



Congress seat allocation using mathematical optimization

Roland Oliver Hales¹ · Sergio García¹

Received: 23 December 2017 / Accepted: 15 April 2019 / Published online: 29 April 2019
© The Author(s) 2019

Abstract

After the 2015 Spanish general election a row erupted over the allocation of physical seats in the Congress of Deputies, with certain parties left feeling they possessed an inferior selection of seats compared to other parties. Using this as motivation, this paper considers how mathematical optimization can be used to generate seating plans for political chambers, an application that has not been considered before. As well as being in some way ‘fair’ to all parties, the seating plan should ensure that each block of seats is well-defined and compact. Two optimization models are formulated and, due to their complexity, heuristic methods are developed to find ‘good’ solutions. Analysis shows that the heuristics are able to produce visually appealing seating plans for basic cases, but problems can occur when there are additional requirements to be satisfied.

Keywords Seat allocation · Mixed integer programming

Mathematics Subject Classification 90C11 Mixed integer programming · 90C27 Combinatorial optimization · 90C90 Applications of mathematical programming

1 Introduction

In recent years there have been arguments regarding the seating plans in the Spanish Congress of Deputies, with some parties believing they have been allocated inferior seats compared to other parties. Indeed, the seating plan following the 2015 general election, shown in the top left of Fig. 1, prompted Podemos spokesperson Íñigo Errejón to claim that, “They are sending the representatives of five million voters to the nosebleed section, separated from the rest.” With this in mind, the aim of this paper

✉ Roland Oliver Hales
roland.hales91@gmail.com

Sergio García
sergio.garcia-quiles@ed.ac.uk

¹ School of Mathematics, University of Edinburgh, Edinburgh, UK

is to investigate how mathematical optimization can assist with decisions regarding how representatives should be seated in a congress chamber and, in particular, to propose methods for generating seating plans that all parties find acceptable. The nature of the problem is inherently subjective, as there is no definitive way of determining whether one seating plan is better than another. However, we propose the following basic criteria that a good seating plan should satisfy:

1. Members of the same party should be seated close to one another, clustered together in a compact way.
2. The plan should abide by any pre-existing rules or traditions regarding where the different parties are seated.
3. No party should be allocated a block of seats that is significantly better or worse than those of other parties.

In particular, we consider the problem as that of optimizing criterion 1, while satisfying constraints that enforce criteria 2 and 3. It should be noted that, except where stated otherwise, we suppose that the number of available seats is exactly equal to the number of representatives.

The aim is to develop a flexible approach that can be applied to any congress, parliament or other political chamber in which each representative is assigned one seat based on the party they represent. This is complicated by the fact that the seats in each congress chamber are arranged differently, with different rules and traditions that need to be adhered to regarding their allocation. Figure 1 shows four examples of seating plans. The top-left diagram¹ shows the Spanish Congress of Deputies, which will be discussed in detail in the case study in Sect. 8. The top-right diagram² shows a seating plan the European Parliament in Strasbourg, which has a semi-circular or fan shape, and it can be seen that each political group is assigned a wedge-shaped selection of seats. The bottom-right diagram³ shows a seating plan for the New Zealand House of Representatives, where the seats are arranged in a horseshoe shape, and we can note that every party is assigned at least one front row seat. The bottom-left diagram⁴ shows the Canadian House of Commons, which consists of two banks of seats separated by an aisle, with the governing party predominantly seated on one side and opposition parties on the other side.

To model the problem we make the assumption that each seat can be represented by a pair of coordinates on the plane, and that these coordinates are known beforehand. Furthermore, we make the assumption that the seats can form a connected graph, with each node representing a seat and edges connecting any two seats that are *adjacent*. This allows us to calculate the shortest path distances between any two seats, which provide a more intuitive measure than, say, the Euclidean distance. A potential drawback of this is that it is not always clear which pairs of seats should be considered adjacent, which in turn means there are many possible graphs, each of which will result in different shortest path distances. Moreover, if the seats are arranged in two banks separated

¹ https://elpais.com/elpais/2016/01/26/inenglish/1453821405_849690.html.

² <http://www.europarl.europa.eu/plenary/en/hemicycle.html>.

³ https://en.wikipedia.org/wiki/New_Zealand_general_election,_2014.

⁴ https://en.wikipedia.org/wiki/Parliament_of_Canada.

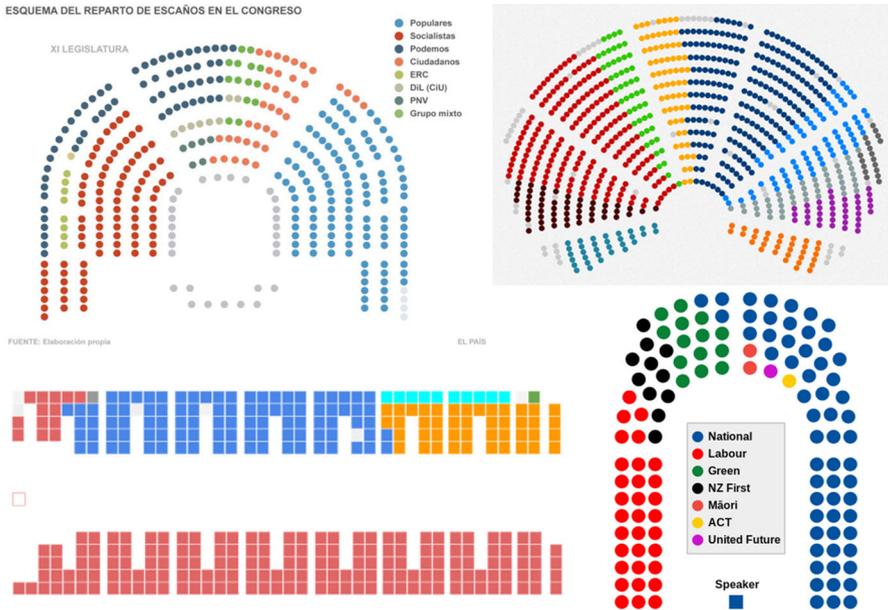


Fig. 1 Examples of seating plans. Top left: Spanish Congress of Deputies; top right: European Parliament in Strasbourg; bottom left: Canadian House of Commons; bottom right: New Zealand House of Representatives

by an aisle then this would result in two disconnected graphs. Despite this issue, we still felt that shortest path distances were the best option, particularly since Euclidean distances can give counter-intuitive results if the seats are arranged in a horseshoe shape, which is a very common arrangement for congress and parliament chambers. Moreover, viewing the seats as a connected graph allows us to view the problem as a type of graph partitioning, which we briefly explore in Sect. 4.

The paper is organized as follows. In the literature review we compare the problem with other well-studied problems. We then develop a facility location model (FLM) and a minimum k -partitioning model (MPM). Since these models are both hard to solve for large instances, in Sect. 5 we develop a location-allocation heuristic (LAH) for the FLM, and in Sect. 6 we develop a geometric cutting heuristic (GCH) for the MPM that uses straight lines to partition the seats into blocks of the correct sizes. In Sect. 7, a computational study is conducted to evaluate and compare the performances of the heuristics, both numerically and visually. The paper concludes with a case study showing the results of applying the heuristics to the Spanish Congress of Deputies.

2 Literature review

Although there exist papers in the literature that apply optimization methods to determine how many seats should be awarded to each party, such as Serafini (2012), to the best of our knowledge this paper is the first application of optimization methods to the allocation of physical seats within the congress. Several seat allocation problems have

been studied previously, see for instance García et al. (2014), but they are concerned with assigning individual people to seats, whereas in our problem the aim is to assign collections of seats to political parties. This fundamental difference means that, in practice, the problems have little in common.

In fact, the problem has more similarities to districting problems, in which the aim is to cluster together small areas of land to form larger districts that satisfy certain desirable properties. These properties typically include that each district be connected and compact, while containing populations of approximately equal sizes. The compactness property is often included to prevent so-called gerrymandering, the practice of manipulating district borders for political gain. Horn et al. (1993) and Niemi et al. (1990) compare different approaches to achieving compactness in districting problems. For an overview of approaches to districting problems see, for example, Ricca et al. (2011).

