On the Approximability of Multistage Min-Sum Set Cover

Panagiotis Kostopanagiotis

National Technical University of Athens

16/07/2021

Outline

Introduction - Static Version(MSSC)

Introduction - Static Version(MSSC)

- Given universe $U = \{1, 2, ..., n\}(w.l.o.g)$

Introduction - Static Version(MSSC)

- Given universe $U = \{1, 2, ..., n\}(w.l.o.g)$
- Collection $S_1, S_2, \ldots, S_m$

Introduction - Static Version(MSSC)

- Given universe $U = \{1, 2, ..., n\}(w.l.o.g)$
- Collection $S_1, S_2, \ldots, S_m$
- Select $\pi \subseteq [n!]$, Minimize: $\sum_{i=1}^{m} \underbrace{\pi(S_i)}_{AccessCost}$.

Introduction - Static Version(MSSC)

- Given universe $U = \{1, 2, ..., n\}(w.l.o.g)$
- Collection $S_1, S_2, \ldots, S_m$
- Select $\pi \subseteq [n!]$, Minimize: $\sum_{i=1}^{m} \underbrace{\pi(S_i)}_{AccessCost}$.
- $\pi(S_i) =$ The first position k s.t $\pi(k) \in S_i$.

Introduction - Static Version(MSSC)

- Given universe $U = \{1, 2, ..., n\}(w.l.o.g)$
- Collection $S_1, S_2, \ldots, S_m$
- Select $\pi \subseteq [n!]$, Minimize: $\sum_{i=1}^{m} \underbrace{\pi(S_i)}_{AccessCost}$.
- $\pi(S_i) =$ The first position k s.t $\pi(k) \in S_i$.
- For example: $U = \{1, 2, 3, 4, 5\}, S_1 = \{3\}, S_2 = \{1, 3, 4\}, S_3 = \{4, 5\}$

Introduction - Static Version(MSSC)

- Given universe $U = \{1, 2, ..., n\}(w.l.o.g)$
- Collection $S_1, S_2, \dots, S_m$
- Select $\pi \subseteq [n!]$, Minimize: $\sum_{i=1}^{m} \underbrace{\pi(S_i)}_{AccessCost}$.
- $\pi(S_i) =$ The first position k s.t $\pi(k) \in S_i$.
- For example: $U = \{1, 2, 3, 4, 5\}, S_1 = \{3\}, S_2 = \{1, 3, 4\}, S_3 = \{4, 5\}$
- Optimal solution:

Introduction - Static Version(MSSC)

- Given universe $U = \{1, 2, ..., n\}(w.l.o.g)$
- Collection $S_1, S_2, \ldots, S_m$
- Select $\pi \subseteq [n!]$, Minimize: $\sum_{i=1}^{m} \underbrace{\pi(S_i)}_{AccessCost}$.
- $\pi(S_i) =$ The first position k s.t $\pi(k) \in S_i$.
- For example: $U = \{1, 2, 3, 4, 5\}, S_1 = \{3\}, S_2 = \{1, 3, 4\}, S_3 = \{4, 5\}$
- Optimal solution: $\pi = \{3, 4, 5, 2, 1\}$,

Introduction - Static Version(MSSC)

- Given universe $U = \{1, 2, ..., n\}(w.l.o.g)$
- Collection $S_1, S_2, \ldots, S_m$
- Select $\pi \subseteq [n!]$, Minimize: $\sum_{i=1}^{m} \underbrace{\pi(S_i)}_{AccessCost}$.
- $\pi(S_i) =$ The first position k s.t $\pi(k) \in S_i$.
- For example: $U = \{1, 2, 3, 4, 5\}, S_1 = \{3\}, S_2 = \{1, 3, 4\}, S_3 = \{4, 5\}$
- Optimal solution: $\pi = \{3, 4, 5, 2, 1\}$, Access Cost $= 1 + 1 + 2 = 4$.

Motivation

Motivation

- Imagine an online store with multiple products.

Motivation

- Imagine an online store with multiple products.
- We have customers who have declared their interests.

Motivation

- Imagine an online store with multiple products.
- We have customers who have declared their interests.
- e.g a customer may be interested in sports and gardening, another in science, chess and videogames.

Motivation

- Imagine an online store with multiple products.
- We have customers who have declared their interests.
- e.g a customer may be interested in sports and gardening, another in science, chess and videogames.
- We want to maintain an ordering of products, so that customers find something they like quickly.

Motivation

- Imagine an online store with multiple products.
- We have customers who have declared their interests.
- e.g a customer may be interested in sports and gardening, another in science, chess and videogames.
- We want to maintain an ordering of products, so that customers find something they like quickly.
- In this example $S_1, S_2, \ldots, S_m$ model the users' interests and π is the ordering we're looking for.

Min Sum Set Cover - First Results

Min Sum Set Cover - First Results

- Min-Sum Set Cover is NP-Hard.

Min Sum Set Cover - First Results

- Min-Sum Set Cover is NP-Hard.
- Greedy 4-Approximation Algorithm [3].

Min Sum Set Cover - First Results

- Min-Sum Set Cover is NP-Hard.
- Greedy 4-Approximation Algorithm [3].

Greedy Algorithm for Min Sum Set Cover

$\pi = [\ ]$
while π doesn't contain all the elements of U **do**
    Append to the end of π the element of U that is contained in the most sets.
end while
return π

Min Sum Set Cover - First Results

- Min-Sum Set Cover is NP-Hard.
- Greedy 4-Approximation Algorithm [3].

Greedy Algorithm for Min Sum Set Cover

$\pi = [\,]$
while π doesn't contain all the elements of U **do**
    Append to the end of π the element of U that is contained in the most sets.
end while
return π

- This is optimal unless P=NP

Min Sum Set Cover - First Results

- Min-Sum Set Cover is NP-Hard.
- Greedy 4-Approximation Algorithm [3].

Greedy Algorithm for Min Sum Set Cover

1: $\pi = [\,]$
2: **while** π doesn't contain all the elements of U **do**
3: Append to the end of π the element of U that is contained in the most sets.
4: **end while**
5: **return** π

- This is optimal unless P=NP (NP-Hard to approximate with factor $4 - \epsilon$).

