Weighted Chairman Assignment and Flow-Time Scheduling

Siyue Liu
Tepper School of Business, Carnegie Mellon University, Pittsburgh, PA, USA

Victor Reis
Microsoft Research, Redmond, WA, USA

Abstract

Given positive integers m, n, a fractional assignment $x \in [0,1]^{m\times n}$ and weights $d \in \mathbb{R}^n_{>0}$, we show that there exists an assignment $y \in \{0,1\}^{m\times n}$ so that for every $i \in [m]$ and $t \in [n]$,

$$\Big|\sum_{j\in[t]} d_j(x_{ij} - y_{ij})\Big| < \max_{j\in[n]} d_j.$$

This generalizes a result of Tijdeman (1973) on the unweighted version, known as the chairman assignment problem. This also confirms a special case of the single-source unsplittable flow conjecture with arc-wise lower and upper bounds due to Morell and Skutella (IPCO 2020). As an application, we consider a scheduling problem where jobs have release times and machines have closing times, and a job can only be scheduled on a machine if it is released before the machine closes. We give a 3-approximation algorithm for maximum flow-time minimization.

2012 ACM Subject Classification Theory of computation → Randomness, geometry and discrete structures

Keywords and phrases prefix discrepancy, flow-time scheduling, unsplittable flow

Digital Object Identifier 10.4230/LIPIcs.ITCS.2026.98

Acknowledgements We would like to thank R. Ravi for numerous discussions. The work was done during an internship of the first author at Microsoft Research.

1 Introduction

Let m, n be positive integers. We say that a matrix $x \in [0,1]^{m\times n}$ is a *fractional assignment* if $\sum_{i\in[m]} x_{ij} = 1$ for every column $j \in [n]$, and that $y \in \{0,1\}^{m\times n}$ is an *(integral) assignment* if $\sum_{i\in[m]} y_{ij} = 1$ for every $j \in [n]$. Consider the following problem introduced by Niederreiter in 1972 [24, 25]: find the minimum value of $\Delta(m)$ such that for every positive integer n and fractional assignment $x \in [0,1]^{m\times n}$, there is an assignment $y \in \{0,1\}^{m\times n}$ so that for every row $i \in [m]$ and column $t \in [n]$,

$$\Big|\sum_{j\in[t]} (x_{ij} - y_{ij})\Big| \le \Delta(m).$$

This is known as the *chairman assignment problem* due to Tijdeman [27]: suppose m states form a union and each state i receives a value x_{ij} from being a member at year j so that $\sum_{i\in[m]} x_{ij} = 1$. Every year a union chairman has to be selected in such a way that at any year t the accumulated number of chairmen $\sum_{j\in[t]} y_{ij}$ from each state i is proportional to its accumulated value $\sum_{j\in[t]} x_{ij}$. Niederreiter showed a bound of $\Delta(m) \le m - 1$ [24] which was subsequently improved by Meijer and Niederreiter to $\Delta(m) \le O(\log m)$ [22] and by Tijdeman to $\Delta(m) \le 1$ [26], who also showed that, for $m > 1$ and any $\delta > 0$, $\Delta(m) \ge 1 - \frac{1}{2m-2} - \delta$ via a family of examples with $n = \Omega(1/\delta)$. The approaches of [22] and [26] are both based on an application of Hall's theorem to a carefully constructed bipartite graph. Finally, Meijer

17th Innovations in Theoretical Computer Science Conference (ITCS 2026).
Editor: Shubhangi Saraf; Article No. 98; pp. 98:1–98:15
Leibniz International Proceedings in Informatics
LIPICS Schloss Dagstuhl – Leibniz-Zentrum für Informatik, Dagstuhl Publishing, Germany

matched this lower bound by showing $\Delta(m) \le 1 - \frac{1}{2m-2}$ [21]. Since then, there have been other proofs of this result through different approaches [27, 3, 16, 8]; see also the survey of Tijdeman [28].

We study the *weighted chairman assignment problem* where each $j \in [n]$ has an associated weight $d_j > 0$ and we want to bound $|\sum_{j\in[t]} d_j(x_{ij} - y_{ij})|$ for every $i \in [m]$ and $t \in [n]$. All of the previous approaches in the unweighted setting rely on the fact that the increments to the integral assignment are integers and do not directly generalize to the weighted setting. Our main theorem generalizes an algorithm of Tijdeman [27] (see also Angel, Holroyd, Martin and Propp [3]) with a new analysis to show the following:

▶ **Theorem 1.** *Given positive integers m, n with $m > 1$, a fractional assignment $x \in [0,1]^{m\times n}$ and weights $d \in \mathbb{R}^n_{>0}$, there is a linear-time algorithm that computes an assignment $y \in \{0,1\}^{m\times n}$ so that for every $i \in [m]$ and $t \in [n]$,*

$$\Big|\sum_{j\in[t]} d_j(x_{ij} - y_{ij})\Big| \le \Big(1 - \frac{1}{2m-2}\Big) \cdot \max_{j\in[n]} d_j.$$

The weighted chairman assignment problem gains renewed interest due to a conjecture by Morell and Skutella [23]: given a flow x from a single source to multiple sinks with varying demands, there is an *unsplittable flow* y where each sink should be served by a single path, such that the discrepancy between the two flows on each arc is at most the maximum demand. The weighted chairman assignment problem with weights $d_1, \dots, d_n$ can be modeled as a single-source unsplittable flow instance with demands $d_1, \dots, d_n$ (see Section 2), and therefore Theorem 1 with discrepancy bound $\max_{j\in[n]} d_j$ would follow if the conjecture were true. When the demands are uniform, i.e., $d_1 = \dots = d_n = 1$, the conjecture is true because by adding capacity constraints $\lfloor x\rfloor \le y \le \lceil x\rceil$ to the flow polytope we can find an *integral* flow y which is automatically unsplittable and satisfies $|x - y|_1 \le 1$. Therefore, the discrepancy bound 1 for the (unweighted) chairman assignment follows as a consequence. The weighted setting corresponds to nonuniform demands, where so far all linear programming techniques fall apart.