The similarities with the congress seating problem are clear, but there are also several key differences. Firstly, in the congress seat allocation problem there are no populations as such, so the population equality condition is not applicable, but instead there is the condition that the number of seats in each block must correspond to the number of seats won by each party. Secondly, the seats in a congress do not necessarily fit together like a jigsaw in the same way that geographical areas of land do, and so some of the techniques used for districting problems may not be applicable; in particular, many of the ways of measuring compactness cannot be applied.

Districting problems are often modelled as facility location problems, also known as plant or warehouse location problems, for example see Hess et al. (1965) and Hojati (1996). The goal of these problems is to choose several locations from a set of candidate locations at which to build facilities. Customers are then allocated to the facilities, with the aim being to minimize the set-up costs of building the facilities together with the transportation costs of delivering goods between the facilities and their customers. The congress seat allocation problem can be thought of as a facility location in the following way. Let each seat be a candidate location for a facility and let each seat also represent a single customer, then specify that we must ‘build’ exactly one facility for each party so that allocating customers to facilities corresponds to allocating seats to parties. It will also be necessary to ensure that each party is allocated the correct number of seats. There are no set-up costs to speak of, but the transportation costs can be thought of as being the sum of distances between each seat and its corresponding facility. Since the goal is to minimize the sum of these distances, in an optimal solution each seat belonging to a given party should be close to that party’s ‘facility’, which should in turn encourage each block of seats to be compact. In Sect. 3 the problem is modelled as a facility location problem, and in Sect. 5 a location-allocation heuristic is developed, similar to that proposed by Hess et al. (1965) and Kalcsics et al. (2005) for political districting problems.

Finally, if the seats are considered as nodes in a graph with adjacent seats connected by edges, then the problem can be viewed as a type of graph partitioning. In particular, a partition of the nodes that minimizes the number of ‘cut edges’ will equate to a partition that minimizes the number of adjacent seats belonging to distinct parties, which seems like a good objective to encourage a compact seating plan. General minimum k -partitioning problems are well-studied and known to be NP-hard, see

for instance Chopra and Rao (1993). Kernighan and Lin (1970) developed a popular heuristic for the problem, and semi-definite programming relaxations can be used to find lower bounds, see for example Wolkowicz and Zhao (1999). In Sect. 4 the problem is formulated as a minimum k -partitioning problem, and in Sect. 6 a geometric cutting heuristic is developed that uses straight-line cuts to partition the graph while cutting as few edges as possible.

3 Facility location model

We first formulate the problem as a facility location model (FLM), which encourages compactness by minimizing the sum of shortest path distances from each seat to their party’s ‘central seat’. To use shortest path distances it is necessary to consider the seats as nodes in a graph with adjacent seats connected by edges. In Sect. 7, we show how such a graph can be constructed. However, it would also be possible to use different distances, such as Euclidean. First we outline the basic model, which simply aims to partition the seats into compact blocks, and then we consider several additional constraints that could be included to promote a sense of fairness or to ensure the solutions adhere to certain rules or traditions. The following notations are used:

- I = set of seats.
- J = set of parties.
- K_j = number of seats to be allocated to party j .
- d_{ik} = shortest path distance between seats i and k .
- $y_k^j = \begin{cases} 1 & \text{if seat } k \text{ is the central seat for party } j, \\ 0 & \text{otherwise.} \end{cases}$
- $x_{ik} = \begin{cases} 1 & \text{if seat } i \text{ is allocated to a central seat } k, \\ 0 & \text{otherwise.} \end{cases}$

The basic model is:

$$\min \sum_{i \in I} \sum_{k \in I} d_{ik} x_{ik} \tag{1}$$

$$\text{s.t. } \sum_{k \in I} x_{ik} = 1 \quad \forall i \in I, \tag{2}$$

$$\sum_{i \in I} x_{ik} = \sum_{j \in J} K_j y_k^j \quad \forall k \in I, \tag{3}$$

$$\sum_{j \in J} y_k^j \leq 1 \quad \forall k \in I, \tag{4}$$

$$\sum_{k \in I} y_k^j = 1 \quad \forall j \in J, \tag{5}$$

$$x_{ik}, y_k^j \in \{0, 1\} \quad \forall i, k \in I, \forall j \in J. \tag{6}$$

The constraints can be interpreted as follows:

- (2) All seats are assigned to a central seat.
- (3) If seat k is the central seat for party j then K_j seats are allocated to it. If seat k is not a central seat then no seats are allocated to it.
- (4) Each seat k can be a central seat for at most one party.
- (5) Exactly one central seat should be chosen for each party $j \in J$.
- (6) All variables are binary.

We now consider several additional properties that might be desirable in a seating plan, either to adhere to tradition or to encourage a degree of fairness, and we show how these properties could be modelled as constraints that can be added to the basic model. In reality, which, if any, of the additional constraints to include will depend upon the specific application.

- Certain parties occupy certain areas of the chamber: let the coordinates of seat i be (p_i, q_i) , and suppose we require that party j be seated on the left-hand side of the congress chamber, or, equivalently, that they must not be allocated any seats on the right-hand side. Suppose that the horizontal coordinates are on a scale from 0 to 100, with a seat i considered to be on the right-hand side if $p_i \geq 50$. Constraints (7) then ensure that no seat on the right-hand side of the congress can be allocated to party j .

$$x_{ik} \leq 1 - y_k^j \quad \forall k \in I, \forall i \in I : p_i \geq 50. \tag{7}$$

The right-hand side of (7) is only equal to zero for the seat k that is the central seat for party j , in which case seat i cannot be assigned to that central seat if $p_i \geq 50$.

- No parties seated too far back: let R_i be the row number of seat i , with the front row being row 1, the next being row 2, and so on. Let R be a constant. In constraints (8), if seat k is not a central seat then the left-hand side is zero, and if k is a central seat then the average row number of the seats assigned to k must be less than or equal to R . Therefore these constraints can be used to ensure no party is seated too far back on average.

$$\sum_{i \in I} (R - R_i)x_{ik} \geq 0 \quad \forall k \in I. \tag{8}$$

- Larger parties have more front row seats: let F be a constant and suppose we want to impose the condition that if a party has more than or equal to F seats then they should have at least one seat in the front row. Moreover, we want to impose that if a party has more than or equal to $2F$ seats then they should have at least two seats in the front row, and so on. Let $f_i = 1$ if seat i is in the front row, and zero otherwise. We model this condition as follows:

$$\sum_{i \in I} f_i x_{ik} \geq \sum_{j \in J} \left\lfloor \frac{K_j}{F} \right\rfloor y_k^j \quad \forall k \in I. \tag{9}$$

For every seat k that is not a central seat the right-hand side of (9) is equal to zero and so no conditions are imposed on the left-hand side. However, if seat k is the central seat corresponding to party j , then the left-hand side of (9) counts the number of front row seats allocated to party j and the right hand imposes a minimum value on this count. Note that for small parties with $K_j < F$ the right-hand side is equal to zero and so they are not required to have any front row seats.

- Every representative must be sat next to another representative of the same party: let n_{il} equal one if seats i and l are next to one another and zero otherwise. Constraints (10) mean that a seat i cannot be allocated to central seat k unless there is another seat next to seat i that is also allocated to central seat k :

$$x_{ik} \leq \sum_{l \in I} n_{il} x_{lk} \quad \forall i, k \in I. \tag{10}$$

4 Minimum k -partitioning model

Let us now consider a graphical representation of the congress in which each node represents a seat and adjacent seats are joined by an edge. We formulate the problem as a model that aims to minimize the number of edges connecting two seats belonging to different parties. Note that this is a very different objective than that of the facility location model, and hence the two models are not addressing the same mathematical problem. This model is a variant of the minimum k -partitioning problem that aims to partition the nodes in a graph into k non-empty subsets, such that the number of edges connecting distinct subsets is minimal. The following notations are used:

I = set of seats.

