



Stochastics and Statistics

## Parliament seating assignment problems

Bart Vangerven<sup>a</sup>, Dirk Briskorn<sup>a,\*</sup>, Dries R. Goossens<sup>b</sup>, Frits C. R. Spieksma<sup>c</sup><sup>a</sup> Schumpeter School of Business and Economics, University of Wuppertal, Gaußstraße 20, Wuppertal 42119, Germany<sup>b</sup> Faculty of Economics and Business Administration, Ghent University, Tweeckerkenstraat 2, Gent 9000, Belgium<sup>c</sup> Department of Mathematics and Computer Science, Eindhoven University of Technology, Eindhoven 5600 MB, the Netherlands

## ARTICLE INFO

## Article history:

Received 22 May 2020

Accepted 4 August 2021

Available online 12 August 2021

## Keywords:

Combinatorial optimization

Complexity theory

Heuristics

Valid inequalities

## ABSTRACT

Motivated by evidence that parliament seatings are relevant for decision making, we consider the problem to assign seats in a parliament to members of parliament. We prove that the resulting *seating assignment problem* is strongly NP-hard in several restricted settings. We present a Mixed Integer Programming formulation of the problem, we describe two families of valid inequalities and we discuss symmetry-breaking constraints. Further, we design a heuristic. Finally, we compare the outcomes of the Mixed Integer Programming formulation with the outcomes of the heuristic in a computational study.

© 2021 Elsevier B.V. All rights reserved.

## 1. Introduction

A key institute in old and modern democracies is the parliament: a collection of persons that have been elected to represent the people and whose main task is to control legislative power. To exercise their right and duty of controlling power, there is a physical location where the elected representatives meet, discuss, and vote: the parliament. Typically, the parliament is a building where locations of seats are given, and hence, after an election, the question arises: who sits where?

While at first sight this question may appear an innocent one, there have been intense debates and rows in several parliaments around the globe about this matter. Before giving an overview of some of these debates, we first motivate that seat allocations are far from innocent, and may actually influence voting behavior. Saia (2018) considers the situation in Iceland, where seats for members of parliament (MPs) are allocated randomly, independent of party affiliation. This has created the opportunity to statistically test whether neighbors of an MP have an impact on the MP's voting behavior, and it is shown in Saia (2018) that the answer is affirmative: not only voting behavior, but even the choice of words is influenced by one's neighbors in the parliament. The existence of this effect is confirmed by Harmon, Fisman, & Kamenica (2019) in a study devoted to the European Parliament. From this, we conclude

that seating allocations are important, and this finding motivates our work.

We now sketch a number of elections where arriving at a seating allocation after an election was a contested matter.

- Following the 2017 elections for the House of Representatives in the Netherlands, newspapers reported a lot of debate on seating assignment. For instance, Kok (2017) describes how the party PvdA, after losing many seats in the elections, is “punished” by losing their front row seats. This shows that not all seats are equally important. Seats in the front have more visibility and allow direct access to the debating spot. Usually, large parties occupy one or more front seats. Abels, Besselink, & Zuidervaart (2017) mention that the seating assignment is a precarious and not to be underestimated matter, and that “redoing the seating assignment puzzle is like doing higher mathematics”. We return to this case in Section 8.
- The 2019 elections for the House of Representatives in Finland. Here, the Swedish People's Party wanted to move closer to the centre, leaving the Finns Party on the right of the parliament, something they were very unhappy about.<sup>1</sup> Many democracies have parties that are labeled somewhere on a “left-right” spectrum. This left-right positioning is often reflected in the allocation of seats. Phrased more generally, MPs from different parties that are considered to hold similar views are allocated to neighboring positions in the parliament.
- The 2017 elections for the Bundestag in Germany. Leading members of Germany's established parties opposed an arrange-

\* Corresponding author.

E-mail addresses: [vangerven@uni-wuppertal.de](mailto:vangerven@uni-wuppertal.de) (B. Vangerven), [briskorn@uni-wuppertal.de](mailto:briskorn@uni-wuppertal.de) (D. Briskorn), [dries.goossens@ugent.be](mailto:dries.goossens@ugent.be) (D.R. Goossens), [f.c.r.spieksma@tue.nl](mailto:f.c.r.spieksma@tue.nl) (F.C.R. Spieksma).<sup>1</sup> See <https://newsnowfinland.fi/politics/new-parliament-seating-plan-solved-but-not-everyone-is-happy>. Accessed: April 28, 2020.

ment where members of the party AfD sit close to the government's bench (Thureau, 2017).

- The 2014 Elections in Flanders, Belgium. The “left-right” spectrum issue came up here, along with the biggest party not wanting to be split up (i.e. have seats separated by corridors).<sup>2</sup>

The problem of finding an acceptable seating assignment is not just limited to examples given above. Seating assignments have, for example, also been the topic of debate in France (Désir, 2016) and the UK.<sup>3</sup>

In the seating assignment problem we assume that each MP receives a particular seat. This is common practice in most parliaments but not in all; notable exceptions are the House of Representatives in the USA, and the House of Commons in the UK. A key property of seating assignments encountered in practice is that in a vast majority of cases MPs from the same party are seated in clusters. This reflects that communication between MPs of the same party is an important factor when determining a seating assignment. More concretely, members of the same party should be seated in each other's vicinity; this allows them to pass information and notes quickly and discretely. Thus, to facilitate intra-party communication, neighboring seats (to be defined later) are as much as possible allocated to MPs of the same party.

We are given a parliament layout, a number of political parties, and for each party, the number of seats it is entitled to. We will model the seating assignment problem by constructing a graph where a node corresponds to a seat, and where neighboring seats are represented by an edge connecting the two corresponding nodes. In our definition of the seating assignment problem, we model the above-described key property by demanding that seats assigned to members of the same party induce a connected subgraph. Our objective is to maximize the number of connections between nodes assigned to the same party, see Section 3 for a precise problem description.

In some parliaments, clear rules exist that yield a seat-allocation. For instance, in the US Senate, senators are ordered by seniority and, starting with the oldest senator, each senator gets to choose a seat with Republicans on the right and Democrats on the left side of the chamber; in Iceland, as mentioned above, a draw determines which MPs sit where. Typically, there is no clear procedure regarding seat assignment; the seating proposals follow from a combination of tradition and negotiations. Hence, a neutral, optimization-based approach can help reduce such discussions. The aim of this paper is to explicitly identify this problem, to formally model it, to derive insights in its complexity, and to test computationally the efficiency of various approaches.

The remainder of this paper is structured as follows. We give an overview of related literature in Section 2. Next, in Section 3, we formally define the seating assignment problem. Section 4 contains computational complexity results. In Section 5, we present a Mixed Integer Programming (MIP) formulation. We also present two classes of valid inequalities, and we illustrate how these strengthen the linear programming relaxation of our MIP formulation. We address the issue of symmetry-breaking constraints. We describe in Section 6 a heuristic that solves an instance of a set partitioning problem using integer programming; here, the user has control over the size of the instance. We test both the mathematical programming formulation using state-of-the-art MIP solver Cplex, as well as the heuristic, in Section 7. We discuss the seating

of the Dutch House of Representatives as a case study in Section 8. Finally, we conclude the paper in Section 9.

## 2. Related literature

Seating assignments can be represented with graphs, where nodes correspond to seats and edges represent seat adjacencies. Assigning seats to parties constitutes a variant of vertex partitioning. Many types of graph partitioning problems are well-known to be NP-complete (see e.g. Garey, Johnson, & Stockmeyer, 1976). Regarding vertex partitioning, one related work is by Dyer & Frieze (1985). They show that finding an arbitrary contiguous node partition into three sets is hard for bipartite, planar graphs. In Section 4 we discuss in more detail the relation between the results in Dyer & Frieze (1985) and this paper.

Another problem that is closely connected to our problem is known as connected MAX- $k$ -CUT, where the nodes of a given graph need to be partitioned into  $k$  connected components such that the number of edges between different parts is maximized. Haglin & Venkatesan (1991) show that this problem is NP-hard already for  $k = 2$ ; connected MAX- $k$ -CUT is studied in Hojny, Joormann, Lüthen, & Schmidt (2021); they mention various applications, study different formulations, and investigate these experimentally. Polyhedral aspects of the connectivity constraints are addressed in Wang, Buchanan, & Butenko (2017) and Oosten, Rutten, & Spijksma (2007). Our setting differs from this class of problems since in our case the size of each of the components is given and we are dealing with a different maximization objective.

Darlay, Brauner, & Moncel (2012) investigate dense and sparse vertex partitioning. Their goal is to maximize the sum of the densities of each subset in the partition. They prove NP-hardness and non-approximability for this problem. Darlay et al. (2012) do not, however, fix the number of subsets in the partition, nor their size, whereas in our problem both are given (and fixed).

Another related work is done by Benati, Puerto, & Rodríguez-Chía (2017). They propose a new model for clustering so-called *attributed graphs*. An attributed graph, aside from having the typical node set  $V$  and edge set  $E$  which correspond to relational data, also contains a matrix  $M$ , which contains information on similarities according to different features. Benati et al. (2017) propose MIP formulations to solve a clustering problem, with the constraint that there must be a path in  $G$  between nodes clustered in the same group. In essence, this is a contiguity-constraint, which is similar to our problem setting. However, the objective considered in Benati et al. (2017) is a typical clustering objective; nodes within each cluster should be similar to each other and nodes from different clusters should be dissimilar with respect to the information in  $M$ . In our problem setting, we want to maximize the number of connections between nodes assigned to the same party. In other words, we try to cluster based on relational data (the number of connections), whereas Benati et al. (2017) cluster based on individual node data. A final difference is that in the work by Benati et al. (2017) all nodes have to be assigned to some cluster, whereas that is not necessarily the case in this work (the number of seats can be greater than the number of MPs).

Clique partitioning and clique covering (see e.g. Pullman, 1983 for a survey), are other related, yet different problems. The problem of finding a minimum clique cover is a well-known NP-hard problem (Karp, 1972). In this paper, we also want to partition (or cover) the nodes of a graph, however we do not require the partitions to be cliques.

There is also a stream of literature that studies problems related to (political) districting or zoning. In districting, a region needs to be divided into a set of districts. In essence this is a type of graph partitioning problem where contiguity is paramount. Shirabe (2009) considers three variations of contiguity-based dis-

<sup>2</sup> See: <https://www.knack.be/nieuws/belgie/onenigheid-in-vlaams-parlement-cd-v-en-n-va-willen-in-het-midden-zitten/article-normal-155667.html> [in Dutch]. Accessed: April 28th, 2020.