As an application, we consider the following scheduling problem. Suppose there are m machines M and n jobs J. Each machine $i \in M$ has a closing time $b_i \ge 0$. Each job $j \in J$ has a release time $r_j \ge 0$ and a processing time $d_j \ge 0$. Job j can be scheduled on machine i if and only if $r_j \le b_i$. The *flow-time* of a job j is defined as the time elapsed from its release to completion. The goal is to schedule jobs to machines to minimize the maximum flow-time of any job. This is a special case of the *restricted assignment* variant of maximum flow-time minimization, where each job j has a subset of machines $M_j \subseteq M$ that it can be scheduled on. In the most general form of the problem, each job j has a potentially different processing time d_{ij} on machines $i \in [m]$. Bansal, Rohwedder and Svensson [6] proved that a natural linear programming relaxation has integrality gap $O(\sqrt{\log n})$, and conjectured that an $O(1)$-approximation should be possible. We confirm this conjecture for the setting of machine closing times:

▶ **Theorem 2.** *There is a $(3 - \frac{1}{m-1})$-approximation algorithm for the maximum flow-time minimization problem in the setting of machine closing times.*

Previously, a $(3 - \frac{2}{m})$-approximation was known via a greedy algorithm (first-in-first-out, or FIFO) for the setting of identical machines where $b_i = \infty$ for all $i \in M$ [7, 20], and this bound is known to be tight for FIFO [19]. We also show that FIFO has approximation ratio at least $\Omega(\log m)$ for the setting of closing times.

Furthermore, we give several lower bound constructions. We provide a construction simpler than Tijdeman's [26] to establish a matching lower bound for the chairman assignment problem:

▶ **Proposition 3.** *For any positive integer $m > 1$ there exists a fractional assignment $x \in [0,1]^{m\times(m-1)}$ so that for any assignment $y \in \{0,1\}^{m\times(m-1)}$, there exists some $i \in [m]$ and $t \in [m-1]$ for which*

$$\Big|\sum_{j\in[t]}(x_{ij} - y_{ij})\Big| \geq 1 - \frac{1}{2m-2}.$$

The assignment given in [26] that achieves $\Delta(m) \leq 1$ yields a strict inequality and has the property that $y_{ij} = 0$ whenever $x_{ij} = 0$. We give a construction showing that Tijdeman's bound is tight under this additional constraint:

▶ **Proposition 4.** *For any $\delta > 0$ there exists a positive integer n and a fractional assignment $x \in [0,1]^{3\times n}$ so that for any assignment $y \in \{0,1\}^{3\times n}$ satisfying $y_{ij} = 0$ whenever $x_{ij} = 0$, there exists some $i \in [3]$ and $t \in [n]$ for which*

$$\Big|\sum_{j\in[t]}(x_{ij} - y_{ij})\Big| \geq 1 - \delta.$$

Finally, we disprove a conjecture in [3] that one may obtain a bound of 1 for all intervals, not just prefixes:

▶ **Proposition 5.** *There exist positive integers m, n and a fractional assignment $x \in [0,1]^{m\times n}$, so that for any assignment $y \in \{0,1\}^{m\times n}$, there exist some $i \in [m]$ and $1 \leq s \leq t \leq n$ for which*

$$\Big|\sum_{j\in[s,t]}(x_{ij} - y_{ij})\Big| > 1.$$

In particular, this implies that we cannot hope to directly apply our rounding approach to obtain a 2-approximation for maximum flow-time minimization.

Organization

In Section 2, we discuss related work. In Section 3, we prove our main Theorem 1. In Section 4, we discuss connections to maximum flow-time scheduling and prove Theorem 2. In Section 5, we give lower bound constructions to prove Propositions 3, 4 and 5. In Section 6, we formulate some open questions.

2 Related work

Theorem 1 resolves a special case of a conjecture of Morell and Skutella [23] which can be formally stated as follows. Let $D = (V, A)$ be a directed acyclic graph with source $s \in V$, sink terminals $t_1, \ldots, t_n \in V$ and associated demands $d \in \mathbb{R}^n_{\geq 0}$. We say a flow $x \in \mathbb{R}^A_{\geq 0}$ satisfies the demands if $x(\delta^-(t_j)) = d_j$ for every $j \in [n]$, and $x(\delta^-(v)) = x(\delta^+(v))$ for every $v \in V \setminus \{s, t_1, ..., t_n\}$. We say a flow $y \in \mathbb{R}^A_{\geq 0}$ is *unsplittable* if for each $j \in [n]$ there is a single path P_j from s to t_j that carries d_j units of flow, so that $y_a = \sum_{j:a\in P_j} d_j$ for every $a \in A$. Morell and Skutella conjectured the following:

▶ **Conjecture 6** ([23]). *For every flow $x \in \mathbb{R}^A_{\geq 0}$ satisfying the demands, there is an unsplittable flow $y \in \mathbb{R}^A_{\geq 0}$ such that $|x_a - y_a| \leq \max_{j\in[n]} d_j$ for all $a \in A$.*

They also showed the one-sided bound $x_a - y_a \leq \max_{j\in[n]} d_j$ for all $a \in A$ may be satisfied, thereby complementing the previously known one-sided bound $y_a - x_a \leq \max_{j\in[n]} d_j$ for all $a \in A$ of Dinitz, Garg and Goemans [12]. So far, Conjecture 6 has been established when the demands have the property that one divides another, i.e., $d_1 \mid d_2 \mid \ldots \mid d_n$ [23] and for special digraphs such as acyclic planar digraphs [29]; see also [2] for stronger guarantees for series-parallel digraphs.

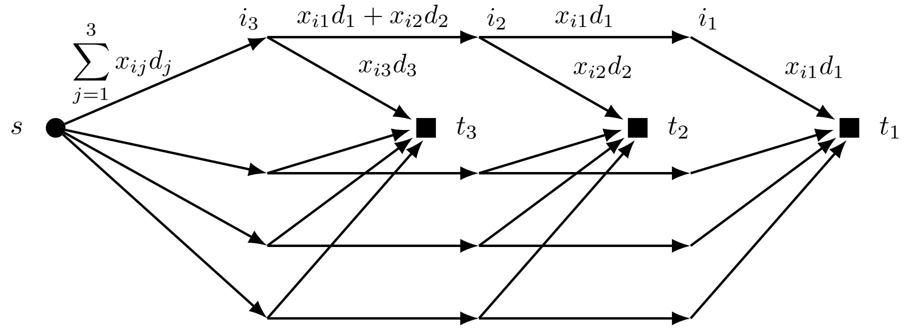

Figure 1 Reduction for $m = 4$, $n = 3$. We create n copies $i_1, \ldots, i_n$ of each $i \in [m]$ to capture all prefixes. Terminal t_j has demand $\sum_{i\in[m]} x_{ij}d_j = d_j$, and is connected to i_j for each $i \in [m]$. The flow value on arc (i_{t+1}, i_t) is $\sum_{j\in[t]} x_{ij}d_j$.

