon)$ under UGC [KR08].

Current State

- Vertex Cover
  - Upper Bound: 2-approx (Folklore)
  - Hardness of Approximation: $(2 - \varepsilon)$ under UGC [KR08].
- Min-Sum Vertex Cover

Current State

- Vertex Cover
  - Upper Bound: 2-approx (Folklore)
  - Hardness of Approximation: $(2 - \varepsilon)$ under UGC [KR08].
- Min-Sum Vertex Cover
  - Upper Bound: 16/9-approx [BBFT21].
  - Hardness of Approximation: 1.014 under UGC [Sta22].
- Ordered Min-Sum Vertex Cover

Current State

- Vertex Cover
  - Upper Bound: 2-approx (Folklore)
  - Hardness of Approximation: $(2 - \varepsilon)$ under UGC [KR08].
- Min-Sum Vertex Cover
  - Upper Bound: 16/9-approx [BBFT21].
  - Hardness of Approximation: 1.014 under UGC [Sta22].
- Ordered Min-Sum Vertex Cover
  - Upper Bound: 8-approx [GGKT08] (All-norm, more general result).
  - Lower Bound: $(2 - \varepsilon)$ under UGC (Captures VC).

However, the techniques for upper bounds are very different across the problems.

Current State

Current State

- Set Cover
  - Upper Bound: $O(\log n)$-approx [Joh73, Sla96].
  - Lower Bound: NP-Hard to approximate within $(1 - \varepsilon) \log_e n$ [DS14]

Current State

- Set Cover
  - Upper Bound: $O(\log n)$-approx [Joh73, Sla96].
  - Lower Bound: NP-Hard to approximate within $(1-\varepsilon)\log_e n$ [DS14]
- Min-Sum Vertex Cover
  - Upper Bound: 4-approx [FLT04].
  - Lower Bound: NP-Hard to approximate within $(4-\varepsilon)$ [FLT04].

Current State

- Set Cover
  - Upper Bound: $O(\log n)$-approx [Joh73, Sla96].
  - Lower Bound: NP-Hard to approximate within $(1 - \varepsilon) \log_e n$ [DS14]
- Min-Sum Vertex Cover
  - Upper Bound: 4-approx [FLT04].
  - Lower Bound: NP-Hard to approximate within $(4 - \varepsilon)$ [FLT04].
- Ordered Set Cover
  - $O(\log n)$-approx. [GGKT08] Upper Bound (Much more general result, All-norm) s
  - General Lower Bound: NP-Hard to approximate within $\Omega(\log n)$ (Captures Set Cover). But for specialised Top-ℓ objectives?

Our Results

Definition

Ordered Min-Sum Vertex Cover(OMSVC): Let $G = (V, E)$ be a graph, and a $w_1 \geq w_2 \geq \ldots w_m \geq 0$ be a sequence of non-negative weights. Design a schedule σ of vertices to minimize $\sum_j w_j Cov_\sigma(e_j)$ where e_j is the ordering of edges in descending order of cover times.

Our Results

Definition

Ordered Min-Sum Vertex Cover(OMSVC): Let $G = (V, E)$ be a graph, and a $w_1 \geq w_2 \geq \ldots w_m \geq 0$ be a sequence of non-negative weights. Design a schedule σ of vertices to minimize $\sum_j w_j Cov_\sigma(e_j)$ where e_j is the ordering of edges in descending order of cover times.

Theorem

There is a $(2+\varepsilon)$-approximation algorithm for OMSVC running in time $n^{O(1/\varepsilon)}$.

Our Results

Definition

Top-ℓ Set Cover: Let E be a set and $\mathcal{I}$ be a collection of subsets of E. Design a schedule σ of sets to minimize $\sum_{j=1}^{\ell} Cov_{\sigma}(e_j)$ for a fixed $\ell \in [n]$ where the elements e_j are ordered in descending order of cover times.

Our Results

Definition

Top-ℓ Set Cover: Let E be a set and $\mathcal{I}$ be a collection of subsets of E. Design a schedule σ of sets to minimize $\sum_{j=1}^{\ell} Cov_{\sigma}(e_j)$ for a fixed $\ell \in [n]$ where the elements e_j are ordered in descending order of cover times.

