Informatik-Bericht Nr. 2013-1

Schriftenreihe Fachbereich Informatik, Fachhochschule Trier

Student Sectioning for Fixed Timetables

M. Dostert[*] A. Politz[*] H. Schmitz[*][†]

March 1, 2013

Abstract

Based on a detailed complexity analysis of various student-sectioning models we show that the following fundamental question in student sectioning can be efficiently decided in polynomial time:

> Is it possible to assign m students to k sectioned courses w.r.t. a given timetable with l timeslots, such that the individual capacities of all sections are not exceeded and no student has more than one appointment per timeslot?

For this setting we provide an algorithm to compute an optimal assignment of students to sections in $O(k^2 l^2 \log(\text{sum}_A))$ where sum_A is the sum of all specified section capacities. We also identify structural properties of solutions that allow their succinct representation. On the other hand, we prove that adding any single of the following constraints turns the above question into an NP-complete problem:

***Course-selection constraint*:** Students select their courses from a larger set and this selection must be respected.

***Timeslot-constraint*:** Students have individual timeslot restrictions.

***Multiple-event constraint*:** Sections may have multiple events, i.e., the same section is assigned to more than one timeslot in the given timetable. There must be no timeslot clashes between all section-events for each student.

Hence our investigation contributes to the practical solvability of student-sectioning problems and it gives insight into the location of the borderline between efficiently solvable and computational hard problem variations.

1 Introduction

We consider an aspect of educational timetabling that appears in schools and universities likewise. Here courses often have *sections* (sometimes called *subevents*) where the same content is taught multiple times during a week, mostly due to capacity requirements. Usually no information about student-to-section assignment is known prior to timetabling because students select courses, not a particular section of a course. Hence sectioned courses put extra challenges on timetabling algorithms because a solution is a timetable *plus* a student-to-section assignment – coming along with additional constraints. The latter typically comprise constraints like maximum section-sizes and conflict-free timetables for the sectioned students. Algorithms on various models with different combinations of constraints have been proposed in the literatur, among them approaches where student sectioning is carried out before [AYH04, AH05, SH10], during [AF89, Car00, MRB04, MM10] or after timetabling [AVCT00, CKL03]. In this paper we

[*]Trier University of Applied Sciences, Germany
[†]Corresponding author: h.schmitz@hochschule-trier.de

take a careful look at sectioning problems for fixed timetables, i.e., we consider ‘pure’ sectioning problems that have a timetable already in the input. In this way we follow the post-timetabling approach to student sectioning and we identify the additional computational effort that originates from student sectioning.

Our models. We give an informal description of student-sectioning models considered in this paper in order to summarize our results below, and begin with the Basic Student Sectioning Problem (BSS), see Section 2 for a formal definition. Here a timetable for k courses is given, and each course consists of a number of sections. There is one event for each section scheduled in the timetable and each section has an individual maximum number of possible participants. We would like to determine the maximum number of students that can attend all k courses and assign them to sections in a way such that

(D) each student has at most one appointment per timeslot (*disjoint-sections constraint*), and

(M) the maximum sizes of all sections are not exceeded (*maximum-capacity constraint*).

Both constraints are very often part of student-sectioning models mentioned in the literature, e.g. [AVCT00, CKL03, MM10]. As a small example consider the following timetable with three sectioned courses, each section given by its capacity:

timeslot	course c_1	course c_2	course c_3
t_1	$10\,{*}{*}$	5	$15{*}$
t_2	$10{*}$	$5\,{*}{*}$	0
t_3	0	0	$5\,{*}{*}$
t_4	10	$15{*}$	10

Without violating (D) or (M) we can assign 10 students to the sections (c_1, t_2), (c_2, t_4) and (c_3, t_1) marked with $*$, and another 5 students to the sections (c_1, t_1), (c_2, t_2) and (c_3, t_3) marked with $**$. This assignment can not be extended any more because the remaining capacities are in timeslots t_1 and t_4 only, but we want to assign students to $k = 3$ courses. However, there exists a feasible assignment with more than 15 students, see example after Corollary 3.4. We call BSS a *basic* model because we assume

- that students have to attend all scheduled courses,
- that students have no timeslot restrictions, and
- that there is only one event per section.

Moreover, it is assumed w.l.o.g. that there is at most one section per course and timeslot. Note that if this is not the case, we may simply add the respective capacities and split the assigned students arbitrarily after sectioning.

Observe that already constraints (D) and (M) catch key issues in student sectioning because they reflect the interests of two main stakeholders: Students certainly want their schedules to be conflict-free, while institutions need to consider their room capacities – the latter being the main reason to have sectioned courses at all. Indeed, if only one of constraints (D) and (M) is present, then it is easy to maximize the number of assigned students, but usually at the expense of massively violating the interests of one of the stakeholders which in turns leads to useless timetables: If only (M) is taken into account then the optimal number of students is the minimum of the sums of all section capacities of each course, which is a trivial bound that no solution can exceed. On the other hand, if only (D) is present then we can assign an arbitrary number of students if and only if we can assign a single student at all, which in turn holds if and only if there are not more courses than non-empty timeslots.

While the basic model BSS is well applicable in some scenarios, e.g., in curriculum-based scheduling where all students attend the same courses, there may well be more constraints that emerge from practice. So we also investigate extended versions of BSS by adding the following input data and constraints, corresponding to the above limitations of the basic model:

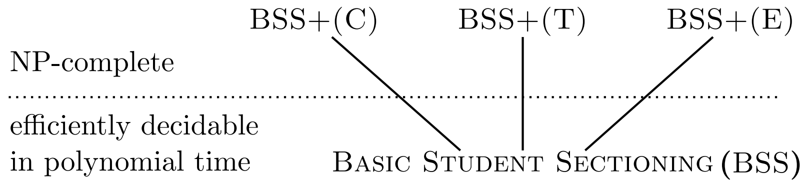
(C) Students select their courses from a larger set of courses offered, and this selection must be respected (*course-selection constraint*).

(T) Students have individual timeslot restrictions, i.e., they must not be assigned to some explicitly excluded timeslots (*timeslot constraint*).

(E) Sections may have multiple events, i.e., the same section is assigned to more than one timeslot in the input timetable. The sectioning of students must be pairwise conflict-free with respect to timeslot clashes between all section-events (*multiple-event constraint*).

We concentrate on these constraints since they (or slight variations of them) appear in most models for student sectioning mentioned in the literature, often in combined form, e.g., [AF89, Car00, AH05]. Adding these contraints to the basic model separately leads in a natural way to the definition of BSS-versions BSS+(C), BSS+(T) and BSS+(E), respectively.

Our results. We begin in Section 2 with a careful modelling of sectioning problems in terms of matchings in bipartite graphs, and describe feasible solutions as matchings between courses and timeslots that must satisfy additional requirements. Our definitions of BSS and the extended versions BSS+X constitute a uniform framework for sectioning problems which also allows to talk about combined problems BSS+X for X$\subseteq \{(C), (T), (E)\}$ in a natural but precise way. Our complexity analysis of these decision problems in Section 3 reveals the borderline between efficiently solvable and NP-hard combinations of constraints, see Theorems 3.7 and 3.11, and Corollary 3.12, which can be summarized as follows:


BSS+(C)
BSS+(T)
BSS+(E)
NP-complete
efficiently decidable in polynomial time
Basic Student Sectioning (BSS)

The polynomial-time decidability of BSS opens the possibility to evaluate candidate-timetables as part of timetabling algorithms efficiently with respect to this measure. To the best of our knowledge, this is the first complexity analysis of student sectioning for fixed timetables – with one exception: In [CKL03] the authors sketch a proof for NP-hardness of their model from which one can conclude that also BSS+(E) is NP-hard. In their reduction, the number of events per section is not limited. We strengthen this result for BSS+(E) and show that it is NP-complete even if sections only have two events.

