

Preemptive and Non-Preemptive Generalized Min Sum Set Cover

Sungjin Im¹, Maxim Sviridenko², and Ruben van der Zwaan³

- 1 Department of Computer Science, University of Illinois, 201 N. Goodwin Ave., Urbana, IL 61801, USA.
im3@illinois.edu
- 2 IBM T.J. Watson Research Center, P.O. Box 218, Yorktown Heights, NY 10598, USA.
sviri@us.ibm.com
- 3 Department of Quantitative Economics, Maastricht University, The Netherlands.
r.vanderzwaan@maastrichtuniversity.nl

Abstract

In the (non-preemptive) Generalized Min Sum Set Cover Problem, we are given n ground elements and a collection of sets $\mathcal{S} = \{S_1, S_2, \dots, S_m\}$ where each set $S_i \in 2^{[n]}$ has a positive requirement $\kappa(S_i)$ that has to be fulfilled. We would like to order all elements to minimize the total (weighted) cover time of all sets. The cover time of a set S_i is defined as the first index j in the ordering such that the first j elements in the ordering contain $\kappa(S_i)$ elements in S_i . This problem was introduced by [1] with interesting motivations in web page ranking and broadcast scheduling. For this problem, constant approximations are known [2, 15].

We study the version where preemption is allowed. The difference is that elements can be fractionally scheduled and a set S is covered in the moment when $\kappa(S)$ amount of elements in S are scheduled. We give a 2-approximation for this preemptive problem. Our linear programming and analysis are completely different from [2, 15]. We also show that any preemptive solution can be transformed into a non-preemptive one by losing a factor of 6.2 in the objective function. As a byproduct, we obtain an improved 12.4-approximation for the non-preemptive problem.

1998 ACM Subject Classification F.2.2. Nonnumerical Algorithms and Problems

Keywords and phrases Set Cover, Approximation, Preemption, Latency, Average cover time

Digital Object Identifier 10.4230/LIPIcs.STACS.2012.465

1 Introduction

The Min Sum Set Cover problem is a minimum latency version of the hitting set problem. We are given as input n elements, $\{1, 2, \dots, n\} = [n]$ and a collection of sets $\mathcal{S} = \{S_1, S_2, \dots, S_m\}$ where each set $S_i \in 2^{[n]}$. The goal is to find a permutation of the elements such that the total sum of (or equivalently average) cover/hitting times of all sets is minimized. For simplicity, we will say that an element e is covered at time slot t or it has cover time $\text{cov}(e) = t$ if it is placed in the t -th position in the permutation. The cover time $\text{cov}(S_i)$ of a set S_i is defined as $\min_{e \in S_i} \text{cov}(e)$ and the goal is to minimize $\sum_{S_i \in \mathcal{S}} \text{cov}(S_i)$. For this problem, a simple greedy algorithm is known to achieve an approximation factor 4 [4, 8]. The greedy algorithm iteratively picks the element that hits the most sets that are not yet hit. Also it is known that the problem cannot be approximated within a factor of $4 - \epsilon$ for any $\epsilon > 0$ unless $P = NP$ [8]. A closely related problem known as Min Sum Coloring was studied before in



© Sungjin Im, Maxim Sviridenko and Ruben van der Zwaan;
licensed under Creative Commons License NC-ND
29th Symposium on Theoretical Aspects of Computer Science (STACS'12).
Editors: Christoph Dürr, Thomas Wilke; pp. 465–476



Leibniz International Proceedings in Informatics
Schloss Dagstuhl – Leibniz-Zentrum für Informatik, Dagstuhl Publishing, Germany



SYMPOSIUM
ON THEORETICAL
ASPECTS
OF COMPUTER
SCIENCE

[4, 5] with applications in scheduling. Also the special case of the Min Sum Vertex Cover was used in [6] as a heuristic for speeding up a solver for semidefinite programs.

The Min Latency Set Cover problem is a variant where the cover time is defined as the time where all elements in the set are covered e.g. $\text{cov}(S_i) = \max_{e \in S_i} \text{cov}(e)$. This problem is in fact equivalent to the precedence-constrained scheduling on a single machine [16], for which various 2-approximation algorithms are known [10, 7, 11]. It was shown that, assuming a variant of the Unique Games Conjecture, unless $P=NP$ there is no $2 - \epsilon$ approximation for any $\epsilon > 0$ [3].

A generalization of the aforementioned problems was introduced by Azar, Gamzu and Yin [1] to provide a better framework for ranking web pages in response to queries that could have multiple intentions. This generalized problem was later named Generalized Min Sum Set Cover [2], and can be stated as follows. Every set S_i has a requirement $\kappa(S_i) \in \{1, 2, \dots, |S_i|\} = [|S_i|]$. For a permutation of the ground set we define $\text{cov}(e)$ as before and S_i is covered at time t if t is the earliest time such that $|\{e \in S_i : \text{cov}(e) \leq t\}| \geq \kappa(S_i)$. Again, the goal is to find a permutation of the elements in $[n]$ minimizing $\sum_{S_i \in \mathcal{S}} \text{cov}(S_i)$. Azar et al. [1] give a modified greedy algorithm that has a performance guarantee of $O(\ln(\max_{S_i \in \mathcal{S}} \kappa(S_i)))$. The question whether there exists an $O(1)$ -approximation was answered affirmatively by Bansal, Gupta and Krishnaswamy [2]. In order to obtain an $O(1)$ -approximation, they used a time indexed linear program together with knapsack cover inequalities and gave a clever randomized rounding scheme. Very recently, their approximation ratio of 485 was improved by Skutella and Williamson to 28 via the same LP but a different rounding scheme [15].

In this paper we study the Preemptive Generalized Min Sum Set Cover. Like the Generalized Min Sum Set Cover problem, when $\kappa(S) = |S|$ for all $S \in \mathcal{S}$ it is a special case (and in fact is equivalent to) single machine scheduling problem with precedence constraints and preemptions: $1|prec, pmtn| \sum w_j C_j$. It is known that preemption does not improve the solution quality for this problem (shown by a simple exchange argument), i.e. the optimal preemptive and non-preemptive schedules have the same optimal value. Hence it follows that there is no $2 - \epsilon$ approximation for any $\epsilon > 0$ assuming a variant of the Unique Games Conjecture and $P \neq NP$ [3].