Theorem

The natural greedy algorithm gives a $(8\log_2(n/\ell) + 16)$*-approx for Top-*ℓ *Set Cover.*

Our Results

Definition

Top-ℓ Set Cover: Let E be a set and $\mathcal{I}$ be a collection of subsets of E. Design a schedule σ of sets to minimize $\sum_{j=1}^{\ell} Cov_{\sigma}(e_j)$ for a fixed $\ell \in [n]$ where the elements e_j are ordered in descending order of cover times.

Theorem

The natural greedy algorithm gives a $(8\log_2(n/\ell) + 16)$*-approx for Top-*ℓ *Set Cover.*

Theorem

It is hard to approximate the Top-ℓ *Set Cover problem within a factor better than* $\max(\Omega(1), \Omega(\log(n/\ell)))$.

Our Results

Definition

Top-ℓ Set Cover: Let E be a set and $\mathcal{I}$ be a collection of subsets of E. Design a schedule σ of sets to minimize $\sum_{j=1}^{\ell} Cov_{\sigma}(e_j)$ for a fixed $\ell \in [n]$ where the elements e_j are ordered in descending order of cover times.

Theorem

The natural greedy algorithm gives a $(8\log_2(n/\ell) + 16)$*-approx for Top-*ℓ *Set Cover.*

Theorem

It is hard to approximate the Top-ℓ *Set Cover problem within a factor better than* $\max(\Omega(1), \Omega(\log(n/\ell)))$.

First ℓ-based approximation ratio and hardness result for Top-ℓ objectives (to our knowledge).

Overview of the Talk

We will talk about our algorithm which gets a $(2+\varepsilon)$-approx. for Top-ℓ Vertex Cover.

Overview of the Talk

We will talk about our algorithm which gets a $(2+\varepsilon)$-approx. for Top-ℓ Vertex Cover.

- We will motivate reducing Top-ℓ Vertex Cover to Discounted Min-Sum Vertex Cover (Standard in Ordered Optimization).

Overview of the Talk

We will talk about our algorithm which gets a $(2+\varepsilon)$-approx. for Top-ℓ Vertex Cover.

- We will motivate reducing Top-ℓ Vertex Cover to Discounted Min-Sum Vertex Cover (Standard in Ordered Optimization).
- We will give a review of the approach of Feige et. al. [FLT04] and why it fails in our problem.

Overview of the Talk

We will talk about our algorithm which gets a $(2+\varepsilon)$-approx. for Top-ℓ Vertex Cover.

- We will motivate reducing Top-ℓ Vertex Cover to Discounted Min-Sum Vertex Cover (Standard in Ordered Optimization).
- We will give a review of the approach of Feige et. al. [FLT04] and why it fails in our problem.
- We will show how a *dependent-rounding*[GKPS06] based approach fixes these problems.

Reduction

Let $\sigma : V \to T$ be a schedule for an instance of Top-ℓ Vertex Cover.

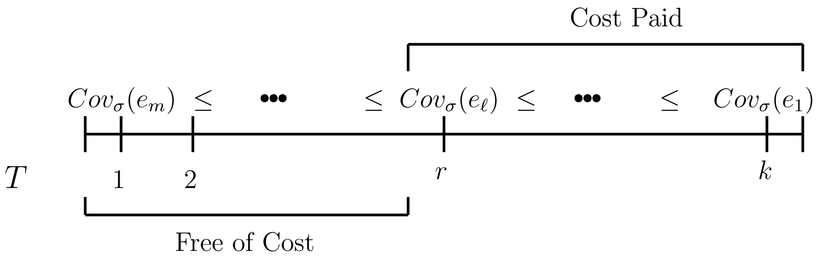


Figure: Top-ℓ Vertex Cover

Reduction

Reduction

Intuition

Reduction

Intuition

- Try to write down an ILP for this problem.

Reduction

Intuition

- Try to write down an ILP for this problem.
- Any schedule should be encoded as a feasible solution to this ILP.

Reduction

Intuition

- Try to write down an ILP for this problem.
- Any schedule should be encoded as a feasible solution to this ILP.
- ILP should know edges which have the Top-ℓ cover times for any schedule σ.