To see the connection with Theorem 1, note that given positive integers m, n we may construct a directed acyclic graph consisting of m paths of length n starting from s, with edges pointing at n terminals along the way. Then any fractional assignment $x \in [0, 1]^{m\times n}$ and weights $d \in \mathbb{R}^n_{>0}$ induce a flow so that each terminal t_j has demand d_j and each prefix is captured by one of the arcs in a path, as illustrated in Figure 1. An unsplittable flow routes each demand d_j to t_j through a path $i \in [m]$, which corresponds to assigning j to i. Assuming Conjecture 6, such an assignment y satisfies $|\sum_{j\in[t]} d_j(x_{ij} - y_{ij})| \leq \max_{j\in[n]} d_j$ for all $i \in [m], t \in [n]$.

A variant of the chairman assignment problem is the *carpooling problem* [14, 1], where $j \in [n]$ can be assigned to $i \in [m]$ if and only if $x_{ij} \neq 0$. The *weighted carpooling problem*, which remains open, can also be viewed as a special case of Conjecture 6 using a similar construction that connects terminal t_j to paths i for which $x_{ij} \neq 0$:

▶ **Conjecture 7.** *Let m, n be positive integers and $d \in \mathbb{R}^n_{>0}$. For every fractional assignment $x \in [0, 1]^{m\times n}$, there is an assignment $y \in \{0, 1\}^{m\times n}$ so that $y_{ij} = 0$ whenever $x_{ij} = 0$, and for every $i \in [m]$ and $t \in [n]$,*

$$\Big|\sum_{j\in[t]} d_j(x_{ij} - y_{ij})\Big| \leq \max_{j\in[n]} d_j.$$

Morell and Skutella [23] outlined the connection to maximum flow-time minimization and noted that Conjecture 7 would imply a 3-approximation for the restricted assignment setting. The best known approximation in polynomial time is $O(\log n)$ due to Bansal and Kulkarni via iterated rounding [5]. Bansal, Rohwedder and Svensson [6] prove a bound of $O(\sqrt{\log n}) \cdot \max_{j\in[n]} d_j$ for Conjecture 7, leveraging a non-constructive argument from convex geometry by Banaszczyk [4], thereby showing that the natural linear programming relaxation for maximum flow-time minimization has integrality gap $O(\sqrt{\log n})$. For identical machines, a simple 3-approximation is known [7, 20]: sort the jobs by release times and assign them one by one to the machine with the least remaining processing time of jobs that have been assigned to it. The special case where release times are all zero corresponds to makespan minimization; the work of Lenstra, Shmoys and Tardos gives a 2-approximation algorithm via rounding [18]. Their approach also solves the Morell-Skutella conjecture for digraphs of diameter 2.

To the best of our knowledge, the weighted chairman assignment problem has not been studied before. The more general weighted carpooling problem fits within the broader framework of *prefix discrepancy* [6], and it is possible to derive a bound of $2m \cdot \max_{j\in[n]} d_j$ for Conjecture 7 via a linear algebraic argument [10, 9]. Together with Banaszczyk's result [4], the current best bound for Conjecture 7 is $\min\{2m, O(\sqrt{\log n})\} \cdot \max_{j\in[n]} d_j$. The special case of Conjecture 7 where each column $\{x_{ij}\}_{i\in[m]}$ has at most two nonzero entries is known as the *2-sparse prefix Beck-Fiala problem* and is equivalent to the general case up to a constant [6]. Proposition 4 shows that unlike Theorem 1, we cannot hope for a $(1-\delta) \cdot \max_{j\in[n]} d_j$ bound for Conjecture 7 for any $\delta > 0$ even for $m = 3$.

For the online version of the chairman assignment problem, where columns of x arrive one at a time and the assignment must be decided irrevocably, it is known that the best possible bound is $\sum_{j=2}^{m} \frac{1}{j} = \Theta(\log m)$ for adaptive adversaries, attainable with a simple greedy algorithm [11], and the analysis readily generalizes for the weighted setting. On the other hand, the best possible bound independent of n for the online version of the carpooling problem is $\frac{m-1}{2}$ [11]. For the online 2-sparse prefix Beck-Fiala problem, there is also an $O(\sqrt{\log n})$ bound [17] and a $\Omega(\sqrt[3]{\log n})$ lower bound [1] for oblivious adversaries. A recent line of work studies the online carpooling problem with *recourse* [15, 13], and it remains open to obtain a constant bound for the online chairman assignment problem with polylogarithmic recourse.

3 Proof of Theorem 1

Given $m, n \in \mathbb{N}$ with $m > 1$, denote $\varepsilon := \frac{1}{2m-2}$. Let $x \in [0,1]^{m\times n}$ be a fractional assignment. We normalize $d \in \mathbb{R}^n_{>0}$ so that $\max_{j\in[n]} d_j = 1$, thereby assuming $d \in (0,1]^n$. For $j \in [n]$, denote by $s_j \in [m]$ the element to which j will be assigned, so that $y_{ij} = 1$ if $s_j = i$ and 0 otherwise. For each $i \in [m]$ and $t \in [n]$, denote

$$P_t(i) := \sum_{j=1}^{t} d_j x_{ij}, \quad N_t(i) := \sum_{j=1}^{t} d_j y_{ij}, \quad \Delta_t(i) := P_t(i) - N_t(i).$$

At time t, the *deadline* of i is defined as

$$D_t(i) := \min\{T \geq t : P_T(i) \geq N_{t-1}(i) + 1 - \varepsilon\}, \tag{1}$$

which is set to ∞ (or $n+1$) if such T does not exist.

The set of *candidates* at time t is defined as

$$C(t) := \big\{i \in [m] : P_t(i) \geq N_{t-1}(i) + \min\{d_t/m, \varepsilon\}\big\}. \tag{2}$$

At time t, the algorithm chooses a candidate with the earliest deadline. The detailed description is in Algorithm 1. We remark that if we only wanted a bound of $\max_{j\in[n]} d_j$ in Theorem 1, simpler definitions of deadline and candidates where setting $\varepsilon = 0$ in both definitions would have been sufficient.

Algorithm 1 EARLIEST DEADLINE ALGORITHM.

Require: Fractional assignment $x \in [0,1]^{m \times n}$, $d \in (0,1]^n$
Ensure: Assignment $y \in \{0,1\}^{m \times n}$ such that $|\Delta_t(i)| \leq 1 - \varepsilon$ for all $i \in [m],\ t \in [n]$

```
for t ∈ [n] do
    C(t) ← {i ∈ [m] : P_t(i) ≥ N_{t−1}(i) + min{d_t/m, ε}}        ▷ candidates
    for i ∈ [m] do
        D_t(i) ← min{T ≥ t : P_T(i) ≥ N_{t−1}(i) + 1 − ε}          ▷ deadlines
    end for
    Choose s_t ∈ arg min_{i∈C(t)} D_t(i)
    y_{s_t,t} ← 1 and y_{i,t} ← 0 for i ∈ [m] \ {s_t}
end for
Output y
```

The feasibility of the algorithm follows from the lemma below.