Preemptive Generalized Min Sum Set Cover is formally defined as follows. Given the ground set of elements $[n]$, sets $\mathcal{S} = \{S_1, S_2, \dots, S_m\}$ and requirement $\kappa(U) \in [|U|]$ for each set $U \in \mathcal{S}$, we should fractionally assign elements of the ground set to the interval $[0, n]$. Formally, we define functions $x_e(t) : [0, n] \rightarrow \{0, 1\}$ where $x_e(t)$ is the indicator function that denotes whether element e is scheduled at time t such that $\int_{t=0}^n x_e(t) dt = 1$ for all $e \in [n]$ and $\sum_{e \in [n]} x_e(t) = 1$ for any time $t \in [0, n]$. Then, the cover time $\text{cov}(S)$ of the set S is defined as the earliest time t such that $\int_{\tau=0}^t \sum_{e \in S} x_e(\tau) d\tau \geq \kappa(S)$ and the goal is to minimize the sum of cover times over all sets. Note that the cover time $\text{cov}(S)$ is not necessarily an integer unlike in the non-preemptive problem.

Our main motivation to study Preemptive Generalized Min Sum Set Cover is the fact that it provides a lower bound for the optimal value of the Generalized Min Sum Set Cover. We decouple finding an approximate solution to the relaxed problem (see Section 2) and the question of the lower bound quality (see Section 3 and Conjecture 1).

Our Results

Our main result is a polynomial time approximation algorithm with performance guarantee of 2 for the Preemptive Generalized Min Sum Set Cover. As we noticed before this result is tight modulo some complexity assumptions [3]. We note that one can easily show that the linear program used in [2, 15] is a valid relaxation for the preemptive problem, thus the

best known approximation for the non-preemptive problem also carries for the preemptive problem as well.

We introduce a *configuration linear program* which completely differs from the linear programming relaxation used in [2, 15]. Interestingly, it is not obvious that our new linear program is a valid relaxation for the preemptive problem, unlike the previous linear program in [2, 15] which can be easily shown to be a valid relaxation for the preemptive (and non-preemptive) problem. Our new LP is provably stronger than the previous LP, for both the preemptive and non-preemptive problems.

Further, we study the “gap” between the preemptive and non-preemptive solutions of the Generalized Min Sum Set Cover Problem, which is of independent interest. With some modifications of the rounding scheme in [15], we show that one can transform any α -approximate preemptive schedule into 6.2α -approximate non-preemptive one. With this transformation, we obtain an 12.4-approximation for the non-preemptive Generalized Min Sum Set Cover Problem, improving upon the previous best 28-approximation by Skutella and Williamson[15]. We conjecture that the gap between optimal preemptive and non-preemptive solutions is precisely two.

All our proofs easily extend to the case where every set S_i has a non-negative weight $w_i \geq 0$ and the objective is to minimize $\sum_{S_i \in \mathcal{S}} w_i \cdot \text{cov}(S_i)$.

Organization

The remainder of this paper is organized as follows. In Section 2 we introduce the configuration linear program $\text{LP}_{\text{primal}}$. First, we prove that our configuration linear program is a valid relaxation for Preemptive Generalized Min Sum Set Cover and that this linear program can be solved in polynomial time. Finally, we design a rounding procedure that results in a randomized 2-approximation (Section 2.4) that can be derandomized. In Section 3 we obtain a transformation from a preemptive schedule to a non-preemptive schedule with a loss of factor 6.2, which immediately implies a 12.4-approximation in expectation to Generalized Min Sum Set Cover. In Section 4 we compare the time indexed linear program in [2, 15] to our own configuration linear program and show our linear program is stronger. Due to space constraints, we omit most parts from Section 3 and 4. We will include the full details and omitted proofs in the full version of this paper.

2 2-Approximation for Preemptive Generalized Min Sum Set Cover

This section is devoted to prove the following theorem.

► **Theorem 1.** *There is a randomized polynomial time 2-approximation algorithm for Preemptive Generalized Min Sum Set Cover.*

Throughout this section, for any integer $t \in [n]$, the t -th time slot will be equivalent to the time interval $[t - 1, t]$.

2.1 Configuration LP

We write a configuration linear program. For a set $S \in \mathcal{S}$, a valid configuration is an (integral) assignment of elements in S to time slots. More formally, such a map can be described as an injective function $f_S : S \rightarrow [n]$. For notational simplicity, we may represent the mapping via a relation $F =_{\text{def}} \{(e, f_S(e)) \mid e \in S\}$. Let $\mathcal{F}(S)$ denote the collection of all possible configurations for set S . Let C_S^F denote the completion time t of set S under the configuration

F , i.e. the first time t' such that $|f_S^{-1}([t'])| \geq \kappa(S)$. Let $x_{e,t}$ denote the fraction of element e we schedule in the t -th time slot. The variable y_S^F is used to indicate which configurations S adheres to. For example, if $y_S^F = 1$, it means all elements in S are scheduled following the configuration F .

Our linear program is formulated as follows.

$$\begin{aligned} \min \quad & \sum_{S \in \mathcal{S}} \sum_{F \in \mathcal{F}(S)} C_S^F y_S^F & (\text{ILP}) \\ \text{s.t.} \quad & \sum_e x_{e,t} = 1 & \forall t \in [n] & (1) \\ & \sum_t x_{e,t} = 1 & \forall e \in [n] & (2) \\ & \sum_{F \in \mathcal{F}(S)} y_S^F = 1 & \forall S \in \mathcal{S} & (3) \\ & \sum_{F \in \mathcal{F}(S), (e,t) \in F} y_S^F = x_{e,t} & \forall e, t \in [n], S : e \in S & (4) \\ & x_{e,t} \in \{0, 1\} & \forall e, t \in [n] \\ & y_S^F \in \{0, 1\} & \forall S \in \mathcal{S}, F \in \mathcal{F}(S) \end{aligned}$$

The constraints (1) and (2) enforce that exactly one element is scheduled at any time slot and that an element can be scheduled only once over all times. The constraint (3) states that each set S has a unique configuration. Finally, (4) says that if an element e is scheduled at time t , then it must align with the configuration of S .