Reduction

Intuition

- Try to write down an ILP for this problem.
- Any schedule should be encoded as a feasible solution to this ILP.
- ILP should know edges which have the Top-ℓ cover times for any schedule σ.
- How to write constraints for this?

Reduction

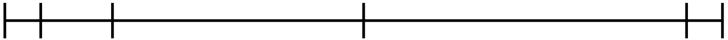
Problem? Do not know the ordering before hand.

Reduction

Shift by a Discount

Add the Discount back later

$(Cov_\sigma(e_m) - \lambda)_+ \leq \quad \bullet\bullet\bullet \quad \leq (Cov_\sigma(e_\ell) - \lambda)_+ \leq \quad \bullet\bullet\bullet \quad \leq (Cov_\sigma(e_1) - \lambda)_+$



$$\forall \lambda \geq 0 \quad \underbrace{\sum_{e \in E}(Cov_\sigma(e) - \lambda)_+}_{\text{Oblivious to Ordering}} + \underbrace{\lambda \cdot \ell}_{\text{Fixed}} \geq \sum_{i=1}^{\ell} Cov_\sigma(e_\ell)$$

Figure: Top-ℓ Vertex Cover

Reduction

Shift by Good Discount

Add the Discount back later

$$\lambda = Cov_{\sigma^*}(e_\ell)$$

$$(Cov_{\sigma^*}(e_m) - \lambda)_+ \leq \cdots \leq (Cov_{\sigma^*}(e_\ell) - \lambda)_+ \leq \cdots \leq (Cov_{\sigma^*}(e_1) - \lambda)_+$$

$$\sum_{e \in E} (Cov_{\sigma^*}(e) - \lambda)_+ + \lambda \cdot \ell = \sum_{i=1}^{\ell} Cov_{\sigma^*}(e_\ell)$$

Oblivious to Ordering — Fixed

Figure: Top-ℓ Vertex Cover

Reduction

Try to minimize:

$$\sum_{e \in E} (Cov_{\sigma}(e) - \lambda)_+$$

over all possible schedules for a correct guess $\lambda = Cov_{\sigma^*}(e_\ell)$.

Reduction

Theorem

[AS, BSS18, CSa, CS19, CSb] Let σ^* be the optimal schedule to an instance of Top-ℓ Vertex Cover. Then, a schedule σ satisfying:

$$\sum_{e \in E} (Cov_\sigma - 2\lambda)_+ \leq 2 \sum_{e \in E} (Cov_{\sigma^*} - \lambda)_+$$

implies that $\sum_{j=1}^{\ell} Cov_\sigma(e_j) \leq 2 \sum_{j=1}^{\ell} Cov_{\sigma^*}(e_j)$.

Definition

Discounted Min-Sum Vertex Cover (DMSVC) Let $G = (V, E)$ be a graph, and a $\lambda \geq 0$ be a discount. Design a schedule σ of vertices which satisfies:

$$\sum_{e \in E} (Cov_\sigma - 2\lambda)_+ \leq 2 \sum_{e \in E} (Cov_{\sigma^*} - \lambda)_+$$

Plan for the Talk

Plan for the Talk

1. In the following, we will try to solve the Discounted Min-Sum Vertex Cover problem via an LP-rounding algorithm.

Plan for the Talk

1. In the following, we will try to solve the Discounted Min-Sum Vertex Cover problem via an LP-rounding algorithm.
2. Let us first try to write down the linear program.

Intuition for Linear Program

Variables: $x_{v,t}$, $u_{e,t}$.
Indicator variables.

Intuition for Linear Program

Variables: $x_{v,t}$, $u_{e,t}$.
Indicator variables.

$$x_{v,t} = \begin{cases} 1 & v \text{ is scheduled at } t \\ 0 & \text{otherwise} \end{cases}$$

Intuition for Linear Program

Variables: $x_{v,t}$, $u_{e,t}$.
Indicator variables.

