



Integrality Gap of Time-Indexed Linear Programming Relaxation for Coflow Scheduling

Takuro Fukunaga  

Faculty of Science and Engineering, Chuo University, Tokyo, Japan

Abstract

Coflow is a set of related parallel data flows in a network. The goal of the coflow scheduling is to process all the demands of the given coflows while minimizing the weighted completion time. It is known that the coflow scheduling problem admits several polynomial-time 5-approximation algorithms that compute solutions by rounding linear programming (LP) relaxations of the problem. In this paper, we investigate the time-indexed LP relaxation for coflow scheduling. We show that the integrality gap of the time-indexed LP relaxation is at most 4. We also show that yet another polynomial-time 5-approximation algorithm can be obtained by rounding the solutions to the time-indexed LP relaxation.

2012 ACM Subject Classification Theory of computation → Scheduling algorithms

Keywords and phrases coflow scheduling, hypergraph matching, approximation algorithm

Digital Object Identifier 10.4230/LIPIcs.APPROX/RANDOM.2022.36

Category APPROX

Funding Takuro Fukunaga: JSPS KAKENHI Grant Numbers JP20H05965, JP21K11759, and JP21H03397, Japan.

1 Introduction

Coflow scheduling was introduced by Chowdhury and Stoica [7]. It is motivated by cluster computation frameworks such as MapReduce and Hadoop. Because these frameworks involve a huge amount of communication within a computer cluster, it is crucial to efficiently schedule this communication to achieve high computation performance. Coflow is an abstraction of data flow created by the processing of a task within the computer cluster. The goal of coflow scheduling is to find the most efficient scheduling of coflows.

Among the many variations of the coflow scheduling problem, weighted completion minimization under a bipartite matching model is the most extensively studied setting. In this setting, a coflow is represented as a bipartite undirected multigraph. An edge in the coflow represents the demand of sending one unit of data from one node to another. We are given a set of coflows F_1, \dots, F_k , all of which are on the same bipartition (X, Y) of the node set. Each coflow F_i is associated with a weight $w_i \geq 0$ and a release time $r_i \in \mathbb{Z}_+$, where \mathbb{Z}_+ is the set of non-negative integers. The required task is to schedule all demands of the coflows under the congestion constraint and the release time constraint. The congestion constraint requires all nodes to send or receive at most one unit of data at any moment, and the release time constraint requires the demand of coflow F_i to not be processed before release time r_i . The completion time C_i of coflow F_i is defined as the time at which all demands of F_i have been processed. The objective of the problem is to minimize the weighted completion time, defined as $\sum_{i=1}^k w_i C_i$. More information on the problem setting is given in Section 2.

This coflow scheduling problem includes the *concurrent open shop scheduling problem*, which corresponds to the special case where $X = \{x_1, \dots, x_n\}$, $Y = \{y_1, \dots, y_n\}$ and each edge of the given coflows joins nodes x_i and y_i for some $i \in \{1, \dots, n\}$. For concurrent open shop scheduling, achieving $(2 - \epsilon)$ -approximation for any $\epsilon > 0$ is known to be NP-hard [16].



© Takuro Fukunaga;

licensed under Creative Commons License CC-BY 4.0

Approximation, Randomization, and Combinatorial Optimization. Algorithms and Techniques (APPROX/RANDOM 2022).

Editors: Amit Chakrabarti and Chaitanya Swamy; Article No. 36; pp. 36:1–36:13



Leibniz International Proceedings in Informatics

Schloss Dagstuhl – Leibniz-Zentrum für Informatik, Dagstuhl Publishing, Germany

Thus, the same approximation hardness holds for coflow scheduling. The best approximation factor for coflow scheduling is achieved by the algorithms proposed by Shafiee and Ghaderi [17] and Ahmadi et al. [2], respectively. The factor is 4 when the release times for all given coflows are identical and 5 when they are not identical. Narrowing the gap between the upper and lower bounds of the approximation factor is an interesting open problem.

The above approximation algorithms [2, 17] for coflow scheduling are both based on linear programming (LP) relaxations of the problem. The algorithm of Shafiee and Ghaderi [17] uses a relaxation with ordering variables and that of Ahmadi et al. [2] uses a relaxation with parallel inequalities. These relaxations are commonly used in the machine scheduling literature. Their algorithms also give upper bounds on the integrality gap of these LP relaxations.

1.1 Our contribution

Our contribution is to investigate the *time-indexed LP relaxation*, which is another standard formulation of LP relaxations for machine scheduling problems. We show that the integrality gap of the time-indexed LP relaxation is at most 4 even for non-identical release times, which is better than the known upper bounds on the integrality gap of other LP relaxations. Our integrality gap analysis relies on Hall's theorem [1] on existence of perfect matchings in bipartite hypergraphs. We show that a 4-approximate solution is obtained by finding a perfect matching in a bipartite hypergraph constructed from an optimal solution solution to the time-indexed LP relaxation.

Unfortunately, our integrality gap analysis does not provide a polynomial-time algorithm of approximation factor that matches the integrality gap bound because there are no known polynomial-time algorithms for computing hypergraph perfect matchings implied by Hall's theorem. Nevertheless, we believe that our analysis is useful for obtaining an improved polynomial-time approximation algorithm in the future.

We would also like to point out that our analysis is a new interesting application of the Hall's theorem on hypergraphs. Previously Hall's theorem on hypergraphs has been used for developing approximation algorithms for min-max allocation problems in a series of studies (see e.g., [4, 5]). We note that this line of studies was initiated by Asadpour, Feige, and Saberi [5], the algorithm given in which is not a polynomial-time algorithm.

In addition to the integrality gap bound, we give a polynomial-time rounding algorithm for the time-indexed LP relaxation. Although our algorithm does not improve upon the currently best approximation algorithms [2, 17], we prove that our algorithm achieves the same approximation factors as them. Namely, its approximation factor is 4 for the identical release times, and 5 for non-identical release times.