The relaxation $\text{LP}_{\text{primal}}$ of ILP is then defined as follows.

$$\begin{aligned} \min \quad & \sum_{S \in \mathcal{S}} \sum_{F \in \mathcal{F}(S)} C_S^F y_S^F & (\text{LP}_{\text{primal}}) \\ \text{s.t.} \quad & \text{Constraints (1),(2),(3) and (4) hold} \\ & x_{e,t} \geq 0 & \forall e, t \in [n] \\ & y_S^F \geq 0 & \forall S \in \mathcal{S}, F \in \mathcal{F}(S) \end{aligned}$$

2.2 Validity of the LP

It is easy to verify that $\text{LP}_{\text{primal}}$ is a valid linear programming relaxation for Generalized Min Sum Set Cover. However, it is not obvious that the $\text{LP}_{\text{primal}}$ is indeed a valid relaxation for the preemptive problem. Since we will use two different types of fractional schedules throughout the analysis, we first clearly define/remind those schedules. The first one is a *continuous* schedule that is defined by indicator functions $x_e(t) : [0, n] \rightarrow \{0, 1\}, e \in [n]$ such that (1) for any $t \in [0, n], \sum_{e \in [n]} x_e(t) = 1$ and (2) for any $e \in [n], \int_{\tau=0}^n x_e(\tau) d\tau = 1$. We say that $x_e(t), e \in [n]$ is a feasible schedule if all these conditions are satisfied. Recall that the completion time C_S of each set S is defined by a continuous schedule as the earliest time t such that $\int_{\tau=0}^t \sum_{e \in S} x_e(\tau) d\tau \geq \kappa(S)$. The other version of schedule, which is somewhat *discretized*, is defined by $x_{e,t}, e, t \in [n]$ that satisfy (1) $\sum_{e \in [n]} x_{e,t} = 1$, (2) $\sum_{t \in [n]} x_{e,t} = 1$ and (3) $0 \leq x_{e,t} \leq 1$ for any $e, t \in [n]$. When these conditions are satisfied, we will say $x_{e,t}, e, t \in [n]$ is feasible. Note that this discretized version of schedule does not immediately define the completion time of sets. Rather, it is used in $\text{LP}_{\text{primal}}$ as a relaxation of continuous schedules. We show the following theorem.

► **Theorem 2.** Consider any feasible continuous schedule $x_e(t), e \in [n]$. Let C_S denote the completion time of S in this schedule. For any $e, t \in [n]$, let $x_{e,t} =_{def} \int_{\tau=t-1}^t x_e(\tau) d\tau$. Then $x_{e,t}$ satisfy constraints (1) and (2). Also there exists y -values that satisfy the other constraints (3) and (4) as well and further satisfy

$$\sum_{S \in \mathcal{S}} \sum_{F \in \mathcal{F}(S)} C_S^F y_S^F \leq \sum_{S \in \mathcal{S}} C_S. \quad (5)$$

The first claim in Theorem 2 that $x_{e,t}$ satisfy constraints (1) and (2) easily follows from the property of continuous schedules and from how $x_{e,t}$ are defined. Due to the space constraints, we defer the proof to the full version of this paper. In fact, it is not difficult to see that there exist y -values that satisfy all constraints (1)-(4). However, we can find an example of y -values satisfying all the constraints but not satisfying the inequality (5) (See full version of this paper). Henceforth, we focus on showing that there exist “good” y -values that also satisfy (5). We will show how to construct a feasible solution y such that the inequality

$$\sum_{F \in \mathcal{F}(S)} C_S^F y_S^F \leq C_S \quad (6)$$

holds for any set $S \in \mathcal{S}$ which will imply the inequality (5). Since setting y_S -values for a specific S does not affect other y -values, we can focus on each $S \in \mathcal{S}$. We will find “good” y_S^F -values that satisfy constraints (3) and (4), and further (6).

To this end, we define two matroids M_1 and M_2 that enforce that any independent set in the intersection of M_1 and M_2 which corresponds to a feasible configuration $F \in \mathcal{F}(S)$. Then we show that the vector $x_{e,t}, e \in S, t \in [n]$ lies in the intersection of the polytopes of the two matroids. Using the fact that such an intersection polytope is integral, we will be able to decompose x into a convex combination of integer points that lie in the intersection of the polytopes of M_1 and M_2 . As already mentioned, due to the structure of the matroids, each integer point will correspond to a configuration $F \in \mathcal{F}(S)$. By setting y -values as suggested by the decomposition, we will guarantee that y satisfy constraints (3) and (4). Finally, we will complete the analysis by showing that such y -values satisfy (6) as well. This is enabled by some additional constraints we impose on the matroids. We refer the reader to Chapters 39-41 in [13] for the extensive overview of algorithmic matroid theory.

We begin with defining each of the two matroids M_1 and M_2 which have the same common ground set, $U = \{(e, t) \mid e \in S, t \in [n]\}$ (Recall that we are focusing on each fixed $S \in \mathcal{S}$ separately). We will call (e, t) a pair in order to distinguish it from elements, $[n]$. The first matroid $M_1 = (U, \mathcal{I}(M_1))$ enforces that each element in S can be scheduled in at most one time slot. Formally, the collection $\mathcal{I}(M_1)$ of independent sets of M_1 is defined as follows: $A \in \mathcal{I}(M_1)$ if and only if for any $e \in S$, $|A \cap \{(e, t) \mid t \in [n]\}| \leq 1$. Observe that M_1 is a partition matroid since pairs in U are partitioned based on each common element, and any independent set collects at most one pair from each group. Hence the polytope $P(M_1)$ of M_1 (polymatroid) is defined as follows.

$$\begin{aligned} \sum_{t \in [n]} x_{e,t} &\leq 1 && \forall e \in S && (P(M_1)) \\ x_{e,t} &\geq 0 && \forall e \in S, t \in [n] \end{aligned}$$

► **Proposition 3.** The vector $x = (x_{e,t}), e \in S, t \in [n]$ is in the polytope $P(M_1)$. Moreover, $\sum_{e \in S, t \in [n]} x_{e,t} = |S|$, i.e. x belongs to the base polymatroid of M_1 .