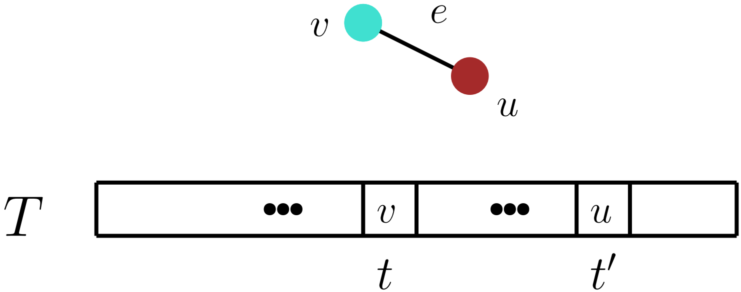
$$x_{v,t} = \begin{cases} 1 & v \text{ is scheduled at } t \\ 0 & \text{otherwise} \end{cases}$$

$$u_{e,t} = \begin{cases} 1 & Cov_{\sigma}(e) \geq t \\ 0 & \text{otherwise} \end{cases}.$$

$u_{e,t}$ tracks at each step if e has not been covered.

Intuition For Linear Program

Variables: $u_{e,t}, x_{v,t}$



$$\textcircled{1} \; \Sigma_{v' \in V} x_{v',t} \leq 1$$

Figure: Intuition for variables and constraints

Intuition For Linear Program

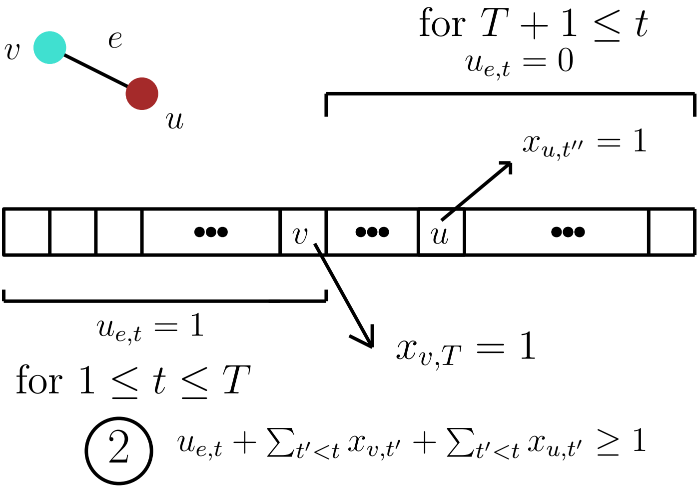


Figure: Intuition for variables and constraints, $Cov_\sigma(e) = \sum_t u_{e,t}$

Intuition For Linear Program

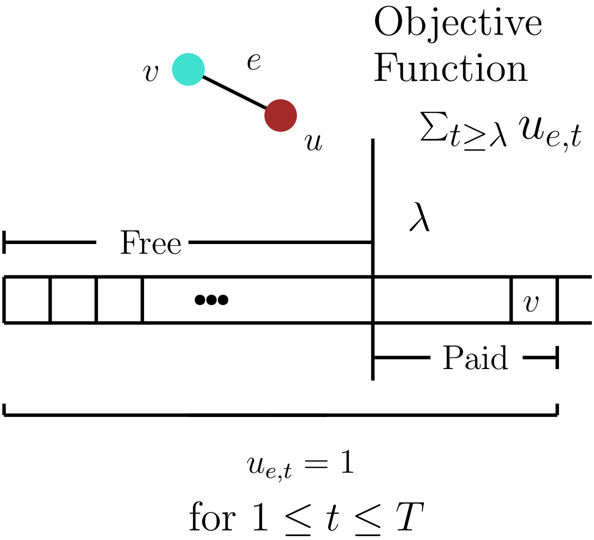


Figure: Intuition for variables and constraints

DMSVC Linear Program

$$\text{Minimize} \quad \sum_{t \geq \lambda} \sum_{e \in E} u_{e,t} \,, \quad s.t.$$

$$\sum_{v \in V} x_{v,t} \leq 1 \,, \quad \forall \; t = 1, 2, \ldots \tag{1}$$

$$u_{e,t} + \sum_{t' < t} x_{u,t'} + \sum_{t' < t} x_{v,t'} \geq 1, \quad \forall \; e = (u, v), t = 1, 2, \ldots \tag{2}$$