▶ **Lemma 8.** *For any $t \in [n]$, there always exists a candidate, i.e., $C(t) \neq \emptyset$.*

Proof. Since

$$\begin{aligned}\sum_{i=1}^{m} \big(P_t(i) - N_{t-1}(i)\big) &= \sum_{i=1}^{m} \Big(\sum_{j=1}^{t} d_j x_{ij} - \sum_{j=1}^{t-1} d_j y_{ij}\Big) \\ &= \sum_{j=1}^{t-1} d_j \Big(\sum_{i=1}^{m} x_{ij} - \sum_{i=1}^{m} y_{ij}\Big) + d_t \sum_{i=1}^{m} x_{ij} \\ &= d_t,\end{aligned}$$

it follows that a candidate always exists as

$$\max_{i \in [m]} \big\{P_t(i) - N_{t-1}(i)\big\} \geq d_t/m \geq \min\{d_t/m, \varepsilon\}.$$ ◀

The following lemma justifies the definition of candidates:

▶ **Lemma 9.** *For any $i \in [m]$ and $t \in [n]$, $\Delta_t(i) \geq -1 + \varepsilon$.*

Proof. If i has not been selected at any time up to t, then $N_t(i) = 0$ and

$$\Delta_t(i) = P_t(i) \geq 0 > -1 + \varepsilon.$$

Otherwise, let $j \in [t]$ be the latest time such that $s_j = i$. Then

$$\begin{aligned}\Delta_t(i) &= P_t(i) - N_t(i) \\ &= P_t(i) - N_{j-1}(i) - d_j \\ &\geq P_j(i) - N_{j-1}(i) - d_j \\ &\geq \min\{d_j/m, \varepsilon\} - d_j \\ &\geq \min\{1/m, \varepsilon\} - 1 \\ &= -1 + \varepsilon,\end{aligned}$$

where the second inequality follows from the fact that i is a candidate at time j. ◀

Picking a candidate with the earliest deadline also ensures the upper bound:

▶ **Lemma 10.** *For any $i \in [m]$ and $t \in [n]$, $\Delta_t(i) \leq 1 - \varepsilon$.*

Proof. Define $I^+ := \{i \in [m] : \Delta_t(i) > 1-\varepsilon\}$ and assume towards contradiction that $I^+ \neq \emptyset$. We will need to study a carefully chosen time interval $(t', t]$ and the set of rows with deadline at most t at time t' to derive a contradiction. Define

$$t' := \max\{j \in [t] : P_t(s_j) < N_{j-1}(s_j) + 1 - \varepsilon\},$$

which by (1) is the first time that we pick some row with deadline later than t.

▷ Claim 11. The time t' is well-defined.

Proof. Let $i_0 \in \arg\min_{i\in[m]} \Delta_t(i)$. Then, since $\sum_{i\in[m]} \Delta_t(i) = 0$ and $I^+ \neq \emptyset$,

$$\Delta_t(i_0) < -\frac{1-\varepsilon}{|[m] \setminus I^+|} \leq -\frac{1-\varepsilon}{m-1} \leq -\varepsilon,$$

where the last inequality holds for $m \geq 2$. Obviously i_0 has been selected by time t, since otherwise $\Delta_t(i_0) = P_t(i_0) \geq 0$. As such let $t_0 \in [t]$ be the latest time with $s_{t_0} = i_0$, and thus

$$P_t(i_0) < N_t(i_0) - \varepsilon = N_{t_0}(i_0) - \varepsilon = N_{t_0-1}(i_0) + d_{t_0} - \varepsilon \leq N_{t_0-1}(i_0) + 1 - \varepsilon.$$

In particular, $t' \geq t_0$ is well-defined. ◁

Define

$$I := \{i \in [m] : P_t(i) \geq N_{t'-1}(i) + 1 - \varepsilon\},$$

which by (1) are precisely the rows with deadlines at most t at time t'.

▷ Claim 12. $s_{t'} \notin I$ and $s_j \in I$ for every $j \in (t', t]$.

Proof. It follows from definitions of t' and I that $s_{t'} \notin I$. For an arbitrary $j \in (t', t]$, it follows from the maximality of t' that

$$P_t(s_j) \geq N_{j-1}(s_j) + 1 - \varepsilon \geq N_{t'-1}(s_j) + 1 - \varepsilon.$$

Therefore, $s_j \in I$. ◁

▷ Claim 13. $I^+ \subseteq I$; in particular $I \neq \emptyset$.

Proof. For every $i \in I^+$,

$$P_t(i) > N_t(i) + 1 - \varepsilon \geq N_{t'-1}(i) + 1 - \varepsilon,$$

which implies $i \in I$. Thus, $I^+ \subseteq I$. ◁

▷ Claim 14. $P_{t'}(i) < N_{t'-1}(i) + \varepsilon$ for every $i \in I$.

Proof. By the definition of t' and deadline, $D_{t'}(s_{t'}) > t$, which means $s_{t'} \notin I$. By the choice of $s_{t'}$ in the algorithm, all candidates $i \in C(t')$ satisfy $D_{t'}(i) > t$. In other words, $I \cap C(t') = \emptyset$. It follows from (2) that for every $i \in I$,

$$P_{t'}(i) < N_{t'-1}(i) + \min\{d_{t'}/m, \varepsilon\} \leq N_{t'-1}(i) + \varepsilon. \quad ◁$$

▷ Claim 15. $P_t(i) \geq N_t(i) - \varepsilon$ for every $i \in I$.

Proof. For an arbitrary $i \in I$, if there does not exist $j \in (t', t]$ with $s_j = i$, then

$$P_t(i) \geq N_{t'-1}(i) + 1 - \varepsilon = N_t(i) + 1 - \varepsilon > N_t(i) - \varepsilon,$$

where we used the fact that $s_{t'} \neq i$ as $s_{t'} \notin I$. Otherwise, let $j \in (t', t]$ be the latest time with $s_j = i$. Then by the maximality of t', we have

$$P_t(i) \geq N_{j-1}(i) + 1 - \varepsilon \geq N_{j-1}(i) + d_j - \varepsilon = N_j(i) - \varepsilon = N_t(i) - \varepsilon. \qquad \triangleleft$$