The second matroid $M_2 = (U, \mathcal{I}(M_2))$ has a more involved structure. It enforces that in each time slot, at most one element in S can be scheduled. Additionally, it enforces that at

most $\kappa(S)$ elements can be scheduled during the first $C - 1$ time slots and at most $|S| - \kappa(S)$ elements can be scheduled during the time slots, $C + 1, C + 2, \dots, n$, where C is an integer such that $C - 1 < C_S \leq C$. This additional constraints will be crucial in finding “good” y -values. Formally, $A \in \mathcal{I}(M_2)$ if and only if A satisfies

- For each integer time $t \in [n]$, $|A \cap \{(e, t) \mid e \in S\}| \leq 1$.
- $|A \cap \{(e, t) \mid e \in S, 1 \leq t \leq C - 1\}| \leq \kappa(S)$.
- $|A \cap \{(e, t) \mid e \in S, C + 1 \leq t \leq n\}| \leq |S| - \kappa(S)$.

We observe that $\mathcal{I}(M_2)$ is a laminar matroid: All pairs in U are partitioned into groups with the same time t , and at most one pair can be chosen from each group to be in an independent set. Further, the second and third constraints put a limit on the number of pairs that can be chosen from the groups of time slots $t = 1, 2, \dots, C - 1$ and from the groups of time slots $t = C + 1, C + 2, \dots, n$, respectively. We define the polymatroid $P(M_2)$ as follows.

$$\begin{aligned} \sum_{e \in S} x_{e,t} &\leq 1 && \forall t \in [n] && (P(M_2)) \\ \sum_{t=1}^{C-1} \sum_{e \in S} x_{e,t} &\leq \kappa(S) \\ \sum_{t=C+1}^n \sum_{e \in S} x_{e,t} &\leq |S| - \kappa(S) \\ x_{e,t} &\geq 0 && \forall e \in S, t \in [n] \end{aligned}$$

► **Proposition 4.** The vector $x = (x_{e,t})$ lies in the polymatroid $P(M_2)$.

It is well known the the intersection of two polymatroids is an integral polytope, i.e. any vertex point is integral. Hence since $(x_{e,t})$ lies in the intersection of two polytopes $P(M_1)$ and $P(M_2)$, it can be decomposed into a linear combination of vertex (hence integer) points in $P(M_1) \cap P(M_2)$. Note that each of such integer points corresponds to an independent set in $\mathcal{I}(M_1) \cap \mathcal{I}(M_2)$, which is of size at most $|S|$ due to the constraints of M_1 . In fact, the size must be exactly $|S|$, since $\sum_{e \in S} \sum_{t \in [n]} x_{e,t} = |S|$. By the constraints of M_1 and the first constraints of M_2 , we conclude that each of such integer points corresponds to a configuration $F \in \mathcal{F}(S)$. Hence we have shown the following lemma.

► **Lemma 5.** *There exist $\mathcal{F}'(S) \subseteq \mathcal{F}(S)$ and positive constants $\theta_S^F, F \in \mathcal{F}'(S)$ that satisfy*

- $\sum_{F \in \mathcal{F}'(S)} \theta_S^F = 1$.
 - For any $e \in S, t \in [n]$, $x_{e,t} = \sum_{F \in \mathcal{F}'(S)} \theta_S^F \cdot \mathbf{1}[(e, t) \in F]$.
- where an indicator variable $\mathbf{1}[(e, t) \in F] = 1$ if and only if $(e, t) \in F$.

We let $y_S^F = \theta_S^F$ for all $F \in \mathcal{F}'(S)$ and $y_S^F = 0$ for all $F \in \mathcal{F}(S) \setminus \mathcal{F}'(S)$. Note that x and y satisfy constraints (3) and (4).

It remains to show that y satisfy (6). Now the second and third constraints of M_2 play a crucial role. We make the following observation.

► **Lemma 6.** *For any $F \in \mathcal{F}'(S)$ exactly one of the following holds.*

- $|F \cap \{(e, t) \mid e \in S, 1 \leq t \leq C - 1\}| = \kappa(S)$.
- $|F \cap \{(e, t) \mid e \in S, 1 \leq t \leq C - 1\}| = \kappa(S) - 1$ and $(e, C) \in F$ for some $e \in S$.

Proof. Recall that $|F| = |S|$. By the third constraints of M_2 , we know that $N_{\geq C+1} =_{def} |F \cap \{(e, t) \mid e \in S, C + 1 \leq t \leq n\}| \leq |S| - \kappa(S)$, hence that $N_{\geq C} =_{def} |F \cap \{(e, t) \mid e \in S, C \leq t \leq n\}| \leq |S| - \kappa(S) + 1$. Therefore, we have $N_{\leq C-1} =_{def} |F \cap \{(e, t) \mid e \in S, 1 \leq t \leq C - 1\}| \geq \kappa(S) - 1$.

$t \leq C - 1$ } $\geq \kappa(S) - 1$. Further, we know $N_{\leq C-1} \leq \kappa(S)$ from the second constraint of M_2 . Thus unless $N_{\leq C-1} = \kappa(S)$, it must be the case that $N_{\leq C-1} = \kappa(S) - 1$. In that case, since $N_{\geq C+1} \leq |S| - \kappa(S)$, we conclude that $(e, C) \in F$ for some $e \in S$. ◀

Motivated by the above lemma, we can now prove that our linear program is a valid relaxation for the preemptive version of the problem.