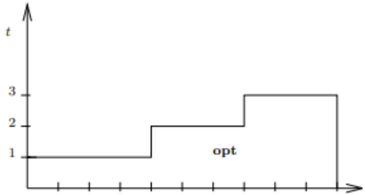
Proof Sketch

Proof Sketch

- Map Greedy and Optimal Solution to Histograms.

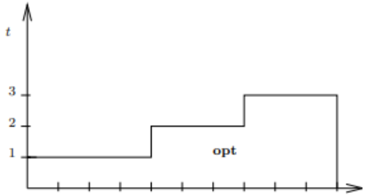
Proof Sketch

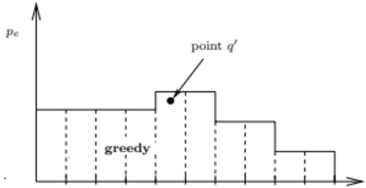
- Map Greedy and Optimal Solution to Histograms.



Proof Sketch

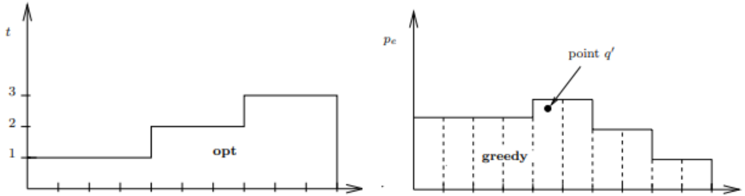
- Map Greedy and Optimal Solution to Histograms.





Proof Sketch

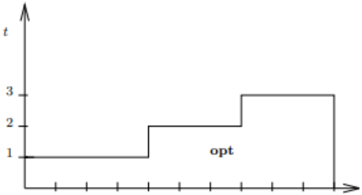
- Map Greedy and Optimal Solution to Histograms.

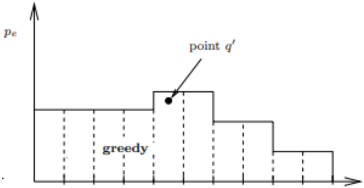


- Shrink Greedy vertically and horizontally by 2.

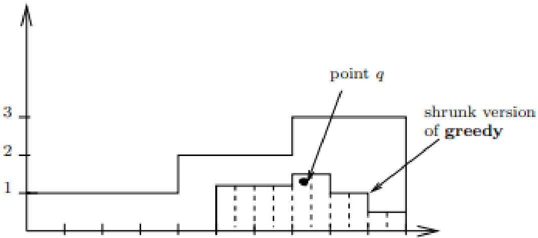
Proof Sketch

- Map Greedy and Optimal Solution to Histograms.





- Shrink Greedy vertically and horizontally by 2.



Generalized MSSC

Generalized MSSC

- What if the customers are demanding?

Generalized MSSC

- What if the customers are demanding?
- Along with $S_1, \ldots, S_m$ we are given: $K(S_i) \leq |S_i|$.

Generalized MSSC

- What if the customers are demanding?
- Along with $S_1, \ldots, S_m$ we are given: $K(S_i) \leq |S_i|$.
- Find π to minimize: $\sum_{i=1}^{m} \pi_{K(S_i)}(S_i)$.

Generalized MSSC

- What if the customers are demanding?
- Along with $S_1, \ldots, S_m$ we are given: $K(S_i) \leq |S_i|$.
- Find π to minimize: $\sum_{i=1}^{m} \pi_{K(S_i)}(S_i)$.
- Here $\pi_{K(S_i)}(S_i) =$ the position of the $K(S_i)$-th element of S_i in π.

Generalized MSSC

- What if the customers are demanding?
- Along with $S_1, \ldots, S_m$ we are given: $K(S_i) \leq |S_i|$.
- Find π to minimize: $\sum_{i=1}^{m} \pi_{K(S_i)}(S_i)$.
- Here $\pi_{K(S_i)}(S_i) =$ the position of the $K(S_i)$-th element of S_i in π.
- Firstly introduced in [1] by Azar et. al. $O(logn)$-apx.

Generalized MSSC

- What if the customers are demanding?
- Along with $S_1, \ldots, S_m$ we are given: $K(S_i) \leq |S_i|$.
- Find π to minimize: $\sum_{i=1}^{m} \pi_{K(S_i)}(S_i)$.
- Here $\pi_{K(S_i)}(S_i) =$ the position of the $K(S_i)$-th element of S_i in π.
- Firstly introduced in [1] by Azar et. al. $O(logn)$-apx.
- Constant factor approximation in [2] by Bansal et al. 485-apx.

Generalized MSSC

- What if the customers are demanding?
- Along with $S_1, \ldots, S_m$ we are given: $K(S_i) \leq |S_i|$.
- Find π to minimize: $\sum_{i=1}^{m} \pi_{K(S_i)}(S_i)$.
- Here $\pi_{K(S_i)}(S_i) =$ the position of the $K(S_i)$-th element of S_i in π.
- Firstly introduced in [1] by Azar et. al. $O(logn)$-apx.
- Constant factor approximation in [2] by Bansal et al. 485-apx.
- LP-Rounding Approach, Skutella and Williamson [6] improve it to 28-apx.

GMSSC - Fractional LP

$$
\begin{aligned}
\text{m}in \quad & \sum_{t=1}^{n}\sum_{i=1}^{m}(1-y_{i,t}) \\
\text{s.t} \quad & \sum_{e\in U} x_{e,t} = 1 \qquad \text{for all } t \leq n \\
& \sum_{t=1}^{n} x_{e,t} = 1 \qquad \text{for all } e \in U \\
& \sum_{e\in S\setminus A}\sum_{t'<t} x_{e,t} \geq (K(S_i) - |A|)\cdot y_{i,t} \qquad \text{for all } i \leq m, A \subseteq S_i, t \leq n \\
& x_{e,t}, y_{i,t} \in [0,1] \qquad \text{for all } e \in U, i \leq m, t \leq n
\end{aligned}
$$