So it turns out that not all variations of student sectioning are hard, and we further exploit this in Section 4 to compute also optimal *solutions* in terms of individual timetables for students efficiently. However, on input of a timetable with section capacities the number of assignable students is usually not polynomially bounded in the input length, which implies that we can not compute solutions efficiently as a list that assigns a timetable to each student. We show that there always exists an optimal solution that has only a small number of *different* timetables for all assignable students. Together with a succinct representation of these solutions we finally get a $O(k^2 l^2 \log(\mathrm{sum}_A))$-time algorithm that computes optimal solutions for BSS, and which is truly polynomial in the input length. As a final result we show that it is NP-hard to determine the minimal number of different timetables for a BSS-instance (Theorem 4.5). On the technical side, our work is inspired by [dW85] where class-teacher models and course scheduling are discussed with an emphasis on graph-theoretical models.

2 Problem Definitions

We fix some notations and formalize our problem definitions. Unless stated otherwise, all variables $i, j, k, l, m, \dots$ are non-negative integers. Let $[k] = \{1, \dots, k\}$ for $k \geq 0$. Sets of courses, timeslots and students are denoted as

$$C = \{c_1, \dots, c_k\}, T = \{t_1, \dots, t_l\} \text{ and } S = \{s_1, \dots, s_p\},$$

respectively. If the context is clear we identify courses, timeslots and students simply by their index, e.g., some course $j \in [k]$.

We model sectioning problems in terms of bipartite graphs with courses and timeslots as node partitions. Let $G = (V, E)$ with $V = C \cup T$ be some undirected bipartite multigraph, i.e., E is a multiset of edges, each edge joining some node in C with some node in T. For $v \in V$ denote the number of edges incident to v by $d(v)$ as the degree of v, and denote by $\Delta(G)$ the maximum degree of the nodes in G. For $e \in E$ we write $m(e)$ for the number of edges e in multiset E. A matching in G is a set of edges $M \subseteq E$ such that no two edges in M share a common node. If $|M| = k$ we say that M is a k-matching. We denote by C_M (T_M, resp.) the set of nodes from C (T, resp.) that appear in M. A set $\mathcal{M}$ of matchings is called edge-disjoint if all $M \in \mathcal{M}$ are pairwise disjoint, i.e., no instance of some edge appears in more than one M. For one of the sectioning problems we also consider hypergraphs $H = (C \cup T, E)$ where each edge e is a subset of $C \cup T$ with $|e| \geq 2$. Such a hypergraph is called bipartite if $|e \cap C| = 1$ for all e. A matching in H is a subset of pairwise disjoint edges. The other notations apply likewise.

All our sectioning problems have in common that (some encoding of) a timetable is part of the input. We may assume in our context that such a timetable only comprises data relevant for sectioning. So we refrain from taking teachers, rooms, equipment and other resources as part of the input. On the other hand, we assume that a given timetable is complete in the sense that all events for all sections are scheduled, and that all capacity data is provided. However, timeslots without any sections can be omitted.

With **P** (**NP**, resp.) we denote the complexity class of all decision problems, that can be decided (accepted, resp.) by some determinstic (nondeterministic, resp.) algorithm whose running time can be upper bounded by some polynomial in the input length. For background on complexity classes and the notion of NP-completeness we refer the reader to standard textbooks, e.g. [GJ80, Pap94].

2.1 Formalization of the Basic Model

Recall that in the basic model each section has exactly one event, which allows to identify the section of a course by its timeslot. So together with the section capacities we may represent the input timetable as a matrix $A = (a_{i,j}) \in \mathbb{N}^{l \times k}$ where l is the number of timeslots, k is the number of courses, and $a_{i,j}$ is the maximum size of the section for course c_j scheduled in timeslot t_i. Note that $a_{i,j} = 0$ if and only if there is no section of course c_j in timeslot t_i. We denote by sum_A the sum of all entries in A. E.g., the matrix of the introductory example is just another representation of the input timetable as

$$A = \begin{pmatrix} 10 & 5 & 15 \\ 10 & 5 & 0 \\ 0 & 0 & 5 \\ 10 & 15 & 10 \end{pmatrix}.$$

Observe that we can understand such a matrix A immediately as the adjacency-matrix of a bipartite multigraph $G(A) = (C \cup T, E)$ where E is the multiset of edges $e = \{c_j, t_i\}$ with multiplicity $m(e) = a_{i,j}$ for $i \in [l], j \in [k]$.

A sectioning solution from a single student's point of view, who is attending all k courses, are just k timeslots $t_i \in T$ (one for each of his courses), which in turn means that this student attends the section of c_j scheduled in t_i. In terms of $G(A)$ this is just a k-matching in $G(A)$ joining all nodes $c_j \in C$ with some timeslot $t_i \in T$ – while consuming one of the $a_{i,j}$ edges $e = \{c_j, t_i\}$. Finally, an entire solution is represented as a set of k-matchings $\mathcal{M}$ which is feasible if no edge is assigned to more than one student. Together, our basic model is formally defined as follows.

Basic Student Sectioning Problem (BSS)

Input: A matrix $A = (a_{i,j}) \in \mathbb{N}^{l \times k}$ with $l \geq k$ and some m.

Question: Is there a set $\mathcal{M}$ of edge-disjoint k-matchings in $G(A)$ with $|\mathcal{M}| = m$?

Observe that we may additionally require $l \geq k$ since otherwise no solution exists due to an unavoidable violation of (D). We define BSS here as a decision problem asking whether at least m students can be sectioned with respect to timetable A. In Section 4 we will also consider an optimization version of BSS where a solution with a maximum value of m needs to be computed.

2.2 Additional Constraints

We now define more comprehensive models by adding constraints (C), (T) and (E) individually. For the course-selection constraint (C) we assume that there is additionally some mapping $c : S \to \mathcal{P}(C)$ specifying for each student the selected courses. To fulfill these requirements there must be some assignment f of students to matchings (i.e., to their individual timetables) such that the matched courses are the selected ones. On the other hand, we do not require any more that each matching has cardinality k.

BSS+(C)

Input: A matrix $A = (a_{i,j}) \in \mathbb{N}^{l \times k}$ with $l \geq k$ and some m. *Additionally a set of students S with $|S| \geq m$ and a mapping $c : S \to \mathcal{P}(C)$ of students to selected courses.*

Question: Is there a set $\mathcal{M}$ of edge-disjoint matchings in $G(A)$ with $|\mathcal{M}| = m$, *such that there is some bijective function $f : S \to \mathcal{M}$ with*

(C) $\qquad \forall_{s \in S} \quad [f(s) = M \Rightarrow C_M = c(s)]$?

Here we ask whether at least m students from S can be sectioned such that (C) holds. Note that f is a partial function if $|S| > m$. Moreover, if all students select all k courses then we can drop the additional input and BSS+(C) is just BSS. Next we extend BSS by the timeslot constraint (T). Again we assume that there is some additional mapping $t : S \to \mathcal{P}(T)$ specifying for each student the available timeslots. This time we need to ensure that all timeslots in each students matching are among the available timeslots for this student.