Proof of Theorem 2. Partition $\mathcal{F}'(S)$ into $\mathcal{F}'_1(S)$ and $\mathcal{F}'_2(S)$ by letting $\mathcal{F}'_1(S)$ to denote all $F \in \mathcal{F}'(S)$ that fall in the first case in the Lemma 6 and letting $\mathcal{F}'_2(S) = \mathcal{F}'(S) \setminus \mathcal{F}'_1(S)$. Let $\theta' = \sum_{F \in \mathcal{F}'_2(S)} \theta_S^F$. Note that for any $F \in \mathcal{F}'_1(S)$, $C_S^F \leq C - 1$ and for any $F \in \mathcal{F}'_2(S)$, $C_S^F = C$. In words, the set S is completed no later than time $C - 1$ for $(1 - \theta')$ fraction of configurations in $\mathcal{F}'(S)$ and exactly at time C for θ' fraction of configurations in $\mathcal{F}'(S)$. Hence we have that

$$\begin{aligned} \sum_{F \in \mathcal{F}'(S)} C_S^F y_S^F &= \sum_{F \in \mathcal{F}'_1(S)} C_S^F \theta_S^F + \sum_{F \in \mathcal{F}'_2(S)} C_S^F \theta_S^F \\ &\leq (1 - \theta')(C - 1) + \theta' C = C - 1 + \theta' \end{aligned} \quad (7)$$

Now we focus on upper-bounding θ' . From the definition of C_S and the fact that $\sum_{e \in S} x_e(\tau) \leq 1$ for any τ , we know that

$$\begin{aligned} \int_{\tau=0}^{C-1} \sum_{e \in S} x_e(\tau) \, d\tau &= \int_{\tau=0}^{C_S} \sum_{e \in S} x_e(\tau) \, d\tau - \int_{\tau=C-1}^{C_S} \sum_{e \in S} x_e(\tau) \, d\tau \\ &\geq \kappa(S) - (C_S - (C - 1)) \end{aligned} \quad (8)$$

On the other hand, it follows that

$$\begin{aligned} &\int_{\tau=0}^{C-1} \sum_{e \in S} x_e(\tau) \, d\tau = \sum_{t=1}^{C-1} \sum_{e \in S} x_{e,t} \quad [\text{By the definition of } x_{e,t}] \\ &= \sum_{t=1}^{C-1} \sum_{e \in S} \sum_{F \in \mathcal{F}'(S)} y_S^F \quad [\text{From the decomposition of } x \text{ into } y_S^F] \\ &= \sum_{F \in \mathcal{F}'_1(S)} y_S^F \sum_{e \in S} \sum_{t=1}^{C-1} \mathbf{1}[(e, t) \in F] + \sum_{F \in \mathcal{F}'_2(S)} y_S^F \sum_{e \in S} \sum_{t=1}^{C-1} \mathbf{1}[(e, t) \in F] \\ &= \sum_{F \in \mathcal{F}'_1(S)} \theta_S^F \cdot \kappa(S) + \sum_{F \in \mathcal{F}'_2(S)} \theta_S^F \cdot (\kappa(S) - 1) \\ &= (1 - \theta') \cdot \kappa(S) + \theta' \cdot (\kappa(S) - 1) = \kappa(S) - \theta' \end{aligned} \quad (9)$$

From (8) and (9), we have $\theta' \leq C_S - (C - 1)$. By combining this with (7), we complete the proof of Theorem 2. ◀

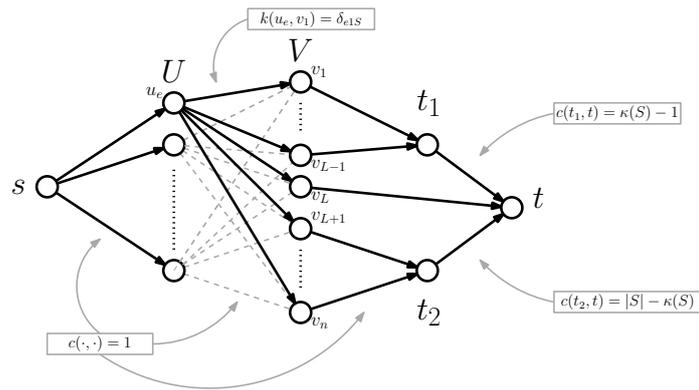
2.3 Solving the LP

The linear programming relaxation $\text{LP}_{\text{primal}}$ has exponentially many variables. Hence, we solve the dual LP and show there are only polynomially many non-zero variables in the primal LP that achieve the optimal LP value. The dual LP is as follows.

$$\max \sum_{t \in [n]} \alpha_t + \sum_{e \in [n]} \beta_e + \sum_{S \in \mathcal{S}} \gamma_S \quad (\text{LP}_{\text{dual}})$$

$$\text{s.t.} \quad \alpha_t + \beta_e - \sum_{S: e \in S} \delta_{eS} \leq 0 \quad \forall e, t \quad (10)$$

$$\gamma_S + \sum_{(e,t) \in F} \delta_{eS} \leq C_S^F \quad \forall S \in \mathcal{S}, F \in \mathcal{F}(S) \quad (11)$$



■ **Figure 1** An illustration of the construction of the graph G , in which we want to find a maximum-value flow.

To solve LP_{dual} with the ellipsoid algorithm, we need a *separation oracle* for finding a violated constraint (see [9]). Since constraints (10) are easy to verify (there are only n^2 of them), we focus on constraints (11). We need a polynomial time algorithm that given γ_S and δ_{eS} -values, finds (if any) $S \in \mathcal{S}$ and $F \in \mathcal{F}(S)$ that violate constraints (11).