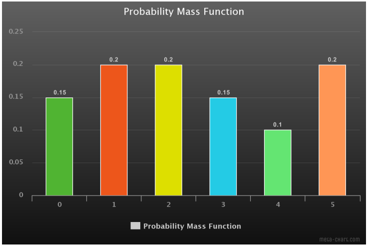
GMSSC - Fractional LP

$$
\begin{aligned}
\text{m}in \quad & \sum_{t=1}^{n}\sum_{i=1}^{m}(1-y_{i,t}) \\
\text{s.t} \quad & \sum_{e\in U} x_{e,t} = 1 \qquad \text{for all } t \leq n \\
& \sum_{t=1}^{n} x_{e,t} = 1 \qquad \text{for all } e \in U \\
& \sum_{e\in S\setminus A}\sum_{t'<t} x_{e,t} \geq (K(S_i) - |A|)\cdot y_{i,t} \qquad \text{for all } i \leq m, A \subseteq S_i, t \leq n \\
& x_{e,t}, y_{i,t} \in [0,1] \qquad \text{for all } e \in U, i \leq m, t \leq n
\end{aligned}
$$

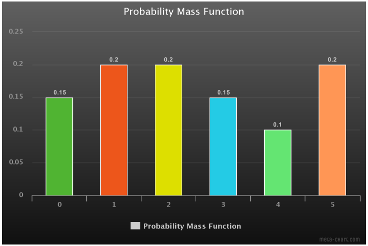
Use $x_{e,t}$ to order the elements and construct π.

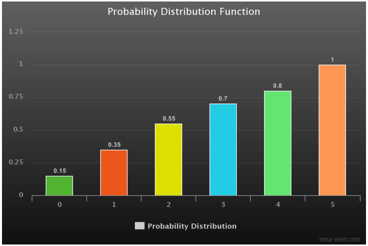
GMSSC - Randomized Rounding

GMSSC - Randomized Rounding

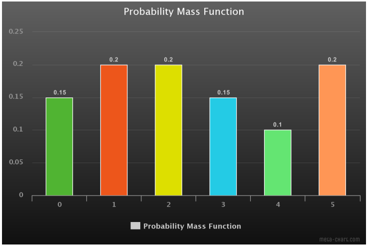


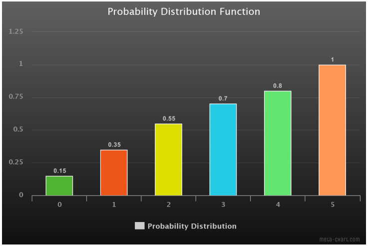
GMSSC - Randomized Rounding





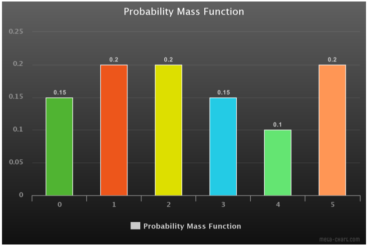
GMSSC - Randomized Rounding

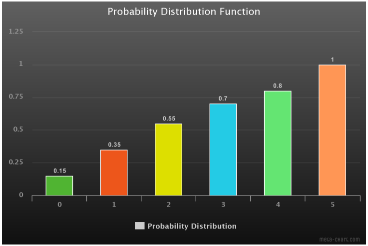




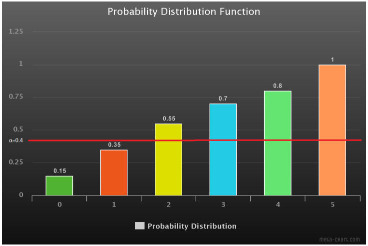
Pick uniformly at random $\alpha \in [0, 1]$.

GMSSC - Randomized Rounding

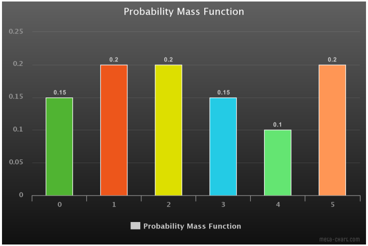


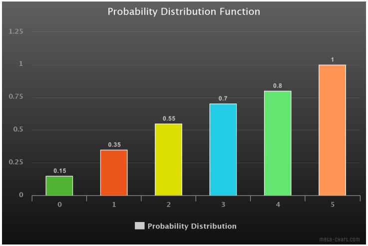


Pick uniformly at random $\alpha \in [0, 1]$.

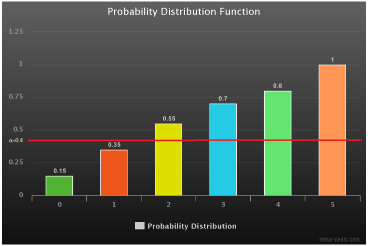


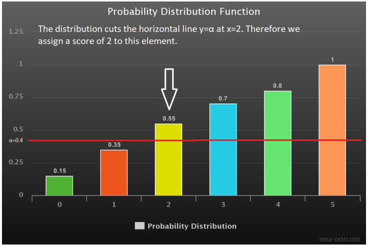
GMSSC - Randomized Rounding



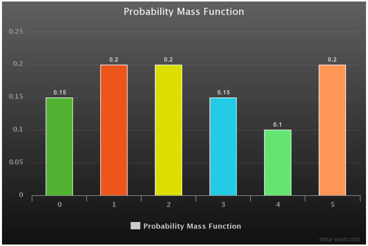


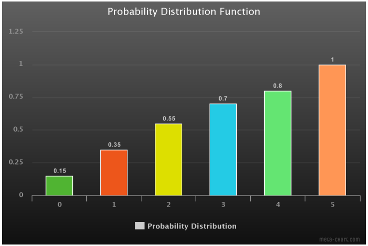
Pick uniformly at random $\alpha \in [0, 1]$.



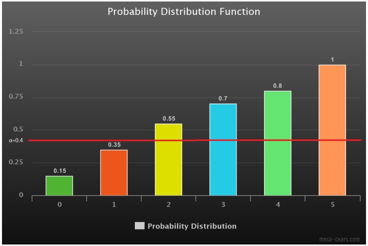


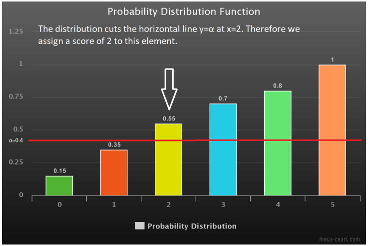
GMSSC - Randomized Rounding





Pick uniformly at random $\alpha \in [0, 1]$.





Sort in order of non-decreasing score.

Multistage Min-Sum Set Cover

Multistage Min-Sum Set Cover

- What if users' interests change over time?