$$u, x \geq 0 \,.$$

Bi-criteria algorithm for DMSVC

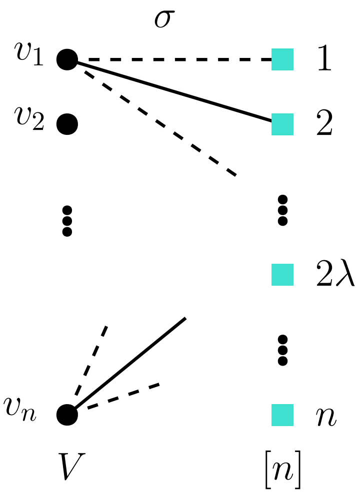


Figure: Assignment Graph

Naive Approach

Solve the above LP.

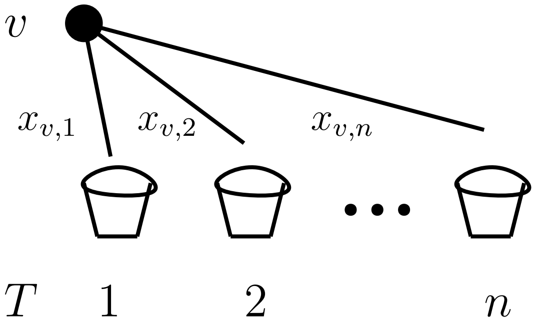


Figure: Try to throw each vertex in bins with probability $x_{v,t}$

Naive Approach

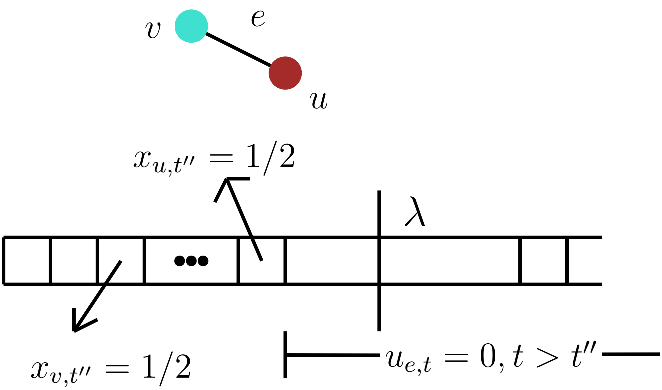


Figure: Try to throw each vertex in bins with probability $x_{v,t}$

Naive Approach

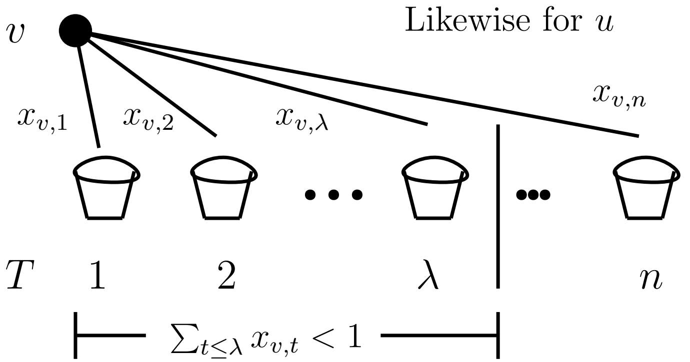


Figure: Try to throw each vertex in bins with probability $x_{v,t}$

Naive Approach

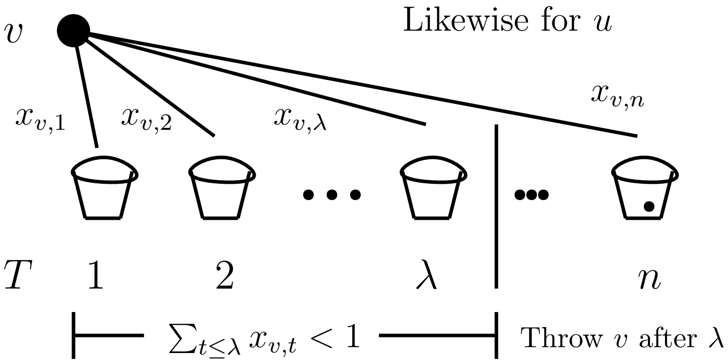


Figure: Ball might be thrown in the last bin

Naive Approach

Independently throw each vertex into the bins using scaled probabilities (x).