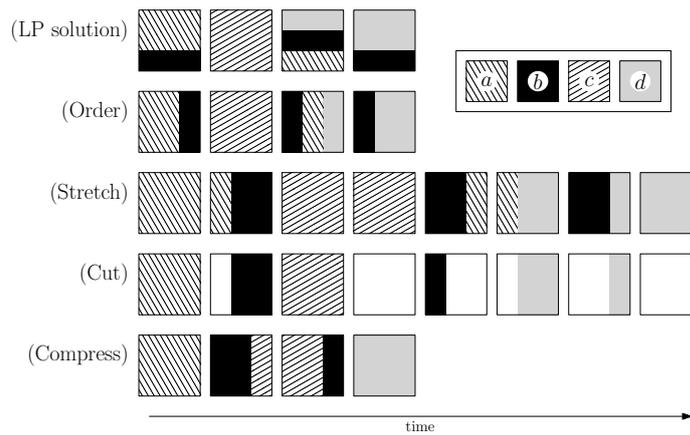
We model this problem as a classical minimum cost s-t flow problem. In this problem, we are given a digraph $G = (V, A)$, a capacity function $c : A \rightarrow \mathbb{Q}_+$, a cost function $k : A \rightarrow \mathbb{Q}$ and the volume $\phi \in \mathbb{Q}_+$. The goal is to send ϕ amount of flow from the source s to the sink t , i.e. to find an s-t flow f of volume ϕ , subject to capacity constraints $0 \leq f(e) \leq c(e)$ for all $e \in A$ and the standard flow conservation constraints, minimizing the costs $\sum_{e \in A} f(e)k(e)$.

It is known that if the volume ϕ and capacities $c_e, e \in E$ are integral then we can test in polynomial time if there is an s-t flow of volume ϕ . Moreover, if there is such a flow (i.e. there is a feasible solution to the problem) then there is an integral minimum-cost s-t flow, and it can be found in polynomial time (see Chapter 12 in [13]).

We now show how to reduce our separation problem for constraints (11) to the minimum cost s-t flow problem. It will be convenient for us to consider an equivalent maximum cost s-t flow problem where the goal is to maximize the value of the objective function $\sum_{e \in A} f(e)k(e)$.

Fix a set S and an integer $L \in [n]$. We will try to find a violated constraint for the constraints (11) corresponding to the set S and configurations $F \in \mathcal{F}(S)$ with $C_S^F = L$. Create a directed complete bipartite graph $G_L = (U, V, A)$ where part U has vertex u_e for each $e \in S$, part V has vertex v_i for each time slot $i \in [n]$. Arc $a = (u_e, v_i) \in A$ has cost $k(e) = \delta_{eiS}$ and capacity $c(e) = 1$. We augment G_L as follows. We add a source vertex s and connect it to all vertices in U . There are two “intermediate” sinks t_1 and t_2 , both connected to the “final” sink t . The vertices v_1, v_2, \dots, v_{L-1} in V are connected to t_1 and the vertices $v_{L+1}, v_{L+2}, \dots, v_n$ in V are connected to the other intermediate sink t_2 . The arcs a between the source s and part U have cost $k(a) = 0$ and capacity $c(a) = 1$. Analogously, all arcs a between part V and intermediate sinks t_1 and t_2 have cost $k(a) = 0$ and capacity $c(a) = 1$. Arcs $a' = (t_1, t)$ and $a'' = (t_2, t)$ have capacities $c(a') = \kappa(S) - 1$ and $c(a'') = |S| - \kappa(S)$ respectively, and all of them have zero costs. The vertex v_L is special and is directly connected to t . The arc (v_L, t) has a unit capacity and zero cost. The goal is to find the s-t flow of volume $\phi = |S|$ of maximum cost. See Figure 1 for an illustration of this construction.

Note that any integral s-t flow f of value $|S|$ in digraph G_L corresponds to a valid configuration F for volume S such that $C_S^F = L$, and vice versa. Hence, if the maximum-cost s-t flow in G_L has cost more than $L - \gamma_S$, the constraint (11) is violated for S and $F \in \mathcal{F}(S)$ that corresponds to the flow. The converse also holds: if the maximum-cost s-t flow has cost less than or equal to $L - \gamma_S$ there is *no* configuration $F \in \mathcal{F}(S)$ with $C_S^F = L$ that violates



■ **Figure 2** In this example the schedule is stretched by a factor of two e.g. $\lambda = \frac{1}{2}$.

(11). With the help of this separation oracle and classical connection between separation and optimization [9], we can solve LP_{dual} in polynomial time.

Then we can optimally solve LP_{primal} by focusing only on y_S^F variables that correspond to the constraints that were considered by the ellipsoid method in solving LP_{dual} . A more formal (and well-known) argument is that the LP_{dual} with the subset of constraints considered by the ellipsoid method is a relaxation of the original problem but it has the same optimal solution. The dual of the relaxed problem is LP_{primal} restricted to the subset of corresponding variables which by the strong duality theorem has the same optimal value.

2.4 Rounding procedure

Let $x_{e,t}$ and y_S^F be a basic optimal solution of the linear programming relaxation LP_{primal} . In particular we know that there are at most $2n + m + n^2m$ non-zero variables (this is the number of constraints (1)-(4)). Let C_S^{LP} denote the completion time of set S in the LP. That is, $C_S^{LP} = \sum_{F \in \mathcal{F}(S)} C_S^F y_S^F$. We create a schedule parameterized by $\lambda \in (0, 1]$, where λ is randomly drawn from $(0, 1]$ according to the density function $f(v) = 2v$.

Create an arbitrary continuous schedule $x_e(t)$, $e \in [n]$, $t \in [0, n]$ from $x_{e,t}$, $e, t \in [n]$ such that for any $e, t \in [n]$, $\int_{\tau=t-1}^t x_e(\tau) d\tau = x_{e,t}$. For example, this can be done by processing each element e for the amount $x_{e,t}$ during the time step t in an arbitrary order between the elements to obtain $x_e(t)$. For notational convenience, let σ denote the continuous schedule $x_e(t)$. The new schedule $\sigma(\lambda)$ is defined as follows. Stretch out the schedule σ by a factor of $\frac{1}{\lambda}$. In other words, map every point τ in time onto τ/λ . For each element e define $\tau_e \in [1, n/\lambda]$ to be the earliest point in time when the element has been processed for one time unit (out of total $1/\lambda$). Leave the machine idle whenever it processes the element e after time τ_e . After repeating this procedure for all elements $e \in [n]$, we shift the whole schedule to the left to eliminate all idle times. The final schedule $\sigma(\lambda)$ has total length n . Let $x_e^{(\lambda)}(t)$, $e \in [n]$, $t \in [0, n]$ be the resulting continuous schedule $\sigma(\lambda)$. Note that similar algorithms were used in scheduling before to design approximation algorithms for various preemptive scheduling problems with total completion time objective [14, 12].