Inspired by our polynomial-time rounding algorithm, we also observe that, if a hypergraph is constructed from the coflow scheduling with identical release times, then a perfect matching can be found in polynomial-time time. This gives an alternative 4-approximation algorithm for the coflow scheduling with identical release times.

Summing up, our contributions can be summarized as follows.

- We show that the rounding of a solution to the time-indexed LP relaxation can be reduced to finding a perfect matching in a hypergraph. This implies that the integrality gap of the time-indexed LP relaxation is at most 4, which improves on the integrality gap upper bounds on LP relaxations for non-identical release times.
- We propose a polynomial-time rounding algorithm for the time-indexed LP relaxation. Its approximation factor is 4 for identical release times and 5 for non-identical release times. These factors match those of the currently known best approximation algorithms for coflow scheduling.

- We propose a polynomial-time algorithm for computing perfect matchings in hypergraphs constructed in the reduction of rounding solutions to the time-indexed LP relaxation with identical release times.

1.2 Organization

The rest of this paper is organized as follows. Section 2 introduces preliminary facts and related studies on coflow scheduling and hypergraph perfect matching. Section 3 formulates the time-indexed LP relaxation. Section 4 presents the analysis of the integrality gap of the time-indexed LP relaxation. Section 5 describes the proposed polynomial-time rounding algorithm for the time-indexed LP relaxation. Section 6 describes the proposed polynomial-time algorithm for computing perfect matchings in hypergraphs constructed from the coflow scheduling with identical release times. Section 7 concludes this work.

2 Preliminary facts and related studies

2.1 Coflow scheduling

Throughout this paper, an edge between two nodes x and y is denoted by xy . The set of integers $1, \dots, n$ is denoted by $[n]$. For an edge set I and a node v , the set of edges in I incident to v is denoted by $\delta_I(v)$. The subscript is omitted when the edge set is clear from the context. The maximum degree of a graph G is denoted by $\Delta(G)$.

As mentioned in Section 1, the inputs of the bipartite matching model of the coflow scheduling problem are coflows F_1, \dots, F_k with weights $w_1, \dots, w_k \geq 0$ and release times $r_1, \dots, r_k \in \mathbb{Z}_+$, where coflows are bipartite multigraphs on the bipartition (X, Y) of the node set. We usually identify a graph with the set of edges. Let F denote $\bigcup_{i=1}^k F_i$.

We denote the time horizon of schedules by T . In this paper, we consider finding a *discrete-time integer* schedule, for which the time interval $[0, T)$ is divided into intervals $[0, 1), [1, 2), \dots, [T-1, T)$ and the data flow does not vary within an interval. We refer to interval $[t-1, t)$ as the t -th round. In contrast to a discrete-time schedule, a *continuous-time* schedule can change the data flow at any moment. In an integer schedule, data flow forms a matching in each round by the congestion constraint. Thus, a schedule is equivalent to a sequence (M_1, \dots, M_T) of matchings such that $\bigcup_{t=1}^T M_t = F$ and $M_t \cap F_i = \emptyset$ for each $i \in [k]$ and $t \in [r_i]$. The completion time C_i of coflow F_i in the schedule is given by $\max\{t: M_t \cap F_i \neq \emptyset\}$. The objective of the problem is to minimize the weighted completion time $\sum_{i=1}^k w_i C_i$. In addition to integer schedules, we can also consider a *fractional* schedule, where data flow within a round forms a fractional matching, i.e., a vector $x \in [0, 1]^E$ such that $\sum_{e \in \delta(v)} x(e) \leq 1$ for each $v \in X \cup Y$.

Since its introduction by Chowdhury and Stoica [7], coflow scheduling has been extensively studied from both practical and theoretical viewpoints [2, 8, 9, 14, 17, 18]. Several extensions of the problem setting have been presented. For example, Im et al. [13] considered the matroid coflow scheduling problem, which replaces the congestion constraint with a constraint that requires the set of elements scheduled in a round to be independent in a given matroid. Note that the bipartite matching model cannot be modeled by the matroid coflow, and hence the result of Im et al. cannot be applied to the bipartite matching model. Chowdhury et al. [6] considered flows in general graphs instead of bipartite matchings in the congestion constraint. Their model is a generalization of the bipartite matching model. However, the algorithm of Chowdhury et al. outputs only a fractional schedule, and thus it cannot be used for computing an integer schedule.

2.2 Hypergraph perfect matching

Let $H = (V, E)$ be a hypergraph with node set V and hyperedge set E . Here, we regard each hyperedge as a set of nodes.

A *matching* M in a hypergraph $H = (V, E)$ is a subset of E such that $|\delta_M(v)| \leq 1$ for all $v \in V$, where we naturally extend the notation δ to hypergraphs. A *transversal* U of $H = (V, E)$ is a subset of V such that $U \cap e \neq \emptyset$ for all $e \in E$. The maximum size of matchings and the minimum size of transversals of H are called the *matching number* and the *transversal number* of H , denoted by $\nu(H)$ and $\tau(H)$, respectively. A *fractional matching* is a function $x: E \rightarrow [0, 1]$ such that $\sum_{e \in \delta(v)} x(e) \leq 1$ for each $v \in V$. The maximum value of $\sum_{e \in E} x(e)$ among all fractional matchings x in H is called the *fractional matching number* of H and is denoted by $\nu^*(H)$. Note that $\nu(H) \leq \nu^*(H) \leq \tau(H)$ holds for any hypergraph H .