BSS+(T)

Input: A matrix $A = (a_{i,j}) \in \mathbb{N}^{l \times k}$ with $l \geq k$ and some m. *Additionally a set of students S with $|S| \geq m$ and a mapping $t : S \to \mathcal{P}(T)$ of students to available timeslots.*

Question: Is there a set $\mathcal{M}$ of edge-disjoint k-matchings in $G(A)$ with $|\mathcal{M}| = m$, *such that there is some bijective function $f : S \to \mathcal{M}$ with*

(T) $\qquad \forall_{s \in S} \quad [f(s) = M \Rightarrow T_M \subseteq t(s)]$?

Observe that we require k-matchings again and that BSS appears as a special case. Finally, we want to model with BSS+(E) that sections may have multiple events in a given timetable. So far we have identified a section of a course by the timeslot of its event in a given timetable. Correspondingly, we now identify a section of a course by the *set of timeslots* of the events of this section in a given timetable. This allows to model solutions in terms of matchings between C and T again. Note that if two sections of the same course have the same set of timeslots then we may safely unify both sections and add their capacities. So if there are at most r sections per course then we may assume that some mapping $g : C \times [r] \to \mathcal{P}(T) \times \mathbb{N}$ specifies for course c_j and its section q with $g(c_j, q) = (T_{j,q}, a_{j,q})$ the set of timeslots $T_{j,q}$ of all events of section number q of course c_j together with the maximum capacity $a_{j,q}$ of this section. This representation of the input timetable replaces matrix A that we used before.

The sets of timeslots lead in a natural way to the notion of a bipartite multi-hypergraph $H(g) = (C \cup T, E)$, where the set E of hyperedges of $H(g)$ contains all edges

$$e = \{c_j\} \cup T_{j,q}$$

for all courses j and their sections q, each having multiplicity $m(e) = a_{j,q}$.

BSS+(E)

Input: A set of courses C, a set of timeslots T, and some m, together with a mapping $g : C \times [r] \to \mathcal{P}(T) \times \mathbb{N}$ specifying the sets of timeslots and capacities of all sections.

Question: Is there a set $\mathcal{M}$ of edge-disjoint k-matchings in $H(g)$ with $|\mathcal{M}| = m$?

Recall that in a matching $M \in \mathcal{M}$ in hypergraph $H(g)$ all edges are pairwise disjoint, i.e., there are no timeslot clashes if we understand M as a student's timetable. So if we find m edge-disjoint matchings we can assign m students to the sections of all k courses such that (E) holds.

Together, we have defined a uniform framework to formally specify sectioning problems. The notion of bipartite multigraphs common to all problem versions allows to talk about 'mixed' versions BSS+X for X $\subseteq \{(\mathrm{C}), (\mathrm{T}), (\mathrm{E})\}$ in a natural but precise way. If (E) is in X then the input timetable is represented as mapping g, and as matrix A otherwise. As an example, if we want to specify that both (C) and (E) are present, we may use notation BSS+$\{(\mathrm{C}), (\mathrm{E})\}$ while the following definition is understood:

BSS+$\{(\mathrm{C}), (\mathrm{E})\}$

Input: A set of courses C, a set of timeslots T, and some m, together with a mapping $g : C \times [r] \to \mathcal{P}(T) \times \mathbb{N}$ specifying the sets of timeslots and capacities of all sections. A set of students S with $|S| \geq m$ and a mapping $c : S \to \mathcal{P}(C)$ of students to selected courses.

Question: Is there a set $\mathcal{M}$ of edge-disjoint k-matchings in $H(g)$ with $|\mathcal{M}| = m$, such that there is some bijective function $f : S \to \mathcal{M}$ with

(C) $\qquad \forall_{s \in S} \quad [f(s) = M \Rightarrow C_M = c(s)]$?

3 Complexity Analysis

We give a complexity analysis of sectioning problems in order to identify 'easy' and 'hard' constraints.

3.1 Good News: The Basic Student Sectioning Problem is in P

A special case of BSS-instances (A, m) appears when all course-nodes $c \in C$ in $G(A)$ have maximum degree. Graphs with this property are called semiregular [CH82].

Definition 3.1 *Let $G = (C \cup T, E)$ be a bipartite multigraph such that $|C| \leq |T|$. Then G is* semiregular *if and only if $d(c_j) = \Delta(G)$ for all $c_j \in C$.*

The following property of semiregular graphs is easy to verify and we state it for further reference.

Proposition 3.2 *Let $G = (C \cup T, E)$ be a semiregular bipartite multigraph such that $|C| \leq |T|$. Furthermore, let $\mathcal{M}$ be a set of matchings in G that form a partitioning of E. If $|\mathcal{M}| = \Delta(G)$, then every $M \in \mathcal{M}$ is a maximum matching with $|M| = |C|$.*

Next we see that for every yes-instance (A, m) of BSS we find a semiregular subgraph of degree m in $G(A)$, and vice versa.

Lemma 3.3 *For all $A \in \mathbb{N}^{l \times k}$ with $l \geq k$ and $m \geq 0$ it holds that $(A, m) \in$ BSS if and only if $G(A)$ has a semiregular subgraph $G' = (C \cup T, E')$ with $\Delta(G') = m$.*

Proof Let (A, m) be some BSS-instance and denote by $G = G(A) = (C \cup T, E)$ its graph representation with $k = |C| \leq |T| = l$. If $(A, m) \in$ BSS then there exists a set $\mathcal{M}$ containing m edge-disjoint k-matchings of $G(A)$. Now define $G' = (C \cup T, E')$ with $E' = \bigcup_{M \in \mathcal{M}} M$. Then $d(c) = m$ for every node $c \in C$ in G' because each matching M adds one edge incident to c. Furthermore, since every $M \in \mathcal{M}$ contains at most one edge incident to some $t \in T$, we also have $d(t) \leq m$ for all $t \in T$ in G'. Therefore it holds that $\Delta(G') = m$.

Conversely, assume that there is some semiregular subgraph $G' = (C \cup T, E')$ of $G(A)$ with $\Delta(G') = m$. So in G' we have $d(c) = \Delta(G')$ for any $c \in C$. We apply König's Lemma to G' and see that it can be partitioned into $\Delta(G')$ edge-disjoint matchings $\mathcal{M}$. We know from Proposition 3.2 that every $M \in \mathcal{M}$ is a k-matching and since $m = \Delta(G')$ we have $(A, m) \in$ BSS. $\square$

This lemma shows that the problem of deciding whether m students can be sectioned into the k courses given in timetable A is equivalent to finding a semiregular subgraph in $G(A)$ that contains all courses and has maximum degree m. So the next question is how we can determine whether or not $G(A)$ has such a subgraph. Obviously, the answer is yes if $G(A)$ itself is semiregular.

Corollary 3.4 *Let $A \in \mathbb{N}^{l \times k}$ with $l \geq k$ and $m \geq 0$. If $G(A)$ is semiregular then $(A, m) \in$ BSS for all $m \leq \Delta(G(A))$.*

Note that it is easy to check whether $G(A)$ is semiregular when A is given: The sums of all columns in A ($= d(c)$ for all $c \in C$) must be equal, and the sums of all rows ($= d(t)$ for all $t \in C$) must not exceed this value. If $G(A)$ is semiregular and we assign $m = \Delta(G(A))$ students to the sections of the k courses, then we have the special case that no capacities $a_{i,j}$ remain unused.