Finally, $|I| \leq m - 1$ since by Claim 13, $I \subseteq [m] \setminus \{s_{t'}\}$. In other words, $0 \leq 1 - 2\varepsilon|I|$. Putting everything together,

$$\begin{aligned}
\sum_{j\in(t',t]} d_j &\leq 1 - 2\varepsilon|I| + \sum_{j\in(t',t]} d_j \\
&= 1 - 2\varepsilon|I| + \sum_{i\in I}(N_t(i) - N_{t'-1}(i)) && \text{(Claim 12)} \\
&= 1 + \sum_{i\in I}(N_t(i) - \varepsilon - N_{t'-1}(i) - \varepsilon) \\
&< \sum_{i\in I}(P_t(i) - N_{t'-1}(i) - \varepsilon) && \text{(Claim 15+Claim 13)} \\
&< \sum_{i\in I}(P_t(i) - P_{t'}(i)) && \text{(Claim 14+Claim 13)} \\
&\leq \sum_{i\in[m]}(P_t(i) - P_{t'}(i)) \\
&= \sum_{j\in(t',t]} d_j,
\end{aligned}$$

This is a contradiction, from which we conclude $I^+ = \emptyset$. ◀

Proof of Theorem 1. The bounds follow from Lemmas 9 and 10. It remains to show that the algorithm can be implemented in linear time. We may precompute $P_t(i)$ for all $t \in [n], i \in [m]$ via prefix sum, and we may update $N_t(i)$ for all $i \in [m]$ after the t-th iteration. The deadlines $D_t(i)$ may also be computed in linear amortized time by incrementing them each iteration until $P_T(i) \geq N_{t-1}(i) + 1 - \varepsilon$ as $T \leq n$. ◀

4 Application for flow-time scheduling

Let $M = [m]$ be a set of machines, and $J = [n]$ be a set of jobs. Each job $j \in J$ can be scheduled on a subset of machines $M_j \subseteq M$. Each job $j \in [n]$ has a processing time d_j and release time r_j. Let C_j be the completion time of j. Order the jobs by their release times $r_1 \leq r_2 \leq \cdots \leq r_n$. We want to schedule each job to a machine such that the maximum flow-time $\max_{j\in[n]}(C_j - r_j)$ is minimized. Consider the following linear programming relaxation for this problem:

$$
\begin{aligned}
\min\ & T \\
s.t.\ & \sum_{i=1}^{m} x_{ij} = 1\ \forall j \in [n] \\
& \sum_{j=s}^{t} x_{ij} d_j \leq (r_t - r_s) + T,\ \forall i \in [m],\ 1 \leq s \leq t \leq n \\
& x \geq 0 \\
& x_{ij} = 0,\ \forall i, j : i \notin M_j.
\end{aligned}
\tag{3}
$$

Morell and Skutella [23] show that Conjecture 7 would imply the integrality gap of (3) is at most 3. Efficiently computing such an assignment y would lead to a 3-approximation algorithm for maximum flow-time scheduling. It follows readily from their proof that upper-bounding $\sum_{j=s}^{t} d_j(y_{ij} - x_{ij})$ for every machine i and interval of jobs defined by $1 \leq s \leq t \leq n$ suffices to prove the desired approximation ratio. We summarize this in the following lemma and give its proof for completeness.

▶ **Lemma 16** ([23]). *Let m, n be positive integers and $d \in \mathbb{R}^n_{>0}$. If for every fractional assignment $x \in [0,1]^{m\times n}$, there is an assignment $y \in \{0,1\}^{m\times n}$ so that $y_{ij} = 0$ whenever $x_{ij} = 0$ and*

$$\sum_{j=s}^{t} d_j(y_{ij} - x_{ij}) \leq \alpha \cdot \max_{j\in[n]} d_j \quad \forall i \in [m], 1 \leq s \leq t \leq n, \tag{4}$$

then the integrality gap of the linear programming relaxation (3) is at most $(\alpha + 1)$. A polynomial time algorithm to compute such an assignment y would lead to an $(\alpha + 1)$-approximation algorithm for maximum flow-time scheduling.

Proof. Denote $d_{\max} := \max_{j\in[n]} d_j$. Let (x, T) be the optimal solution to the linear program (3). Let OPT be the minimum maximum flow-time. Suppose y is the assignment satisfying (4). Let $i \in [m]$ and $1 \leq s \leq t \leq n$. The total processing time of jobs assigned to machine i and released between r_s and r_t is

$$
\begin{aligned}
\sum_{j=s}^{t} y_{ij} d_j \leq & \sum_{j=s}^{t} x_{ij} d_j + \alpha \cdot d_{\max} \\
\leq & (r_t - r_s) + T + \alpha \cdot \mathrm{OPT} \\
\leq & (r_t - r_s) + (1 + \alpha)\,\mathrm{OPT},
\end{aligned}
$$

where the first inequality follows from (4). The second inequality follows from the fact that the job of size $d_{\max}$ has flow-time at least $d_{\max}$. The last inequality follows from the fact that $T \leq \mathrm{OPT}$. Suppose job t is scheduled on i. Let r_s be the latest time before t such that i is idle when some job s scheduled to i is released (which exists because i is idle when the first job scheduled to i is released). Then, the completion time of t is $C_t = r_s + \sum_{j=s}^{t} y_{ij} d_j$. The flow-time of t is $C_t - r_t = r_s + \sum_{j=s}^{t} y_{ij} d_j - r_t \leq (1 + \alpha)\,\mathrm{OPT}$. Therefore, the maximum flow-time is at most $(1 + \alpha)\,\mathrm{OPT}$. ◀

Previously, a $(3 - \frac{2}{m})$-approximation algorithm was known for identical machines [7, 20]. Combining Theorem 1 and Lemma 16, we get a $(3 - \frac{1}{m-1})$-approximation algorithm for this setting, matching the current best approximation ratio asymptotically. Yet, their algorithm

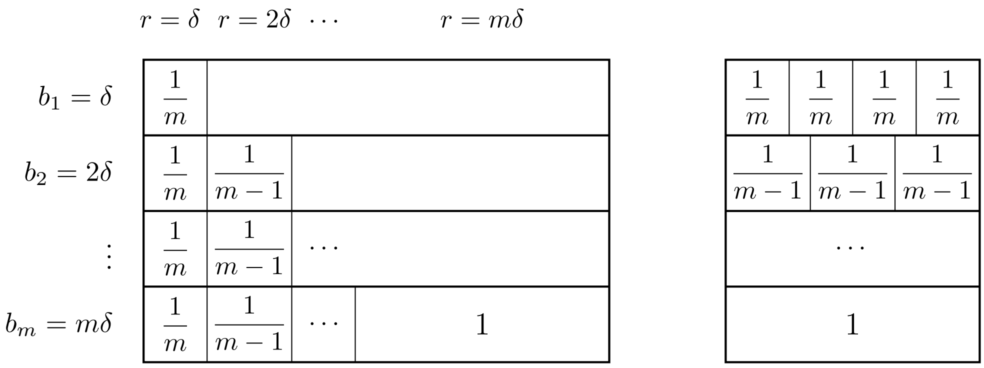

Figure 2 There are m machines $M = [m]$ with closing times $b_i = i \cdot \delta$ for some $\delta \ll \frac{1}{m}$. There are m batches of jobs, released at times $\delta, 2\delta, ..., m\delta$. The j-th batch has $(m - j + 1)$ jobs, each with processing time $\frac{1}{m-j+1}$. Left: FIFO schedules the j-th batch to machines $j, j+1, ..., m$, one for each, because those are the machines that are not closed yet. The maximum flow-time is $\left(\sum_{j\in[m]} \frac{1}{j}\right) - m\delta = \Omega(\log m)$. Right: OPT schedules the j-th batch to machine j, which is feasible because the j-th batch is released no later than machine j is closed. The maximum flow-time is $1 - m\delta \le 1$.