H is said to be *r-uniform* if $|e| = r$ for each $e \in E$, and is said to be *bipartite* if its node set has a bipartition (A, B) such that $|A \cap e| = 1$ for all $e \in E$. Hereafter, we suppose that H is an *r-uniform bipartite hypergraph* with bipartition (A, B) . We denote the nodes in A by *A-nodes* and those in B by *B-nodes*. A *perfect matching* in H is a matching whose size is $|A|$ (i.e., all *A-nodes* are covered by some hyperedge in the matching). For $X \subseteq A$, let H_X represent the hypergraph with the node set B and the hyperedge set $E_X := \{e \setminus A: e \in E, e \cap X \neq \emptyset\}$. The following sufficient conditions for the existence of perfect matching are known.

► **Theorem 1** (Haxell [11]). *If an r-uniform bipartite hypergraph H with the node set bipartition (A, B) satisfies*

$$\tau(H_X) > (2r - 3)(|X| - 1) \text{ for any } X \subseteq A, \quad (1)$$

then H has a perfect matching.

► **Theorem 2** (Aharoni and Haxell [1]). *If an r-uniform bipartite hypergraph H with the node set bipartition (A, B) satisfies*

$$\nu(H_X) > (r - 1)(|X| - 1) \text{ for any } X \subseteq A, \quad (2)$$

then H has a perfect matching.

Note that these two theorems extend the sufficient condition implied by Hall's theorem to the existence of perfect matchings in bipartite graphs (although the condition in Hall's theorem is necessary and sufficient, the conditions in the above two theorems are not).

The proofs of these theorems are not algorithmic. Nevertheless, Annamalai [3] gave an algorithmic proof of Haxell's theorem by introducing a small amount of slack into the condition. More concretely, Annamalai showed that, if there exists a constant $\epsilon > 0$ such that the hypergraph H satisfies $\tau(H_X) > (2r - 3 + \epsilon)(|X| - 1)$ for any $X \subseteq A$, then there exists a polynomial-time algorithm for finding a perfect matching in H . There is no known polynomial-time algorithm for finding a perfect matching in a hypergraph that satisfies condition (2). Note that finding perfect matchings in 3-uniform bipartite hypergraphs is NP-hard in general because it includes 3-dimensional matching [15].

3 Time-indexed LP relaxation

In this section, we introduce the time-indexed LP relaxation for the coflow scheduling problem.

We set T to an upper bound on the time horizon of optimal coflow scheduling. For example, T can be set to $|F|$. Indeed, we can see that $2\Delta(F) + \max_{i \in [k]} r_i$ is also an upper bound because of the observations explained below in Lemma 6.

In the time-indexed LP, we have a variable $x_{t,e} \in [0, 1]$ for each $t \in [T]$ and $e \in F$, and a variable c_i for each $i \in [k]$. When the variables take integer values, variable $x_{t,e}$ indicates whether the demand e is processed in the t -th round (i.e., time interval $[t-1, t)$), and variable c_i is the completion time of coflow F_i .

The time-indexed LP is formulated as follows.

$$\begin{aligned} \text{minimize} \quad & \sum_{i \in [k]} w_i c_i \\ \text{subject to} \quad & \sum_{t \in [T]} t x_{t,e} \leq c_i, & \forall i \in [k], \forall e \in F_i, \end{aligned} \quad (3)$$

$$\sum_{e \in \delta_F(v)} x_{t,e} \leq 1, \quad \forall t \in [T], \forall v \in V, \quad (4)$$

$$\sum_{t \in [T]} x_{t,e} = 1, \quad \forall i \in [k], \forall e \in F_i, \quad (5)$$

$$\begin{aligned} x_{t,e} &= 0, & \forall i \in [k], \forall e \in F_i, \forall t \in [r_i], \\ x_{t,e} &\geq 0, & \forall e \in F, \forall t \in [T]. \end{aligned} \quad (6)$$

Constraint (3) requires c_i to be at least the time of processing $e \in F_i$. Constraint (4) requires at most one edge incident to a node v to be processed within the t -th round. Constraint (5) requires each demand e in coflow F_i to be processed in some round. Constraint (6) requires the demands in coflow F_i to not be processed before the release time r_i .

Each solution for the time-indexed LP relaxation represents a discrete-time fractional schedule that consists of fractional matchings $x_1, \dots, x_T \in [0, 1]^F$. Let C_i be the completion time of coflow F_i in this schedule, expressed as

$$C_i = \max\{t \in [T] : x_{t,e} > 0 \text{ for some } e \in F_i\}.$$

Thus, the weighted completion time of this fractional schedule is $\sum_{i \in [k]} w_i C_i$. Note that this value is possibly larger than the objective value $\sum_{i \in [k]} w_i c_i$ of the relaxation.

In the bipartite matching model, the discrete-time fractional schedule can be transformed into a continuous-time integer schedule without increasing the completion time of each coflow as follows. By the integrality of the fractional matching polytope, the fractional matching x_t can be represented as a convex combination of (integer) matchings. Namely, there exists a set of matchings M_1, \dots, M_m and nonnegative numbers $\lambda_1, \dots, \lambda_m$ such that $x_t = \sum_{j=1}^m \lambda_j \chi_{M_j}$ and $\sum_{j=1}^m \lambda_j = 1$ hold, where χ_{M_j} is the characteristic vector of matching M_j . A continuous-time integer schedule is obtained by scheduling the matching M_j for time λ_j within the t -th round.

Conversely, a continuous-time integer schedule can be transformed into a discrete-time fractional schedule. Let λ_M be the time spent for processing a matching M in the t -th round of the integer schedule. Then, the convex combination of matchings with coefficients λ_M is a fractional matching. A discrete-time fractional schedule is obtained by scheduling this fractional matching in the t -th round. If the completion time of coflow F_i in the integer schedule is C'_i , the completion time of F_i in the constructed fractional schedule is $\lceil C'_i \rceil$.

► **Remark.** The size of the time-indexed LP linearly depends on T , and hence running time for solving the LP is at least a polynomial with regards to T . Although this running time is polynomial in the input size of the instance of the coflow scheduling problem, it may be a disadvantage compared with other LP relaxations such as those used in [2, 17]. However, the size of the time-indexed LP can be reduced using a commonly used technique (see e.g., [12]) so that it depends on $O(\log T)$ with a loss of $1 + \epsilon$ in the approximation factor for any constant $\epsilon > 0$.