Recall matrix A from the introductory example and consider a submatrix A' with

$$A = \begin{pmatrix} 10 & 5 & 15 \\ 10 & 5 & 0 \\ 0 & 0 & 5 \\ 10 & 15 & 10 \end{pmatrix} \quad \text{and} \quad A' = \begin{pmatrix} 5 & 0 & 15 \\ 10 & 5 & 0 \\ 0 & 0 & 5 \\ 5 & 15 & 0 \end{pmatrix} .$$

We easily verify that $G(A')$ is semiregular with $\Delta(A') = 20$, hence, by Lemma 3.3, we can section 20 students w.r.t. timetable A such that (D) and (M) hold. We show next how to determine for arbitrary instances whether $G(A)$ has a suitable subgraph with help of a flow network.

Definition 3.5 *Let (A, m) be some instance of* BSS. *Then $F(A, m) = (V, E, c)$ is a flow network with source v_s and sink v_t defined as*

$$\begin{aligned} V &= C \cup T \cup \{v_s, v_t\} , \\ E &= C \times T \cup \{v_s\} \times C \cup T \times \{v_t\} \text{ and capacities} \end{aligned}$$

$$c(u, v) = \begin{cases} a_{i,j} & \text{if } u = c_j \wedge v = t_i \\ \min(m, \sum_{i=1}^{l} a_{i,j}) & \text{if } u = v_s \wedge v = c_j \\ \min(m, \sum_{j=1}^{k} a_{i,j}) & \text{if } u = t_i \wedge v = v_t \end{cases}$$

Observe that $F(A, m)$ can be constructed efficiently from (A, m).

Lemma 3.6 *Let (A, m) be a* BSS*-instance with $A \in \mathbb{N}^{l \times k}$ and $l \geq k$, and let f denote some maximum flow in $F(A, m)$ with value $v(f)$. Then*

$$(A, m) \in \text{BSS} \Leftrightarrow v(f) = k \cdot m .$$

Proof We start by stating the capacity and flow-conservation constraints resulting from our definition of the network.

i. capacity constraints

$$\forall_{j \in [k]} : 0 \leq f(v_s, c_j) \leq \min(m, \sum_{i \in [l]} a_{i,j}) \tag{1}$$

$$\forall_{i \in [l]} : 0 \leq f(t_i, v_t) \leq \min(m, \sum_{j \in [k]} a_{i,j}) \tag{2}$$

$$\forall_{i \in [l], j \in [k]} : 0 \leq f(c_j, t_i) \leq a_{i,j} \tag{3}$$

ii. flow-conservation constraints

$$\forall_{j \in [k]} : f(v_s, c_j) = \sum_{i \in [l]} f(c_j, t_i) \tag{4}$$

$$\forall_{i \in [l]} : f(t_i, v_t) = \sum_{j \in [k]} f(c_j, t_i) \tag{5}$$

The value of some flow f in this network is determined by

$$v(f) = \sum_{j\in[k]} f(v_s, c_j).$$

First, suppose $(A, m) \in \mathrm{BSS}$. By Lemma 3.3 there is a semiregular subgraph $G' = (C \cup T, E')$ in $G(A)$ with $\Delta(G') = m$. So in G' we have $d'(c) = m$ for any $c \in C$ and $d'(t) \leq m$ for any $t \in T$. We construct a flow f as follows:

$$\begin{aligned} f(v_s, c_j) &= d'(c_j) && \text{for } j \in [k], \\ f(t_i, v_t) &= d'(t_i) && \text{for } i \in [l] \text{ and} \\ f(c_j, t_i) &= m'(\{c_j, t_i\}) && \text{for } j \in [k],\ i \in [l]. \end{aligned}$$

Capacity constraint (1) holds because $\sum_{i\in[l]} a_{i,j} = d(c_j) \geq d'(c_j) = m$, so for all $j \in [k]$ we have

$$f(v_s, c_j) = d'(c_j) = m = \min(m, \sum_{i\in[l]} a_{i,j}) \ . \tag{6}$$

For constraint (2) note that for all $i \in [l]$ it holds that $\sum_{j\in[k]} a_{i,j} = d(t_i) \geq d'(t_i)$ and $m \geq d'(t_i)$, so also in this case we have

$$f(t_i, v_t) = d'(t_i) \leq \min(m, \sum_{j\in[k]} a_{i,j}) \ .$$

Constraint (3) holds because for all $i \in [l]$, $j \in [k]$ we have

$$f(c_j, t_i) = m'(\{c_j, t_i\}) \leq m(\{c_j, t_i\}) = a_{i,j} \ .$$

To see that conservation constraint (4) holds note that

$$f(v_s, c_j) = d'(c_j) = \sum_{i\in[l]} m'(\{c_j, t_i\}) = \sum_{i\in[l]} f(c_j, t_i) \ .$$

The same is true for constraint (5) with a similar argument. The value of f is

$$v(f) = \sum_{j\in[k]} f(v_s, c_j) = \sum_{j\in[k]} d'(c_j) = k \cdot m$$

which is maximum, because we see in Eq. (6) that f saturates all capacities of all outgoing edges of the source v_s.

Now assume conversely that f is some maximum flow in $F(A, m)$ with $v(f) = km$. We define a matrix $A' \in \mathbb{N}^{l\times k}$ by

$$a'_{i,j} = f(c_j, t_i)$$

for $j \in [k]$ and $i \in [l]$, and consider graph $G' = G(A')$. Capacity constraint (3) guarantees $a'_{i,j} = f(c_j, t_i) \leq a_{i,j}$ so G' is a subgraph of $G(A)$. Next we show that G' is semiregular. Observe that for all $j \in [k]$ it holds that

$$d'(c_j) = \sum_{i\in[l]} a'_{i,j} = \sum_{i\in[l]} f(c_j, t_i) = f(v_s, c_j)$$

where the latter is due to constraint (4). From $v(f) = \sum_{j\in[k]} f(v_s, c_j) = km$ and $c(v_s, c_j) \leq m$ by construction of $F(A, m)$ we get $f(v_s, c_j) = m$. So together we see that $d'(c_j) = m$ for all $j \in [k]$.

It remains to show that $d'(t_i)$ is at most m for all $t_i \in T$. With constraint (5) we have

$$d'(t_i) = \sum_{j\in[k]} a'_{i,j} = \sum_{j\in[k]} f(c_j, t_i) = f(t_i, v_t) \ .$$

We apply constraint (2) and get

$$d'(t_i) = f(t_i, v_t) \leq \min(m, \sum_{j\in[k]} a_{i,j}) \leq m \ .$$

Since $\Delta(G') = m$ we obtain from Lemma 3.3 that $(A, m) \in$ BSS. □

Since a maximum flow can be computed in polynomial time, we have the following theorem.

Theorem 3.7 BSS *is in* **P**.

Proof We summarize our investigations with the decision procedure BSS($\cdot$) given in the following listing.