► **Example 7.** See Figure 2 for an illustration. Consider an instance with 4 elements $\{a, b, c, d\}$, with the LP solution $x_{a,1} = 2/3$, $x_{b,1} = 1/3$, $x_{c,2} = 1$, $x_{d,3} = 1/3$, $x_{b,3} = 1/3$, $x_{a,3} = 1/3$, $x_{d,4} = 2/3$, $x_{b,4} = 1/3$. Construct a continuous schedule by randomly ordering the elements in each time step. For example in time step 3, three elements, a, b, d are scheduled seamlessly, each for $1/3$ time steps. Then stretch the whole schedule by a factor two ($\lambda = 1/2$). cut out each element after being scheduled by a unit amount. Finally, compress the schedule, by shifting everything to the left removing the idle times.

Let $C_S(\lambda)$ denote the completion time of S in the new schedule $\sigma(\lambda)$. Order all configurations $F \in \mathcal{F}(S)$ for $y_S^F > 0$ in non-decreasing order of C_S^F . Let F_1, F_2, \dots, F_j be such an ordering. Define $\tilde{C}_S(\lambda) =_{def} C_S^{F_j}$ where $\sum_{i=1}^{j-1} y_S^{F_i} < \lambda$ and $\sum_{i=1}^j y_S^{F_i} \geq \lambda$. Let $\mathbf{1}[\phi]$ be an indicator function such that $\mathbf{1}[\phi] = 1$ if and only if ϕ is true and zero otherwise.

► **Lemma 8.** For any $S \in \mathcal{S}$ and $0 < \lambda \leq 1$, $C_S(\lambda) \leq \frac{1}{\lambda} \cdot \tilde{C}_S(\lambda)$.

Proof. To simplify the proof we assume that there exists j such that $\sum_{1 \leq l \leq j} y_S^{F_l} = \lambda$. Otherwise, let j be the lowest index such that $\sum_{1 \leq l \leq j} y_S^{F_l} > \lambda$, then we define two copies F_j' and F_j'' of configuration F_j , with $y_S^{F_j'} = \lambda - \sum_{1 \leq l \leq j-1} y_S^{F_l}$ and $y_S^{F_j''} = \sum_{1 \leq l \leq j} y_S^{F_l} - \lambda$. Here F_j' and F_j'' are the same configurations as F_j . Now, $\sum_{1 \leq l \leq j-1} (y_S^{F_l}) + y_S^{F_j'} = \lambda$.

We will show the following inequality:

$$\int_{\tau=0}^{\tilde{C}_S(\lambda)/\lambda} \sum_{e \in S} x_e^{(\lambda)}(\tau) \, d\tau \geq \kappa(S) \quad (12)$$

since it would imply that the completion time $C_S(\lambda)$ of the set S in the schedule $\sigma(\lambda)$ must be no later than $\tilde{C}_S(\lambda)/\lambda$. Since for every $e \in S$ we have $\int_{\tau=0}^{\tilde{C}_S(\lambda)/\lambda} x_e^{(\lambda)}(\tau) \, d\tau \geq \min \left\{ 1, \int_{\tau=0}^{\tilde{C}_S(\lambda)} x_e(t)/\lambda \, d\tau \right\} \geq \min \left\{ 1, \sum_{t \leq \lfloor \tilde{C}_S(\lambda) \rfloor} x_{e,t}/\lambda \right\}$, and $\tilde{C}_S(\lambda)$ is integral by definition for any $\lambda \in (0, 1]$, it is sufficient to show the inequality

$$\sum_{e \in S} \min \left\{ \lambda, \sum_{t \leq \tilde{C}_S(\lambda)} x_{e,t} \right\} \geq \lambda \kappa(S) \quad (13)$$

to derive (12). We now derive the inequality (13).

$$\begin{aligned} \sum_{e \in S} \min \left\{ \lambda, \sum_{t \leq \tilde{C}_S(\lambda)} x_{e,t} \right\} &\geq \sum_{e \in S} \min \left\{ \lambda, \sum_{l=1}^j y_S^{F_l} \cdot \mathbf{1}[(e, t) \in F_l \text{ for some } t \leq \tilde{C}(\lambda)] \right\} \\ &= \sum_{e \in S} \sum_{l=1}^j y_S^{F_l} \cdot \mathbf{1}[(e, t) \in F_l \text{ for some } t \leq \tilde{C}(\lambda)] \\ &= \sum_{l=1}^j y_S^{F_l} \sum_{e \in S} \mathbf{1}[(e, t) \in F_l \text{ for some } t \leq \tilde{C}(\lambda)] \\ &\geq \sum_{l=1}^j y_S^{F_l} \kappa(S) = \lambda \kappa(S) \end{aligned}$$

The first inequality follows from constraints (4). The first equality holds because $\sum_{l=1}^j y_S^{F_l} = \lambda$. The last inequality holds because for any F_l , $l \leq j$, $C_S^{F_l} \leq \tilde{C}_S(\lambda)$. ◀

The following lemma can be easily shown from the definition of $\tilde{C}_S(\lambda)$.

► **Lemma 9.** For any $S \in \mathcal{S}$, $\int_{\lambda=0}^1 \tilde{C}_S(\lambda) \, d\lambda = C_S^{LP}$.

Proof of Theorem 1. By Theorem 2, LP_{primal} is a valid relaxation, and we now show how to round to obtain a 2-approximation (in expectation).

$$\begin{aligned} \mathbb{E}[C_S(\lambda)] &= \int_{\lambda=0}^1 C_S(\lambda) \cdot 2\lambda \, d\lambda \quad [\text{By definition}] \\ &\leq \int_{\lambda=0}^1 \frac{1}{\lambda} \cdot \tilde{C}_S(\lambda) \cdot 2\lambda \, d\lambda \quad [\text{By Lemma 8}] \\ &= 2 \int_{\lambda=0}^1 \tilde{C}_S(\lambda) \, d\lambda = 2C_S^{LP} \quad [\text{By Lemma 9}] \end{aligned}$$

◀

We shortly indicate how our approximation algorithm can be derandomized. The function $\tilde{C}_S(\lambda)$ is a piecewise constant function, with at most a polynomial number of pieces since there are at most polynomially many non-zero variables y_S^F for each S . This implies that there are at most polynomially many “interesting” λ -values that we need to consider, among which at least one that gives the desired approximation ratio.