4 Integrality gap analysis

This section proves that the integrality gap of the time-indexed LP relaxation is at most 4 for the bipartite matching model. In the proof, we first show that there exists a discrete-time fractional schedule whose weighted completion time is at most twice the optimal objective value of the relaxation. Then, this fractional schedule is rounded into an integer schedule that is subject to the completion time of each coflow being at most twice that in the fractional schedule. This rounding is done by finding a perfect matching in a hypergraph constructed from the fractional schedule.

4.1 Random stretching of fractional schedule

As mentioned in Section 3, a solution (x, c) for the time-indexed LP relaxation represents a discrete-time fractional schedule, but the completion time C_i of coflow F_i in this schedule is possibly larger than c_i . However, as studied in [6, 13], random stretching gives another fractional schedule wherein the expected completion time of F_i is at most $2c_i$. The details are as follows.

For $e \in F$ and $t \in [T]$, let $v_e(t) = \sum_{t' \in [t]} x_{t', e}$. Furthermore, we extend the definition of $v_e(t)$ to any $t \in [0, T]$ via linear interpolation. Namely, if $t \in [t' - 1, t')$ for some $t' \in [T]$, then $v_e(t) := v_e(t' - 1) + (t - t' + 1)(v_e(t') - v_e(t' - 1))$.

For $i \in [k]$ and $\theta \in [0, 1]$, we define $C_i(\theta)$ as the time at which θ -fraction of coflow F_i is completed in the discrete-time fractional schedule implied by the solution (x, c) to the relaxation. That is, $C_i(\theta)$ is the minimum value of $t \in [0, T]$ such that $v_e(t) \geq \theta$ for all $e \in F_i$.

In the random stretching operation, we randomly sample θ from $[0, 1]$ according to the probability density function $f(\theta) := 2\theta$. Then, we stretch the schedule by the factor $1/\theta$. This means that if a demand is processed in a time interval $[t', t'']$, then it is processed in $[t'/\theta, t''/\theta]$. The processing of a demand is truncated when the processing time reaches one unit of time. This gives a continuous-time fractional schedule such that the completion time of a coflow F_i is $C_i(\theta)/\theta$.

The continuous-time fractional schedule can be transformed into a discrete-time fractional schedule as follows. For $t \in [T]$ and $e \in F$, let $\bar{x}_{t, e}$ be the fraction of e processed in time $[t - 1, t)$ of the continuous-time fractional schedule. Then, it can be verified that $\{\bar{x}_{t, e} : e \in F\}$ forms a fractional matching for any $t \in [T]$, and thus it gives a discrete-time fractional schedule. In this discrete-time schedule, the process of coflow F_i is within an interval $[r_i, \lceil C_i(\theta)/\theta \rceil]$.

We have thus obtained a discrete-time fractional schedule by stretching the schedule represented by the LP optimal solution. The following lemma shows that the expected completion time in this schedule can be bounded by twice the objective value of the time-indexed LP.

► **Lemma 3.** *For each $i \in [k]$, $\mathbb{E}[\lceil C_i(\theta)/\theta \rceil] \leq 2c_i$.*

This lemma is proven in [6, 13] for other variations of the coflow scheduling problem, and these proofs also apply to our problem. We omit the proof of Lemma 3 in this paper.

In the rest of the paper, we let \bar{C}_i denote $\lceil C_i(\theta)/\theta \rceil$.

4.2 Reduction to hypergraph perfect matching

By Lemma 3, a schedule of processing coflow F_i within the interval $[r_i, \bar{C}_i]$ achieves a weighted completion time that is at most twice the optimal objective value of the relaxation. Moreover, the discrete-time fractional schedule implied by \bar{x} does so. What remains is to round this fractional schedule into a discrete-time integer schedule.

For the matroid coflow scheduling problem, Im et al. [13] showed that this rounding process can be done without loss of the approximation factor. This is because the fractional schedule is included in the intersection of a matroid polytope and a base polytope, where the matroid polytope is defined based on a constraint that requires demands processed in each round to be independent in the given matroid. Because the intersection forms an integer polytope, the fractional schedule can be represented as a convex combination of integer schedules, any of which processes coflow F_i within $[r_i, \bar{C}_i]$. This approach is not available for our problem because bipartite matchings do not form a matroid but a matroid intersection; thus the set of the fractional schedules is the intersection of two matroid polytopes and a base polytope, that is not integer in general.

Instead, we reduce the rounding process to hypergraph perfect matching. We first construct a hypergraph as follows. We prepare T copies of the node set, each of which corresponds to a round. We let V_t denote the copy corresponding to the t -th round for each $t \in [T]$, and let v_t denote the node in V_t corresponding to $v \in X \cup Y$. In addition, we introduce a node a_e corresponding to each demand $e \in F$. Let $A := \{a_e : e \in F\}$ and $B := \bigcup_{t \in [T]} V_t$. A hyperedge in the hypergraph is defined by an edge $e = xy \in F_i$ and time $t \in [r_i + 1, \bar{C}_i]$ as $h_{e,t} := \{x_t, y_t, a_e\}$. Let $H = (V_H, E_H)$ denote the hypergraph with the node set $V_H = A \cup B$ and the hyperedge set $E_H = \{h_{e,t} : i \in [k], e \in F_i, t \in [r_i + 1, \bar{C}_i]\}$. Note that H is a 3-uniform bipartite hypergraph with bipartition (A, B) .