<sup>3</sup> See: <https://www.bbc.com/news/uk-scotland-scotland-politics-32802374>. Accessed: April 28, 2020.

tricting problems, and models these using integer programming. Next, he uses the 48 contiguous US states as an instance.

King, Jacobson, Sewell, & Cho (2012), King, Jacobson, & Sewell (2015) also analyze political districting. They primarily focus on so-called *geo-graphs*, a graph model that provides a scale-invariant method for enforcing contiguity constraints in local search methods.

### 3. The seating assignment problem

In this section we formalize the seating assignment problem and emphasize some relevant special graph structures. We denote the set of seats in the parliament by  $V = \{1, \dots, n\}$ . The set  $P$  represents the political parties. Every party  $p \in P$  is entitled to a number  $\beta^p$  of seats. We consider the weighted graph  $G = (V, E, w)$ , which represents the seats and their adjacency relations. There is an edge  $e = \{i, j\} \in E$  between nodes  $i \in V$  and  $j \in V$  if the corresponding two seats are adjacent in the parliament. Adjacency can occur when two seats are next to each other on the same row; in addition, seats on consecutive rows can also be adjacent. There is a given weight  $w_e \in \mathbb{R}$ ,  $e \in E$ , which reflects the degree to which seats  $i$  and  $j$  allow communication. We restrict ourselves to  $0 < w_e \leq 1$ , where higher weights reflect better communication opportunities.

**Definition 1.** Given a weighted graph  $G = (V, E, w)$ , and given a set  $P$  with numbers  $\beta^p$  for each  $p \in P$ , a *seating assignment* is an assignment of nodes to parties, such that (i) each party  $p \in P$  gets assigned  $\beta^p$  nodes and (ii) every node in  $V$  is assigned at most once.

Members of the same political party want to sit in a way that is *contiguous*.

**Definition 2.** A seating assignment is *contiguous* if, for each political party  $p \in P$ , the sub-graph induced by the nodes assigned to  $p$  is connected.

In other words, in a contiguous seating assignment there exists at least one path between every two nodes assigned to the same party, using only nodes assigned to that same party.

In the following we introduce one more constraint motivated by interests of political parties in the real world. As made clear in the article by Abels et al. (2017), seats on the front row are usually of special interest to political parties. They are desired, valued above other seats, for reasons of exposure: the seats are in plain view of the camera, and are usually situated close to the speaker microphones. It follows that assigning a fair number of front row seats to every party is a desirable. Let  $F \subseteq V$  be the set of front row seats/nodes. A fair distribution of front row seats assigns a given number of  $r^p$  seats in  $F$  to each party  $p \in P$ . There are several methods available to determine  $r^p$ , e.g. the D'Hondt method (Gallagher, 1991).

**Definition 3.** Given numbers  $r^p$  for each  $p \in P$ , a seating assignment is *front-fair* if each political party  $p \in P$  gets assigned at least  $r^p$  seats in  $F$ .

**Definition 4.** Given a graph  $G = (V, E, w)$ , a set  $P$  of parties, and numbers  $\beta^p, r^p$  for each party  $p \in P$ , the *Seating Assignment Problem* (SAP) asks for a contiguous and front-fair seating assignment that maximizes the total weight of edges between nodes assigned to the same party.

### 4. Computational complexity results

We gather and present the computational complexity results in this section. While we state the results here, we relegate the

technical proofs to [Appendix A](#). The results can be summarized as follows. First, we prove that finding an optimal contiguous seating assignment is strongly NP-hard even for considerably restricted classes of graphs and even if front row seats do not need to be accounted for. Second, we show that it is strongly NP-complete to determine whether a contiguous and front-fair seating assignment exists even on a considerably restricted class of graphs.

**Theorem 1.** SAP is strongly NP-hard even for connected planar graphs with maximum degree of at most 3 and  $w_e = 1$  for each edge  $e \in E$ , if  $r^p = 0$  for each party  $p \in P$ , and if each seat is occupied.

**Corollary 1.** SAP is strongly NP-hard even for planar graphs with maximum degree of at most 2 and  $w_e = 1$  for each edge  $e \in E$ , if  $r^p = 0$  for each party  $p \in P$ , and if each seat is occupied.

These results relate to the results of Dyer & Frieze (1985) as follows. Dyer & Frieze (1985) show (Theorem 2.1(a)) that finding an arbitrary contiguous node partition into three sets is hard for bipartite, planar graphs. They do not consider partition sizes, hence, our results are incomparable.

However, they also show that (Theorem 2.2) the problem to decide whether a feasible solution to SAP exists is strongly NP-complete even for bipartite graphs and even for two parties. We show in [Theorem 1](#) that SAP is strongly NP-hard even for connected planar graphs with maximum degree of at most 3 and even if each seat is occupied. In our results, planarity comes into play. Additionally, we strengthen the result by bounding the degree. [Corollary 1](#) strengthens this again, but sacrifices connectedness of the graph in the process.

We continue by considering SAP on a further considerably restricted class of graphs. Some parliament seating schematics exhibit a grid-like shape. If this is the case, the corresponding graph results in a *grid graph*, see Itai, Papadimitriou, & Szwarcfiter (1982), from which we take the following definition. Let  $G^\infty$  be the infinite graph whose vertex set consists of all points of the plane with integer coordinates and in which vertices are connected if and only if the (Euclidian) distance between them is equal to 1. A grid graph is a finite, node-induced subgraph of  $G^\infty$ . In the following we consider a particular type of grid graphs.

**Definition 5.**  $G(m, m)$ ,  $m \in \mathbb{N}^+$ , is the grid graph which has exactly the set of nodes with both coordinates in  $[1, m]$ .

**Theorem 2.** SAP is strongly NP-hard even for  $G(m, m)$ ,  $m \in \mathbb{N}^+$ , with  $w_e = 1$  for each edge  $e \in E$ , if  $r^p = 0$  for each party  $p \in P$ , and if each seat is occupied.

Dyer & Frieze (1985) consider bipartite graphs and [Theorem 1](#) considers planar graphs. Grid graphs are both bipartite and planar. Note that while [Theorem 1](#) restricts to a maximum degree of 3, [Theorem 2](#) restricts to a proper special case of connected planar graphs namely grid graphs (which have maximum degree of 4). Thus, the results in [Theorem 1](#) and [2](#) are not comparable.

Finally, we state that existence of proper seating assignment becomes a non-trivial issue if front row seats come into play.

**Theorem 3.** Deciding whether a contiguous and front-fair seating assignment exists is strongly NP-complete even for connected planar graphs with maximum degree of at most 3 and if each seat is occupied.

We point out that this result does not only hold for arbitrary numbers of front row seats granted for parties but also if the D'Hondt method is used to determine these numbers. We provide more details in [Appendix A](#).

### 5. A mixed-integer programming formulation of the SAP

We provide a mathematical programming formulation for the SAP in Section 5.1. In Section 5.2, we present two sets of valid inequalities, and in Section 5.3 we discuss symmetry-breaking constraints.

#### 5.1. Main mixed-integer programming formulation

There are different ways to model the SAP as a mixed-integer program; to enforce the contiguity constraints, we opt for a multicommodity flow formulation - this is in the spirit of formulations given in Hojny et al. (2021). We consider a directed graph  $G = (V, A)$ , where, like before, the nodes in  $V$  correspond to the seats. There is one node for every seat in the parliament. Observe that one can always (arbitrarily) number the nodes in  $V$ , e.g. from 1 to  $|V|$ , which implies an ordering of the nodes. We use a similar procedure to order the parties in  $P$ . Since our mathematical formulation ensures contiguity using a flow-based formulation, we use arcs instead of edges. There are arcs between nodes corresponding to adjacent seats in the set  $A$ . If and only if  $\{i, j\} \in E$ , then  $(i, j) \in A$  and  $(j, i) \in A$ . In other words, we replace every edge by two arcs.

There is a binary variable  $x_i^p$  which equals 1 if node  $i$  is assigned to party  $p$  and 0 otherwise. A binary variable  $y_e^p$  indicates whether nodes  $i$  and  $j$  defined by  $e = \{i, j\}$  are both assigned to party  $p$ .

We use a flow-based formulation to ensure contiguity: every party has one source node, from which an amount of flow emanates, that can only flow via existing arcs. To that end, we introduce binary variables  $z_i^p$ , that decide which node is the source node from which an amount  $\beta^p - 1$  of flow emanates, for every party  $p \in P$ . The variables  $f_{(i,j)}^p$  indicate the amount of flow that goes from a node  $i$  to another node  $j$  for a party  $p \in P$ .

$$\max \sum_{p \in P} \sum_{e \in E} w_e y_e^p \tag{1}$$

$$\text{s.t. } \sum_{p \in P} x_i^p \leq 1 \quad \forall i \in V, \tag{2}$$

$$\sum_{i \in V} x_i^p = \beta^p \quad \forall p \in P, \tag{3}$$

$$\sum_{i \in V} z_i^p = 1 \quad \forall p \in P, \tag{4}$$

$$z_i^p \leq x_i^p \quad \forall p \in P, i \in V, \tag{5}$$

$$y_e^p \leq x_i^p \quad \forall p \in P, e \in E, i \in e, \tag{6}$$

$$\sum_{j: (i,j) \in A} f_{(i,j)}^p \geq \beta^p z_i^p + \sum_{j: (j,i) \in A} f_{(j,i)}^p - 1 \quad \forall p \in P, i \in V, \tag{7}$$

$$0 \leq f_{(i,j)}^p \leq \beta^p y_e^p \quad \forall p \in P, e = \{i, j\} \in E, \tag{8}$$

$$\sum_{i \in F} x_i^p \geq r^p \quad \forall p \in P, \tag{9}$$

$$y_e^p \in \{0, 1\} \quad \forall p \in P, e \in E, \tag{10}$$

$$x_i^p, z_i^p \in \{0, 1\} \quad \forall p \in P, i \in V. \tag{11}$$

**Table 1**

Values of  $x$ -variables and objective function contribution of the LP-relaxation of the MIP formulation for the example in Fig. 1.