3 Gap between Preemptive and Non-preemptive Schedules

In this section, we study the lower bound quality of the preemptive problem for the non-preemptive problem. The goal is to show that there exists a small gap between the preemptive and non-preemptive solutions (schedules) of Generalized Min Sum Set Cover. Note that if we show a way to convert any given preemptive schedule into a non-preemptive one losing a factor of η , it would immediately obtain a 2η -approximation algorithm for the non-preemptive Generalized Min Sum Set Cover.

Our scheme for transforming a preemptive schedule into non-preemptive one is similar to the one by Skutella and Williamson [15]. We obtain a better gap by utilizing several additional tricks *and* starting from a preemptive schedule. Formally we prove the following theorem, of which the proof is deferred to the full version of this paper.

► **Theorem 10.** *Given a preemptive schedule with cost C , then there exists a non preemptive schedule with expected cost at most $6.2C$. Furthermore, this transformation can be done in polynomial time.*

Combining Theorem 1 and Theorem 10 we derive

► **Theorem 11.** *There exists a polynomial time 12.4 -approximation algorithm for Generalized Min Sum Set Cover.*

We have shown an upper bound on the gap of 6.2 , and any gap lower than 2 would result in an approximation factor strictly less than 4 for the non-preemptive problem, which is impossible unless $P=NP$ [8]. We believe that our gap is not tight. In fact, we make the following bold conjecture:

► **Conjecture 1.** *Given a preemptive schedule with cost C then there is a non-preemptive schedule with cost at most $2C$. Further, such a non-preemptive schedule can be found in polynomial time.*

It would be also interesting to show if the optimal gap between values of preemptive and non-preemptive schedules depends on parameter $\xi = \min_S \{\kappa(S)/|S|\}$. For example, we know if $\xi = 1$ then there is no advantage for preemptive schedules, i.e. $\eta = 1$ in this case.

4 Comparison of our LP and the previous one in [2, 15]

In this section we compare our configuration LP and the LP considered in [2, 15], which is time-indexed and based on knapsack covering inequalities. We show that our configuration LP (LP_{Primal}) is *stronger* for the non-preemptive problem than the LP (LP_{BGK}) considered in [2, 15]. We first provide an instance for which LP_{Primal} has an objective value strictly larger than LP_{BGK} . Secondly, we show that for any instance LP_{Primal} has an objective no smaller than the objective of LP_{BGK} . Together with the fact that both LPs are valid relaxations, we establish that LP_{Primal} is stronger than LP_{BGK} . We will include the proofs in the full version of this paper.

References

- 1 Yossi Azar, Iftah Gamzu, and Xiaoxin Yin. In *STOC*, pages 669–678, 2009.
- 2 Nikhil Bansal, Anupam Gupta, and Ravishankar Krishnaswamy. A constant factor approximation algorithm for generalized min-sum set cover. In *SODA*, pages 1539–1545, 2010.
- 3 Nikhil Bansal and Subhash Khot. Optimal long code test with one free bit. In *FOCS*, pages 453–462, 2009.
- 4 Amotz Bar-Noy, Mihir Bellare, Magnús M. Halldórsson, Hadas Shachnai, and Tami Tamir. On chromatic sums and distributed resource allocation. *Inf. Comput.*, 140(2):183–202, 1998.
- 5 Amotz Bar-Noy, Magnús M. Halldórsson, and Guy Kortsarz. A matched approximation bound for the sum of a greedy coloring. *Inf. Process. Lett.*, 71(3):135–140, 1999.
- 6 S. Burer and R. Monteiro. A projected gradient algorithm for solving the maxcut sdp relaxation. *Optimization Methods and Software*, 15:175–200, 2001.
- 7 Chandra Chekuri and Rajeev Motwani. Precedence constrained scheduling to minimize sum of weighted completion times on a single machine. *Discrete Applied Mathematics*, 98(1-2):29–38, 1999.
- 8 Uriel Feige, László Lovász, and Prasad Tetali. Approximating min sum set cover. *Algorithmica*, 40(4):219–234, 2004.
- 9 Martin Grottschel, László Laszlo Lovász, and Alexander Schrijver. *Geometric algorithms and combinatorial optimization. Second edition. Algorithms and Combinatorics, 2*. Springer-Verlag, Berlin, 1993.
- 10 Leslie A. Hall, Andreas S. Schulz, David B. Shmoys, and Joel Wein. Scheduling to minimize average completion time: Off-line and on-line approximation algorithms. *Mathematics of Operations Research*, 22(3):513–544, 1997.
- 11 François Margot, Maurice Queyranne, and Yaoguang Wang. Decompositions, network flows, and a precedence constrained single-machine scheduling problem. *Operations Research*, 51(6):981–992, 2003.
- 12 Maurice Queyranne and Maxim Sviridenko. A $(2+\varepsilon)$ -approximation algorithm for the generalized preemptive open shop problem with minsum objective. *J. Algorithms*, 45(2):202–212, 2002.
- 13 Alexander Schrijver. *Combinatorial optimization: polyhedra and efficiency*. Springer-Verlag, Berlin, 2003.
- 14 Andreas Schulz and Martin Skutella. Random-based scheduling: new approximations and lp lower bounds. In *RANDOM*, pages 119–133, 1997.
- 15 Martin Skutella and David P. Williamson. A note on the generalized min-sum set cover problem. *Operations Research Letters*, To appear, 2011.
- 16 Gerhard J. Woeginger. On the approximability of average completion time scheduling under precedence constraints. *Discrete Applied Mathematics*, 131(1):237–252, 2003.