From a perfect matching in H , we define a discrete-time integer schedule so that a demand $e = uv$ is processed in the t -th round whenever the hyperedge $h_{e,t}$ is included in the matching. Because each node in B is incident to at most one hyperedge in a matching, the demands processed in each round of the schedule form a matching. Moreover, because each node $a_e \in A$ is covered by exactly one hyperedge in the perfect matching, and because all hyperedges incident to a_e are defined only for the t -th rounds with $t \in [r_i + 1, \bar{C}_i]$ if $e \in F_i$, the demand $e \in F_i$ is processed within an interval $[r_i, \bar{C}_i]$ in the schedule. Therefore, the defined integer schedule is feasible.

Based on this discussion, it suffices to find a perfect matching in H . However, we do not know whether H has a perfect matching. To ensure the existence of a perfect matching, we modify H so as to satisfy the Aharoni-Haxell condition (2). For this purpose, let us bound $\nu(H_X)$ for $X \subseteq A$. First, observe that H_X is a bipartite graph, with the node set $B = \bigcup_{t \in [T]} V_t$ and the edge set $\{u_t v_t : a_{uv} \in X, t \in [r_i + 1, \bar{C}_i] \text{ for } i \text{ with } uv \in F_i\}$. Therefore, $\tau(H_X) = \nu^*(H_X) = \nu(H_X)$. Moreover, $\bar{x}_{t,uv}$ can be regarded as a weight assigned to edge $u_t v_t$ in H_X . It forms a fractional matching in H_X . Because $\sum_{t \in [T]} \bar{x}_{t,uv} = 1$, H_X has a fractional matching of size $|X|$. These facts indicate that $\nu(H_X) \geq |X|$.

This bound is insufficient to satisfy the Aharoni-Haxell condition, which requires satisfying $\nu(H_X) > 2(|X| - 1)$ since $r = 3$ in our case. Thus, we modify H as follows. In the original definition, for each round $t \in [T]$, we have the corresponding node set V_t , and the node set of H is defined as $A \cup (\bigcup_{t \in [T]} V_t)$. For each $i \in [k]$, $e = uv \in F_i$, and $t \in [r_i + 1, \bar{C}_i]$, H has a hyperedge $\{a_e, u_t, v_t\}$. In the new definition, for each round $t \in [T]$, we define two node sets V_{2t-1} and V_{2t} , and define the node set as $A \cup (\bigcup_{t \in [T]} V_{2t-1} \cup V_{2t})$. Hyperedges $\{a_e, u_{2t-1}, v_{2t-1}\}$ and $\{a_e, u_{2t}, v_{2t}\}$ are defined for each $i \in [k]$, $e = uv \in F_i$, and $t \in [r_i + 1, \bar{C}_i]$. Let H' denote the obtained hypergraph.

► **Lemma 4.** *H' has a perfect matching.*

Proof. H' is still a 3-uniform bipartite hypergraph, with the bipartition $(A, \bigcup_{t=1}^{2T} V_t)$. Let us show that H'_X satisfies $\nu(H_X) \geq 2|X|$ for any $X \in A$, which indicates the existence of a perfect matching in H' by Lemma 2.

Note that each edge $u_t v_t$ in H'_X is defined by a hyperedge $\{a_e, u_t, v_t\}$ incident to an A -node $a_e \in X$. We define $x'_{u_t v_t}$ as $\bar{x}_{\lceil t/2 \rceil, e}$ for each edge $u_t v_t$ in H'_X . Then, x' is a fractional matching in H'_X because \bar{x}_t is a fractional matching for each $t \in [T]$. Moreover, because $\sum_{t \in [T]} \bar{x}_{t, e} = 1$, $\sum_{t \in [2T]} x'_{u_t v_t} = 2$ holds. Thus, the size of the fractional matching x' is $2|X|$, and hence $\nu^*(H'_X) \geq 2|X|$. Note that H'_X is a bipartite graph, and hence $\nu(H'_X) = \nu^*(H'_X)$. Therefore, the claim is proven. ◀

We can define a discrete-time integer schedule from a perfect matching in H' ; if a_e is covered by a hyperedge $\{a_e, u_t, v_t\}$ in the perfect matching, then demand e is processed in the t -th round. Because each A -node a_e has incident hyperedges corresponding to rounds in $[2(r_i + 1) - 1, 2\bar{C}_i]$ if $e \in F_i$, the constructed integer schedule satisfies the release time constraint and all demands of coflow F_i are completed by time $2\bar{C}_i$. Therefore, the weighted completion time of this schedule is at most $2 \sum_{i \in F} w_i \bar{C}_i$. This fact and Lemma 3 prove the following theorem.

► **Theorem 5.** *The integrality gap of the time-indexed LP relaxation is at most 4.*

As for a lower bound on the integrality gap of the time-indexed LP, the following simple instance shows that it is at least 2. Suppose that there is a single coflow that consists of M parallel edges, and its weight and release time are 1 and 0. The minimum weighted completion time of integer schedules for this instance is M . On the other hand, the fractional schedule that processes $1/M$ unit of all edges in each round achieves the weighted completion time $(M + 1)/2$. The ratio of this value to M approaches 2 as M grows. We are aware of no instance that indicates integrality gap larger than 2.

As mentioned in Section 2.2, the Aharoni-Haxell condition ensures the existence of a perfect matching but does not provide a polynomial-time algorithm for finding it. The algorithm of Annamalai [3] finds a perfect matching in a hypergraph that satisfies the Haxell condition with a constant slack, i.e., $\tau(H'_X) > (2r - 3 + \epsilon)(|X| - 1)$ for any $X \subseteq A$ and any constant $\epsilon > 0$ (again, recall that $r = 3$ in our case). Using this algorithm gives us a polynomial-time rounding algorithm, but making the hypergraph satisfy the condition results in an approximation factor of 6, which is worse than that for existing coflow scheduling algorithms.

5 Polynomial-time rounding algorithm

In this section, we present a polynomial-time rounding algorithm for the time-indexed LP. It achieves 4-approximation for identical release times and 5-approximation for non-identical release times.