Party $p$	$x_1^p$	$x_2^p$	$x_3^p$	$x_4^p$	$x_5^p$	$x_6^p$	$x_7^p$	$x_8^p$	$x_9^p$	$\sum y^p$
1	1/9	1/9	1/9	1/9	1/9	1/9	1/9	1/9	1/9	10/9
2	1/9	1/9	1/9	1/9	1/9	1/9	1/9	1/9	1/9	10/9
3	1/9	1/9	1/9	1/9	1/9	1/9	1/9	1/9	1/9	10/9
4	2/3	2/3	2/3	2/3	2/3	2/3	2/3	2/3	2/3	20/3

The objective function (1) represents the goal to maximize the weighted number of arcs between nodes assigned to the same party. Constraints (2) ensure that every node is assigned at most once, implying that there are seats that can potentially be left empty, and constraints (3) make sure that every party gets assigned the correct number of nodes. Constraints (4) and (5) take care of the flow source selection: one source node needs to be selected for every party, and that source node needs to be assigned to that party. Constraints (6) restrict the  $y_e^p$  variables. Specifically, variable  $y_e^p$  with  $e = \{i, j\}$  can only equal 1 if both nodes  $i$  and  $j$  are assigned to party  $p$ . Finally, Constraints (7) and (8) take care of the flow requirements, implying that only contiguous seating assignments are feasible. Constraints (7) ensure that the source node of party  $p$  has a supply of  $\beta^p - 1$  while non-source nodes have a flow demand of 1. Constraints (8) make sure that only arcs connecting nodes of party  $p$  can be used to establish a flow for  $p$ . Notice that for each edge  $e$  both corresponding arcs are restricted that way. Now it is easy to verify that a flow of  $\beta^p - 1$  originating from the source node can be established only if each node can be reached from the source node on a path only visiting nodes assigned to  $p$ . Constraints (9) require that each party  $p \in P$  gets at least its granted number of front row seats. Constraints (10) and (11) are the integrality constraints.

Finally, we remark that we can relax the constraint that  $y_e^p$  is binary, because this is implied by the integrality of  $x_i^p$  and the nature of the objective function. Similarly, we can also relax the constraint that  $z_i^p$  is binary.

#### 5.2. Valid inequalities

We present two different sets of inequalities valid for formulation (2)–(11). The first type of valid inequalities is graph-specific. Let  $\Delta(G)$  be the maximum degree of a node  $v \in V$  in the original graph (with edges), that is the maximum number of seats any seat is adjacent to. Then, the following inequalities are valid:

$$\sum_{e \in E} y_e^p \leq \frac{\beta^p}{2} \cdot \min\{(\beta^p - 1), \Delta(G)\} \quad \forall p \in P. \tag{12}$$

The idea behind these inequalities is as follows. Consider the  $\beta^p$  nodes assigned to party  $p \in P$ . Since, in a solution to the SAP, an individual node is connected to at most  $\min(\beta^p - 1, \Delta(G))$  other nodes, it follows that the number of edges that can be selected in the graph induced by nodes assigned to party  $p$  (i.e., the left-hand side of (12)), is bounded by  $\frac{\beta^p}{2} \cdot \min(\beta^p - 1, \Delta(G))$  (where we divide by 2 to account for double counting of each individual edge).

To see the potential impact of inequalities (12), consider the instance depicted in Fig. 1. There are 9 nodes, and there are four parties with  $\beta^1 = \beta^2 = \beta^3 = 1$  and  $\beta^4 = 6$ . The optimal integer solution has an objective function value of 6 and assigns either nodes 1 to 3 or nodes 7 to 9 to parties 1, 2, and 3, and the remaining nodes to party 4. An optimal solution of the LP-relaxation of (1)–(11) has an objective function value of 10 (i.e. the number of edges); the corresponding values of the  $x$ -variables are given in Table 1. After adding inequalities (12), the objective function value drops to



**Table 2**

Values of  $x$ -variables and objective function contribution of the LP-relaxation of the MIP formulation for the example in Fig. 1 including inequalities (12).

Party $p$	$x_1^p$	$x_2^p$	$x_3^p$	$x_4^p$	$x_5^p$	$x_6^p$	$x_7^p$	$x_8^p$	$x_9^p$	$\sum y^p$
1	0	1/3	0	0	0	1/3	1/3	0	0	0
2	0	0	1/3	0	1/3	0	0	0	1/3	0
3	1/3	0	0	1/3	0	0	0	1/3	0	0
4	2/3	2/3	2/3	2/3	2/3	2/3	2/3	2/3	2/3	20/3

**Table 3**

Values of  $x$ -variables and objective function contribution of the LP-relaxation of the MIP formulation for the example in Fig. 1 including inequalities (13).

Party $p$	$x_1^p$	$x_2^p$	$x_3^p$	$x_4^p$	$x_5^p$	$x_6^p$	$x_7^p$	$x_8^p$	$x_9^p$	$\sum y^p$
1	0	0	0	0	0	0	1/3	1/3	1/3	1
2	1/6	1/6	1/6	0	0	0	1/6	1/6	1/6	1
3	1/3	1/3	1/3	0	0	0	0	0	0	1
4	1/2	1/2	1/2	1	1	1	1/2	1/2	1/2	6

**Table 4**

Values of  $x$ -variables and objective function contribution of the LP-relaxation of the MIP formulation for the example in Fig. 1 including both inequalities (12) and (13).

Party $p$	$x_1^p$	$x_2^p$	$x_3^p$	$x_4^p$	$x_5^p$	$x_6^p$	$x_7^p$	$x_8^p$	$x_9^p$	$\sum y^p$
1	0	0	0	0	0	0	0	1/3	2/3	0
2	1/3	0	0	1/5	1/5	1/5	0	0	0	0
3	0	1/4	1/5	0	0	0	1/5	1/3	0	0
4	2/3	2/3	4/5	4/5	4/5	4/5	4/5	1/3	1/3	31/5

$\frac{20}{3}$ , and the corresponding values of the  $x$ -variables when (12) is added, are given in Table 2.

The second type of valid inequalities is contiguity-specific. They can be obtained by first calculating a shortest path (with arcs having unit length) between every pair of distinct nodes in  $G = (V, A)$ . The all-pairs shortest path problem is a well-known and well-researched problem that be solved in polynomial time (e.g. by applying the Floyd–Warshall algorithm (Cormen, Leiserson, Rivest, & Stein, 2009)). We denote by  $d_{(i,j)}$  the length of the shortest path from  $i \in V$  to  $j \in V$ . The following inequalities are then valid:

$$x_i^p + x_j^p \leq 1 \quad \forall p \in P, i, j \in V : i < j \text{ and } d_{(i,j)} \geq \beta^p. \quad (13)$$

Indeed, two nodes  $i, j \in V$  that are separated in  $G$  by more than  $\beta^p$  nodes cannot be assigned to the same party.

To see the potential impact of inequalities (13), consider again the instance depicted in Fig. 1. After adding inequalities (13) for the fourth party, the value of an optimal solution of the LP-relaxation equals 9. The corresponding values of the  $x$ -variables are given in Table 3.

Clearly, adding inequalities (12) or adding inequalities (13) to the MIP formulation (1) to (11) strengthens the LP-relaxation. Even more, we point out that when adding inequalities (12) as well as inequalities (13) to the MIP formulation describing the instance depicted in Fig. 1, the optimal value of the LP-relaxation drops to 6.2, which is lower than the value resulting from only adding (12) or only adding (13). The corresponding values of the  $x$ -variables are given in Table 4. This example thus shows that combining the two classes of inequalities may lead to a bound that is stronger than when considering a single class of inequalities.

5.3. Symmetry-breaking constraints

Another issue that is relevant when efficiently solving the MIP formulation is the presence of symmetry. We distinguish two types of symmetry in our formulation. The first one relates to the selection of the source node among the nodes/seats assigned to the same party, while the second one aims to eliminate symmetry with respect to parties of equal size.

**Table 5**  
Computational results of the MIP formulations.

Instances	R	P	s	Var.	MIP standard			MIP cuts			MIP cuts + (15)			MIP cuts + (16)		
					Obj.	Gap	Time	F/B/O	Obj.	Gap	Time	F/B/O	Obj.	Gap	Time	F/B/O
4	5	31.7	81.5	0.0%	12.9	0.0%	10/10/10	81.5	0.0%	12.5	0.0%	10/10/10	81.5	0.0%	11.7	10/10/10
4	5	10	17.2	0.0%	38.4	0.0%	10/10/10	78.0	0.0%	38.1	0.0%	10/10/10	78.0	0.0%	67.3	10/10/10
4	5	15	2.6	0.0%	155.6	0.0%	10/10/10	74.9	0.0%	67.5	0.0%	10/10/10	74.9	0.0%	103.3	10/10/10
4	6	5	11.7	0.0%	144.4	0.0%	10/10/10	74.9	0.0%	111.4	0.0%	10/10/10	74.6	0.0%	124.7	10/10/10
4	6	10	6.8	0.0%	262.0	0.0%	10/10/10	71.9	0.0%	239.3	0.0%	10/10/10	71.9	0.0%	451.4	10/10/10
4	6	15	0.7	0.0%	36.7	0.0%	10/10/10	72.0	0.0%	46.5	0.0%	10/10/10	72.0	0.0%	205.1	10/10/10
4	7	5	14.0	0.7%	1029.1	0.0%	10/10/9	69.7	0.0%	806.1	0.0%	10/10/10	69.7	0.0%	754.4	10/10/10
4	7	10	2.6	3.9%	1658.3	3.1%	10/9/3	65.9	3.1%	1521.7	8.0%	10/10/4	65.6	7.0%	1954.8	10/8/1
5	7	5	37.4	114.1	5.9%	1918.1	10/4/2	114.0	6.2%	2000.3	8.2%	10/1/0	112.7	8.6%	2000.2	10/3/0
5	7	10	5.6	101.9	19.3%	2000.2	10/3/0	98.4	23.3%	2000.3	22.4%	2000.1	100.9	21.3%	2000.1	9/2/0
5	8	5	20.7	102.8	19.1%	2000.3	10/1/0	105.7	15.1%	2000.2	18.1%	2000.1	104.1	17.2%	2000.2	10/3/0
5	8	10	1.5	91.7	30.7%	2000.2	8/0/0	91.6	30.9%	2000.3	30.0%	2000.1	139.0	32.8%	2000.1	4/0/0
5	9	5	19.4	95.4	28.8%	2000.2	10/0/0	96.3	26.2%	2000.2	33.5%	2000.3	92.1	32.9%	2000.2	6/0/0
5	9	10	0.3	89.0	28.9%	2000.2	6/0/0	-	-	2000.2	-	2000.1	-	-	2000.1	0/0/0
6	8	5	30.7	150.4	21.5%	2000.1	8/0/0	149.0	23.4%	2000.1	22.3%	2000.1	150.0	20.7%	2000.1	4/0/0
6	8	10	2.8	-	2000.1	0/0/0	128.5	40.9%	2000.1	33.7%	2000.1	135.5	31.4%	2000.1	1/1/0	
6	9	5	15.6	144.8	26.5%	2000.1	2/0/0	143.5	27.7%	2000.1	32.8%	2000.2	138.5	31.6%	2000.1	6/0/0
6	9	10	0.9	-	2000.1	0/0/0	131.0	37.0%	2000.1	-	-	2000.1	-	-	2000.1	0/0/0
6	10	5	13.5	127.2	45.0%	2000.1	3/0/0	132.5	38.4%	2000.1	-	2000.1	-	-	2000.1	0/0/0
Average							77.4%/45.8%/38.9%					66.8%/42.1%/36.8%				68.4%/45.8%/36.8%

In order to reduce the number of choices of the source node for a given assignment of seats to a party we propose to choose the lowest-indexed seat as the source node. The following constraints achieve this choice:

$$\sum_{j=1}^i z_j^p \geq x_i^p \quad \forall p \in P, i \in V. \tag{14}$$

These constraints ensure that whenever a node is assigned to a party, the corresponding source node has an index that is lower or equal. We use these inequalities (14) in all further computations.