J = set of parties.

$$x_{ij} = \begin{cases} 1 & \text{if seat } i \text{ is assigned to party } j, \\ 0 & \text{otherwise.} \end{cases}$$

K_j = number of seats to be allocated to party j .

$$a_{ik} = \begin{cases} 1 & \text{if seats } i \text{ and } k \text{ are adjacent and therefore connected by an edge,} \\ 0 & \text{otherwise.} \end{cases}$$

The basic model is:

$$\min \sum_{j \in J} \sum_{i \in I} \sum_{\substack{k \in I: \\ k \leq i-1}} a_{ik} x_{ij} (1 - x_{kj}) \tag{11}$$

$$\text{s.t. } \sum_{i \in I} x_{ij} = K_j \quad \forall j \in J, \tag{12}$$

$$\sum_{j \in J} x_{ij} = 1 \quad \forall i \in I, \tag{13}$$

$$x_{ij} \in \{0, 1\} \quad \forall i \in I, \forall j \in J. \tag{14}$$

As can be seen, this is a quadratic assignment problem. Constraints (12) ensure that there is a total of K_j seats allocated to party j , and constraints (13) ensure that every seat i is allocated to one and only one party. The additional constraints discussed for the FLM in the previous section can easily be modified to be applied to this model.

Finally, note that the model does not explicitly aim to minimize the distances between seats belonging to the same party, although this is likely to be encouraged indirectly by minimizing the number of edges connecting seats belonging to different parties. However, if it is found that this model generates solutions where the seats belonging to a party are very spread out, it could be necessary to include a distance component to the model. This could either be a constraint that limits the distance between seats belonging to the same party, or else an additional quadratic component in the objective function, such as:

$$\min \sum_{j \in J} \sum_{i \in I} \sum_{\substack{k \in I: \\ k \leq i-1}} a_{ik} x_{ij} (1 - x_{kj}) + \sum_{j \in J} \sum_{i \in I} \sum_{\substack{k \in I: \\ k \leq i-1}} d_{ik} x_{ij} x_{kj}. \tag{15}$$

5 Location-allocation heuristic

A simple approach for finding good solutions to Facility Location problems is to separate the processes of choosing where the locations are built (the location phase) and choosing how to allocate the customers to these locations (the allocation phase). Indeed, this was the approach adopted by Hess et al. (1965), in what is considered as the earliest Operational Research paper on political districting (Ricca et al. 2011).

Although this heuristic was developed for Facility Location models like the one we considered in Sect. 3, our overall purpose is to produce good seating plans rather than solving any particular model, and so in the computational study we will also examine how well the solutions produced by the heuristic perform for the minimum k -partitioning model from Sect. 4.

Suppose a set of central seats $\{c_j : j \in J\}$ has already been chosen. That is, we have fixed the variables y_k^j from the Facility Location model in Sect. 3 so that $y_k^j = c_j$. The allocation problem is:

$$\begin{aligned} \min \quad & \sum_{i \in I} \sum_{j \in J} d_{ic_j} x_{ij} \\ \text{s.t.} \quad & \sum_{j \in J} x_{ij} = 1 \quad \forall i \in I, \\ & \sum_{i \in I} x_{ij} = K_j \quad \forall j \in J, \\ & x_{ik} \in \{0, 1\} \quad \forall i \in I, \forall j \in J, \\ & \text{Additional constraints.} \end{aligned} \tag{16}$$

The additional constraints could be any extra conditions we want to impose on the solutions generated by the heuristic, similar to those described in (7)–(10). Note that unlike the FLM, there is no constraint preventing one seat being the central seat for more than one party. This allocation problem can be solved very quickly by commercial solvers for all sizes of problem that are relevant to our application. After finding an optimal solution to the allocation problem, the next step is to choose a new set of central seats. This is done by fixing the variables x_{ij} and finding the central seats $\{c_j : j \in J\}$ that minimize the objective of (16).

By iteratively repeating the allocation and location phases, the objective value is necessarily non-increasing from one iteration to the next. The algorithm is terminated when the new set of central seats is identical to the previous set, which means that no further reduction of the objective value will be achieved. The full location-allocation heuristic (LAH) is shown in Algorithm 1.

Algorithm 1 Location-allocation heuristic

- Step 1:** Randomly designate one initial central seat, $c_j \in I$ for each party $j \in J$.
- Step 2:** Solve the allocation problem (16) to obtain the solution $\{x_{ij} : i \in I, j \in J\}$.
- Step 3:** Calculate the new central seats, given by

$$\tilde{c}_j = \arg \min_{k=1, \dots, n} \sum_{i \in I} d_{ik} x_{ij}.$$

- Step 4:** If $\tilde{c}_j = c_j$ for all $j \in J$ then output the solution and stop. Else, set $c_j = \tilde{c}_j$ for all $j \in J$ and return to step 2.
-

In its current form the objective function of the allocation problem minimizes the sum of total distances from seats to their corresponding central seat. Therefore, parties with more seats will contribute more to the objective function, and so in a sense this prioritizes compact blocks for the larger parties over the smaller parties. This effect can be mitigated by scaling the objective function. By dividing each term by K_j we obtain the sum of *average* distances from each central seat to the seats assigned to it. However, the average distance will still be larger for larger parties, and so we can scale further by dividing by $\sqrt{K_j}$ in an attempt to counteract this effect. The *scaled objective function* is therefore given by

$$\min \sum_{i \in I} \sum_{j \in J} \frac{d_{ic_j}}{K_j \sqrt{K_j}} x_{ij}. \tag{17}$$

Figure 2 shows two example solutions generated by LAH. The first was produced by the original, unscaled objective function, while the second was produced by the scaled version. In the first solution, the smaller parties are all located around the edge of the chamber, whereas in the second solution they tend to be embedded within the larger parties. Which of the two solutions is best is ultimately a matter of opinion. On the one hand, the second solution seems less biased against the smaller parties because they are not forced to the edges of the chamber, but on the other hand it does not seem

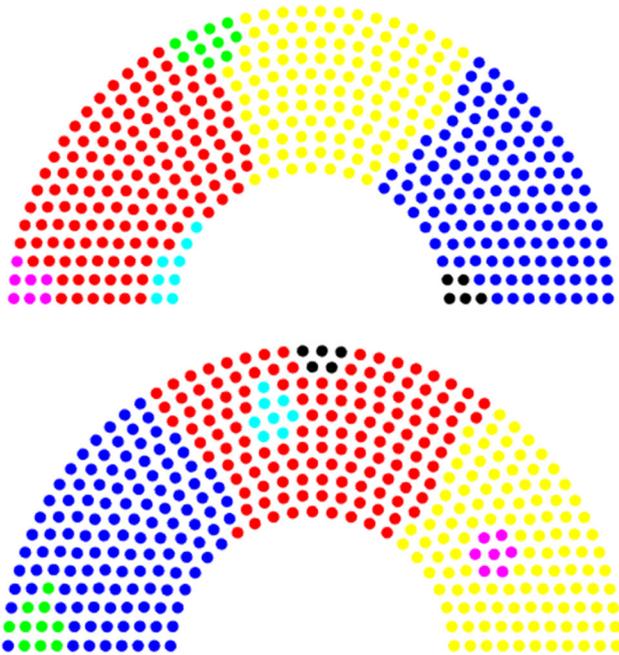


Fig. 2 Solutions from the unscaled (top) and scaled (bottom) LAH

ideal for the larger parties to be ‘punctured’ by the smaller parties. For now, we put the scaled version to one side and continue using the unscaled version, although it is worth bearing in mind the potential problems that this could cause.

6 Geometric cutting heuristic

In this section, we outline a geometric cutting heuristic (GCH) inspired by the computational geometry heuristic for political districting proposed by Kalcsics et al. (2005). The heuristic requires that the seats be considered as a connected graph, with each seat assigned a pair of coordinates, and adjacent seats connected by edges. The heuristic then proceeds as follows:

- The parties are randomly split into two groups. Suppose the parties in the first group have a total of N_1 seats between them, and the parties in the second group have N_2 seats.
- An algorithm finds the straight line that partitions the seats into one group of N_1 seats and one group of N_2 seats, while cutting the fewest possible edges in the graph. Performing this cut separates the graph into two subgraphs, which are then considered separately.
- If a subgraph contains the seats for only one party, then those seats are assigned to that party. Otherwise, the process is repeated for the subgraph. By continuing in this way, eventually the original graph will be broken down into one subgraph for each party, each containing the correct number of seats.