Multistage Min-Sum Set Cover

- What if users' interests change over time?
- Given $S_1, S_2, \ldots, S_T \subseteq \{1, 2, \ldots, n\}$, construct sequence of permutations $\pi^1, \pi^2, ..., \pi^T$

Multistage Min-Sum Set Cover

- What if users' interests change over time?
- Given $S_1, S_2, \dots, S_T \subseteq \{1, 2, \dots, n\}$, construct sequence of permutations $\pi^1, \pi^2, ..., \pi^T$

$$\text{Minimize} \underbrace{\sum_{t=1}^{T} \pi^t(S_t)}_{\text{Access Cost}} + \underbrace{\sum_{t=1}^{T} \#inversions(\pi^t, \pi^{t-1})}_{\text{Moving Cost}}$$

Multistage Min-Sum Set Cover

- What if users' interests change over time?
- Given $S_1, S_2, \ldots, S_T \subseteq \{1, 2, \ldots, n\}$, construct sequence of permutations $\pi^1, \pi^2, ..., \pi^T$

$$\text{Minimize} \underbrace{\sum_{t=1}^{T} \pi^t(S_t)}_{\text{Access Cost}} + \underbrace{\sum_{t=1}^{T} \# inversions(\pi^t, \pi^{t-1})}_{\text{Moving Cost}}$$

- W.L.O.G $\pi^0 = [1, 2, \ldots, n]$.

Mult-MSSC - **The Online Setting**

Mult-MSSC - **The Online Setting**

- We don’t know how the users’ interests evolve over time(realistic scenario).

Mult-MSSC - **The Online Setting**

- We don't know how the users' interests evolve over time(realistic scenario).
- W.L.O.G Consider the r-uniform case: $|S_t| = r, \forall t \leq T$

Mult-MSSC - **The Online Setting**

- We don't know how the users' interests evolve over time(realistic scenario).
- W.L.O.G Consider the r-uniform case: $|S_t| = r, \forall t \leq T$
- In [4] the authors provide the following results:

Mult-MSSC - **The Online Setting**

- We don't know how the users' interests evolve over time(realistic scenario).
- W.L.O.G Consider the r-uniform case: $|S_t| = r, \forall t \leq T$
- In [4] the authors provide the following results:
  1. A simple combinatorial online algorithm (Move-All-Equally) which is $\Theta(r\sqrt{(}n))$-competitive.

Mult-MSSC - **The Online Setting**

- We don't know how the users' interests evolve over time(realistic scenario).
- W.L.O.G Consider the r-uniform case: $|S_t| = r, \forall t \leq T$
- In [4] the authors provide the following results:
  1. A simple combinatorial online algorithm (Move-All-Equally) which is $\Theta(r\sqrt(n))$-competitive.
  2. An $\Omega(r)$ lower bound on the competitive ratio of any online algorithm.

Mult-MSSC - **The Online Setting**

- We don't know how the users' interests evolve over time(realistic scenario).
- W.L.O.G Consider the r-uniform case: $|S_t| = r, \forall t \leq T$
- In [4] the authors provide the following results:
  1. A simple combinatorial online algorithm (Move-All-Equally) which is $\Theta(r\sqrt(n))$-competitive.
  2. An $\Omega(r)$ lower bound on the competitive ratio of any online algorithm.
- Bridge the Gap? $\rightarrow$

Mult-MSSC - **The Online Setting**

- We don't know how the users' interests evolve over time(realistic scenario).
- W.L.O.G Consider the r-uniform case: $|S_t| = r, \forall t \leq T$
- In [4] the authors provide the following results:
    1. A simple combinatorial online algorithm (Move-All-Equally) which is $\Theta(r\sqrt(n))$-competitive.
    2. An $\Omega(r)$ lower bound on the competitive ratio of any online algorithm.
- Bridge the Gap? $\rightarrow$ Construct approximation algorithms for the offline version.

Mult-MSSC - **The Offline Version**

- **INPUT:** A sequence of request sets $S_1, S_2, \ldots, S_T \subseteq U$

Mult-MSSC - **The Offline Version**

- **INPUT:** A sequence of request sets $S_1, S_2, \ldots, S_T \subseteq U$
- **OUTPUT:** A Sequence of permutations minimizing

Mult-MSSC - **The Offline Version**

- **INPUT:** A sequence of request sets $S_1, S_2, \ldots, S_T \subseteq U$
- **OUTPUT:** A Sequence of permutations minimizing

$$\underbrace{\sum_{t=1}^{T} \pi^t(S_t)}_{AccessCost} + \underbrace{\sum_{t=1}^{T} d_{KT}(\pi^t, \pi^{t-1})}_{MovingCost}$$

Mult-MSSC - **The Offline Version**

- **INPUT:** A sequence of request sets $S_1, S_2, \ldots, S_T \subseteq U$
- **OUTPUT:** A Sequence of permutations minimizing

$$\underbrace{\sum_{t=1}^{T} \pi^t(S_t)}_{AccessCost} + \underbrace{\sum_{t=1}^{T} d_{KT}(\pi^t, \pi^{t-1})}_{MovingCost}$$

- Here d_{KT} denotes the Kendall-Tau distance - number of inv. to transform π^{t-1} to π^t.

Mult-MSSC - **Our Results**

- Bad News: Approximation preserving reduction from Set Cover

Mult-MSSC - **Our Results**

- Bad News: Approximation preserving reduction from Set Cover
  1. There is no $o(logn)$-approximation algorithm for Mult-MSSC unless $P = NP$.

Mult-MSSC - **Our Results**

- Bad News: Approximation preserving reduction from Set Cover
  1. There is no $o(logn)$-approximation algorithm for Mult-MSSC unless $P = NP$.
  2. There is no $o(r)$-approximation algorithm for **r-bounded** request sequences(unless there is one for r-bounded version of Set Cover).

Mult-MSSC - **Our Results**

- Bad News: Approximation preserving reduction from Set Cover
  1. There is no $o(logn)$-approximation algorithm for Mult-MSSC unless $P = NP$.
  2. There is no $o(r)$-approximation algorithm for **r-bounded** request sequences(unless there is one for r-bounded version of Set Cover).
- Good News: Almost Matching Upper Bounds

Mult-MSSC - **Our Results**

- Bad News: Approximation preserving reduction from Set Cover
  1. There is no $o(logn)$-approximation algorithm for Mult-MSSC unless $P = NP$.
  2. There is no $o(r)$-approximation algorithm for **r-bounded** request sequences(unless there is one for r-bounded version of Set Cover).
- Good News: Almost Matching Upper Bounds
  1. There exists a randomized $O(log^2 n)$-approximation algorithm for Mult-MSSC.