The algorithm first sorts the coflows in the non-decreasing order of c . Then, it schedules the demands greedily, giving higher priority to demands of earlier coflows. The details of this algorithm are given in Algorithm 1.

► **Lemma 6.** *The completion time of coflow F_i in the schedule output by Algorithm 1 is at most $r_i + 2\Delta(\bigcup_{j=1}^i F_j) - 1$ for each $i \in [k]$.*

Algorithm 1 Rounding Algorithm.

- 1 solve the time-indexed LP to obtain an optimal solution (x, c) ;
 - 2 sort the coflows so that $c_1 \leq c_2 \leq \dots \leq c_k$;
 - 3 $M_t := \emptyset$ for each $t \in [T]$;
 - 4 **for** $i = 1, \dots, k$ **do**
 - 5 **for** $uv \in F_i$ **do**
 - 6 find the minimum $t \in [r_i + 1, T]$ such that $\delta_{M_t}(u) = \delta_{M_t}(v) = \emptyset$;
 - 7 $M_t := M_t \cup \{uv\}$
 - 8 output (M_1, \dots, M_t)
-

Proof. Let uv be a demand in F_i that is processed last, and let t be the round in which uv is processed (i.e., t is the completion time of F_i). Then, in each round in $[r_i + 1, \dots, t - 1]$, a demand incident to u or v is processed. This means that $t - 1 - r_i \leq |\delta_{\bigcup_{j=1}^i F_j}(u)| - 1 + |\delta_{\bigcup_{j=1}^i F_j}(v)| - 1 \leq 2\Delta(\bigcup_{j=1}^i F_j) - 2$ holds. Therefore, the completion time of F_i is at most $r_i + 2\Delta(\bigcup_{j=1}^i F_j) - 1$. ◀

Now, we prove the following.

► **Lemma 7.** For each $i \in [k]$, $\Delta(\bigcup_{j=1}^i F_j) \leq 2c_i$.

Proof. Suppose that the indices of coflows indicate those after sorting in line 3 of the algorithm. Namely, $c_1 \leq c_2 \leq \dots \leq c_k$. We fix $i \in [k]$ and $v \in X \cup Y$, and we prove that the degree of v in the graph $\bigcup_{j=1}^i F_j$ is at most $2c_i$.

Since $\sum_{t \in [T]} x_{t,e} = 1$ holds for any e by (5), we have

$$\sum_{j \in [i]} \sum_{e \in \delta_{F_j}(v)} \sum_{t \in [T]} x_{e,t} = \sum_{j \in [i]} \sum_{e \in \delta_{F_j}(v)} 1 = \sum_{j \in [i]} |\delta_{F_j}(v)|.$$

It suffices to show that this value is at most $2c_i$. For arriving at a contradiction, suppose that this is more than $2c_i$, i.e.,

$$2c_i < \sum_{j \in [i]} \sum_{e \in \delta_{F_j}(v)} \sum_{t \in [T]} x_{e,t}. \quad (7)$$

Let $e \in F_j$ for some $j \leq i$. Then, (3) and the assumption of $c_j \leq c_i$ show that

$$\sum_{t \in [T]} tx_{t,e} \leq c_j \leq c_i. \quad (8)$$

Moreover, since $\sum_{t \in [T]} x_{t,e} = 1$ holds by (5), we have

$$\begin{aligned} c_i - \sum_{t \in [T]} tx_{t,e} &= \sum_{t \in [T]} c_i x_{t,e} - \sum_{t \in [T]} tx_{t,e} \\ &= \sum_{t \in [T]} (c_i - t)x_{t,e} \\ &= \sum_{1 \leq t \leq c_i} (c_i - t)x_{t,e} + \sum_{c_i < t \leq T} (c_i - t)x_{t,e}. \end{aligned}$$

36:10 Integrality Gap of Time-Indexed LP for Coflow Scheduling

Since (8) indicates that this is at least 0, we have

$$\sum_{c_i < t \leq T} (t - c_i)x_{t,e} \leq \sum_{1 \leq t \leq c_i} (c_i - t)x_{t,e}.$$

Summing this inequality over all $j \in [i]$ and $e \in \delta_{F_j}(v)$ gives

$$\sum_{j \in [i]} \sum_{e \in \delta_{F_j}(v)} \sum_{c_i < t \leq T} (t - c_i)x_{t,e} \leq \sum_{j \in [i]} \sum_{e \in \delta_{F_j}(v)} \sum_{1 \leq t \leq c_i} (c_i - t)x_{t,e}. \quad (9)$$

Since $\sum_{e \in \delta_F(v)} x_{t,e} \leq 1$ for each $t \in [T]$ by (4), the right-hand side of (9) is bounded as

$$\sum_{j \in [i]} \sum_{e \in \delta_{F_j}(v)} \sum_{1 \leq t \leq c_i} (c_i - t)x_{t,e} \leq \sum_{1 \leq t \leq c_i} (c_i - t) \sum_{e \in \delta_F(v)} x_{t,e} \leq \sum_{1 \leq t \leq c_i} (c_i - t) = \frac{c_i(c_i - 1)}{2}. \quad (10)$$

On the other hand, from (7), we have

$$\sum_{c_i < t \leq T} \sum_{j \in [i]} \sum_{e \in \delta_{F_j}(v)} x_{e,t} > 2c_i - \sum_{1 \leq t \leq c_i} \sum_{j \in [i]} \sum_{e \in \delta_{F_j}(v)} x_{e,t} \geq c_i.$$

Thus the left-hand side of (9) is bounded as

$$\sum_{j \in [i]} \sum_{e \in \delta_{F_j}(v)} \sum_{c_i < t \leq T} (t - c_i)x_{t,e} = \sum_{c_i < t \leq T} \sum_{j \in [i]} \sum_{e \in \delta_{F_j}(v)} (t - c_i)x_{t,e} > \sum_{c_i < t \leq 2c_i} (t - c_i) = \frac{c_i(c_i + 1)}{2}. \quad (11)$$

(9), (10), and (11) give a contradiction. \blacktriangleleft

Combining Lemmas 6 and 7 proves the following theorem.