(FIFO) follows a much simpler greedy approach, which orders jobs by their release times and assigns them online to the machine with the least remaining processing time of jobs that have been assigned to it.

We study a variation where each machine i has a closing time b_i, and a job can be scheduled on machine i only if $r_j \le b_i$. We begin by observing that FIFO has a lower bound $\Omega(\log m)$ for the setting of closing times (see Figure 2). We give a $(3 - \frac{1}{m-1})$-approximation algorithm for the setting of closing times. We start by observing that Algorithm 1 has the following property.

▶ **Proposition 17.** *Given $d \in \mathbb{R}^n_{>0}$, $a \in [n]^m$ and a fractional assignment $x \in [0,1]^{m\times n}$ so that $x_{ij} = 0\ \forall i, j : j < a_i$, there is a linear-time algorithm that computes an assignment $y \in \{0,1\}^{m\times n}$ so that $y_{ij} = 0\ \forall i, j : j < a_i$ and*

$$\Big|\sum_{j=1}^{t} d_j(x_{ij} - y_{ij})\Big| \le \left(1 - \frac{1}{2m-2}\right) \cdot \max_{j\in[n]} d_j \quad \forall i \in [m], t \in [n].$$

Proof. Note that for every $t < a_i$, $P_t(i) = \sum_{j=1}^{t} d_j x_{ij} = 0$, since $x_{ij} = 0\ \forall j < a_i$. Therefore, for every $t < a_i$, $P_t(i) = 0 < N_{t-1}(i) + \min\{d_t/m, \varepsilon\}$, which means i is not a candidate at time t. Thus, the assignment y returned by Algorithm 1 satisfies $y_{ij} = 0\ \forall j < a_i$. ◀

▶ **Lemma 18.** *Given release times $r \in \mathbb{R}^n_{\ge 0}$ of jobs, closing times $b \in \mathbb{R}^m_{\ge 0}$ of machines, $d \in \mathbb{R}^n_{>0}$ and a fractional assignment $x \in [0,1]^{m\times n}$ so that $x_{ij} = 0\ \forall i, j : r_j > b_i$, there is a linear-time algorithm that computes an assignment $y \in \{0,1\}^{m\times n}$ so that $y_{ij} = 0\ \forall i, j : r_j > b_i$ and*

$$\sum_{j=s}^{t} d_j(y_{ij} - x_{ij}) \le \left(2 - \frac{1}{m-1}\right) \cdot \max_{j\in[n]} d_j \quad \forall i \in [m], 1 \le s \le t \le n.$$

Proof. For every $i \in [m]$, define $a_i := \max\{j : r_j \le b_i\}$. Then, assignment x satisfies $x_{ij} = 0\ \forall j > a_i$. Applying Proposition 17 with $x'_{ij} := x_{i,n-j+1}$ and $a'_i := n - a_i + 1$, we conclude that there exists an assignment y' satisfying $y'_{ij} = 0\ \forall j < a'_i$, and thus $y_{ij} := y'_{i,n-j+1}$ satisfies $y_{ij} = 0\ \forall j > a_i$. Moreover, for every $i \in [m]$ and $1 \le s \le t \le n$,

$$\begin{aligned}
\sum_{j=s}^{t} d_j(y_{ij} - x_{ij}) &= \sum_{j=n-t+1}^{n-s+1} d_j(y'_{ij} - x'_{ij}) \\
&= \sum_{j=1}^{n-s+1} d_j(y'_{ij} - x'_{ij}) - \sum_{j=1}^{n-t} d_j(y'_{ij} - x'_{ij}) \\
&\le 2\Big(1 - \frac{1}{2m-2}\Big) \cdot \max_{j\in[n]} d_j \\
&= \Big(2 - \frac{1}{m-1}\Big) \cdot \max_{j\in[n]} d_j.
\end{aligned}$$ ◀

Proof of Theorem 2. For every $j \in J$, let $M_j = \{i \in [m] : r_j \le b_i\}$. It follows from Lemmas 18 and 16 that there is a $(3 - \frac{1}{m-1})$-approximation algorithm for maximum flow-time scheduling with closing times. ◀

5 Lower bounds

Proof of Proposition 3. If $m = 2$ we can simply take $x_{1,1} = x_{2,1} = \frac{1}{2}$. Otherwise, let $x \in \mathbb{R}^{m\times(m-1)}$ be defined as $x_{i,1} = \frac{1}{2m-2}$ for $i < m$ and $x_{m,1} = \frac{1}{2}$, $x_{m,j} = 0$ for $j > 1$ and $x_{i,j} = \frac{1}{m-1}$ otherwise (see below). Assume towards contradiction that there exists some assignment $y \in \{0,1\}^{m\times(m-1)}$ so that $|\sum_{j\in[t]}(x_{ij} - y_{ij})| < 1 - \frac{1}{2m-2}\ \forall i \in [m], t \in [m-1]$. If $y_{i,1} = 1$ for some $i \ne m$ then already $|x_{i,1} - y_{i,1}| = 1 - \frac{1}{2m-2}$. Otherwise, $y_{m,1} = 1$ and yet each $i \in [m-1]$ must be assigned at least once, since $\sum_{j\in[m-1]} x_{ij} = 1 - \frac{1}{2m-2}$. But there are only $m-2$ columns left, so one of $i \in [m-1]$ must never be assigned, which is a contradiction. ◀

$$\begin{array}{c|cccc|}
 & 1 & 2 & \cdots & m-1 \\
\hline
1 & \frac{1}{2m-2} & \frac{1}{m-1} & \cdots & \frac{1}{m-1} \\
2 & \frac{1}{2m-2} & \frac{1}{m-1} & \cdots & \boxed{\frac{1}{m-1}} \\
\vdots & \vdots & \vdots & \ddots & \vdots \\
m-1 & \frac{1}{2m-2} & \boxed{\frac{1}{m-1}} & \cdots & \frac{1}{m-1} \\
m & \boxed{\frac{1}{2}} & 0 & \cdots & 0 \\
\hline
\end{array}$$