Mult-MSSC - **Our Results**

- Bad News: Approximation preserving reduction from Set Cover
  1. There is no $o(logn)$-approximation algorithm for Mult-MSSC unless $P = NP$.
  2. There is no $o(r)$-approximation algorithm for **r-bounded** request sequences(unless there is one for r-bounded version of Set Cover).
- Good News: Almost Matching Upper Bounds
  1. There exists a randomized $O(log^2 n)$-approximation algorithm for Mult-MSSC.
  2. There exists a deterministic $O(r^2)$-approximation algorithm for the **r-bounded** version of Mult-MSSC.

Mult-MSSC - **The Move-To-Front Version**

Mult-MSSC - **The Move-To-Front Version**

INPUT: A Request Sequence $S_1, S_2, \ldots, S_T \subseteq U$.

Mult-MSSC - **The Move-To-Front Version**

INPUT: A Request Sequence $S_1, S_2, \ldots, S_T \subseteq U$.
OUTPUT: A Sequence of permutations $\pi^1, \pi^2, \ldots, \pi^T$ s.t.

Mult-MSSC - **The Move-To-Front Version**

INPUT: A Request Sequence $S_1, S_2, \ldots, S_T \subseteq U$.
OUTPUT: A Sequence of permutations $\pi^1, \pi^2, \ldots, \pi^T$ s.t.

1. $\pi^t(1) \in S_t \ \forall \ t \leq T$.

INPUT: A Request Sequence $S_1, S_2, \ldots, S_T \subseteq U$.
OUTPUT: A Sequence of permutations $\pi^1, \pi^2, \ldots, \pi^T$ s.t.

1. $\pi^t(1) \in S_t \ \forall \ t \leq T$.
2. The moving cost $\sum_{t=1}^{T} d_{KT}(\pi^t, \pi^{t-1})$ is minimized.

Lemma

Mult-MSSC and Move-To-Front are approximately the same. That is:

Mult-MSSC - **The Move-To-Front Version**

INPUT: A Request Sequence $S_1, S_2, \ldots, S_T \subseteq U$.
OUTPUT: A Sequence of permutations $\pi^1, \pi^2, \ldots, \pi^T$ s.t.

1. $\pi^t(1) \in S_t \; \forall \; t \leq T$.
2. The moving cost $\sum_{t=1}^{T} d_{KT}(\pi^t, \pi^{t-1})$ is minimized.

Lemma

Mult-MSSC and Move-To-Front are approximately the same. That is:

- $OPT_{\text{Mult-MSSC}} \leq OPT_{MTF}$

Mult-MSSC - **The Move-To-Front Version**

INPUT: A Request Sequence $S_1, S_2, \ldots, S_T \subseteq U$.
OUTPUT: A Sequence of permutations $\pi^1, \pi^2, \ldots, \pi^T$ s.t.

1. $\pi^t(1) \in S_t \ \forall \ t \leq T$.
2. The moving cost $\sum_{t=1}^{T} d_{KT}(\pi^t, \pi^{t-1})$ is minimized.

Lemma

Mult-MSSC and Move-To-Front are approximately the same. That is:

- $OPT_{\text{Mult-MSSC}} \leq OPT_{MTF}$
- $OPT_{MTF} \leq 4 \cdot OPT_{\text{Mult-MSSC}}$

Move-To-Front - Linear Relaxation

Move-To-Front - Linear Relaxation

Fact

Any permutation can be represented with a 0-1 doubly stochastic matrix.

Move-To-Front - Linear Relaxation

Fact

Any permutation can be represented with a 0-1 doubly stochastic matrix.

LP Relaxation of Move-to-Front

$$
\begin{aligned}
\min \quad & \sum_{t=1}^{T} \mathrm{d}_{\mathrm{FR}}(A^t, A^{t-1}) \\
\text{s.t} \quad & \sum_{i=1}^{n} A^t_{ei} = 1 \qquad e \in U \text{ and } t = 1, \dots, T \\
& \sum_{e \in U} A^t_{ei} = 1 \qquad i = 1, \dots, n \text{ and } t = 1, \dots, T \\
& \sum_{e \in S_t} A^t_{e1} = 1 \quad t = 1, \dots, T \\
& A^0 = \pi^0 \\
& A^t_{ei} \geq 0 \qquad\quad e \in U,\ i = 1, \dots, n \text{ and } t = 1, \dots, T
\end{aligned}
$$

Move-To-Front - Linear Relaxation

Fact

Any permutation can be represented with a 0-1 doubly stochastic matrix.

LP Relaxation of Move-to-Front

$$
\begin{aligned}
\min \quad & \sum_{t=1}^{T} \mathrm{d_{FR}}(A^t, A^{t-1}) \leftarrow \textbf{Extension of Kendall-Tau Distance} \\
\text{s.t} \quad & \sum_{i=1}^{n} A^t_{ei} = 1 \qquad e \in U \text{ and } t = 1, \ldots, T \\
& \sum_{e \in U} A^t_{ei} = 1 \qquad i = 1, \ldots, n \text{ and } t = 1, \ldots, T \\
& \sum_{e \in S_t} A^t_{e1} = 1 \quad t = 1, \ldots, T \\
& A^0 = \pi^0 \\
& A^t_{ei} \geq 0 \qquad e \in U,\ i = 1, \ldots, n \text{ and } t = 1, \ldots, T
\end{aligned}
$$

Move-To-Front - Linear Relaxation

Fact

Any permutation can be represented with a 0-1 doubly stochastic matrix.