The second type of constraints account for two parties of the same size being exchangeable in a solution (which yields a different solution with the same objective value):

$$z_i^p \leq \sum_{i'=1}^{i-1} z_{i'}^{p'} \quad \forall p', p \in P, p' < p, \beta^p = \beta^{p'}, i \in V, i > 1, \tag{15}$$

$$\sum_{i \in V} i \cdot z_i^p \leq \sum_{i \in V} i \cdot z_i^{p'} - 1 \quad \forall p', p \in P, p' < p, \beta^p = \beta^{p'}. \tag{16}$$

Each of these constraints enforce that the source nodes of parties that have the same size are assigned to seats that are indexed in the same order as the order of the parties. A similar type of symmetry-breaking constraints can be found in Grimm, Kleinert, Liers, Schmidt, & Zöttl (2019) and Méndez Díaz & Zabala (2001).

### 6. A heuristic for the SAP

In this section, we present a heuristic in order to determine good solutions for SAP instances. While the heuristic can be applied to any type of graph, it has been developed with graphs that represent the structure of a typical parliament in mind (see Section 7). The heuristic consists of two phases. First, we generate for each party  $p$  a set of connected subgraphs of the graph  $G = (V, E)$ . Each such subgraph corresponds to a contiguous subset of  $\beta^p$  seats (nodes in  $G$ ), including the required number of front row seats  $r^p$ . Note that settings without front row seat requirements can also be handled with this approach, by simply setting  $r^p = 0$  for each party  $p \in P$ . Second, we select one subgraph per party such that there is no overlap, maximizing the weighted number of edges between nodes assigned to the same party. The result is a contiguous seating assignment.

*Phase 1.* For each party  $p$ , we generate a set  $\mathcal{N}^p$  of connected subgraphs of  $G$  as follows. We make a distinction depending on whether or not the party is entitled to front row seats.

- If the party  $p$  is not entitled to front row seats ( $r^p=0$ ), we create connected subgraphs for every node  $n \in V$  as a starting node. We initialize a new connected subgraph implied by node-set  $N = \{n\}$ . Next, we consider the nodes adjacent to a node in  $N$ , but not including any front row seats (nor nodes that are already in  $N$ ). From this set of neighboring nodes, denoted by  $\delta_{V \setminus F}(N)$ , we select one node using uniform probabilities and add it to  $N$ . We continue doing this until  $N$  has the required size  $\beta^p$ . Finally, we add  $N$  to  $\mathcal{N}^p$ , and keep repeating this procedure until we have a given number of  $Q^p$  (chosen as a parameter depending on the size of party  $p$ ) connected subgraphs for every node  $n \in V$  as starting point.
- If the party is entitled to  $r^p > 0$  front row seats, we generate all sets of  $r^p$  connected front row seats. For each such set  $D$ , we initialize  $N$  to  $D$ , and we iteratively add neighboring nodes to  $N$  as we did for the case of  $r^p=0$ . We repeat this  $Q^p$  times for each  $D$ , and add each resulting  $N$  to  $\mathcal{N}^p$ . The motivation behind using a set of connected front row seats, is twofold: (i) ensuring a connected subgraph is made trivial, and (ii) using disconnected

(sets of) front row seats can lead to potentially “exotic” subgraphs, which, given graphs that reflect the layout of a typical parliament, are likely to contribute less to the objective function value than subgraphs in which all front row seats are connected.

The pseudo code of the procedure for phase 1 is given in Algorithm 1.

```

generate subgraphs
for q=1 to Qp do
  while |N| < βp do
    Draw {n'} from δV\F(N) using probabilities pn' =
      1 / |δV\F(N)| ;
    N ← N ∪ {n'} ;
  end
  Np ← Np ∪ {N} ;
end

main procedure
for each p in P do
  Np = ∅ ;
  if rp = 0 then
    for each n in V \ F do
      N = {n};
      generate subgraphs;
    end
  end
  else
    for each D ⊆ F such that |D| = rp and D is
      connected do
      N = D;
      generate subgraphs;
    end
  end
end
end
    
```

**Algorithm 1:** Pseudo code for phase 1.

*Phase 2.* We denote  $\mathcal{N}$  as the superset including all subgraphs in  $\mathcal{N}^p$  over all parties. Each such  $N \in \mathcal{N}$  can be characterized by (i) the corresponding party  $p_N$  and (ii) its weighted number of edges  $w_N$  (i.e.,  $w_N = \sum_{e=\{i,j\} \in E: i,j \in N} w_e$ ). We declare a binary variable  $x_N$  for each  $N \in \mathcal{N}$ , which takes value 1 if party  $p_N$  will be seated according to  $N$ , and 0 otherwise. Solving the following integer programming (IP) formulation then provides the best solution we can obtain by assigning each party  $p$  a subgraph in  $\mathcal{N}^p$ .

$$\max \sum_{N \in \mathcal{N}} w_N x_N \tag{17}$$

$$\text{s.t.} \quad \sum_{N \in \mathcal{N}^p} x_N = 1 \quad \forall p \in P, \tag{18}$$

$$\sum_{N \in \mathcal{N}: n \in N} x_N \leq 1 \quad \forall n \in V, \tag{19}$$

$$x_N \in \{0, 1\} \quad \forall N \in \mathcal{N}. \tag{20}$$

Notice that the solution to the above IP formulation is indeed the optimum solution to the instance of SAP if for each party  $p$  set  $\mathcal{N}^p$  contains all subgraphs with  $\beta^p$  nodes and  $r^p$  front row seats.

The quality of the solution generated by the heuristic, and even whether the heuristic results in a feasible solution, depends on the subgraphs generated in phase 1. Most parliaments have a number

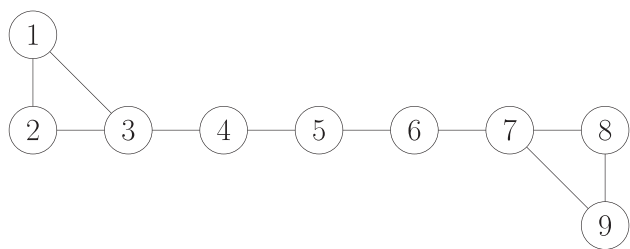


Fig. 1. An instance of the SAP with 9 seats where each edge has a weight of 1.

of seats per row that increases towards the back of the room. For graphs based on this layout and parties that have at least one front row seat, phase 1 is likely to result in favorable pie slice or drop shaped sets of seats, as the number of neighbors, and hence the chance of expanding in that direction, increases towards the back of the parliament. If the subgraphs present in an optimal solution are generated, then the heuristic will output an optimal solution as well. Herein lies the flexibility of the heuristic: adding more subgraphs may improve the quality of the solution, however, this comes at the cost of computation time. Finally, observe that, when implementing Phase 2, one may choose not to solve the IP to optimality, but instead be content with a solution that is, say within 5% of the optimum. Although we did experiment with this idea, the outcomes were very similar to the outcomes we got when solving the IP to optimality; therefore we do not report here on these former outcomes.

### 7. Computational study

In this section, we discuss a computational study which compares the MIP model from Section 5 with the heuristic developed in Section 6 on a set of artificially generated instances.

The instances are generated according to the seating layout depicted in Fig. 2, which represents a fictitious parliament consisting of three blocks of seats. According to a book by Cohen de Lara & Mulder (2016), this layout is quite common in parliaments all over the world. The left- and rightmost blocks each have 2 seats on the front row, whereas the middle block is larger and has 3 front row seats. Rows increase in size as they are more to the back of the parliament. The weights used in the objective function are 1 for in-block connections (depicted with full lines in Fig. 2) and 0.5 for between-block connections (depicted as dotted lines in Fig. 2). Note that for seats on different blocks, only connections on the same row are considered. Each instance consist of the first 4, 5 or 6 rows, leading to settings with 46, 65, or 87 seats, respectively. Preliminary computational experiments showed that instances with 2 or 3 rows are trivial for both the exact and heuristic approaches: optimal solutions are always found in at most a couple of seconds. Hence, the results for these instances are not reported in this paper.

For every number of rows, we examine 3 different numbers of parties, ranging from 5 to 10. The party sizes are generated such that each party takes between  $s\%$  and  $50\%$  of the seats, and there is at least one party with size in  $[s\%, (s + 5)\%]$ , for values of  $s$  in  $\{5, 10, 15\}$ . A large value of  $s$  corresponds with a parliament filled with large parties of similar size, while a small value of  $s$  will result in at least one small party and more variance in party sizes. Note that  $s = 15$  is not possible for settings with more than 6 parties.

In total, our computational study involves 190 instances. Each line in Table 5 corresponds with a set of 10 instances with the indicated number of rows, number of parties, and  $s$ -value; the average variance of party sizes is given in the forth column.

Table 6  
Computational results of the heuristic.