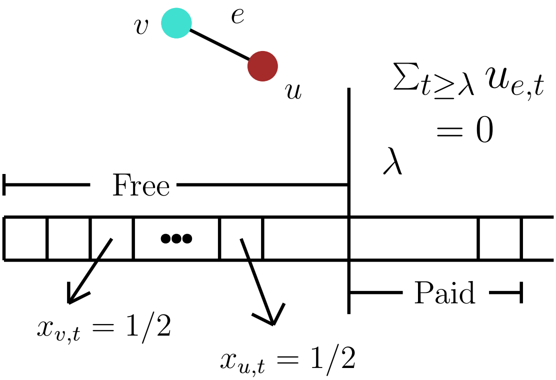


Figure: Linear Program does not pay anything, While we pay a lot.

Approach of [FLT04]

If $e = \{v, u\}$ is fractionally scheduled before λ, then either $\sum_{t<\lambda} x_{v,t} \geq 1/2$ or $\sum_{t<\lambda} x_{u,t} \geq 1/2$.
(Since fractional covering means $\sum_{t<\lambda} x_{v,t} + \sum_{t<\lambda} x_{u,t} \geq 1$.)

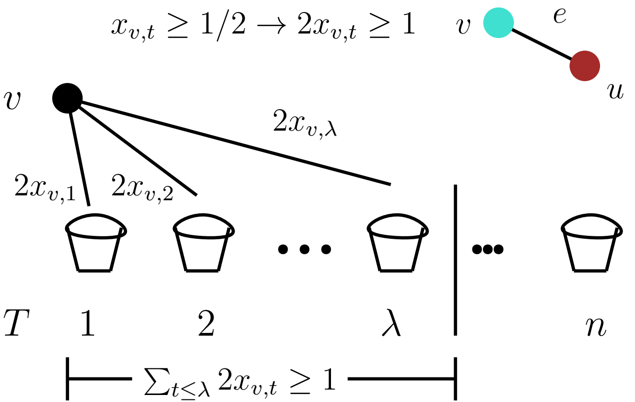


Figure: [FLT04]

Approach of [FLT04]

Independently throw each vertex into the bins using scaled probabilities $(2x)$.

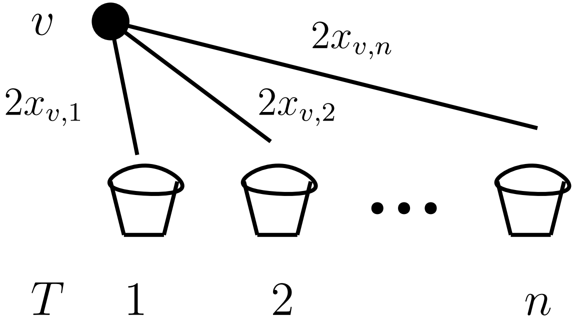


Figure: [FLT04]

Approach of [FLT04]

Break the bins and schedule a uniformly random permutation of vertices inside the bin.

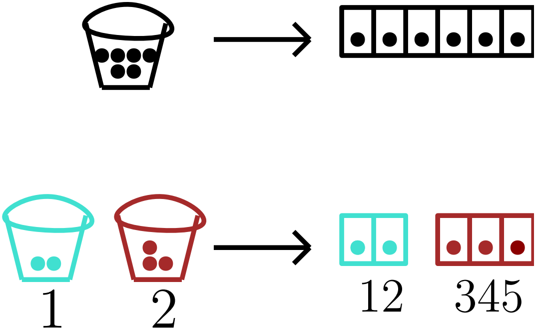


Figure: [FLT04]

Problem?

Recall, $\sum_{v \in V} x_{v,t} \le 1$. Thus
$\mathbb{E}[\#\text{vertices in a bin}] = 2 \sum_{v \in V} x_{v,t} \le 2$.