► **Theorem 8.** *Algorithm 1 is a 4-approximation algorithm for identical release times and a 5-approximation algorithm for non-identical release times.*

Proof. By Lemmas 6 and 7, the schedule output by Algorithm 1 processes the coflow F_i by time $r_i + 4c_i - 1$. Note that $r_i \leq c_i$ holds for each $i \in [k]$. Therefore, the weighted completion time of the schedule is at most $5 \sum_{i \in [k]} w_i c_i$, which means that the algorithm achieves 5-approximation. In the identical release time case, we can assume that $r_i = 0$ for all $i \in [k]$. Then, the weighted completion time of the schedule is at most $4 \sum_{i \in [k]} w_i c_i$, which means that it achieves 4-approximation. \blacktriangleleft

► **Remark.** The above analysis does not depend on the assumption that coflows F_1, \dots, F_k are bipartite. Thus, it applies to the general graph model, where given coflows are not bipartite graphs and the congestion constraint requires that the demands processed in each round form a (non-bipartite) matching. Although this is not mentioned in previous works, similar analysis shows that the approximation algorithms of [2, 17] can also work for the general graph model. In other words, these approximation algorithms do not make full use of the assumption that the coflows are bipartite. In contrast, the integrality gap analysis given in Section 4 uses the bipartiteness.

6 Finding perfect matchings in hypergraphs

We proved Theorem 5 by showing that the hypergraph H' (defined in Section 4.2) has a perfect matching. Unfortunately, we do not know how to find the perfect matching in polynomial time even though its existence is implied by Theorem 2. In this section, we present a polynomial-time algorithm for finding a perfect matching in H' when $r_i = 0$ for all $i \in [k]$. This gives an alternative proof of the statement for identical release times in Theorem 8.

■ **Algorithm 2** Perfect Matching Algorithm.

```

1 sort the coflows so that  $\bar{C}_1 \leq \bar{C}_2 \leq \dots \leq \bar{C}_k$ ;
2  $M := \emptyset$ ;
3 for  $i = 1, \dots, k$  do
4   for  $uv \in F_i$  do
5     find the minimum  $t \in [2\bar{C}_i]$  such that both  $u_t$  and  $v_t$  have no incident
     hyperedge in  $M$ ;
6     add hyperedge  $\{a_{uv}, u_t, v_t\}$  to  $M$ 
7 output  $M$ 

```

The algorithm is given in Algorithm 2. The next theorem shows that it finds a perfect matching.

► **Theorem 9.** *Algorithm 2 outputs a perfect matching in polynomial time.*

Proof. On line 5 of Algorithm 2, there always exists $t \in [2\bar{C}_i]$ such that both u_t and v_t have no incident hyperedge in M . If this claim is true, M is a perfect matching in H' at the termination of the algorithm. Because the algorithm runs in polynomial time, this proves the theorem.

To prove the above claim, we first show that $\sum_{j \in [i]} |\delta_{F_j}(v)| \leq \bar{C}_i$ holds for each $i \in [k]$ and $v \in V$. Recall that there exists $\bar{x}_{t,e} \in [0, 1]$ ($e \in F_j$, $t \in \bar{C}_j$) such that $\sum_{t \in [\bar{C}_j]} \bar{x}_{t,e} = 1$ for each $e \in F_j$, and $\sum_{j \in [k]} \sum_{e \in \delta_{F_j}(v)} \bar{x}_{t,e} \leq 1$ for each $t \in [T]$ and $v \in V$. Then,

$$\begin{aligned}
 \sum_{j \in [i]} |\delta_{F_j}(v)| &= \sum_{j \in [i]} \sum_{e \in \delta_{F_j}(v)} 1 = \sum_{j \in [i]} \sum_{e \in \delta_{F_j}(v)} \sum_{t \in [\bar{C}_j]} \bar{x}_{t,e} \\
 &= \sum_{t \in [\bar{C}_i]} \sum_{j \in [i]} \sum_{e \in \delta_{F_j}(v)} \bar{x}_{t,e} \leq \sum_{t \in [\bar{C}_i]} 1 = |\bar{C}_i|.
 \end{aligned}$$

Here, the third equality uses the fact that $\bar{C}_j \leq \bar{C}_i$ for all $j \in [i]$.

Then, when $uv \in F_i$ is chosen on line 4 of Algorithm 2, the number of hyperedges in M incident to nodes $u_1, \dots, u_{2\bar{C}_i}$ is at most $\sum_{j \in [i]} |\delta_{F_j}(u)| - 1 \leq \bar{C}_i - 1$. Similarly, the number of hyperedges in M incident to nodes $v_1, \dots, v_{2\bar{C}_i}$ is at most $\sum_{j \in [i]} |\delta_{F_j}(v)| - 1 \leq \bar{C}_i - 1$. Therefore, among $2\bar{C}_i$ pairs of $\{u_t, v_t\}$ ($t \in [2\bar{C}_i]$), there exist at least $2\bar{C}_i - 2(\bar{C}_i - 1) = 2$ pairs such that no hyperedge in M is incident to nodes in the pairs. ◀

7 Conclusion

We showed that the integrality gap of the time-indexed LP relaxation for the coflow scheduling problem is at most 4. We also proposed a polynomial-time rounding algorithm that achieves 4-approximation for identical release times and 5-approximation for non-identical release

times. In addition, we proposed a polynomial-time algorithm for finding a perfect matching in the bipartite hypergraph constructed from a solution for the time-indexed LP relaxation with identical release times.