Proof of Proposition 4. Consider the fractional assignment $x \in [0,1]^{3\times n}$ below where there are $n = 1 + 2\cdot p$ columns, with $p := \lceil \frac{1-\delta}{2\delta} \rceil$ repeating pairs of columns after the first. Assume towards contradiction that there exists some assignment $y \in \{0,1\}^{3\times n}$ so that $|\sum_{j\in[t]}(x_{ij} - y_{ij})| < 1 - \delta$ for all $t \in [n]$. We have to assign $y_{3,1} = 1$, for otherwise already

$|x_{3,1} - y_{3,1}| = 1 - \delta$. Since $\sum_{j \in [2s]} x_{1,j} = s - \delta$, to keep the discrepancy of the first row bounded, we must assign $y_{1,2s} = 1$ and $y_{1,2s+1} = 0$ for $s \in [p]$. Then we are forced to assign $y_{3,2s+1} = 1$ for every $s \in [p]$ because $x_{2,2s+1} = 0$ implies $y_{2,2s+1} = 0$. Yet the second row sums to $2\delta \cdot p \geq 1 - \delta$ without ever being assigned, a contradiction. ◀

$$\begin{bmatrix} \delta & 1-2\delta & 2\delta & \cdots & 1-2\delta & 2\delta \\ 0 & 2\delta & 0 & \cdots & 2\delta & 0 \\ 1-\delta & 0 & 1-2\delta & \cdots & 0 & 1-2\delta \end{bmatrix}$$

Proof of Proposition 5. A construction is given by $x_{i,j} = v_i$ for all $i \in [3], j \in [100]$ where $v = (0.01, 0.48, 0.51)$, which can be computationally verified to achieve interval discrepancy equal to 1.32. See Appendix A. ◀

6 Open problems

We ask for a version with non-uniform weights which may shed light on the *unrelated machines* scheduling, where job j has processing time d_{ij} on machine i:

▶ **Conjecture 19.** *Let m, n be positive integers and $d \in \mathbb{R}^{m\times n}_{>0}$. For every fractional assignment $x \in [0,1]^{m\times n}$, there is an assignment $y \in \{0,1\}^{m\times n}$ so that for every $i \in [m]$ and $t \in [n]$,*

$$\Big|\sum_{j\in[t]} d_{ij}(x_{ij} - y_{ij})\Big| \leq \max_{i\in[m], j\in[n]} d_{ij}.$$

The construction in Proposition 4 applied to $d_{ij} \in \{0, d_j\}$ also shows a lower bound of $(1-\delta)\cdot \max_{i\in[m],j\in[n]} d_{ij}$ for all $\delta > 0$ for this setting. The argument in [18] shows that the bound holds for the non-prefix version, i.e. $t = n$.

We also formulate a *weighted committee assignment problem*:

▶ **Conjecture 20.** *Let m, n be positive integers and $d \in \mathbb{R}^{n}_{>0}$. For every $x \in [0,1]^{m\times n}$ so that $n_j := \sum_{i\in[m]} x_{ij} \in \mathbb{N}$ for all $j \in [n]$, there is $y \in \{0,1\}^{m\times n}$ so that $\sum_{i\in[m]} y_{ij} = n_j$ for all $j \in [n]$ and*

$$\Big|\sum_{j\in[t]} d_j(x_{ij} - y_{ij})\Big| \leq \max_{j\in[n]} d_j \quad \forall i \in [m], t \in [n].$$

It is plausible that Conjectures 7, 19 and 20 may all be combined:

▶ **Conjecture 21.** *Let m, n be positive integers and $d \in \mathbb{R}^{m\times n}_{>0}$. For every $x \in [0,1]^{m\times n}$ so that $n_j := \sum_{i\in[m]} x_{ij} \in \mathbb{N}$ for all $j \in [n]$, there is $y \in \{0,1\}^{m\times n}$ so that $\sum_{i\in[m]} y_{ij} = n_j$ for all $j \in [n]$, $y_{ij} = 0$ whenever $x_{ij} = 0$, and*

$$\Big|\sum_{j\in[t]} d_{ij}(x_{ij} - y_{ij})\Big| \leq \max_{i\in[m], j\in[n]} d_{ij} \quad \forall i \in [m], t \in [n].$$

References

1 Miklós Ajtai, James Aspnes, Moni Naor, Yuval Rabani, Leonard J. Schulman, and Orli Waarts. Fairness in scheduling. In *Proceedings of the Sixth Annual ACM–SIAM Symposium on Discrete Algorithms (SODA)*, pages 477–485. SIAM, 1995. URL: http://dl.acm.org/citation.cfm?id=313651.313796.

2 Mohammed Majthoub Almoghrabi, Martin Skutella, and Philipp Warode. Integer and unsplittable multiflows in series-parallel digraphs. In *Proceedings of the 26th International Conference on Integer Programming and Combinatorial Optimization (IPCO 2025)*, volume 15620 of *Lecture Notes in Computer Science*, Baltimore, MD, USA, 2025. Springer. Also appears as arXiv preprint arXiv:2412.05182v2, July 22, 2025. `doi:10.48550/arXiv.2412.05182`.

3 Omer Angel, Alexander E. Holroyd, James B. Martin, and James Propp. Discrete low-discrepancy sequences, 2009. `arXiv:0910.1077`.

4 Wojciech Banaszczyk. On series of signed vectors and their rearrangements. *Random Structures & Algorithms*, 40(3):301–316, 2012. `doi:10.1002/rsa.20373`.

5 Nikhil Bansal and Janardhan Kulkarni. Minimizing flow-time on unrelated machines. In *Proceedings of the Forty-Seventh Annual ACM Symposium on Theory of Computing*, STOC '15, pages 851–860, New York, NY, USA, 2015. Association for Computing Machinery. `doi:10.1145/2746539.2746601`.

6 Nikhil Bansal, Lars Rohwedder, and Ola Svensson. Flow time scheduling and prefix Beck–Fiala. In *Proceedings of the 54th Annual ACM SIGACT Symposium on Theory of Computing (STOC)*, pages 331–342. ACM, 2022. `doi:10.1145/3519935.3520077`.

7 Michael A. Bender, Soumen Chakrabarti, and S. Muthukrishnan. Flow and stretch metrics for scheduling continuous job streams. In *Proceedings of the 9th Annual ACM–SIAM Symposium on Discrete Algorithms (SODA)*, pages 270–279, San Francisco, CA, 1998. SIAM. URL: `https://dl.acm.org/doi/10.5555/314613.314715`.