LP Relaxation of Move-to-Front

$$
\begin{aligned}
\min \quad & \sum_{t=1}^{T} \mathrm{d_{FR}}(A^t, A^{t-1}) \leftarrow \textbf{Extension of Kendall-Tau Distance} \\
\text{s.t} \quad & \sum_{i=1}^{n} A^t_{ei} = 1 \quad\quad e \in U \text{ and } t = 1, \dots, T \\
& \sum_{e \in U} A^t_{ei} = 1 \quad\quad i = 1, \dots, n \text{ and } t = 1, \dots, T \\
& \sum_{e \in S_t} A^t_{e1} = 1 \quad\quad t = 1, \dots, T \leftarrow \textbf{Element of } S_t \textbf{ in the first position} \\
& A^0 = \pi^0 \\
& A^t_{ei} \geq 0 \quad\quad e \in U,\ i = 1, \dots, n \text{ and } t = 1, \dots, T
\end{aligned}
$$

Move-to-Front - Linear Relaxation

Move-to-Front - Linear Relaxation

Footrule Distance d_{FR} - Optimal Value of the Transportation Problem:

Move-to-Front - Linear Relaxation

Footrule Distance d_{FR} - Optimal Value of the Transportation Problem:

$$
\begin{aligned}
\min \quad & \sum_{e \in U} \sum_{i=1}^{n} \sum_{j=1}^{n} |i-j| \cdot f_{ij}^{e} \\
\text{s.t} \quad & \sum_{i=1}^{n} f_{ij}^{e} = B_{ej} \quad \text{for all } e \in U \text{ and } j = 1, \ldots, n \\
& \sum_{j=1}^{n} f_{ij}^{e} = A_{ei} \quad \text{for all } e \in U \text{ and } i = 1, \ldots, n
\end{aligned}
$$

Move-to-Front - Linear Relaxation

Footrule Distance d_{FR} - Optimal Value of the Transportation Problem:

$$\begin{aligned} \mathrm{m}in \quad & \sum_{e\in U}\sum_{i=1}^{n}\sum_{j=1}^{n}|i-j|\cdot f_{ij}^{e} \\ \text{s.t} \quad & \sum_{i=1}^{n} f_{ij}^{e} = B_{ej} \quad \text{for all } e\in U \text{ and } j=1,\dots,n \\ & \sum_{j=1}^{n} f_{ij}^{e} = A_{ei} \quad \text{for all } e\in U \text{ and } i=1,\dots,n \end{aligned}$$

Let the stochastic matrices $A = \begin{pmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{pmatrix}$, $B = \begin{pmatrix} 1/3 & 1/3 & 1/3 \\ 1/2 & 1/2 & 0 \\ 1/4 & 0 & 3/4 \end{pmatrix}$.

Move-to-Front - Linear Relaxation

Footrule Distance d_{FR} - Optimal Value of the Transportation Problem:

$$\begin{aligned} \mathrm{m}in \quad & \sum_{e \in U} \sum_{i=1}^{n} \sum_{j=1}^{n} |i-j| \cdot f_{ij}^{e} \\ \text{s.t} \quad & \sum_{i=1}^{n} f_{ij}^{e} = B_{ej} \quad \text{for all } e \in U \text{ and } j = 1, \ldots, n \\ & \sum_{j=1}^{n} f_{ij}^{e} = A_{ei} \quad \text{for all } e \in U \text{ and } i = 1, \ldots, n \end{aligned}$$

Let the stochastic matrices $A = \begin{pmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{pmatrix}$, $B = \begin{pmatrix} 1/3 & 1/3 & 1/3 \\ 1/2 & 1/2 & 0 \\ 1/4 & 0 & 3/4 \end{pmatrix}$.

The FootRule distance $\mathrm{d}_{\mathrm{FR}}(A, B) =$

Move-to-Front - Linear Relaxation

Footrule Distance d_{FR} - Optimal Value of the Transportation Problem:

$$\begin{aligned}
\min \quad & \sum_{e \in U} \sum_{i=1}^{n} \sum_{j=1}^{n} |i-j| \cdot f_{ij}^{e} \\
\text{s.t} \quad & \sum_{i=1}^{n} f_{ij}^{e} = B_{ej} \quad \text{for all } e \in U \text{ and } j = 1, \ldots, n \\
& \sum_{j=1}^{n} f_{ij}^{e} = A_{ei} \quad \text{for all } e \in U \text{ and } i = 1, \ldots, n
\end{aligned}$$

Let the stochastic matrices $A = \begin{pmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{pmatrix}$, $B = \begin{pmatrix} 1/3 & 1/3 & 1/3 \\ 1/2 & 1/2 & 0 \\ 1/4 & 0 & 3/4 \end{pmatrix}$.

The FootRule distance $\mathrm{d}_{\mathrm{FR}}(A, B) = \underbrace{(0 \cdot 1/3 + 1 \cdot 1/3 + 2 \cdot 1/3)}_{\text{first row}}$

Move-to-Front - Linear Relaxation

Footrule Distance d_{FR} - Optimal Value of the Transportation Problem:

$$\begin{aligned} \min \quad & \sum_{e\in U}\sum_{i=1}^{n}\sum_{j=1}^{n} |i-j|\cdot f_{ij}^{e} \\ \text{s.t} \quad & \sum_{i=1}^{n} f_{ij}^{e} = B_{ej} \quad \text{for all } e \in U \text{ and } j = 1,\dots,n \\ & \sum_{j=1}^{n} f_{ij}^{e} = A_{ei} \quad \text{for all } e \in U \text{ and } i = 1,\dots,n \end{aligned}$$

Let the stochastic matrices $A = \begin{pmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{pmatrix}$, $B = \begin{pmatrix} 1/3 & 1/3 & 1/3 \\ 1/2 & 1/2 & 0 \\ 1/4 & 0 & 3/4 \end{pmatrix}$.

The FootRule distance $\mathrm{d}_{\mathrm{FR}}(A, B) = \underbrace{(0\cdot 1/3 + 1\cdot 1/3 + 2\cdot 1/3)}_{\text{first row}} +$

$\underbrace{(1\cdot 1/2 + 0\cdot 1/2 + 1\cdot 0)}_{\text{second row}}$

Move-to-Front - Linear Relaxation

Footrule Distance d_{FR} - Optimal Value of the Transportation Problem:

$$\begin{aligned} \min \quad & \sum_{e\in U}\sum_{i=1}^{n}\sum_{j=1}^{n} |i-j|\cdot f_{ij}^{e} \\ \text{s.t} \quad & \sum_{i=1}^{n} f_{ij}^{e} = B_{ej} \quad \text{for all } e \in U \text{ and } j = 1,\dots,n \\ & \sum_{j=1}^{n} f_{ij}^{e} = A_{ei} \quad \text{for all } e \in U \text{ and } i = 1,\dots,n \end{aligned}$$

Let the stochastic matrices $A = \begin{pmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{pmatrix}$, $B = \begin{pmatrix} 1/3 & 1/3 & 1/3 \\ 1/2 & 1/2 & 0 \\ 1/4 & 0 & 3/4 \end{pmatrix}$.