Due to the line direction being chosen to cut the fewest possible edges, the process can be considered primarily as a heuristic for the minimum k -partition model from Sect. 4, however in the computational study we will also examine how well it performs as a heuristic for the Facility Location model from Sect. 3.

Figure 3 shows an example of how the geometric cutting heuristic produces a solution for a congress with 100 seats. Suppose parties 1, 2, 3 and 4 have 47, 32, 17 and 4 seats, respectively. Firstly, the parties are randomly split into two groups, in this case parties 2 and 3 form one group with a total of 49 seats, and parties 1 and 4 form the other group with a total of 51 seats. The seats are partitioned into two groups of sizes 49 and 51 by the straight line that cuts the fewest possible edges. This produces two subgraphs, each containing the seats for two parties. Therefore, the process is repeated for each subgraph separately, dividing each into two further subgraphs. This results in four subgraphs, each corresponding to exactly one party. Therefore, the seats in each subgraph are assigned to the corresponding party, producing the solution shown at the bottom of Fig. 3.

The detailed explanation of the heuristic is divided into two parts. The Bisection Algorithm (Algorithm 2) explains the process used to perform each division of the graph into two subgraphs. Figure 4 shows visually how the bisection is performed. The Geometric Cutting Algorithm (Algorithm 3) then explains how the Bisection Algorithm is repeated iteratively until the original graph is divided into one subgraph for each party.

The key differences between this heuristic and that presented by Kalcsics et al. (2005) are as follows:

- The heuristic presented by Kalcsics et al. aims to partition the space such that each district achieves approximately equal values of a given performance measure (e.g. population size). In our problem, the blocks need to be different sizes determined by the numbers of seats won by each party. To achieve this, we randomly split the values $\{K_j\}$ into two non-empty sets, then consider cuts that partition the seats into two blocks of sizes given by the total number of seats in each set. After each cut has been performed, we consider each block of seats separately and further subdivide them in the same way.
- Compactness is encouraged by Kalcsics et al. (2005) in two different ways, neither of which are adopted by the heuristic in this paper. Instead, we encourage compactness by choosing the straight line that minimizes the number of cut edges in the adjacency graph.

The parameter d , which determines the number of line directions to try on each iteration, needs to be fixed beforehand. Larger values of d provide the possibility of generating better solutions, at the expense of a longer running time. Experiments suggest that $d = 32$ works well, with larger values resulting in little or no improvement, and this is the value we use in the computational study.

By only allowing the seats to be partitioned by straight-line cuts, we dramatically reduce the number of solutions that can be generated by the heuristic. However, the solutions that it does produce are likely to be visually appealing because the boundaries between parties will not be too meandering or ‘wiggly’.

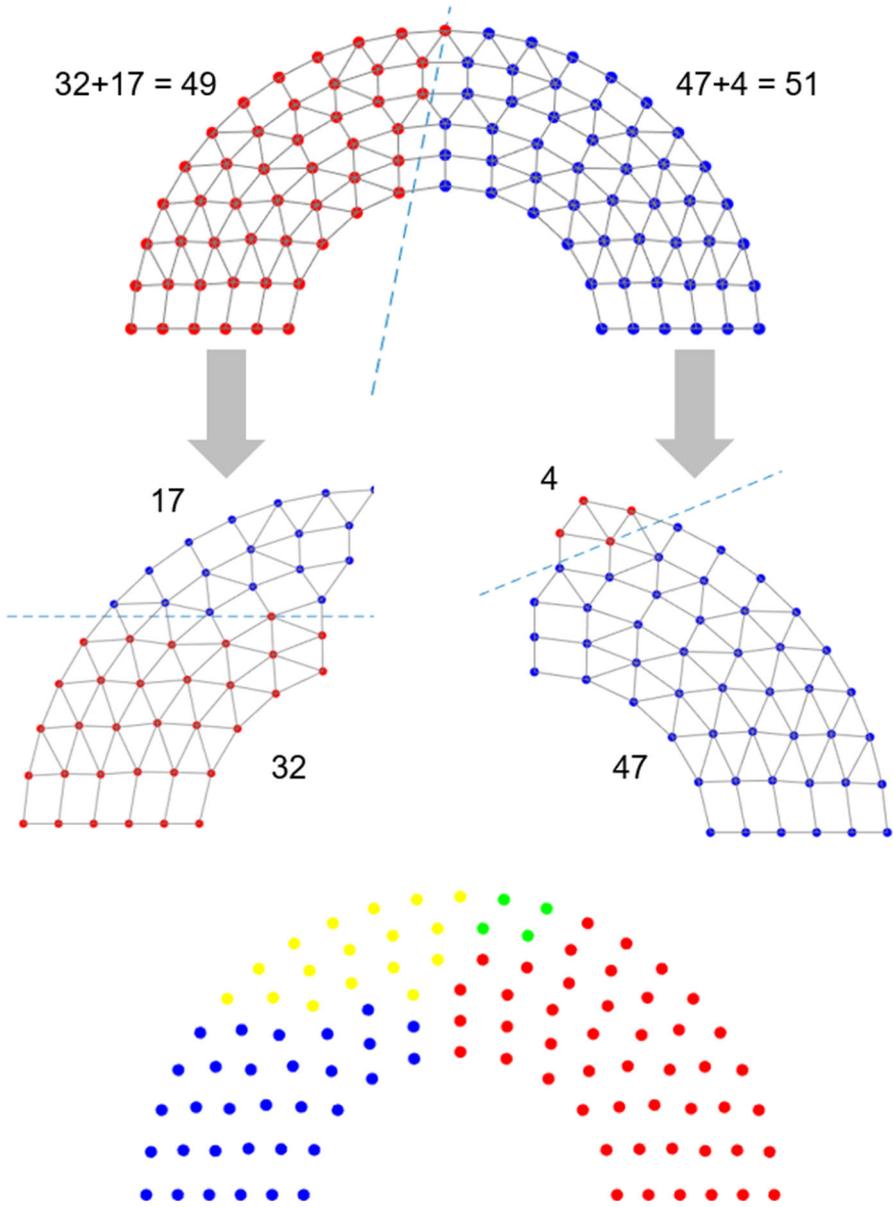


Fig. 3 A visual representation of how the geometric cutting heuristic produces solutions for a congress with 100 seats and seat breakdown $K = \{47, 32, 17, 4\}$

A key drawback of the heuristic is that it could be difficult to modify the algorithm to produce solutions that satisfy additional conditions, such as those discussed in Sect. 3. However, this is not to say it is not possible; for example, we could prohibit any cuts that would result in solutions that violate particular conditions.

Algorithm 2 Bisection Algorithm

Step 1: Start with set of seats I , a set of parties J , and the number of representatives in each party $\{K_j : j \in J\}$. Suppose each seat $i \in I$ has coordinates (x_i, y_i) , and that a graph is formed by connecting adjacent seats with edges. Use a random procedure to separate the parties J into two groups J_1 and J_2 . Let $N_1 = \sum_{j \in J_1} K_j$ and $N_2 = \sum_{j \in J_2} K_j$.

Step 2: Select a number d of line directions to try for the bisection, and for each $k = 1, \dots, d$:

- Let $\theta_k := \left(\frac{k-1}{d}\right) 2\pi$. Then, for all seats $i \in I$, let

$$p_{ik} := x_i \cos \theta_k - y_i \sin \theta_k$$

so that p_{ik} is the x -coordinate of seat i rotated anticlockwise by an angle of θ_k about the origin.

- Sort the values p_{ik} in non-decreasing order and then find a value τ_k such that

$$\begin{aligned} p_{ik} < \tau_k & \quad \text{for the first } N_1 \text{ elements and} \\ p_{ik} > \tau_k & \quad \text{for the last } N_2 \text{ elements.} \end{aligned}$$

If the N_1 th and $N_1 + 1$ th elements are equal, meaning no τ_k can be found, then continue to the next value of k . Otherwise, let C_k be the number of edges connecting seats in the set $\{i \in I : p_{ik} < \tau_k\}$ with seats in the set $\{i \in I : p_{ik} > \tau_k\}$.