Input: An instance (A, m) of BSS with $A \in \mathbb{N}^{l\times k}$, $l \geq k$ and $m \geq 0$.
Output: False, if $(A, m) \notin$ BSS, and a witnessing maximum flow in $F(A, m)$ otherwise.

```
function BSS($A, l, k, m$)
    $F \leftarrow$ MAKE_FLOWNETWORK($A, m, l, k$)          ▷ According to Def. 3.5
    $f_{\max} \leftarrow$ MAX_FLOW($F, v_s, v_t$)      ▷ E.g. by the algorithm of Sleator and Tarjan [ST83]
    if $v(f_{\max}) \neq k * m$ then                   ▷ Correctness follows from Lemma 3.6
        return False
    return $f_{max}$
```

Observe that $F(A, m)$ has $O(k + l)$ nodes and $O(kl)$ edges. The algorithm of Sleator and Tarjan [ST83] has running time $O(kl(k+l)\log(k+l)) \subseteq O(k^2l^2\log(k+l))$. Since the bit-length of the input is at least $k \cdot l$, the overall running time is polynomial in the input length. □

Now consider that for any semiregular subgraph G' of $G(A)$ we have

$$\Delta(G') \leq \min_{j\in[k]}(\sum_{i\in[l]} a_{i,j}) \leq \mathrm{sum}_A$$

because every student has to be assigned to all courses. Thus we may compute the maximum number m^* of assignable students for a given timetable A in polynomial time by a binary search over the interval $m \in [\mathrm{sum}_A]$ with repeated calls to the previous algorithm. To compute solutions in terms of timetables for these students (= k-matchings in $G(A)$) efficiently is more subtle because the number of matchings is usually not polynomially bounded. We come back to this point in Section 4.

3.2 Bad News: Adding (C), (T) or (E) makes BSS NP-complete

We show the NP-hardness of the extended BSS-versions with polynomial-time many-one reductions from the NP-complete problem 3-Dimensional Matching [Kar72]. For finite, disjoint sets X, Y and Z we say that $M \subseteq X \times Y \times Z$ is a 3-dimensional matching if for all distinct triples $(x_1, y_1, z_1), (x_2, y_2, z_2) \in M$ it holds that $x_1 \neq x_2$, $y_1 \neq y_2$ and $z_1 \neq z_2$. It is known that

3-Dimensional Matching is NP-complete even in the special case when $|X| = |Y| = |Z| = u$ and M is a perfect matching with $|M| = u$.

3-Dimensional Matching (3-DM)

Input: Finite and disjoint sets X, Y, Z with $|X| = |Y| = |Z|$ and a subset $J \subseteq X \times Y \times Z$.

Question: Is there a perfect 3-dimensional matching $M \subseteq J$?

In all proofs of the following lemmas let $X = \{x_i \mid i \in [u]\}$, $Y = \{y_i \mid i \in [u]\}$, $Z = \{z_i \mid i \in [u]\}$ for some $u \geq 1$ and $J \subseteq X \times Y \times Z$ be an arbitrary instance of 3-DM.

Lemma 3.8 3-DM *is polynomial-time reducible to* BSS+(C)*.*

Proof The idea of the reduction is to encode each of the sets X, Y, Z separately as a set of u courses, each of them having a single section with capacity 1. All the sections of the courses for X go to timeslot t_1, all sections of Y to timeslot t_2 and all sections of Z to t_3. This can be achieved if we define $A = (a_{i,j})^{3\times 3u}$ with $a_{i,j} = 1$ for $i \in [3]$ and $(i-1) \cdot u + 1 \leq j \leq i \cdot u$, and $a_{i,j} = 0$ otherwise. So $G(A)$ is a bipartite graph with node partitions $C = \{c_j \mid j \in [3u]\}$ and $T = \{t_i \mid i \in [3]\}$. Each $(x, y, z) \in J$ is regarded as a student with course-selection (x, y, z). So we set $S = J$ and define $c(s) = (c_p, c_{u+q}, c_{2u+r})$ for all $s = (x_p, y_q, z_r) \in S$. Note that the courses for X, Y and Z are disjoint due to offsets u and $2u$. Moreover, we ask whether $m = u$ students can be sectioned. Clearly, the reduction function h with

$$h(X, Y, Z, J) = (A, u, J, c)$$

is computable in polynomial time.

Now assume that $(X, Y, Z, J) \in$ 3-DM and let $M \subseteq J$ be a witnessing 3-dimensional matching. For each $(x_p, y_q, z_r) \in M$ we define a set

$$M(p, q, r) = \{(c_p, t_1), (c_{u+q}, t_2), (c_{2u+r}, t_3)\}$$

of edges in $G(A)$ which is also a 3-matching. So $\mathcal{M} = \{M(p, q, r) \mid (x_p, y_q, z_r) \in M\}$ is a set of matchings in $G(A)$ with $|\mathcal{M}| = u$. All sets in $\mathcal{M}$ are edge-disjoint because any two elements in M differ in all three components. The mapping $f : J \to \mathcal{M}$ with $f((x_p, y_q, z_r)) = M(p, q, r)$ for all $(x_p, y_q, z_r) \in M$ is bijective and we have $C_{M(p,q,r)} = \{c_p, c_{u+q}, c_{2u+r}\} = c((x_p, y_q, z_r))$, hence $h(X, Y, Z, J) \in$ BSS+(C).

Conversely, if $(A, u, J, c) \in$ BSS+(C) let $\mathcal{M}$ be an edge-disjoint set of u matchings in $G(A)$ such that (C) holds, witnessed by some bijective $f : J \to \mathcal{M}$. Define $M = f^{-1}(\mathcal{M}) \subseteq J$. Then $|M| = u$ and any two triples in M differ in all three components because the matchings in $\mathcal{M}$ are edge-disjoint and all edges in $G(A)$ have multiplicity 1. So M is a perfect 3-dimensional matching and hence $(X, Y, Z, J) \in$ 3-DM. □

Observe that the reduction function maps only to instances where each course has just one section and where no timeslot clashes can occur, i.e., the disjoint-section constraint (D) is always fulfilled. So to obtain NP-hardness it is enough if (M) and (C) are present, and if students are limited to choose three courses - even if there are no sectioned courses at all.

Lemma 3.9 3-DM *is polynomial-time reducible to* BSS+(T)*.*

Proof The idea of the reduction is very similar to the previous proof, we just exchange the roles of courses and timeslots. So A is the transposed matrix of the one before and $G(A)$ has node partitions $C = \{c_1, c_2, c_3\}$ and $T = \{t_i | i \in [3u]\}$. Each $(x, y, z) \in J$ is then regarded as a student again, whose available timeslots are the timeslots of sections x, y, and z. So we set $S = J$ and define $t(s) = (t_p, t_{u+q}, t_{2u+r})$ for all $s = (x_p, y_q, z_r) \in S$. Again, the reduction function h with

$$h(X, Y, Z, J) = (A, u, J, t)$$

is computable in polynomial-time. We omit the rest of the proof since it is equal to the previous proof. □

Notice that this time h maps only to instances with 3 courses and again no timeslot clashes can occur. So to obtain NP-hardness it is enough if (M) and (T) are present, and if the number of courses is limited to three.

Lemma 3.10 3-DM *is polynomial-time reducible to* BSS+(E)*.*

Proof The intention of the reduction is to encode X as a set of u courses and the sets Y and Z are regarded as two disjoint sets of u timeslots. Each triple $(x, y, z) \in J$ represents a section of course x with two events scheduled in timeslots y and z and having capacity 1. Hence the number of sections of some course x is equal to the number of triples in J with x as first component. For all $x_p \in X$ and $i \in [v]$ with $v = |\{(x, y, z) \in J \mid x = x_p\}|$ we set $g(c_p, i) = (\{t_q, t_{u+m}\}, 1)$ if (x_p, y_q, z_m) is the i-th tripel in J with first component x_p. Observe that in this case $H(g)$ is a bipartite hypergraph with node partitions $C = \{c_j \mid j \in [u]\}$ and $T = \{t_i \mid i \in [2h]\}$. We define the reduction function h as

$$h(X, Y, Z, J) = (X, Y \cup Z, 1, g)$$

which asks whether a single student can be sectioned, and which is computable in polynomial time.