The FootRule distance $\mathrm{d_{FR}}(A,B) = \underbrace{(0\cdot 1/3 + 1\cdot 1/3 + 2\cdot 1/3)}_{\text{first row}} + \underbrace{(1\cdot 1/2 + 0\cdot 1/2 + 1\cdot 0)}_{\text{second row}} + \underbrace{(2\cdot 1/4 + 1\cdot 0 + 0\cdot 3/4)}_{\text{third row}} = 2.$

Move-To-Front - Randomized Rounding

Move-To-Front - Randomized Rounding

Fact

From the Linear Relaxation of MTF we know:
$\sum_{t=1}^{T} d_{FR}(A_t, A_{t-1}) \leq 4 \cdot OPT_{\text{Mult-MSSC}}$.

Move-To-Front - Randomized Rounding

Fact

From the Linear Relaxation of MTF we know:
$\sum_{t=1}^{T} d_{FR}(A_t, A_{t-1}) \leq 4 \cdot OPT_{\text{Mult-MSSC}}$.

A Randomized Algorithm for $\mathrm{Mult\text{-}MSSC}$

Find the optimal solution $A^0 = \pi^0, A^1, \ldots, A^T$ for $\mathrm{Fractional - MTF}$.
for each element $e \in U$ **do**
  Select α_e uniformly at random in $[0, 1]$.
end for
for $t = 1 \ldots T$ **do**
  for all elements $e \in U$ **do**
    $I_e^t := \mathrm{argmin}_{1 \leq i \leq n}\{\log n \cdot \sum_{s=1}^{i} A_{es}^t \geq \alpha_e\}$.
  end for
  $\pi^t :=$ sort elements according to I_e^t with ties being broken lexicographically.
end for

Analysis of Randomized Rounding

Analysis of Randomized Rounding

Theorem - Approximation Ratio

The Randomized Rounding Algorithm 18 is $O(log^2 n)$ approximation for Mult-MSSC.

Analysis of Randomized Rounding

Theorem - Approximation Ratio

The Randomized Rounding Algorithm 18 is $O(log^2 n)$ approximation for $\mathrm{Mult\text{-}MSSC}$.

- Proof Sketch: We can prove that

Analysis of Randomized Rounding

Theorem - Approximation Ratio

The Randomized Rounding Algorithm 18 is $O(log^2 n)$ approximation for Mult-MSSC.

- Proof Sketch: We can prove that
  1. $\underbrace{\mathbb{E}[d_{KT}(\pi^t, \pi^{t-1})]}_{\text{Expected Moving Cost}} \leq 4log^2 n \cdot \underbrace{d_{FR}(A_t, A_{t-1})}_{\text{moving cost of matrices}}$.

Analysis of Randomized Rounding

Theorem - Approximation Ratio

The Randomized Rounding Algorithm 18 is $O(log^2 n)$ approximation for Mult-MSSC.

- Proof Sketch: We can prove that
  1. $\underbrace{\mathbb{E}[d_{KT}(\pi^t, \pi^{t-1})]}_{\text{Expected Moving Cost}} \leq 4log^2 n \cdot \underbrace{d_{FR}(A_t, A_{t-1})}_{\text{moving cost of matrices}}$.
  2. $\underbrace{\mathbb{E}[\pi^t(S_t)]}_{\text{Access Cost of set } S_t} \leq 2 \cdot \underbrace{\pi_*^t(S_t)}_{\text{Access Cost of the Optimal Permutation}}$.

Analysis of Randomized Rounding

Theorem - Approximation Ratio

The Randomized Rounding Algorithm 18 is $O(log^2 n)$ approximation for Mult-MSSC.

- Proof Sketch: We can prove that
  1. $\underbrace{\mathbb{E}[d_{KT}(\pi^t, \pi^{t-1})]}_{\text{Expected Moving Cost}} \leq 4log^2 n \cdot \underbrace{d_{FR}(A_t, A_{t-1})}_{\text{moving cost of matrices}}$.
  2. $\underbrace{\mathbb{E}[\pi^t(S_t)]}_{\text{Access Cost of set } S_t} \leq 2 \cdot \underbrace{\pi^t_*(S_t)}_{\text{Access Cost of the Optimal Permutation}}$.
- Basic idea: Decompose matrix to sequence of neighboring matrices(matrices that differ only in 2 entries).

The Deterministic Rounding Algorithm for MTF

The Deterministic Rounding Algorithm for MTF

- Sequence of requests $S_1, S_2, \ldots, S_T$ is r-bounded iff $|S_t| \leq r,$ for all t.

The Deterministic Rounding Algorithm for MTF

- Sequence of requests $S_1, S_2, \ldots, S_T$ is r-bounded iff $|S_t| \leq r,$ for all t.
- Reminder: $\sum_{e \in S_t} A^t_{e1} = 1 \quad t = 1, \ldots, T$.

The Deterministic Rounding Algorithm for MTF

- Sequence of requests $S_1, S_2, \ldots, S_T$ is r-bounded iff $|S_t| \leq r,$ for all t.
- Reminder: $\sum_{e \in S_t} A^t_{e1} = 1 \quad t = 1, \ldots, T$.
- There is at least an element e with $A_{e1} \geq 1/r$.

The Deterministic Rounding Algorithm for MTF

- Sequence of requests $S_1, S_2, \ldots, S_T$ is r-bounded iff $|S_t| \leq r$, for all t.
- Reminder: $\sum_{e \in S_t} A^t_{e1} = 1 \quad t = 1, \ldots, T$.
- There is at least an element e with $A_{e1} \geq 1/r$.

Deterministic Rounding Algorithm for r-bounded sequences

Input: A request sequence $R_1, \ldots, R_T$ with $|R_t| \leq r$ and an initial permutation π^0.
Output: A sequence of permutations $\pi^1, \ldots, \pi^T$.