Step 3: If no τ_k was found for any $k = 1, \dots, d$ then return to Step 2 and increase the number of directions d . Otherwise, let

$$k^* = \arg \min_{k=1, \dots, d} C_k$$

so that k^* denotes the bisection line that cuts the fewest edges of the graph. Perform the bisection by separating the seats into the sets $\{i \in I : p_{ik^*} < \tau_{k^*}\}$ and $\{i \in I : p_{ik^*} > \tau_{k^*}\}$.

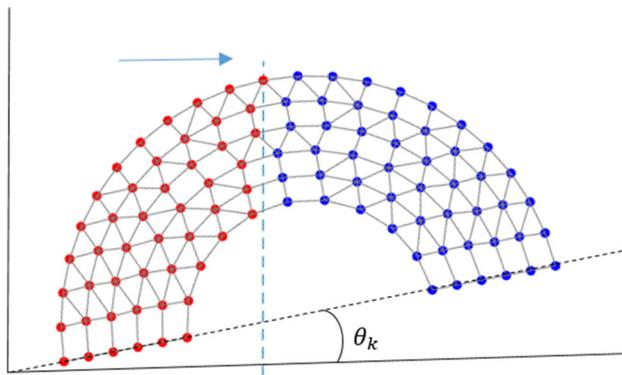


Fig. 4 To find the line direction that cuts the fewest edges, for each $k = 1 \dots d$ we rotate the seats by $\theta_k = 2\pi(k - 1)/d$ degrees and count the number of edges cut by a vertical line

7 Computational study

This computational study examines the performance of the location-allocation heuristic (LAH) and geometric cutting heuristic (GCH), assuming the basic cases in which they do not include any additional constraints. We compare them to each other as well

Algorithm 3 Geometric Cutting Algorithm

- Step 1:** Begin with the graph containing all seats in the congress. Apply Algorithm 2 to divide the graph into two subgraphs, each of which corresponds to a particular subset of parties. Set both of these subgraphs to *active*.
- Step 2:** For every *active* subgraph:
- If the subgraph corresponds to only one party then set the subgraph to *complete*.
 - Otherwise, apply Algorithm 2 to the subgraph to create two new *active* subgraphs, and set the original subgraph to *inactive*.
- Step 3:** If there are any remaining *active* subgraphs then return to step 2. If there are no *active* subgraphs then stop. The *complete* subgraphs all correspond to an individual party, and the seats in the subgraph are assigned to that party to form the seating plan.
-

as to the solutions found for the facility location model (FLM) (1)–(6) and the minimum k -partition model (MPM) (11)–(14) by Xpress Optimization software. We also show some of the best and worst solutions generated by each heuristic and analyse them visually.

7.1 Procedure

We considered four different sizes of congress: 50, 100, 200 and 400 seats. The coordinates of the seats were generated by an online tool for generating parliament diagrams⁵ which were then used in Matlab to produce the diagrams shown in this paper. The adjacency graphs were formed by drawing edges between pairs of seats whose Euclidean distance was less than a threshold, which was determined as the smallest distance such that every seat, where applicable, is connected to both seats on either side of it and at least one seat from the row in front and one from the row behind. Figure 5 shows the resulting graphs. For the GCH, the parameter d , which determines the number of line directions, was set to $d = 32$ in all cases.

As well as varying the number of seats in the congress, the number of parties and number of representatives from each party were varied. It is typical for there to be two, three or four larger parties that win the majority of seats, followed by several much smaller parties that have only a handful of seats each. We considered these cases, as well as the case in which the number of seats belonging to parties decreases exponentially, with the largest party having half the seats, the next largest having a quarter of the seats, and so on. Table 1 shows all combinations of congress size and seat breakdowns that we considered.

For each variation, five sets of solutions were generated by each heuristic. For the congresses with 50, 100 and 200 seats, each set consists of 100 solutions, while for the congress of 400 seats each set consists of generating 30 solutions. For each solution generated by either of the heuristics, we recorded the objective value of the solution under both the FLM and MPM models. The results tables display the following information:

- Best best: The lowest value achieved by any solution in any of the five sets.

⁵ <http://tools.wmflabs.org/parliamentdiagram/parliamentinputform.html>.

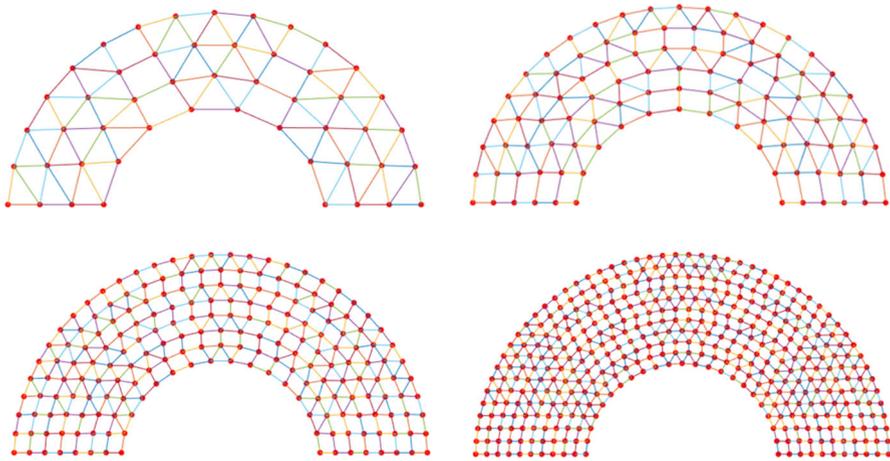


Fig. 5 Adjacency graphs for the four congress sizes

Table 1 Seat breakdowns that were studied

	50 seats	100 seats	200 seats	400 seats
Exponential	25, 13, 6, 3, 2, 1	50, 25, 13, 6, 3, 2, 1	100, 50, 25, 13, 6, 3, 2, 1	200, 100, 50, 25, 13, 6, 3, 2, 1
Two large parties	25, 20, 3, 2	50, 40, 5, 3, 2	100, 80, 8, 6, 4, 2	200, 160, 13, 10, 8, 6, 3
Three large parties	–	–	65, 60, 55, 9, 6, 5	113, 123, 114, 10, 8, 7, 5
Four large parties	–	–	–	105, 99, 93, 88, 7, 5, 3

- Mean best: The mean of the lowest values from each set.
- Mean mean: The mean value from all solutions in all five sets.
- Mean worst: The mean of the highest values from each set.
- Worst worst: The highest value from any of the five sets.

For both heuristics, the computational time required to generate each individual solution is negligible, particularly when considered in the context of the application, which would not require solutions to be generated very quickly.

We also recorded the best solutions found for the FLM and MPM by running the models in Xpress solver for 30 min (best found). The percentage gap between the best lower bound and best feasible solution found by the solver are shown. A gap of 0% indicates that the model was solved to optimality, in which case the time taken to find an optimal solution is stated in parentheses. Xpress Optimizer Version 29.01.10 was used on a Core i5 machine with clock speed 3.2 GHz and 8 GB of RAM run on Windows 7 OS.

Table 2 50 seats, exponential breakdown

	FLM		MPM	
Results from solver				
Best found	79		20	
Gap	0% (6s)		0% (188s)	
	FLM		MPM	
	LAH	GCH	LAH	GCH
Results from heuristics				
Best best	79	79	20	20
Mean best	79	79	21.2	20
Mean mean	81.8	81.0	25.3	22.2
Mean worst	95.8	83.8	35.2	25.2
Worst worst	107	85	40	26

Table 3 50 seats, two large parties

	FLM		MPM	
Results from solver				
Best found	91		14	
Gap	0% (9s)		0% (11s)	
	FLM		MPM	
	LAH	GCH	LAH	GCH
Results from heuristics				
Best best	91	91	14	14
Mean best	91	91	14	14
Mean mean	93.2	92.5	15.9	14.6
Mean worst	98.4	94	21.2	16
Worst worst	100	94	25	16

7.2 Results and analysis

The results of the computational study are shown in Tables 2, 3, 4, 5, 6, 7, 8, 9, 10, 11 and 12. We can make the following three key observations:

1. In 30 min, Xpress solver was able to find optimal solutions for the FLM with 50 and 100 seats and for the MPM with 50 seats. In both cases with 50 seats, both heuristics were able to produce solutions that were optimal for either model. For 100 seats, both heuristics were able to produce solutions that were very close to optimal for the FLM.
2. Under the FLM objective function, the mean values produced by both heuristics are very similar. However, the best scores from the LAH are consistently slightly better than the best scores from the GCH, while the worst scores from the LAH are usually worse than the worst scores from the GCH.