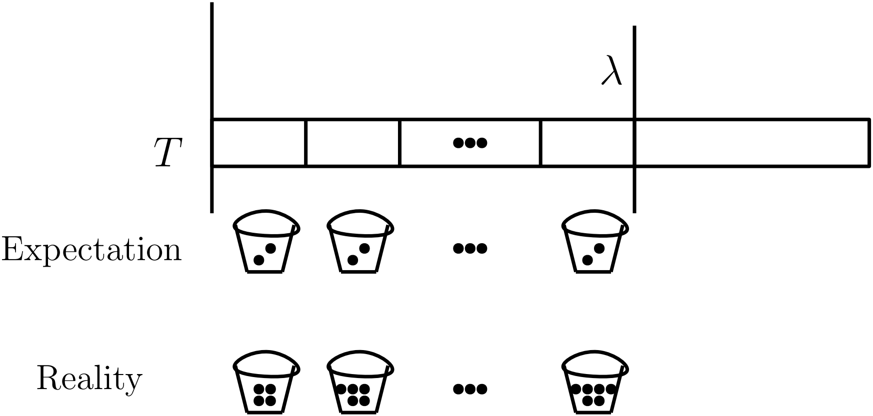


Figure: [FLT04]

Problem?

Want to ensure with high probability that at most 2λ vertices are scheduled within first λ slots.

Ingredients We Need

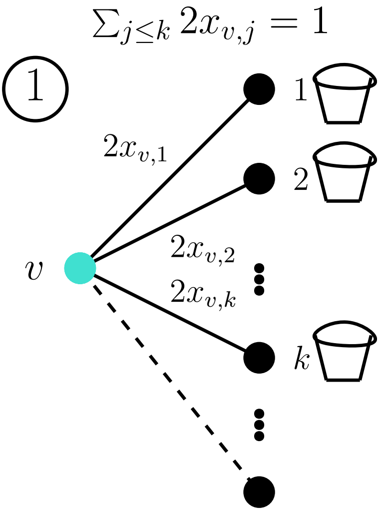


Figure: Marginal Preservation: Throw v into a bin with probability $2x_{v,t}$

Ingredients We Hope for

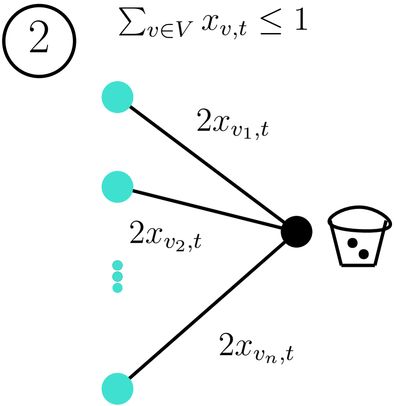


Figure: Dependent Rounding on Side of Slots

- ≤ 2 vertices in each bin with probability 1.
- Thus, $\leq 2\lambda$ vertices in first λ bins with probability 1.

Fix?

Dependent Rounding.

Dependent Rounding [GKPS06]

Dependent Rounding [GKPS06]

Theorem

Given a weighted bipartite graph $(V \cup T, F, z : F \to [0,1])$ *(*$z = 2x$*), it is possible to sample a subset of edges* $S \subseteq F$ *satisfying the following criterion:*

Dependent Rounding [GKPS06]

Theorem

Given a weighted bipartite graph $(V \cup T, F, z : F \to [0,1])$ *(*$z = 2x$*), it is possible to sample a subset of edges* $S \subseteq F$ *satisfying the following criterion:*

1. $\Pr((v,t) \in S) = 2x_{v,t}$ *(Marginal Preservation),*

Dependent Rounding [GKPS06]

Theorem

Given a weighted bipartite graph $(V \cup T, F, z : F \to [0,1])$ *($z = 2x$), it is possible to sample a subset of edges* $S \subseteq F$ *satisfying the following criterion:*

1. $\Pr((v,t) \in S) = 2x_{v,t}$ *(Marginal Preservation),*
2. $deg_S(t) = \lceil \sum_{e \in \delta(t)} 2x_{v,t} \rceil \leq 2$ *(Ensures that the load is at most 2).*

Dependent Rounding [GKPS06]

Theorem

Given a weighted bipartite graph $(V \cup T, F, z : F \to [0,1])$ $(z = 2x)$*, it is possible to sample a subset of edges* $S \subseteq F$ *satisfying the following criterion:*