8 Valérie Berthé, Olivier Carton, Nicolas Chevallier, Wolfgang Steiner, and Reem Yassawi. A dynamical view of Tijdeman's solution of the chairman assignment problem. *Combinatorics and Number Theory*, 13(4):377–412, 2024. `doi:10.2140/cnt.2024.13.377`.

9 Imre Bárány. On the power of linear dependencies. In *Building Bridges: Between Mathematics and Computer Science*, volume 19 of *Bolyai Society Mathematical Studies*, pages 31–45. Springer, 2010.

10 Imre Bárány and V. S. Grinberg. On some combinatorial questions in finite-dimensional spaces. *Linear Algebra and its Applications*, 41:1–9, 1981.

11 Don Coppersmith, Tomasz J. Nowicki, Giuseppe A. Paleologo, Charles Philippe Tresser, and Chai Wah Wu. The optimality of the online greedy algorithm in carpool and chairman assignment problems. *ACM Transactions on Algorithms*, 7(3):37:1–37:22, 2011. `doi:10.1145/1978782.1978792`.

12 Yefim Dinitz, Naveen Garg, and Michel X. Goemans. On the single-source unsplittable flow problem. *Combinatorica*, 19(1):17–41, 1999. `doi:10.1007/s004930050043`.

13 Yuval Efron, Shyamal Patel, and Cliff Stein. A simple algorithm for dynamic carpooling with recourse. In *Proceedings of the 2025 Symposium on Simplicity in Algorithms (SOSA)*, pages 196–201, 2025. arXiv preprint arXiv:2411.07553.

14 Ronald Fagin and John H. Williams. A fair carpool scheduling algorithm. *IBM Journal of Research and Development*, 27(2):133–139, March 1983. `doi:10.1147/rd.272.0133`.

15 Anupam Gupta, Vijaykrishna Gurunathan, Ravishankar Krishnaswamy, Amit Kumar, and Sahil Singla. Online discrepancy with recourse for vectors and graphs. In *Proceedings of the 2022 ACM–SIAM Symposium on Discrete Algorithms (SODA)*, 2022. arXiv preprint arXiv:2111.06308. `arXiv:2111.06308`.

16 Alexander E. Holroyd and James Propp. Rotor walks and Markov chains. In Manuel E. Lladser, Robert S. Maier, Marni Mishna, and Andrew Rechnitzer, editors, *Algorithmic Probability and Combinatorics*, volume 520 of *Contemporary Mathematics*, pages 105–126. American Mathematical Society, Providence, RI, 2010. `arXiv:0904.4507`.

17 Janardhan Kulkarni, Victor Reis, and Thomas Rothvoss. Optimal online discrepancy minimization. In *Proceedings of the 56th Annual ACM Symposium on Theory of Computing*, STOC 2024, pages 1832–1840, New York, NY, USA, 2024. Association for Computing Machinery. `doi:10.1145/3618260.3649720`.

18 Jan Karel Lenstra, David B Shmoys, and Éva Tardos. Approximation algorithms for scheduling unrelated parallel machines. *Mathematical programming*, 46(1):259–271, 1990. doi:10.1007/BF01585745.

19 Monaldo Mastrolilli. Notes on max flow time minimization with controllable processing times. *Computing*, 71(4):375–386, 2003. doi:10.1007/s00607-003-0029-z.

20 Monaldo Mastrolilli. Scheduling to minimize max flow time: Off-line and on-line algorithms. *International Journal of Foundations of Computer Science*, 15(2):385–401, 2004. doi:10.1142/S0129054104002480.

21 H. G. Meijer. On a distribution problem in finite sets. *Indagationes Mathematicae*, 35:9–17, 1973. doi:10.1016/1385-7258(73)90015-2.

22 H. G. Meijer and H. Niederreiter. On a distribution problem in finite sets. *Compositio Mathematica*, 25(2):153–160, 1972. URL: https://www.numdam.org/item/CM_1972__25_2_153_0/.

23 Sarah Morell and Martin Skutella. Single source unsplittable flows with arc-wise lower and upper bounds. In *Proceedings of IPCO 2020*, volume 12125 of *Lecture Notes in Computer Science*, pages 294–306. Springer, 2020. doi:10.1007/978-3-030-45771-6_23.

24 H. Niederreiter. On the existence of uniformly distributed sequences in compact spaces. *Compositio Mathematica*, 25(1):93–99, 1972. URL: https://www.numdam.org/item/CM_1972__25_1_93_0/.

25 Harald Niederreiter. A distribution problem in finite sets. In S. K. Zaremba, editor, *Applications of Number Theory to Numerical Analysis*, pages 237–248. Academic Press, New York, 1972. Proc. Symposium, Université de Montréal, 1971.

26 R. Tijdeman. On a distribution problem in finite and countable sets. *Journal of Combinatorial Theory, Series A*, 15:129–137, 1973. doi:10.1016/S0097-3165(73)80002-0.

27 Robert Tijdeman. The chairman assignment problem. *Discrete Mathematics*, 32(3):323–330, 1980. doi:10.1016/0012-365X(80)90269-1.

28 Robert Tijdeman. A progress report on discrepancy. *Astérisque*, pages 175–185, 1982.

29 Vera Traub, Laura Vargas Koch, and Rico Zenklusen. Single-source unsplittable flows in planar graphs. In *Proceedings of the 2024 Annual ACM–SIAM Symposium on Discrete Algorithms (SODA)*, pages 639–668. SIAM, 2024. doi:10.1137/1.9781611977912.24.

A Computer program

We use the following Julia function to verify Proposition 5:

Listing 1 Input: $m, n \in \mathbb{N}$ and a fractional assignment $x \in [0,1]^{m\times n}$. Output: the smallest Δ for which there exists an assignment $y \in \{0,1\}^{m\times n}$ so that $|\sum_{j\in[s,t]}(x_{ij} - y_{ij})| \leq \Delta$ for all $i \in [m]$ and $1 \leq s \leq t \leq n$.

```julia
using JuMP, Gurobi

function interval_disc(m,n,x)
    opt = Model(Gurobi.Optimizer)
    @variable(opt, Delta >= 0) # interval discrepancy
    @variable(opt, y[1:m,1:n], Bin) # assignment

    for j in 1:n
        @constraint(opt, sum(y[:,j]) == 1)
    end
    for i in 1:m
        for s in 1:n
            for t in s:n
                @constraint(opt, sum(y[i,s:t]) - sum(x[i,s:t]) <= Delta)
                @constraint(opt, sum(y[i,s:t]) - sum(x[i,s:t]) >= -Delta)
            end
        end
    end

    @objective(opt, Min, Delta)
    optimize!(opt)
    return value(Delta)
end
```