Table 4 100 seats, exponential breakdown

	FLM		MPM	
Results from solver				
Best found	218		33	
Gap	0% (114s)		81%	
	FLM		MPM	
	LAH	GCH	LAH	GCH
Results from heuristics				
Best best	219	220	30	29
Mean best	219.4	220.8	31.8	29.6
Mean mean	228.7	227.7	38.5	32.9
Mean worst	244.6	236.2	48.4	26.2
Worst worst	251	237	56	37

Table 5 100 seats, two large parties

	FLM		MPM	
Results from solver				
Best found	251		22	
Gap	0% (71s)		66%	
	FLM		MPM	
	LAH	GCH	LAH	GCH
Results from heuristics				
Best best	251	251	20	20
Mean best	251.2	251.4	20.2	20
Mean mean	256.6	255.7	24.1	21.1
Mean worst	268.4	264.2	31.2	24
Worst worst	272	266	33	24

3. Under the MPM objective function, the GCH performs significantly better than the LAH. Indeed, the best, mean and worst scores from the GCH are all better than the corresponding values from the LAH.

Of course, examining the numerical results is only useful if they correspond to our visual intuition regarding what constitutes a good seating plan. Therefore we will now show some examples of the best and worst solutions generated by each heuristic. This serves two purposes; firstly, we can judge whether the best and worse solutions numerically correspond to what we would consider to be good and bad solutions visually, and, secondly, we can visually compare and contrast the solutions generated by the two heuristics. Recall that we are only studying the basic cases, without imposing any additional conditions, and so at this point we are only looking to see whether the seats are arranged in a sensible and visually appealing way. In particular, the seats should form compact shapes that are not too ‘spread out’ and ideally each party’s

Table 6 200 seats, exponential breakdown

	FLM		MPM	
	LAH	GCH	LAH	GCH
Results from solver				
Best found	658		55	
Gap	9%		96%	
Results from heuristics				
Best best	660	664	43	41
Mean best	661.8	666.2	45.2	41
Mean mean	686.9	684.1	52.9	45.1
Mean worst	717.6	715	62.4	50.2
Worst worst	721	719	65	51

Table 7 200 seats, two large parties

	FLM		MPM	
	LAH	GCH	LAH	GCH
Results from solver				
Best found	744		38	
Gap	59%		93%	
Results from heuristics				
Best best	745	746	29	25
Mean best	747	747.2	29.4	25.20
Mean mean	764.8	763	34.8	29.2
Mean worst	857.2	777.6	48	33
Worst worst	1102	779	68	33

seats should be connected or contiguous. For purposes of brevity, we only look at the solutions for the congress with 400 seats. Each figure shows four solutions, which correspond to the four types of seat breakdown.

Figure 6 shows the best solutions generated by the LAH under the FLM objective function. The blocks appear reasonably compact. There is a small problem with the bottom-left solution because the black seats are not connected. The boundaries between the blocks are a little irregular and wiggly, but are not too bad.

Figure 7 shows the worst solutions generated by the LAH under the FLM objective function. In the top-left solution, the purple party are entirely surrounded by the yellow party, which is not ideal. In the top-right solution, the yellow block is a long, thin shape which is therefore not visually compact. In the bottom-left solution, the red and yellow blocks are not visually compact shapes, and their boundary is very irregular. In the bottom-right solution, the blue block does not appear visually compact. In general,

Table 8 200 seats, three large parties

	FLM		MPM	
	LAH	GCH	LAH	GCH
Results from solver				
Best found		614		46
Gap		52%		98%
Results from heuristics				
Best best	594	596	37	35
Mean best	595.6	596.4	38.2	35.2
Mean mean	612.4	609.7	44.3	38
Mean worst	676.8	631.2	55.6	41.4
Worst worst	742	632	61	42

Table 9 400 seats, exponential breakdown

	FLM		MPM	
	LAH	GCH	LAH	GCH
Results from solver				
Best found		1996		83
Gap		77%		99%
Results from heuristics				
Best best	1909	1914	66	60
Mean best	1917	1925	68.4	61.4
Mean mean	1961.2	1962.5	77.1	66.4
Mean worst	2022.4	2016.2	86.4	71
Worst worst	2036	2022	89	72

these solutions are visually inferior to the solutions in Fig. 6, which suggests that there is a correlation between the numerical quality and the visual quality of solutions.

Figure 8 shows the best solutions generated by the GCH under the MPM objective function. In all of the solutions, the blocks of seats all appear compact and connected, with well-defined borders.

Figure 9 shows the worst solutions generated by the GCH under the MPM objective function. The blocks are all connected and seem reasonably compact, with the possible exception of the pink party in the bottom-left solution. It is noticeable that the worst solutions generated by the GCH are visually much better than the worst solutions generated by the LAH. The worst solutions generated by the GCH were also numerically better than the worst solutions generated by the LAH, which again suggests that there is a correlation between the numerical and visual quality of solutions.

Table 10 400 seats, two large parties

	FLM		MPM	
Results from solver				
Best found	2333		65	
Gap	78%		100%	
	FLM	MPM		
	LAH	GCH	LAH	GCH
Results from heuristics				
Best best	2158	2163	46	42
Mean best	2165.4	2169.6	47.2	42.4
Mean mean	2199.1	2198.7	55.0	48.2
Mean worst	2237.6	2243.8	65	53.2
Worst worst	2250	2251	69	54

Table 11 400 seats, three large parties

	FLM		MPM	
Results from solver				
Best found	1899		65	
Gap	74%		100%	
	FLM	MPM		
	LAH	GCH	LAH	GCH
Results from heuristics				
Best best	1727	1739	53	50
Mean best	1732.8	1744.2	54.2	50.2
Mean mean	1768.3	1769.9	59.7	58.8
Mean worst	2019.8	1805.4	80.6	56.8
Worst worst	2177	1813	94	57

In summary, the results suggest that visually the GCH performs better than the LAH. However, the LAH has the advantage that it is easier to incorporate additional requirements, because they can simply be included as constraints in the allocation phase. With the GCH, on the other hand, it is not so easy to include additional requirements.

8 Case study: Congress of Deputies

We now consider a case study of the Spanish Congress of Deputies. We include several requirements that are specific to the application and discuss how these can be accommodated by the heuristics.

In the Congress of Deputies there are 350 elected representatives. There are 368 physical seats in the chamber, and nine cross-party members are chosen to sit at the

Table 12 400 seats, four large parties

	FLM		MPM	
Results from solver				
Best found	1791		67	
Gap	74%		100%	
	FLM		MPM	
	LAH	GCH	LAH	GCH
Results from heuristics				
Best best	1641	1650	65	62
Mean best	1647.4	1653.8	67	62
Mean mean	1679.9	1675.4	75.3	66
Mean worst	1815.8	1697	87.6	70.4
Worst worst	1855	1703	95	71

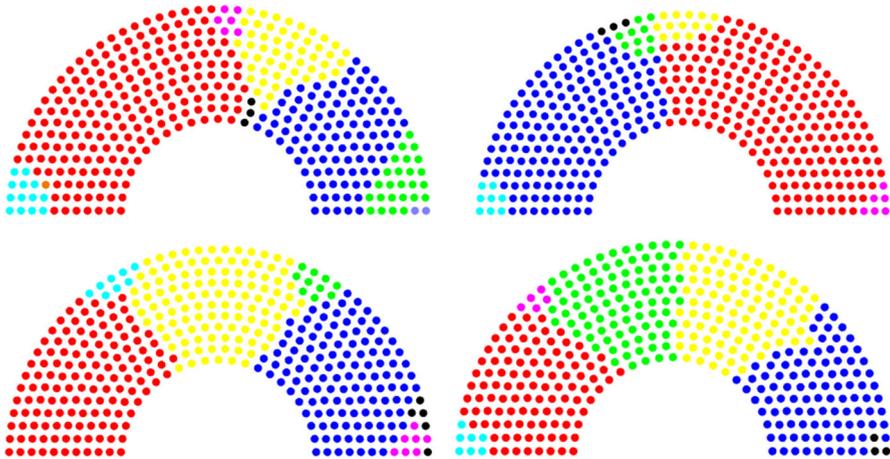


Fig. 6 The best solutions from the LAH under the FLM objective (color figure online)

Mesa del Congreso at the front, meaning that in practice there is a surplus of 27 seats. To simplify matters, we remove the members at the Mesa del Congreso from consideration, so that the problem involves assigning 341 members to 368 seats. The size of the problem makes finding optimal solutions to the FLM or MPM impractical, and so it is necessary to use the heuristics.