1. $\Pr((v,t) \in S) = 2x_{v,t}$ *(Marginal Preservation),*
2. $deg_S(t) = \left\lceil \sum_{e \in \delta(t)} 2x_{v,t} \right\rceil \leq 2$ *(Ensures that the load is at most 2).*
3. $deg_S(v) = \sum_{e \in \delta(v)} 2x_{v,t} = 1$ *(Ensures each vertex is thrown into exactly 1 bin)*

Dependent Rounding [GKPS06]

Theorem

Given a weighted bipartite graph $(V \cup T, F, z : F \rightarrow [0,1])$ $(z = 2x)$*, it is possible to sample a subset of edges* $S \subseteq F$ *satisfying the following criterion:*

1. $\Pr((v,t) \in S) = 2x_{v,t}$ *(Marginal Preservation),*
2. $deg_S(t) = \left\lceil \sum_{e \in \delta(t)} 2x_{v,t} \right\rceil \leq 2$ *(Ensures that the load is at most 2).*
3. $deg_S(v) = \sum_{e \in \delta(v)} 2x_{v,t} = 1$ *(Ensures each vertex is thrown into exactly 1 bin)*
4. *(Negative Correlation): For any time t,* $\Pr((v,t) \in S, (u,t) \in S) \leq \Pr((v,t) \in S) \Pr((u,t) \in S)$. *[However, we do not use it]*

Our Algorithm:

Our Algorithm:

1. Guess the appropriate discount given ℓ: ℓ^{th}-largest cover time of OPT ($Cov_{\sigma^*}(e_\ell)$).

Our Algorithm:

1. Guess the appropriate discount given ℓ: ℓ^{th}-largest cover time of OPT ($Cov_{\sigma^*}(e_\ell)$).
2. Solve the linear program for DMSVC with the discount chosen above to obtain solution (x, u).

Our Algorithm:

1. Guess the appropriate discount given ℓ: ℓ^{th}-largest cover time of OPT $(Cov_{\sigma^*}(e_\ell))$.
2. Solve the linear program for DMSVC with the discount chosen above to obtain solution (x, u).
3. Run Dependent rounding on the bipartite graph $(V \cup T, F, 2x : F \to [0,1])$. (The sampled edges give an assignment of vertices to time slot).

Our Algorithm:

1. Guess the appropriate discount given ℓ: ℓ^{th}-largest cover time of OPT ($Cov_{\sigma^*}(e_\ell)$).
2. Solve the linear program for DMSVC with the discount chosen above to obtain solution (x, u).
3. Run Dependent rounding on the bipartite graph $(V \cup T, F, 2x : F \rightarrow [0,1])$. (The sampled edges give an assignment of vertices to time slot).
4. Break bins of time slots t (Each bin has at most 2 vertices: cost goes up by at most 2).

The End

Thank you.

Open Directions

Open Directions

- Getting a 2-approx. for All-Norm Vertex Cover (Improving from 8 [GGKT08])

Open Directions

- Getting a 2-approx. for All-Norm Vertex Cover (Improving from 8 [GGKT08])
- Getting ℓ-dependent or w-dependent approximation algorithms for other problems (such as Load Balancing or k-Median).

References

A. A. Ageev and M. I. Sviridenko.
Pipage rounding: a new method of constructing algorithms with proven performance guarantee.
Journal of Combinatorial Optimization, 8:2004.

Nikhil Bansal, Jatin Batra, Majid Farhadi, and Prasad Tetali.
Improved approximations for min sum vertex cover and generalized min sum set cover.
SODA '21, page 986–1005. Society for Industrial and Applied Mathematics, 2021.

Jarosław Byrka, Krzysztof Sornat, and Joachim Spoerhase.
Constant-factor approximation for ordered k-median.
In *Proceedings of the 50th Annual ACM SIGACT Symposium on Theory of Computing*, STOC 2018, page 620–631, New York, NY, USA, 2018. Association for Computing Machinery.
doi:10.1145/3188745.3188930.

Deeparnab Chakrabarty and Chaitanya Swamy.
Interpolating between k-median and k-center: Approximation