Now assume that $(X, Y, Z, J) \in$ 3-DM and let $M \subseteq J$ be a witnessing 3-dimensional matching. We define a set

$$B = \{\{c_p, t_q, t_{u+m}\} \mid (x_p, y_q, z_m) \in M\}$$

of edges in $H(g)$ which is also a u-matching in $H(g)$ because all triples in M differ in all components. So $h(X, Y, Z, J) \in$ BSS+(C) witnessed by $\mathcal{M} = \{B\}$.

Conversely, if $(X, Y \cup Z, 1, g) \in$ BSS+(C) let $\mathcal{M} = \{B\}$ with u-matching B in $H(g)$. We define a set

$$M = \{(x_p, y_q, z_{r-u}) \mid \{c_p, t_q, t_r\} \in B\}$$

and see that any two triples in M differ in all three components because B is a matching in $H(g)$. So M is a perfect 3-dimensional matching and hence $(X, Y, Z, J) \in$ 3-DM. □

This time reduction function h maps only to instances where each section has exactly two events and only one student is considered, i.e., the maximum-capacity constraint (M) can not be violated for the existing sections. So to obtain NP-hardness it is enough if (D) and (E) are present, if the number of sections is limited to two and if a single student needs to be sectioned.

Theorem 3.11 *All problems* BSS+(X) *for* X$\in$ {C, T, E} *are NP-complete.*

Proof We have shown the lower complexity bounds for these problems in the preceeding lemmas and it remains to argue that they are all members of **NP**. To do so we sketch a nondeterministic algorithm only for BSS+(C) since there are similar algorithms for the other problems. We use the notations from the definition of BSS+(C) to denote the input data.

1. Choose nondeterministically a subset $S' \subseteq S$ with $|S'| = m$. $O(|S|)$
2. For each student $s \in S'$ construct a set M_s as follows: $O(m \cdot |C| \cdot |T|)$
   (a) For each course $c \in c(s)$ select nondeterministically a timeslot $t \in T$ and add (c, t) to M_s.
   (b) Reject if some timeslot appears twice in M_s.
3. For all $(c_j, t_i) \in C \times T$ count the occurences of (c_j, t_i) in all sets M_s. Reject if this number exceeds $a_{i,j}$. $O(|C|^2 \cdot |T| \cdot m)$
4. Accept.

Note that the running time on a single computation path is polyomially bounded in the input length, so BSS+(C) $\in$ **NP**. □

Observe that if we add more constraints to one of these problems, then an NP-complete problems appears as a special case. The upper bound can be shown similarly to algorithm in the previous theorem. So we have:

Corollary 3.12 *All problems* BSS+X *for non-empty* X$\subseteq \{\mathrm{C}, \mathrm{T}, \mathrm{E}\}$ *are NP-complete.*

4 Succinct Solutions for BSS

We investigate in this section how to efficiently compute solutions for BSS in terms of individual timetables for the optimal number of students. We have already mentioned in Section 2 that we can interpret a k-matching $M \in \mathcal{M}$ in $G(A)$ as such a single students timetable. Let us first recall from Section 3 what we can do efficiently so far if the input timetable A is given:

- Compute the optimal number of students m^* with a binary search over m and subsequent calls to algorithm BSS($\cdot$) stated in Theorem 3.7.
- Get the maximum flow $f^*_{\max}$ in $F(A, m^*)$ from this algorithm.

Furthermore, we can construct from $f^*_{\max}$ a submatrix A' of A such that $G(A')$ is semiregular and $m^* = \Delta(G(A'))$. For this we just set

$$a'_{i,j} = f^*_{\max}(c_j, t_i)$$

for all $(c_j, t_i) \in C \times T$. The advantage of A' over the original A is that the solution $\mathcal{M}$ we are looking for is a partitioning of the edges in $G(A')$, i.e., no capacities in A' remain unused. Coming back to our running example, this gives us matrix A' stated after Corollary 3.4.

It is easy to see that in this case edge-partitioning is the EDGE-COLORING problem with $\Delta(G(A'))$-many colors (cf. Proposition 3.2), so we could fall back on some standard algorithm for the latter, see, e.g. [Sch98]. However, there is a major drawback with this approach if complexity is considered. This is because all known EDGE-COLORING algorithms have factor $|E|$ in their running time, but the number of edges in $G(A')$ is not polynomially bounded in the bit-length of input A. Note that there can be as many as $O(\mathrm{sum}_A)$ edges which can only be bounded pseudo-polynomially in the input length. This is a substantial aspect since in practice rather large sums of capacities occur. In our small example $G(A')$ has already 60 edges. Moreover, as result we would get a long list of matchings for all students, whose size can only be bounded pseudo-polynomially as well. For our example this is

$$\begin{array}{rcl} s_1 & \to & ((c_1,t_2),(c_2,t_4),(c_1,t_1)) \\ s_2 & \to & ((c_1,t_2),(c_2,t_4),(c_1,t_1)) \\ & \vdots & \\ s_{11} & \to & ((c_1,t_4),(c_2,t_2),(c_1,t_1)) \\ & \vdots & \\ s_{20} & \to & ((c_1,t_1),(c_2,t_4),(c_1,t_3)) \end{array}$$

So our goal in this section is to find an algorithm for optimal solutions that is truly polynomial in the bit-length of input A. For this, a succinct representation of solutions $\mathcal{M}$ is needed.

4.1 Efficient Computation of Solutions

It will turn out that there is always an optimal solution that has only a small number of *different* student timetables $M \in \mathcal{M}$. Together with the fact that in the BSS setting students are indistinguishable, it will be enough to compute the number of students that follow each of these timetables. We say that two matchings $M, M' \in \mathcal{M}$ in multigraph $G(A)$ are *distinct* if they do not comprise the same set of edges.

Definition 4.1 (Succinct solution) *Let A be some* BSS*-matrix such that $G(A)$ is semiregular and let $\mathcal{M}$ be a set of distinct k-matchings in $G(A)$. A function $f : \mathcal{M} \to \mathbb{N}$ is called a* succinct solution *for A, if $\sum_{M:\{c_j,t_i\}\in M} f(M) = a_{i,j}$ for all $j \in [k]$, $i \in [l]$.*

This condition ensures that the sum of all students that follow the distinct student timetables in $\mathcal{M}$ is exactly the maximum number of students that can be assigned to A. A succinct solution for our running example A' is

$$\begin{array}{rcl} ((c_1,t_2),(c_2,t_4),(c_1,t_1)) & \to & 10 \\ ((c_1,t_4),(c_2,t_2),(c_1,t_1)) & \to & 5 \\ ((c_1,t_1),(c_2,t_4),(c_1,t_3)) & \to & 5 \end{array}$$

assigning the optimal number of 20 students to a small number of different student timetables. We now state the algorithm to compute such a succinct solution efficiently. Note that the algorithm is underspecified in the sense that it is not clear yet how to compute in line 9 a matching that saturates all nodes of maximal degree. We come back to this in the theorem below.