The number of representatives from each party is obtained from the results of the election of the 26th June 2016, shown in Table 13. The table also shows the index that will be used for each party in the model and the colour that will be used to represent each party in the solutions. Traditionally, PSOE sit on the left-hand side, PP sit on the right, and Ciudadanos and smaller parties somewhere in the centre. As a group that identifies as being on the left of the political spectrum, we also make the

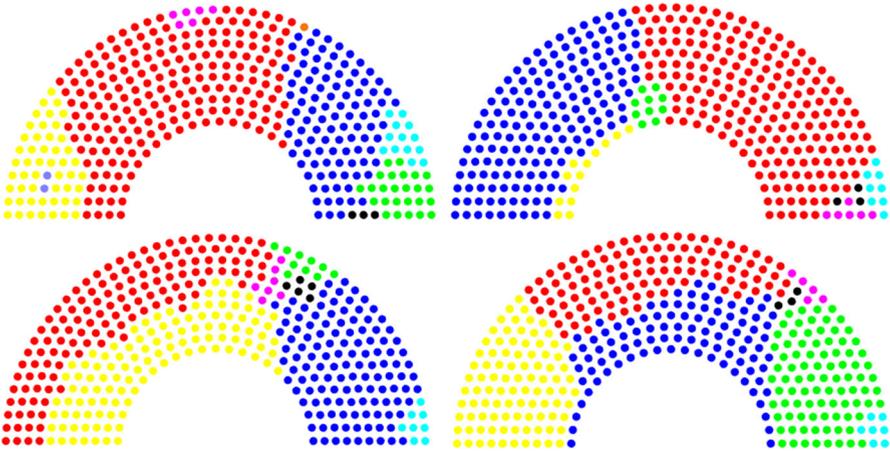


Fig. 7 The worst solutions from the LAH under the FLM objective (color figure online)

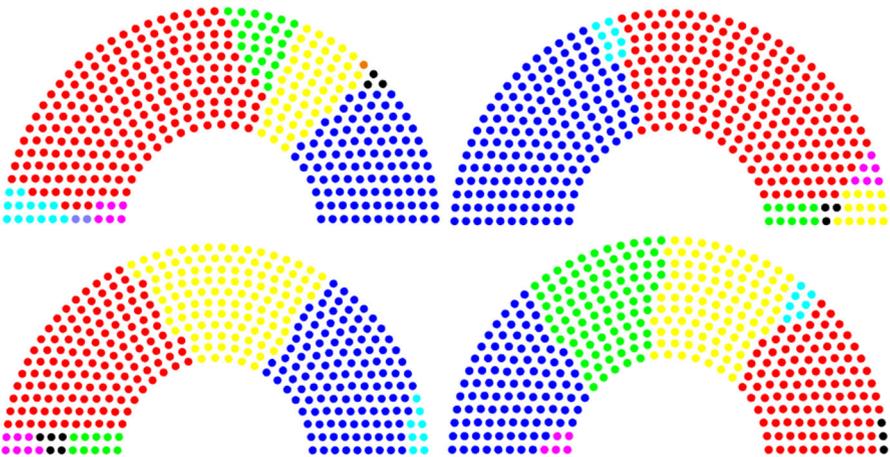


Fig. 8 The best solutions from the GCH under the MPM objective function (color figure online)

assumption that Podemos should sit on the left. The front row of seats is only occupied by government ministers, in this case meaning only the PP can sit in the front row.

The coordinates of the seats used to produce all the diagrams in this section were obtained from a pre-existing diagram of the Spanish Congress,⁶ and the coordinates were then used in Matlab to produce the diagrams shown.

⁶ https://commons.wikimedia.org/wiki/File:Congreso_de_los_Diputados_de_la_XII_Legislatura_de_Espa%C3%B1a.svg.

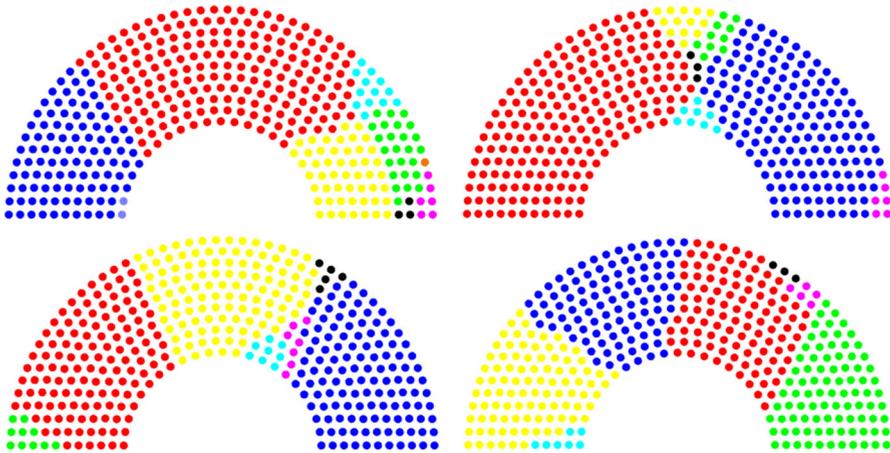


Fig. 9 The worst solutions from the GCH under the MPM objective function (color figure online)

Table 13 The breakdown of seats in the Spanish Congress

Index	Group	Seats	Colour
1	Partido Popular (PP)	131	Red
2	Partido Socialista Obrero Español (PSOE)	82	Blue
3	Unidos Podemos	65	Yellow
4	Ciudadanos	30	Green
5	Esquerra Republicana de Catalunya (ERC)	9	Aqua
6	Partido Nacionalista Vasco (PNV)	5	Pink
7	Mixto	19	Black
		341	

8.1 The LAH with additional constraints

The location phase of the heuristic functions in exactly the same way as described in Sect. 5, but we include several additional constraints in the allocation model in an attempt to (a) encourage fairness and (b) adhere to the rules and traditions of the congress. The full allocation model is shown below, followed by an explanation:

$$\min \sum_{j=1}^7 \sum_{i=1}^{368} d_{ic_j} x_{ij} \tag{18}$$

$$\text{s.t.} \quad \sum_{j=1}^7 x_{ij} \leq 1 \quad i = 1, \dots, 368, \tag{19}$$

$$\sum_{i=1}^{368} x_{ij} = K_j \quad j = 1, \dots, 7, \tag{20}$$

$$x_{ij} \in \{0, 1\} \quad i = 1, \dots, 368, j = 1, \dots, 7, \quad (21)$$

$$x_{ij} \leq \sum_{k=1}^n n_{ik}x_{kj} \quad i = 1, \dots, 368, j = 1, \dots, 7, \quad (22)$$

$$\sum_{i=1}^n f_i^{(2)}x_{ij} \geq \frac{K_j - 20}{20} \quad j = 1, \dots, 7, \quad (23)$$