Instances				Heuristic			
Rows	Parties	$s$	Var.	Obj.	Time	F	B
4	5	5	31.7	81.3	53.2	10	8
4	5	10	17.2	78.0	53.7	10	10
4	5	15	2.6	74.9	54.0	10	10
4	6	5	11.7	74.9	45.9	10	10
4	6	10	6.8	71.9	40.8	10	10
4	6	15	0.7	72.0	22.7	10	10
4	7	5	14.0	69.7	28.1	10	10
4	7	10	2.6	65.9	14.8	10	10
5	7	5	37.4	114.2	160.6	10	7
5	7	10	5.6	104.6	237.1	10	10
5	8	5	20.7	107.8	129.2	10	9
5	8	10	1.5	97.0	174.1	10	10
5	9	5	19.4	102.5	97.6	10	9
5	9	10	0.3	91.0	63.9	10	10
6	8	5	30.7	157.0	379.6	10	10
6	8	10	2.8	143.4	1457.1	6	6
6	9	5	15.6	151.2	295.8	10	10
6	9	10	0.9	140.6	318.4	10	10
6	10	5	13.5	145.3	288.2	10	10
Average						97.9%	94.2%

We tackled all instances using four MIP formulations:

- MIP standard:** formulation (1)–(11) and (14),
- MIP cuts:** formulation (1)–(14),
- MIP cuts + (15):** formulation (1)–(14) and (15), and
- MIP cuts + (16):** formulation (1)–(14) and (16).

Instances were solved using the state-of-the-art MIP solver Cplex 12.10, with a time limit of 2000 seconds. We compare the performance of the MIP models with an implementation of the SAP heuristic, where we set  $Q^p = 15 \cdot \beta^p$ . All the experiments were carried out on a laptop with a 3.70 Gigahertz processor and 64 gigabyte of RAM. The results are summarized in Tables 5 and 6.

In Tables 5 and 6 each line represents averages over 10 instances, with a given number of rows, number of parties, and  $s$ -value. We outline the variance of the party sizes, achieved objective function value, required computation time in seconds, and the number of instances for which a feasible, a best, and a proven optimal (F/B/O) solution was found by the corresponding method. Note that B is to be interpreted as best over the results by the MIP models (Table 5) and the heuristic (Table 6), and that the number of instances for which the time limit was hit is given by 10–0.

According to the results in Table 5, the performance difference between the various MIP models is small. Optimal solutions are found for nearly all instances with 4 rows by each of the MIP models. While adding valid inequalities (12) and (13) to the standard MIP reduces the computation time from 417.2 seconds to 355.2 seconds on average, no further reduction is obtained by also adding symmetry-breaking constraints (15) or (16). Looking at the instances with 5 or 6 rows, the MIP approaches hit the time limit for each instance. While this results at least in a feasible solution for most of the cases with 5 rows, this is not the case for the instances with 6 rows. In these cases, the cuts offer no consistent added value, since the gap as well as the percentage of instances for which a feasible and a best solution was found is very similar. Adding symmetry-breaking constraints (15) or (16) also does not offer a clear advantage.

Table 6 shows that, for the setting with 4 rows, the heuristic finds best solutions for all but two instances, where it terminates with a close-to-optimal solution. In fact, for all but six instances we can conclude the solution to be optimal since Cplex proved the corresponding objective value to be optimal using at least one MIP formulation. The heuristic manages to obtain these results in less

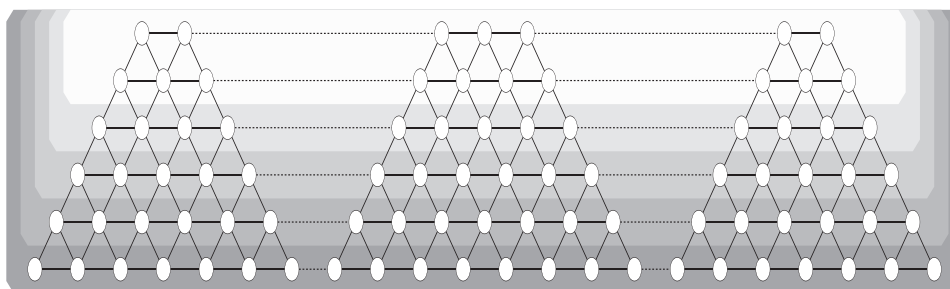


Fig. 2. Graph structure of the generated instances.

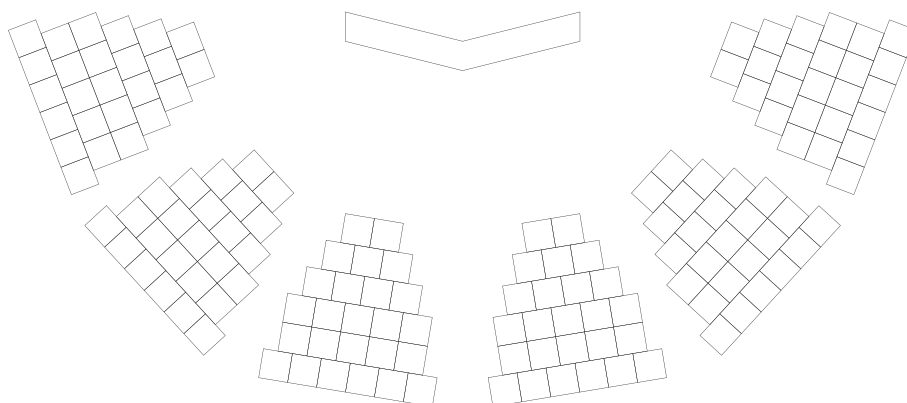


Fig. 3. Seating schematic of the Dutch House of Representatives (based on Cohen de Lara & Mulder, 2016).

than 40 seconds on average, and its computation times show very little variability. For the larger instances, and in contrast with the MIP formulations, the heuristic almost always finds a feasible solution. With an average computation time of 143.8 seconds for 5 row instances, and 547.8 seconds for 6 row instances, its computation time remains limited. The bulk of this computation time (98.8%) is spent on the second phase; the first phase accounts for more than 5% of the computation time in only one instance.

The bottom lines of the tables show that the heuristic clearly outperforms the MIP approaches with respect to the number of instances for which a feasible solution was found. Furthermore, for nearly 95% of the instances, the best found solution was obtained with the heuristic.

We have also experimented with setting the  $x$ -variables in the MIP-cuts model to the values implied by the solution obtained by the heuristic, as a (partial) initial solution. While this resulted in a significant decrease of computation time (often 30%) for the smaller instances, it did not improve the objective function value for the instances that could not be solved to optimality within 2000 seconds. Furthermore, the MIP could only very rarely improve the heuristic solution it started from (and never substantially). Given these very modest results, we opted not to include detailed results in this paper.

### 8. Case study: the Dutch House of Representatives

The House of Representatives in the Netherlands counts 150 seats, divided over six identical pie slice shaped blocks of seats. Each block has six rows, and includes two front-row seats, which are closest to the microphones and are prominently visible on TV broadcastings. Fig. 3 contains a schematic representation of the seating layout in the House of Representatives. The 2017 election gave rise to a rather dispersed composition of the parliament, involving no less than 15 parties. The largest party (VVD) holds 32 seats, followed by PVV with 20 seats; CDA and D66 each can claim

19 seats. There are several very small parties, including two one-person fractions (which, at some point had dissociated from larger parties).

The current seating assignment is depicted in Fig. 4 (top), and was highly contested, as described by several newspaper articles (e.g. Abels et al., 2017; Adriaanse & de Witt Wijnen, 2017; Kok, 2017; Mesdag, 2017; de Witt Wijnen, 2017). For example, the political party FvD, claiming to be a progressive middle party, did not want to be seated next to the right-wing populist party PVV. As reported in Section 1, the party PvdA was frustrated not to have a front-row seat, which is however not unreasonable given that the election left them with only 9 seats. The party DENK was complaining that they did not have access to a corridor, as they found themselves surrounded by other parties, making it difficult to reach the microphones. One MP of the party D66 did not like to be separated by a corridor from the rest of the party. This also happened with CDA and VVD, the MPs of the latter party were even spread over three blocks.

If we associate a weight of one for each depicted edge that connects neighboring seats that are assigned to the same party, and we use a weight of 0.5 for each connection between blocks (represented by dashed lines), the current solution has an objective value of 242.5. The middle row of Fig. 4 shows the result of running “MIP cuts” for little over 35 hours. The objective function value is 272.5, beating the current seating assignment. The bottom row of Fig. 4 depicts the result of our heuristic approach. In less than 30 minutes, we obtain an objective function value of 275, further improving the MIP result. Furthermore, this solution has other attractive properties: (i) each party has access to a corridor, and (ii) the contesting parties FvD and PVV do not occupy neighboring seats. No party is split over more than one block, except of course VVD, which is too large to be seating in a single block. Hence, our approach shows that it can alleviate most of the concerns that were expressed in the media.



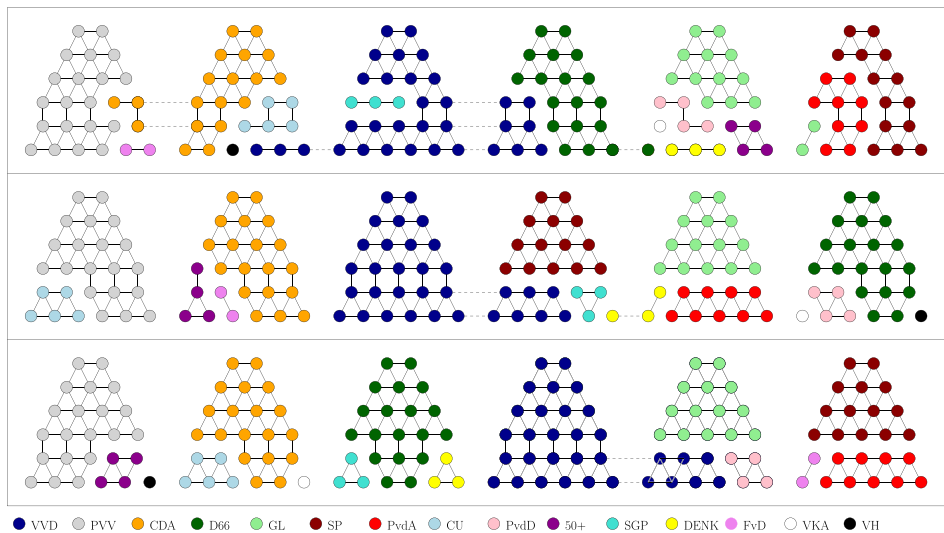


Fig. 4. Current seating assignment in the House of Representatives (top), the best found MIP solution (middle) and the solution resulting from our heuristic (bottom).