1: Find the optimal solution $A^0 = \pi^0, A^1, \ldots, A^T$ for $\mathrm{Fractional-MTF}$.
2: **for** $t = 1 \ldots T$ **do**
3: $\quad \pi^t :=$ in π^{t-1}, move to the first position an element $e \in R_t$ such that $A^t_{e1} \geq 1/r$
4: **end for**

Analysis of Deterministic Rounding Algorithm 20

Analysis of Deterministic Rounding Algorithm 20

Theorem - Approximation Ratio

Algorithm 20 is a $O(r^2)$-approximation algorithm for $\mathrm{Mult\text{-}MSSC}$.

Analysis of Deterministic Rounding Algorithm 20

Theorem - Approximation Ratio

Algorithm 20 is a $O(r^2)$-approximation algorithm for $\mathrm{Mult\text{-}MSSC}$.

Proof Sketch: We can prove that

Analysis of Deterministic Rounding Algorithm 20

Theorem - Approximation Ratio

Algorithm 20 is a $O(r^2)$-approximation algorithm for $\mathrm{Mult\text{-}MSSC}$.

Proof Sketch: We can prove that

$$\sum_{t=1}^{T} \underbrace{d_{KT}(\pi^t, \pi^{t-1})}_{\text{Moving Cost of Permutations Produced}} \leq 2r^2 \cdot \underbrace{d_{FR}(A^t, A^{t-1})}_{\text{Moving Cost of Matrices}}$$

.

Analysis of Deterministic Rounding Algorithm 20

Theorem - Approximation Ratio

Algorithm 20 is a $O(r^2)$-approximation algorithm for $\mathrm{Mult\text{-}MSSC}$.

Proof Sketch: We can prove that

$$\sum_{t=1}^{T} \underbrace{d_{KT}(\pi^t, \pi^{t-1})}_{\text{Moving Cost of Permutations Produced}} \leq 2r^2 \cdot \underbrace{d_{FR}(A^t, A^{t-1})}_{\text{Moving Cost of Matrices}}$$

.

- Prove $O(r^2)$-approximation if matrices A^t are semi-integral.

Analysis of Deterministic Rounding Algorithm 20

Theorem - Approximation Ratio

Algorithm 20 is a $O(r^2)$-approximation algorithm for $\mathrm{Mult\text{-}MSSC}$.

Proof Sketch: We can prove that

$$\sum_{t=1}^{T} \underbrace{d_{KT}(\pi^t, \pi^{t-1})}_{\text{Moving Cost of Permutations Produced}} \leq 2r^2 \cdot \underbrace{d_{FR}(A^t, A^{t-1})}_{\text{Moving Cost of Matrices}}$$

.

- Prove $O(r^2)$-approximation if matrices A^t are semi-integral.
- Prove that we can transform any sequence of matrices to semi-integral with the same moving cost.

Concluding Remarks

Concluding Remarks

- We presented the results of [5] examining the approximability of Mult-MSSC.

Concluding Remarks

- We presented the results of [5] examining the approximability of $\mathrm{Mult\text{-}MSSC}$.
- Bridging the quadratic gap between lower and upper bounds?

Concluding Remarks

- We presented the results of [5] examining the approximability of Mult-MSSC.
- Bridging the quadratic gap between lower and upper bounds?
- Maybe we can prove $O(log^2 n)$ and $O(r^2)$ lower bounds?

Concluding Remarks

- We presented the results of [5] examining the approximability of Mult-MSSC.
- Bridging the quadratic gap between lower and upper bounds?
- Maybe we can prove $O(log^2 n)$ and $O(r^2)$ lower bounds?
- Can we use MTF and the rounding schemes to attack the Online Version of the Problem?

Concluding Remarks

- We presented the results of [5] examining the approximability of Mult-MSSC.
- Bridging the quadratic gap between lower and upper bounds?
- Maybe we can prove $O(log^2 n)$ and $O(r^2)$ lower bounds?
- Can we use MTF and the rounding schemes to attack the Online Version of the Problem?
- Our paper can be found here.

Concluding Remarks

- We presented the results of [5] examining the approximability of Mult-MSSC.
- Bridging the quadratic gap between lower and upper bounds?
- Maybe we can prove $O(log^2 n)$ and $O(r^2)$ lower bounds?
- Can we use MTF and the rounding schemes to attack the Online Version of the Problem?
- Our paper can be found here.
- You can find the implementation of our algorithms and relevant experiments in this github repo.

Yossi Azar, Iftah Gamzu, and Xiaoxin Yin.
Multiple intents re-ranking.
In *STOC*, pages 669–678, 2009.

Nikhil Bansal, Anupam Gupta, and Ravishankar Krishnaswamy.
A constant factor approximation algorithm for generalized min-sum set cover.
In *SODA*, pages 1539–1545, 2010.

Uriel Feige, László Lovász, and Prasad Tetali.
Approximating min-sum set cover.
In *Approximation Algorithms for Combinatorial Optimization, 5th International Workshop, APPROX 2002, Rome, Italy, September 17-21, 2002, Proceedings*, pages 94–107, 2002.
doi:10.1007/3-540-45753-4_10.

Dimitris Fotakis, Loukas Kavouras, Grigorios Koumoutsos, Stratis Skoulakis, and Manolis Vardas.
The online min-sum set cover problem.

In *Proc. of the 47th International Colloquium on Automata, Languages and Programming (ICALP 2020)*, LIPIcs, 2020.

Dimitris Fotakis, Panagiotis Kostopanagiotis, Vasileios Nakos, Georgios Piliouras, and Stratis Skoulakis.
On the Approximability of Multistage Min-Sum Set Cover.
In Nikhil Bansal, Emanuela Merelli, and James Worrell, editors, *48th International Colloquium on Automata, Languages, and Programming (ICALP 2021)*, volume 198 of *Leibniz International Proceedings in Informatics (LIPIcs)*, pages 65:1–65:19, Dagstuhl, Germany, 2021. Schloss Dagstuhl – Leibniz-Zentrum für Informatik.
URL:
`https://drops.dagstuhl.de/opus/volltexte/2021/14134`,
`doi:10.4230/LIPIcs.ICALP.2021.65`.

Martin Skutella and David P. Williamson.
A note on the generalized min-sum set cover problem.
Oper. Res. Lett., 39(6):433–436, 2011.