Lemma 4.2 *Algorithm* OPT_SOLUTION *is correct and the while-loop in line* 8 *has at most $k(l+1)$ iterations.*

Proof Since we have already shown the correctness of the first part of the algorithm, we are left with the while-loop. We show by induction on the maximal degree $\Delta(G)$ of $G = G(A)$ in line 8 that the while-loop computes a succinct solution for A if G is semiregular. If $\Delta(G) = 0$, then $f = \emptyset$ is the only feasible solution.

Input: A BSS-timetable A with $A \in \mathbb{N}^{l \times k}$ and $l \geq k$.
Output: A succinct solution f for the maximum number of assignable students.

```
function OPT_SOLUTION($A, l, k$)
    $d \leftarrow \min_{j \in [k]}(\sum_{i \in [l]} a_{i,j})$          ▷ upper bound for number of students
    $m^*, f^*_{max} \leftarrow$ BIN_SEARCH($0 \leq m \leq d$, BSS($A, l, k, m$))     ▷ optimal number of students
    $A \leftarrow \mathbf{0}_{l \times k}$
    for each $(c_j, t_i)$ in $C \times T$                  ▷ reduce matrix to semiregular subgraph
        $a_{i,j} \leftarrow f^*_{max}(c_j, t_i)$
    $f \leftarrow \emptyset$
    while $\Delta(G(A)) \neq 0$ do
        $M \leftarrow$ some $k$-matching in $G(A)$ saturating all nodes of maximal degree
        $n \leftarrow \min(\min_{\{c_j,t_i\} \in M}(a_{i,j}), \min_{t_i \in T \setminus T_M}(\Delta(G(A)) - d(t_i)))$
        for each $\{c_j, t_i\}$ in $M$
            $a_{i,j} \leftarrow a_{i,j} - n$
        $f(M) \leftarrow n$
    return $f$
```

Now suppose that the while-loop computes a succinct solution for any semiregular graph $G' = G(A')$ with $\Delta(G') < \Delta(G)$ and we want to show this fact for G.

Since G is semiregular, there exist $\Delta(G)$-many k-matchings that partition the edge-set of G. There are at most k nodes $t_i \in T$ with $d(t_i) = \Delta(G)$, so matching M in line 9 exists and we have $d(t_i) < \Delta(G)$ for those t_i that are not saturated by M. So with the fact that $a_{i,j} \geq 1$ for any edge $\{c_j, t_i\}$ occuring in M, we have $n > 0$ in line 10.

In lines 11 and 12 we construct a new matrix A' by removing from A the k edges from M, each edge n times. And since this n is not greater than the number of times any edge $e \in M$ occurs in multiset E, we see that $G' = G(A')$ is a subgraph of G. Next we show that G' is again semiregular. First observe that for the degree of any $c_j \in C$ in G' we have

$$d'(c_j) = \Delta(G) - n$$

since all these nodes are saturated by M. For any $t_i \in T_M$ it holds that

$$d'(t_i) = d(t_i) - n \leq \Delta(G) - n$$

because this node is saturated and $d(t_i) \leq \Delta(G)$. Finally, consider some node $t_i \in T \setminus T_M$. Here we have $d'(t_i) = d(t_i)$ and thus by the definition of n it holds that

$$n \leq \Delta(G) - d(t_i) = \Delta(G) - d'(t_i)$$

and therefore $d'(t_i) \leq \Delta(G) - n$. Hence G' is a semiregular subgraph of G with

$$\Delta(G') = \Delta(G) - n < \Delta(G) \ . \tag{7}$$

We can apply the hypothesis and obtain that the remaining iterations of the while-loop compute a succinct solution $f' : \mathcal{M}' \to \mathbb{N}$ for a set $\mathcal{M}'$ of distinct k-matchings in $G(A')$, i.e., for all $j \in [k]$, $i \in [l]$ we have $\sum_{M' : \{c_j,t_i\} \in M'} f'(M') = a'_{i,j}$. It remains to show that $\mathcal{M} = \mathcal{M}' \cup \{M\}$ and $f = f' \cup \{M \to n\}$ is a succinct solution for A. We first argue that $M \notin \mathcal{M}'$. By definition of n, at least one of the following two conditions holds:

i. $n = a_{i,j}$ for some $\{c_j, t_i\} \in M$.

Then $a'_{i,j} = 0$ and therefore no matching in $\mathcal{M}'$ can contain an edge $\{c_j, t_i\}$.

ii. $n = \Delta(G) - d(t_i)$ for some $t_i \in T \setminus T_M$

So M has no edge $\{c_j, t_i\}$, but all matchings in $\mathcal{M}'$ must saturate t_i because by Eq. (7) we have $\Delta(G') = d(t_i) = d'(t_i)$.

Hence f is well-defined. Now observe that for all edges $\{c_j, t_i\} \in M$ we have

$$f(M) + \sum_{M' : \{c_j, t_i\} \in M'} f'(M') = f(M) + a'_{i,j} = a_{i,j}$$

which shows that f is a succinct solution for A. This completes the induction.

We conclude our analysis and show a bound on the number of iterations of the while-loop. To do so we count how often conditions i. or ii. can hold before the loop terminates. In case i. only $(k \cdot l)$ entries of A can be set to zero before the loop terminates with $A = \mathbf{0}$. If case ii. applies then t_i is a node of maximum degree for all matrices A' in all forthcoming iterations. Since a semiregular matrix can have at most k nodes from T with maximum degree, this means that case ii. can occur over all iterations at most k times. Together, this gives the stated bound on the number of iterations. □

Theorem 4.3 *Let A be a* BSS*-timetable with $A \in \mathbb{N}^{l \times k}$ and $l \geq k$, and let $a_{\max}$ denote the maximum entry in A. Algorithm* OPT_SOLUTION *computes a succinct solution with the optimal number of assignable students in $O(k^2 l^2 \log(\text{sum}_A))$.*

Proof It remains to establish the upper bound on the running time. The value of d in line 2 is computable in $O(kl)$ and the value of d is bounded by $O(\text{sum}_A)$. So by Theorem 3.7 the binary search in line 3 can be carried out in $O(\log(\text{sum}_A) k^2 l^2 \log(k + l))$. Lines 4 to 7 are in $O(kl)$ and we already know that the while-loop has at most $k(l + 1)$ iterations. Next we need to argue how we can find matching M in line 9. Since we only need a single k-matching we can construct a graph G' from A which has a single edge (i, j) if $a_{i,j} > 0$. Note that the number of edges in G' is bounded by $O(kl)$ as opposed to the number of edges in $G(A')$. We use the algorithm by Cole and Hopcroft [CH82] to find a k-matching in G' that saturates all nodes of maximum degree. The running time of this algorithm on input G' is bounded by $O(kl \log k)$. The other statements in the body of the while-loop have smaller running times, so the while-loop is in $O(k^2 l^2 \log k)$. Hence the overall running time is dominated by the binary search which can be simplified to $O(k^2 l^2 \log(\text{sum}_A))$. □

Observe that the given bound is polynomial in the bit-length of A. But the theorem also gives some insight into the structure of BSS-solutions: Although the number of sectioned students can be large, we can efficiently compute an optimal solution with not more than $k(l + 1)$ many distinct student-timetables.

Corollary 4.4 *For any* BSS*-timetable A there exists an optimal succinct solution $f : \mathcal{M} \to \mathbb{N}$ with $|\mathcal{M}| \leq k(l + 1)$.*

4.2 Minimizing the Number of Different Student-Timetables is NP-hard

A small number of different student-timetables is a reasonable goal in student sectioning since it has some advantages for students (study groups, car pools) and it simplifies administration. We make this question precise with the following problem definition.