### 9. Conclusion

The seating assignment problem is a conceptually simple, but computationally challenging problem in combinatorics. Just like some politicians, we find that answering the question of who sits where is surprisingly difficult. Yet it is not impossible. The computational experiments show that even when using additional valid inequalities, exact MIP solvers have difficulties coming up with optimal, or at times even feasible, solutions within reasonable computation times. Heuristics, however, prove to be a promising avenue to tackle such problems. The case study on the House of Representatives in the Netherlands further illustrates that our heuristic is able to solve a real-life problem adequately within a reasonable computation time.

This research can also have merits in practical applications other than seating assignments in parliaments. For example, there is a resemblance to assigning professors and research assistants to offices in a building. Naturally, one would want to maximize the communication possibilities between members of the same research group (which would correspond to a political party in this paper). It also makes sense to not spread these people randomly over a building. The concept of seat adjacency carries over to office adjacency. If multiple people can share the same offices, then there is an additional extension that does not arise in a parliament seating setting: office capacities. This type of assignment of persons (linked to certain groups) to offices, taking into account communication possibilities, undoubtedly also arises in private companies, government building, etc.

### Acknowledgments

The research of Frits C.R. Spieksma was partly funded by the Netherlands Organization for Scientific Research (NWO) through Gravitation grant NETWORKS 024.002.003.

### Appendix A. Proofs

**Theorem 1.** *SAP is strongly NP-hard even for connected planar graphs with maximum degree of at most 3 and  $w_e = 1$  for each edge  $e \in E$ , if  $r^p = 0$  for each party  $p \in P$ , and if each seat is occupied.*

The proof is based on a reduction from 3-partition which is well-known to be NP-hard in the strong sense (Garey & Johnson, 1979). The 3-partition problem (3PP) is as follows. You are

given  $3b + 1$  positive integers  $a_1, \dots, a_{3b}, B$  with  $\frac{B}{4} < a_q < \frac{B}{2}$  for each  $q = 1, \dots, 3b$  and  $\sum_{q=1}^{3b} a_q = bB$ . Does there exist a partition of set  $\{1, 2, \dots, 3b\}$  into  $b$  subsets  $A_1, \dots, A_b$  of three integers each such that  $\sum_{q \in A_r} a_q = B$  for each  $r = 1, \dots, b$ ?

**Proof.** First, we argue that we can restrict ourselves to a special case of 3PP where each number is a multiple of 3. We can transform each instance of 3PP into an instance of this special case by multiplying each number  $a_1, \dots, a_{3b}, B$  by 3. The answer to the new instance is yes if and only if the answer to the original instance is yes.

We consider the decision problem whether or not a feasible solution to SAP with a target objective value (or more) exists. Given an instance  $I$  of 3PP with  $a_q$  being a multiple of 3 for each  $q = 1, \dots, 3b$  we construct an instance  $I'$  of SAP as follows. We have two types of nodes that make up the node set  $V = V^t \cup V^c$ .

- We have  $bB$  triangle nodes  $V^t = V_1^t \cup \dots \cup V_b^t$ ,  $V_r^t = \{(r-1)B + 1, \dots, rB\}$  for each  $r = 1, \dots, b$ . In Fig. A.5, these nodes are not colored.
- We have  $b - 1$  connecting nodes  $V^c = \{bB + 1, \dots, bB + (b - 1)\}$ . In Fig. A.5, these nodes are shaded grey.

We have three types of edges. Each edge has unit weight.

- The first type of edges is such that each triple  $3s + 1, 3s + 2, 3s + 3$  of nodes with  $s = 0, \dots, bB/3 - 1$  forms a triangle. Since we restricted ourselves to a special case of 3PP where each number is positive and a multiple of 3,  $bB/3 - 1$  is integer. We refer to these edges as triangle edges.
- The second type of edges directly connects triangles. Node  $Bg + 3s + 3$  is connected to node  $Bg + 3s + 4$  for every  $g = 0, \dots, b - 1$  and  $s = 0, \dots, B/3 - 2$ . Again, since  $B$  is a positive multiple of 3,  $B/3 - 2$  is integer.
- The third type of edges indirectly connects triangles via nodes in  $V^c$ . Node  $Bg$  is connected to node  $bB + g$  which in turn is connected to node  $Bg + 1$  for every  $g = 1, \dots, b - 1$ . In Fig. A.5, these edges are depicted as fat grey lines.

By construction, the graph has maximum degree of 3 and is planar. We have  $4b - 1$  parties. Party  $q = 1, \dots, 3b$  needs to get  $a_q$  seats. Parties  $3b + 1, \dots, 4b - 1$  each need to get a single seat. This completely specifies an instance  $I'$  of SAP.

The reduction is in pseudo-polynomial time. We claim that we can achieve a total weight of edges between nodes assigned to the

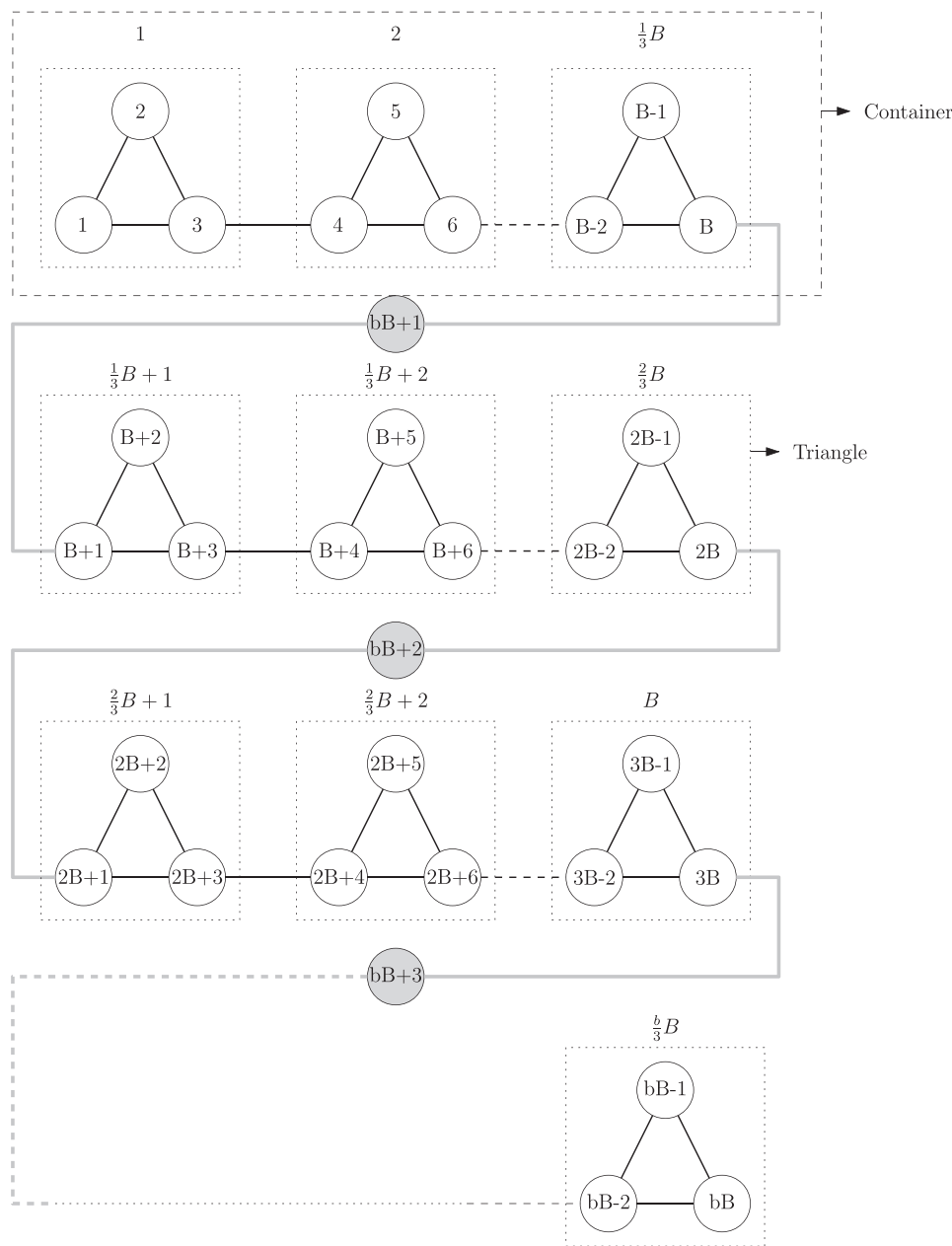


Fig. A.5. Reduction graph.

same party of  $\sum_{q=1}^{3b} (a_q + a_q/3 - 1) = 4bB/3 - 3b$  if and only if the answer to *I* is yes.

First, let us assume that we have a solution with solution value  $4bB/3 - 3b$ . We are first going to argue that each party  $q = 1, \dots, 3b$  can contribute at most  $4a_q/3 - 1$  to the objective value and it does so if and only if it gets assigned a set of  $a_q/3$  triangles which are directly connected to each other by edges in the second subset. Assume that exactly  $k < a_q/3$  full triangles are assigned to party  $q$ . Then, it contributes to the objective value as follows. We have at most  $3k + (k - 1)$  edges from the triangles and this value is achieved if the triangles are chosen such that they are connected by edges of the second subset. Each of the remaining  $a_q - 3k > 0$  seats can have at most two adjacent seats of party  $q$  (otherwise they occupy a triangle). Furthermore, two seats can have only one adjacent seat of party  $q$  (otherwise the seats imply a cycle which is only possible if a triangle is occupied). This gives an upper bounds of  $(2(a_q - 3k - 2) + 2)/2 = a_q - 3k - 1$  for the contribution to the

objective value. Overall, the contribution of party  $q$  cannot exceed  $3k + (k - 1) + (a_q - 3k - 1) = a_q + k - 2 < 4a_q/3 - 2$ .

Since each party  $3b + 1, \dots, 4b - 1$  does not contribute a positive value to the objective function, solution value

$$\sum_{q=1}^{3b} (4a_q/3 - 1) = 4bB/3 - 3b$$

can be achieved only if each party  $q = 1, \dots, 3b$  gets assigned  $a_q/3$  triangles which are directly connected to each other by edges in the second subset. Now we see that each maximum subset of triangles which are directly connected to each other constitutes a container to be packed with parties. The capacity of each container equals  $B$  while party  $q = 1, \dots, 3b$  occupies  $a_q$  units of capacity. Thus, the partition of parties into subsets to be seated in the same maximum subset of triangles constitutes a yes certificate to *I*.