There are many interesting directions of further study. One of them is to improve the approximation factor, in particular for non-identical release times. Based on our integrality gap analysis, this can be achieved by developing a polynomial-time algorithm for finding perfect matchings in 3-uniform bipartite hypergraphs that satisfy the Aharoni-Haxell condition (2). However, designing such an algorithm is regarded as a difficult problem. Indeed, it is mentioned in [10] as “Thus algorithmic versions of these results would also be very interesting and useful, but currently seem out of reach.” We believe that it is interesting to investigate algorithms for hypergraphs constructed in our rounding of solutions to the time-indexed LP relaxation with non-identical release times.

References

- 1 Ron Aharoni and Penny Haxell. Hall’s theorem for hypergraphs. *Journal of Graph Theory*, 35(2):83–88, 2000.
- 2 Saba Ahmadi, Samir Khuller, Manish Purohit, and Sheng Yang. On scheduling coflows. *Algorithmica*, 82(12):3604–3629, 2020.
- 3 Chidambaram Annamalai. Finding perfect matchings in bipartite hypergraphs. *Combinatorica*, 38(6):1285–1307, 2018.
- 4 Chidambaram Annamalai, Christos Kalaitzis, and Ola Svensson. Combinatorial algorithm for restricted max-min fair allocation. *ACM Transactions on Algorithms*, 13(3):37:1–37:28, 2017.
- 5 Arash Asadpour, Uriel Feige, and Amin Saberi. Santa Claus meets hypergraph matchings. *ACM Transactions on Algorithms*, 8(3):24:1–24:9, 2012.
- 6 Mosharaf Chowdhury, Samir Khuller, Manish Purohit, Sheng Yang, and Jie You. Near optimal coflow scheduling in networks. In Christian Scheideler and Petra Berenbrink, editors, *The 31st ACM on Symposium on Parallelism in Algorithms and Architectures, SPAA 2019, Phoenix, AZ, USA, June 22–24, 2019*, pages 123–134. ACM, 2019.
- 7 Mosharaf Chowdhury and Ion Stoica. Coflow: a networking abstraction for cluster applications. In Srikanth Kandula, Jitendra Padhye, Emin Gün Sirer, and Ramesh Govindan, editors, *11th ACM Workshop on Hot Topics in Networks, HotNets-XI, Redmond, WA, USA – October 29 – 30, 2012*, pages 31–36. ACM, 2012.
- 8 Mosharaf Chowdhury and Ion Stoica. Efficient coflow scheduling without prior knowledge. In Steve Uhlig, Olaf Maennel, Brad Karp, and Jitendra Padhye, editors, *Proceedings of the 2015 ACM Conference on Special Interest Group on Data Communication, SIGCOMM 2015, London, United Kingdom, August 17–21, 2015*, pages 393–406. ACM, 2015.
- 9 Mosharaf Chowdhury, Yuan Zhong, and Ion Stoica. Efficient coflow scheduling with Varys. In Fabián E. Bustamante, Y. Charlie Hu, Arvind Krishnamurthy, and Sylvia Ratnasamy, editors, *ACM SIGCOMM 2014 Conference, SIGCOMM’14, Chicago, IL, USA, August 17–22, 2014*, pages 443–454. ACM, 2014.
- 10 Alessandra Graf and Penny Haxell. Finding independent transversals efficiently. *Combinatorics, Probability & Computing*, 29(5):780–806, 2020.
- 11 Penny E. Haxell. A condition for matchability in hypergraphs. *Graphs and Combinatorics*, 11(3):245–248, 1995.
- 12 Sungjin Im and Shi Li. Better unrelated machine scheduling for weighted completion time via random offsets from non-uniform distributions. In Irit Dinur, editor, *IEEE 57th Annual Symposium on Foundations of Computer Science, FOCS 2016, 9–11 October 2016, Hyatt Regency, New Brunswick, New Jersey, USA*, pages 138–147. IEEE Computer Society, 2016.

- 13 Sungjin Im, Benjamin Moseley, Kirk Pruhs, and Manish Purohit. Matroid coflow scheduling. In Christel Baier, Ioannis Chatzigiannakis, Paola Flocchini, and Stefano Leonardi, editors, *46th International Colloquium on Automata, Languages, and Programming, ICALP 2019, July 9-12, 2019, Patras, Greece*, volume 132 of *LIPICs*, pages 145:1–145:14. Schloss Dagstuhl – Leibniz-Zentrum für Informatik, 2019.
- 14 Hamidreza Jahanjou, Erez Kantor, and Rajmohan Rajaraman. Asymptotically optimal approximation algorithms for coflow scheduling. In Christian Scheideler and Mohammad Taghi Hajiaghayi, editors, *Proceedings of the 29th ACM Symposium on Parallelism in Algorithms and Architectures, SPAA 2017, Washington DC, USA, July 24-26, 2017*, pages 45–54. ACM, 2017.
- 15 Richard M. Karp. Reducibility among combinatorial problems. In Raymond E. Miller and James W. Thatcher, editors, *Proceedings of a symposium on the Complexity of Computer Computations, held March 20-22, 1972, at the IBM Thomas J. Watson Research Center, Yorktown Heights, New York, USA*, The IBM Research Symposia Series, pages 85–103. Plenum Press, New York, 1972.
- 16 Sushant Sachdeva and Rishi Saket. Optimal inapproximability for scheduling problems via structural hardness for hypergraph vertex cover. In *Proceedings of the 28th Conference on Computational Complexity, CCC 2013, K.lo Alto, California, USA, 5-7 June, 2013*, pages 219–229. IEEE Computer Society, 2013.
- 17 Mehrnoosh Shafiee and Javad Ghaderi. An improved bound for minimizing the total weighted completion time of coflows in datacenters. *IEEE/ACM Transactions on Networking*, 26(4):1674–1687, 2018.
- 18 Yue Zeng, Baoliu Ye, Bin Tang, Songtao Guo, and Zhihao Qu. Scheduling coflows of multi-stage jobs under network resource constraints. *Computer Networks*, 184:107686, 2021.