MIN-BSS

Input:	A matrix $A = (a_{i,j}) \in \mathbb{N}^{l \times k}$ with $l \geq k$ such that $G(A)$ is semiregular, and some $m \leq k(l+1)$.
Question:	Is there a succinct solution $f : \mathcal{M} \to \mathbb{N}$ for A with $|\mathcal{M}| = m$?

It seems that the problem of finding a solution with a minimum number of different timetables is much harder than finding a solution with a small number.

Theorem 4.5 MIN-BSS *is NP-complete.*

Proof Observe that we can nondeterministically guess all sets $\mathcal{M}$ of k-matchings of size m and assign some value $f(M) \leq \mathrm{sum}_A$ to all $M \in \mathcal{M}$. To check whether the condition from the definition of succinct solutions holds for this choice of f can be done deterministically in polynomial time, so MIN-BSS is in **NP**. We show hardness with a polynomial-time many-one reduction from the NP-complete problem Sum of Subsets (SOS) [GJ80]. Let (a, k) be some SOS-instance with $a = (a_1, \ldots, a_m) \in \mathbb{N}^m$, $a_i > 0$ and $k \in \mathbb{N}$. Then $(a, k) \in \mathrm{SOS}$ if and only if there exists some $I \subseteq [m]$ with $\sum_I a_i = k$. We may assume w.l.o.g. that $0 < k \leq s$ since otherwise (a, k) is trivial.

Denote $s = \sum_{i \in [m]} a_i$ as the sum of the components of a. Define reduction function h as $h(a, k) = (B, m)$ such that $B \in \mathbb{N}^{(m+2)\times 3}$ with rows b_i given by

$$b_i = \begin{cases} (0, \quad a_i, \quad 0) & \text{for } i \in [m], \\ (k, \quad 0, \; s-k) & \text{for } i = m+1, \text{ and} \\ (s-k, \; 0, \quad k) & \text{for } i = m+2. \end{cases} \tag{8}$$

Obviously, h may be computed in polynomial time and $G(B)$ is semiregular. We refer to a 3-matching in $G(B)$ for short by $M(x, y, z) = \{\{c_1, t_x\}, \{c_2, t_y\}, \{c_3, t_y\}\}$ using the saturated nodes from T.

First assume $(a, k) \in \mathrm{SOS}$ witnessed by $I \subseteq [m]$ and define a function f on

$$\mathcal{M} = \{M(m+1, i, m+2) \mid i \in I\} \cup \{M(m+2, i, m+1) \mid i \notin I\}$$

with $f(M(m+1, i, m+2)) = f(M(m+2, i, m+1)) = a_i$ for $i \in [m]$. Note that each $M \in \mathcal{M}$ is a 3-matching and that all elements in $\mathcal{M}$ are pairwise distinct, so it remains to show, that

$$\sum_{\{c_j, t_i\} \in M} f(M) = b_{i,j} \tag{9}$$

for all $j \in [k]$, $i \in [l]$. Consider some $i \in I$ and fix $j = 2$. Then

$$\sum_{\{c_2, t_i\} \in M} f(M) = f(M(m+1, i, m+2)) = a_i = b_{i,2}.$$

For $j = 1$ only egdes $i = m+1$ exist and

$$\sum_{\{c_1, t_{m+1}\} \in M} f(M) = \sum_{i \in I} a_i = k = b_{1,m+1} = b_{3,m+2}.$$

The same is true for $j = 3$ and $i = m + 2$. Similar arguments apply when $i \notin I$ since $b_{1,m+2} = b_{3,m+1} = \sum_{i \notin I} a_i = s - k$.

Now assume $(B, m) \in$ MIN-BSS and let $f : \mathcal{M} \to \mathbb{N}$ be a succinct solution with $|\mathcal{M}| = m$. Consider the second column of B and let $i \in [m]$. Since Eq. (9) holds we have

$$\sum_{\{c_2, t_i\} \in M} f(M) = b_{i,2} = a_i \ .$$

Because $a_i > 0$ there is at least one M in each of the i sums and no two M in different sums can be the same. Since $|\mathcal{M}| = m$ there must be exactly one such matching per i.

Now by the definition of B we have either $M = M(m+1, i, m+2)$ or $M = M(m+2, i, m+1)$ for any $M \in \mathcal{M}$. The index set

$$I = \{i \mid M(m+1, i, m+2) \in \mathcal{M}\}$$

is a feasible solution for the SOS-instance because with Eq. (9) we have

$$\sum_{i \in I} a_i = \sum_{i \in I} f(M(m+1, i, m+2)) = \sum_{\{c_1, t_{m+1}\} \in M} f(M) = b_{m+1,1} = k \ .$$

□

We see from this reduction that the problem remains NP-complete even if the number of courses is fixed to $k = 3$ in all instances.

References

[AF89] J Aubin and J A Ferland. A large scale timetabling problem. *Computers and Operations Research*, 16(1):67–77, 1989.

[AH05] M Amintoosi and J Haddadnia. Feature selection in a fuzzy student sectioning algorithm. *Practice and Theory of Automated Timetabling V*, pages 147–160, 2005.

[AVCT00] R Alvarez-Valdes, E Crespo, and J M Tamarit. Assigning students to course sections using tabu search. *Annals of Operations Research*, 96(1):1–16, 2000.

[AYH04] M Amintoosi, H S Yazdi, and J Haddadnia. Fuzzy student sectioning. *PATAT'04: Proceedings of the 5th international conference on Practice and Theory of Automated Timetabling*, pages 421–425, 2004.

[Car00] Michael W Carter. A Comprehensive Course Timetabling and Student Scheduling System at the University of Waterloo. In *PATAT '00: Selected papers from the Third International Conference on Practice and Theory of Automated Timetabling III*. Springer-Verlag, August 2000.

[CH82] R Cole and J Hopcroft. On edge coloring bipartite graphs. *SIAM Journal on Computing*, 1982.

[CKL03] E Cheng, S Kruk, and M Lipman. Flow formulations for the student scheduling problem. *Practice and Theory of Automated Timetabling IV*, pages 299–309, 2003.

[dW85] D de Werra. An introduction to timetabling. *European Journal of Operational Research*, 19(2):151–162, 1985.

[GJ80] Michael R Garey and David S Johnson. Computers and Intractability, 1980.

[Kar72] R M Karp. Reducibility among combinatorial problems. *Complexity of Computer Computations*, January 1972.

[MM10] Tomáš Müller and Keith Murray. Comprehensive approach to student sectioning. *Annals of Operations Research*, 181(1):249–269, March 2010.

[MRB04] Tomáš Müller, Hana Rudová, and Roman Barták. Minimal perturbation problem in course timetabling. In *PATAT'04: Proceedings of the 5th international conference on Practice and Theory of Automated Timetabling.* Springer-Verlag, August 2004.

[Pap94] Christos H Papadimitriou. *Computational complexity.* Addison Wesley, 1994.

[Sch98] A Schrijver. Bipartite Edge Coloring in O(m) Time. *SIAM Journal on Computing*, 1998.

[SH10] Heinz Schmitz and Christian Heimfarth. Cross-Curriculum Scheduling with Themis. *PATAT'10: Proceedings of the 8th international conference on Practice and Theory of Automated Timetabling*, pages 385–391, July 2010.

[ST83] D D Sleator and R Endre Tarjan. A data structure for dynamic trees. *Journal of computer and system sciences*, 1983.