Second, if the answer to *I* is yes we can pack parties into containers such that each party  $q$  occupies exactly  $a_q/3$  triangles

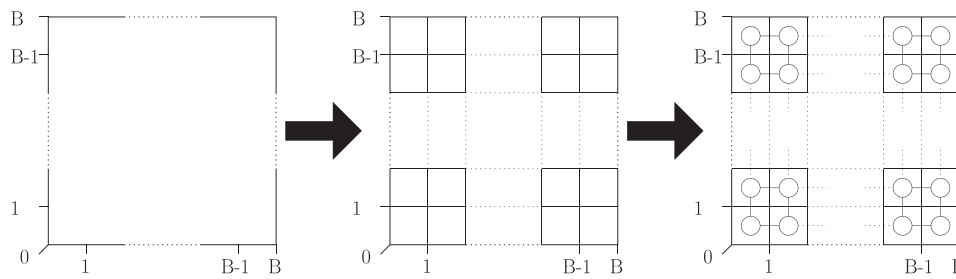


Fig. A.6. Boundary square (left), boundary square with grid overlay (middle) and boundary square with grid graph overlay (right).

which achieves an objective value of  $4bB/3 - 3b$ . This completes the proof.  $\square$

**Corollary 1.** *SAP is strongly NP-hard even for planar graphs with maximum degree of at most 2 and  $w_e = 1$  for each edge  $e \in E$ , if  $r^p = 0$  for each party  $p \in P$ , and if each seat is occupied.*

The reduction can be done in a similar way as in the proof of Theorem 1. We can construct a graph of  $b$  isolated paths of length  $B$  each. We consider only parties  $1, \dots, 3b$  and, by packing parties into paths, can achieve an objective value of  $\sum_{q=1}^{3b} (a_q - 1) = b(B - 3)$  if and only if the answer to  $I$  is yes.

**Theorem 2.** *SAP is strongly NP-hard even for  $G(m, m)$ ,  $m \in \mathbb{N}^+$ , with  $w_e = 1$  for each edge  $e \in E$ , if  $r^p = 0$  for each party  $p \in P$ , and if each seat is occupied.*

The proof is based on a reduction from the Packing Squares in a Square problem (PSS). PSS is known to be strongly NP-complete (see Li & Cheng, 1990 and Leung, Tam, Wong, Young, & Chin, 1990). PSS is defined as follows. Given a boundary square with edge length  $B \in \mathbb{N}^+$  and a set of smaller squares with edge lengths  $S = \{s_1, s_2, \dots, s_b\}$ , where  $s_i \in \mathbb{N}^+$ ,  $\forall i = 1, \dots, b$ . Does there exist an orthogonal packing of the squares in  $S$  in the boundary square  $B$ , such that no squares overlap?

**Proof.** We consider the corresponding decision problem of SAPGG, which asks whether a feasible solution with a target objective value of  $A$  (or more) exists. We will transform an arbitrary instance of PSS to an instance of SAPGG. We assume that  $\sum_{i=1}^b s_i^2 = B^2$  since we can add extra squares of size 1 otherwise.

The boundary square of surface  $B \times B$  in the PSS instance will correspond with a grid graph  $G$  that has  $B \times B$  nodes in the SAPGG instance, that is  $G(B, B)$ . First, we show how the boundary square with side length  $B$  corresponds to a grid graph  $G(B, B)$ . Fig. A.6 shows how the boundary square can be divided in a grid. Using a Cartesian coordinate system with its origin in the left-bottom corner of the boundary square, a vertical grid line goes from  $(x, 0)$  to  $(x, B)$  for each  $x \in \{0, \dots, B\}$  and a horizontal grid line goes from  $(0, x)$  to  $(B, x)$  for each  $x \in \{0, \dots, B\}$ . Next, as Fig. A.6 clarifies, that grid trivially corresponds to a grid graph  $G(B, B)$  where a node is centered in each cell.

We have  $b$  parties and party  $i = 1, \dots, b$  needs to get  $\beta^i = s_i^2$  seats. Finally, let  $A = 2 \sum_{i=1}^b (s_i^2 - s_i)$ . This completes the reduction in pseudo-polynomial time.

We claim that we can reach the target objective value of  $A$  in the SAPGG instance if and only if the answer to the PSS instance is yes.

We remark that a connected sub-graph induced by an assignment of nodes to a party in a grid graph corresponds to a so-called polyomino in the corresponding grid overlay. We present a polyomino definition, that is slightly adapted from the original book on the topic by Golomb (1996).

**Definition 6.** Polyominoes are shapes made by connecting certain numbers of equal-sized squares, each joined together with at least one other square along an edge. Chess players might call this “rookwise connection”; that is, a rook - which can travel either horizontally or vertically in any move, but never diagonally - placed on any square of the polyomino must be able to travel to any other square in that polyomino in a finite number of moves.

Imagine a polyomino of size  $s_i^2$  embedded in an infinite grid graph. Edges that connect nodes in that polyomino, we refer to as *inner edges*. A polyomino has  $e_{\text{inner}}$  inner edges. Edges that connect a node in the polyomino with a node that is not part of the polyomino, we refer to as *cut edges*. A polyomino has  $e_{\text{cut}}$  cut edges. Observe now that  $4s_i^2 = 2e_{\text{inner}} + e_{\text{cut}}$  holds, because every node has a degree equal to 4. By considering each edge incident to each node in the polyomino we account for inner edges twice. Hence, for a polyomino of fixed size minimizing the number of cut edges is equivalent to maximizing the number of inner edges.

Next, we note that the perimeter of a polyomino equals  $e_{\text{cut}}$ . From Harary & Harborth (1976) and Kurz (2008), it follows that given a size  $s_i^2 = \beta^i$ , the unique polyomino of that size that minimizes the perimeter is a square with side length  $s_i$ . Hence, square-shaped polyominoes minimize the number of cut edges, which in turn maximizes the number of inner edges. In our setting, polyominoes are to be positioned in a boundary square with grid graph overlay, instead of an infinite grid graph. However, this does not impact the result that the number of inner edges is maximized if and only if all corresponding polyominoes are square-shaped. In particular, this results in  $A = 2 \sum_{i=1}^b (s_i^2 - s_i)$  inner edges.

We show how a yes certificate for the PSS instance corresponds to a solution of SAPGG which meets target value  $A$ . If a square  $i$  covers a grid cell, then the corresponding node in  $G(B, B)$  is assigned to party  $i$ . Obviously, the implied seating assignment is contiguous and each party  $i$  gets  $\beta^i$  seats. Due to the square shape of subgraphs assigned to parties, the number of inner edges in  $G(B, B)$  equals  $A$ .

Conversely, a yes answer for an instance of SAPGG corresponds to a yes answer for the instance of PSS. Recall, that  $A = 2 \sum_{i=1}^b (s_i^2 - s_i)$  is the number of inner edges obtained if and only if all corresponding polyominoes are square shaped. Hence, subgraphs assigned to parties correspond to squares and because no seat is assigned to more than one party it follows that these squares do not overlap.  $\square$

We give an extra example in Fig. A.7, where there are 6 squares to be packed in a boundary square with side length  $B = 3$ . In this example we have  $A = 4$ . The corresponding SAPGG solution is optimal, because the assignments are all square shaped, minimizing the total number of cut edges, which in turn maximizes the inner edges.

**Theorem 3.** *Deciding whether a contiguous and front-fair seating assignment exists is strongly NP-complete even for connected planar*

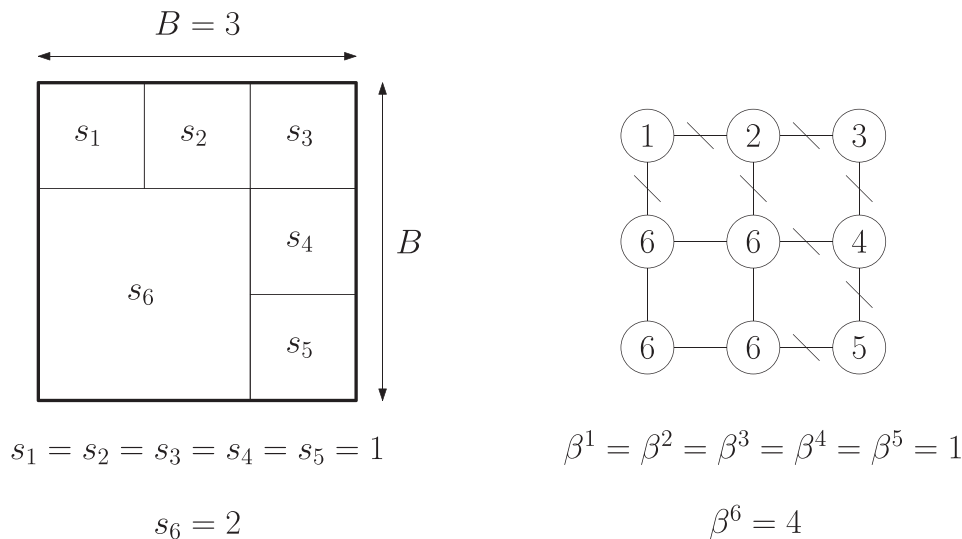


Fig. A.7. A solution of a PSS instance (left) and its corresponding SAPGG solution (right) with  $A = 4$ .

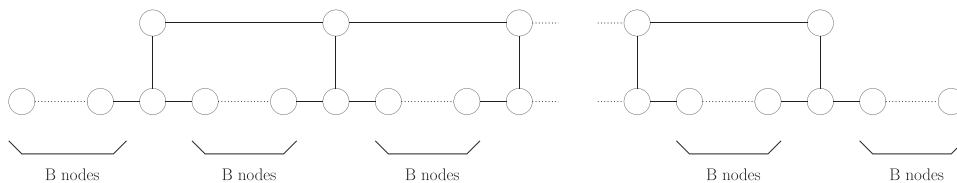


Fig. A.8. Sketch of the graph construction used in the proof of Theorem 3.

graphs with maximum degree of at most 3 and if each seat is occupied.

The proof is again by reduction from 3PP.

**Proof.** Given an instance  $I$  of 3PP we construct an instance  $I'$  of SAP as follows. We construct a graph  $G$  with  $V = \{1, \dots, b(B+2) - 2\}$  and  $F = \{1, \dots, b(B+1) - 1\}$ . The set  $E$  of edges is determined as  $E = E^F \cup E^{-F} \cup E'$  with

- $E^F = \{\{i, i + 1\} \mid i = 1, \dots, b(B+1) - 2\}$ ,
- $E^{-F} = \{\{i, i + 1\} \mid i = b(B+1), \dots, b(B+1) - 3\}$ , and
- $E' = \{\{k(B+1), b(B+1) - 1 + k\} \mid k = 1, \dots, b - 1\}$ .

Note that  $E^F$  constitute a path on nodes in  $F$ ,  $E^{-F}$  constitute a path on nodes in  $V \setminus F$ , and  $E'$  connect each node in  $V \setminus F$  with exactly one node in  $F$ . A sketch of this construction is depicted in Fig. A.8.

We have  $4b - 1$  parties in set  $P = \{1, \dots, 4b - 1\}$ . Each party  $p \in \{1, \dots, 3b\}$  gets  $\beta^p = a_q$  seats. Each party  $p \in \{3b + 1, \dots, 4b - 1\}$  gets  $\beta^p = 2$  seats. For parties  $1, \dots, 3p$  we set  $r^p = \beta^p - 1$ , and for parties  $3b + 1, \dots, 4b - 1$  we set  $r^p = 1$ . This completely specifies instance  $I'$  of SAP.

We claim that for  $I'$  there is a contiguous and front-fair seating assignment if and only if the answer to  $I$  is yes.

Suppose we have a contiguous seating assignment satisfying front row seat requirements. We can see that parties  $3b + 1, \dots, 4b - 1$  occupy seats  $1 \cdot (B + 1), 2 \cdot (B + 1), \dots, (b - 1) \cdot (B + 1)$  since these are the only front row seats connected to a seat in  $V \setminus F$ . Since  $\beta^p = 2$  for each  $p \in \{3b + 1, \dots, 4b - 1\}$ , we can assume that each party  $p \in \{3b + 1, \dots, 4b - 1\}$  gets seats  $(p - 3b) \cdot (B + 1)$  and  $b(B + 1) + p - 4b$  (by renumbering). Note that seats in  $F$  are separated by seats taken by these parties into subsets of  $B$  seats connected by a path each. No pair of seats in different subsets is connected by an edge. Hence, remaining parties  $1, \dots, 3b$  are assigned to these subsets such that the total number of seats

of parties assigned to the same subset does not exceed  $B$ . Since there are only  $bB$  seats in  $F$  available for these parties and their total number of seats is  $bB$ , as well, the total number of seats of parties assigned to the same subset is exactly  $B$ . This constitutes a yes certificate for  $I$ . Finally, given a yes certificate for  $I$ , we can arrange a seating with the structure described above.  $\square$

We point out that the numbers of front row seats for parties in the proof of Theorem 3 might be obtained using the D'Hondt method although it is not a unique outcome. Each party  $p \in \{1, \dots, 4b - 1\}$  gets granted at least  $\beta^p - 1$  front row seats which leaves  $(b(B + 1) - 1) - ((bB - 3b) + (b - 1)) = 3b$  front row seats to be distributed. All parties are tied for an additional front row seat and, thus, depending on the tie breaker, each party  $p \in \{1, \dots, 3b\}$  might get only front row seats while each party  $p \in \{3b + 1, \dots, 4b - 1\}$  might get only one front row seat (and, thus, one seat in  $V \setminus F$ ).

### References

Abels, R., Besselink, N., & Zuidervaart, B. (2017). Wie Krijgen de Gewilde Plekken op de Eerste Rij in de Kamer? Newspaper article. Accessed: 28-04-2020. URL: <https://www.trouw.nl/home/wie-krijgen-de-gewilde-plekken-op-de-eerste-rij-in-de-kamer~a50c3f1b/>[in Dutch].

Adriaanse, M. L., & de Witt Wijnen, P. (2017). Baudet wil in de Kamer niet achter de PVV zitten. Newspaper article. Accessed: 28-04-2020. URL: <https://www.nrc.nl/nieuws/2017/03/16/baudet-wil-niet-in-de-kamerbankjes-achter-de-pvv-zitten-want-forum-is-niet-rechts-7415533-a1550735>[in Dutch].

Benati, S., Puerto, J., & Rodríguez-Chía, A. M. (2017). Clustering data that are graph connected. *European Journal of Operational Research*, 261(1), 43–53.

Cormen, T. H., Leiserson, C. E., Rivest, R. L., & Stein, C. (2009). *Introduction to algorithms* (3rd). Cambridge: MIT press.

Darlay, J., Brauner, N., & Moncel, J. (2012). Dense and sparse graph partition. *Discrete Applied Mathematics*, 160(16), 2389–2396.

Désir, A. (2016). Apartheid au conseil régional : Personne ne veut s'asseoir à côté des élus FN. Newspaper article. Accessed: 28-04-2020. URL: <http://fr.novopress.info/197604/apartheid-au-conseil-regional-personne-veut-sasseoir-cote-elus-fn>[in French].

Dyer, M., & Frieze, A. (1985). On the complexity of partitioning graphs into connected subgraphs. *Discrete Applied Mathematics*, 10(2), 139–153.



- Gallagher, M. (1991). Proportionality, disproportionality and electoral systems. *Electoral Studies*, 10(1), 33–51.
- Garey, M., Johnson, D., & Stockmeyer, L. (1976). Some simplified NP-complete graph problems. *Theoretical Computer Science*, 1(3), 237–267.
- Garey, M. R., & Johnson, D. S. (1979). *Computers and intractability: A guide to the theory of NP-completeness*. W. H. Freeman and Company.
- Golomb, S. W. (1996). *Polyominoes : Puzzles, patterns, problems, and packings*. Princeton University Press.
- Grimm, V., Kleinert, T., Liers, F., Schmidt, M., & Zöttl, G. (2019). Optimal price zones of electricity markets: A mixed-integer multilevel model and global solution approaches. *Optimization Methods and Software*, 34(2), 406–436.
- Haglin, D., & Venkatesan, S. (1991). Approximation and intractability results for the maximum cut problem and its variants. *IEEE Transactions on Computers*, 40(1), 110–113.
- Harary, F., & Harborth, H. (1976). Extremal animals. *Journal of Combinatorics, Information & System Sciences*, 1(1), 1–8.
- Harmon, N., Fisman, R., & Kamenica, E. (2019). Peer effects in legislative voting. *American Economic Journal: Applied Economics*, 11(4), 156–180.
- Hojny, C., Joormann, I., Lüthen, H., & Schmidt, M. (2021). Mixed-integer programming techniques for the connected max-k-cut problem. *Mathematical Programming Computation*, 13, 75–132.
- Itai, A., Papadimitriou, C., & Szwarcfiter, J. (1982). Hamilton paths in grid graphs. *SIAM Journal on Computing*, 11(4), 676–686.
- Karp, R. M. (1972). *Reducibility among combinatorial problems* (pp. 85–103). Boston, MA: Springer US.
- King, D. M., Jacobson, S. H., & Sewell, E. C. (2015). Efficient geo-graph contiguity and hole algorithms for geographic zoning and dynamic plane graph partitioning. *Mathematical Programming*, 149(1), 425–457.
- King, D. M., Jacobson, S. H., Sewell, E. C., & Cho, W. K. T. (2012). Geo-graphs: An efficient model for enforcing contiguity and hole constraints in planar graph partitioning. *Operations Research*, 60(5), 1213–1228.
- Kok, L. (2017). Stoelendans om juiste plek in Tweede Kamer. Newspaper article. Accessed: 28-04-2020. URL: [https://www.ad.nl/politiek/stoelendans-om-juiste-plek-in-tweede-kamer~ae5fac63/\[in Dutch\]](https://www.ad.nl/politiek/stoelendans-om-juiste-plek-in-tweede-kamer~ae5fac63/[in Dutch]).
- Kurz, S. (2008). Counting polyominoes with minimum perimeter. *Ars Combinatoria*, 88, 161–174.
- Cohen de Lara, M., & Mulder, D. (2016). *Parliament*. XML. Amsterdam.
- Leung, J. Y.-T., Tam, T. W., Wong, C., Young, G. H., & Chin, F. Y. (1990). Packing squares into a square. *Journal of Parallel and Distributed Computing*, 10(3), 271–275.
- Li, K., & Cheng, K. H. (1990). Static job scheduling in partitionable mesh connected systems. *Journal of Parallel and Distributed Computing*, 10(2), 152–159.
- Mesdag, M. (2017). DENK huilt om plek in kamer. Newspaper article. Accessed: 28-04-2020. URL: [https://www.powned.tv/artikel/denk-boos-om-plek-in-kamer \[in Dutch\]](https://www.powned.tv/artikel/denk-boos-om-plek-in-kamer [in Dutch]).
- Méndez Díaz, I., & Zabala, P. (2001). A polyhedral approach for graph coloring. *Electronic Notes in Discrete Mathematics*, 7, 178–181.
- Oosten, M., Rutten, J., & Spieksma, F. (2007). Disconnecting graphs by removing vertices: A polyhedral approach. *Statistica Neerlandica*, 61(1), 35–60.
- Pullman, N. J. (1983). Clique coverings of graphs — A survey. In L. R. A. Casse (Ed.), *Combinatorial mathematics x* (pp. 72–85). Berlin, Heidelberg: Springer Berlin Heidelberg.
- Saia, A. (2018). Random interactions in the chamber: Legislators' behavior and political distance. *Journal of Public Economics*, 164, 225–240.
- Shirabe, T. (2009). Districting modeling with exact contiguity constraints. *Environment and Planning B: Planning and Design*, 36(6), 1053–1066.
- Thurau, J. (2017). Far-right, right? where will the AfD sit in the Bundestag? Newspaper article. Accessed: 28-04-2020. URL: [http://www.dw.com/en/far-right-right-where-will-the-afd-sit-in-the-bundestag/a-40807847\[in Dutch\]](http://www.dw.com/en/far-right-right-where-will-the-afd-sit-in-the-bundestag/a-40807847[in Dutch]).
- Wang, Y., Buchanan, A., & Butenko, S. (2017). On imposing connectivity constraints in integer programs. *Mathematical Programming*, 166(1–2), 241–271.
- de Witt Wijnen, P. (2017). Stoelendans in nieuwe Tweede Kamer blijft stuiten op bezwaren Denk en Forum. Newspaper article. Accessed: 28-04-2020. URL: [https://www.nrc.nl/nieuws/2017/03/22/stoelendans-in-nieuwe-tweede-kamer-blijft-stuiten-op-bezwaren-denk-en-forum-7516272-a1551485\[in Dutch\]](https://www.nrc.nl/nieuws/2017/03/22/stoelendans-in-nieuwe-tweede-kamer-blijft-stuiten-op-bezwaren-denk-en-forum-7516272-a1551485[in Dutch]).