$$\sum_{i=1}^n (7 - R_i)x_{ij} \geq 0 \quad j = 1, \dots, 7, \quad (24)$$

$$f_i^{(1)}x_{ij} = 0 \quad i = 1, \dots, 368, j = 2, \dots, 7, \quad (25)$$

$$x_{ij} = 0 \quad j = 1, \forall i : p_i \leq 50, \quad (26)$$

$$x_{ij} = 0 \quad j = 2, 3, \forall i : p_i \geq 50, \quad (27)$$

$$x_{ij} = 0 \quad j = 4, 5, 6, 7, \forall i : q_i \leq 35. \quad (28)$$

- The basic model is contained in Eqs. (18)–(21), with the only difference being that what were previously equalities in constraints (19) are now inequalities due to the fact in this case there are more physical seats than there are party members, so some seats must be left unassigned.
- Constraints (22) ensure that each representative is sat next to another representative from the same party.
- Let $f_i^{(2)} = 1$ if seat i is in the second row, and zero otherwise. Then constraints (23) ensure that any party with more than 20 seats has at least one seat in the second row, any party with more than 40 seats has at least 2 seats in the second row, and so on.
- Let R_i be equal to the row number of seat i , with the front row being row 1, and the back row being row 8. Constraints (24) mean that the average row number of each party must be less than or equal to seven.
- Let $f_i^{(1)} = 1$ if seat i is in the front row and zero otherwise. Then constraints (25) mean that only PP can occupy seats in the front row.
- Let (p_i, q_i) be the coordinates of seat i , as shown in Fig. 10. Constraints (26)–(28) ensure that each party is restricted to the correct part of the congress.

After running the LAH ten times, the two best and two worst solutions based on their objective values are shown in Fig. 11. In each of the four solutions there are notable problems with the Ciudadanos seats (in green). Since there are 30 green seats, constraints (23) mean there must be at least one green seat in the second row, and due to constraints (22) every member must be sat next to another person from the same party, and so there must be at least two green seats in the second row. The problem is that in all of the solutions these two green seats in the second row are separated from the rest of the party’s seats.

As was discussed in Sect. 5, one option to attempt to improve the solutions is to scale the objective function so that the model encourages all blocks to be compact, regardless of the number of seats they contain. Figure 12 shows the two best and two worst solutions after running the LAH with a scaled objective function (17). In three

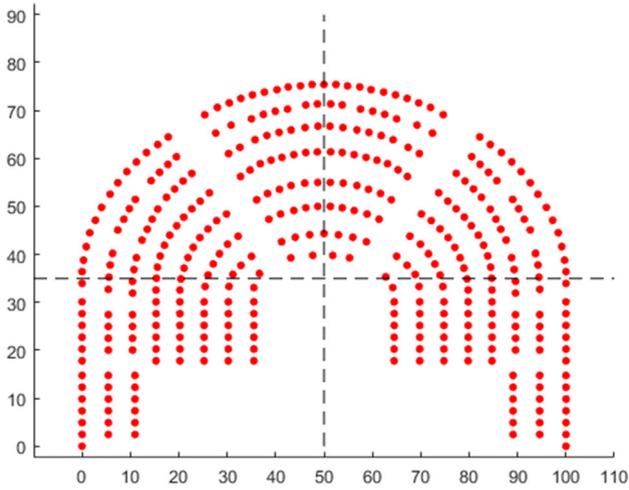


Fig. 10 Use seat coordinates to restrict the areas occupied by each party (color figure online)

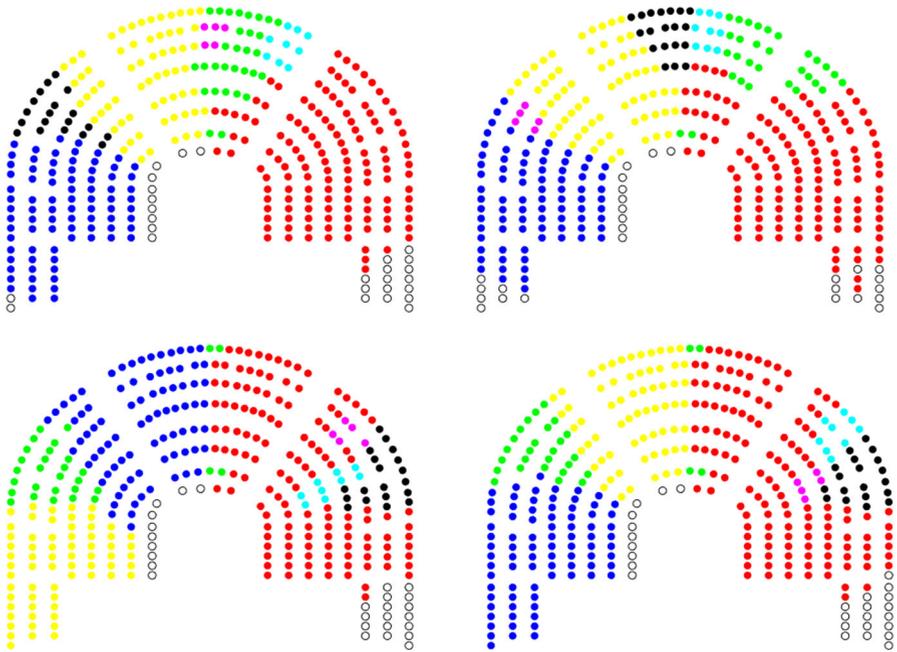


Fig. 11 Four of ten solutions generated by the LAH. Clockwise from top left: best solution (1466), second best solution (1472), second worst solution (1590), worst solution (1612) (color figure online)

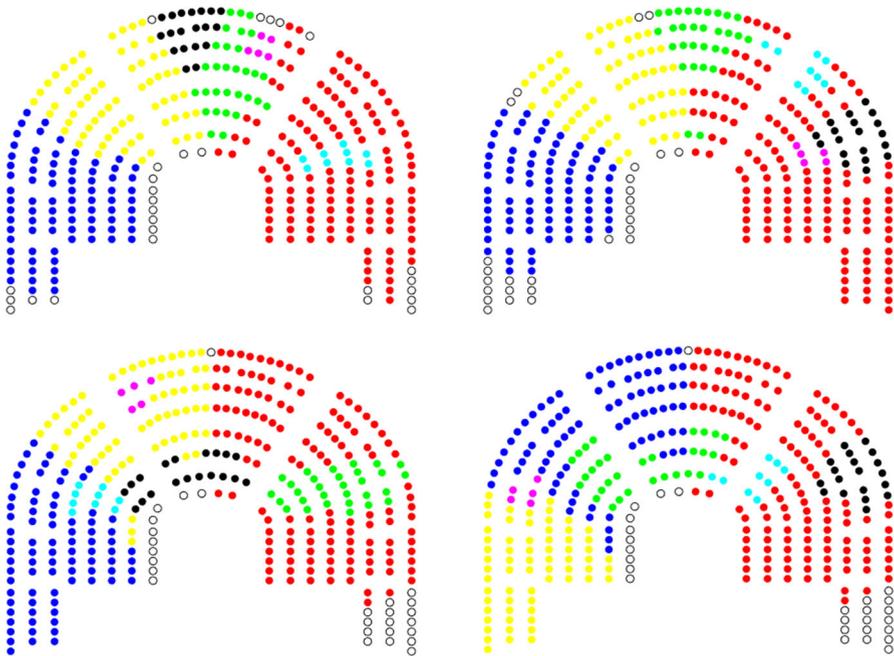


Fig. 12 Four of ten solutions generated by the scaled LAH. Clockwise from top left: best solution (3.271), second best solution (3.354), second worst solution (3.634), worst solution (3.731) (color figure online)

of the solutions, the scaling has prevented the previous problems with the green block, although now the PP seats (in red) do not form very compact shapes.

8.2 Filtering solutions from the GCH

The geometric cutting heuristic (GCH) does not have the same flexibility to incorporate additional requirements as the LAH. However, the results from the computational study suggest that it is capable of producing solutions that are visually very good.

Recall that in the Congress of Deputies we need to assign 341 representatives to 368 seats and only the PP are permitted to sit in the front row. The GCH requires an equal number of representatives and seats, therefore we supposed that we have 368 representatives by increasing the number of representatives from each party, so that the breakdown is {139, 89, 70, 33, 10, 6, 21}. We then generated a solution, which we deem infeasible if it does not satisfy the party location constraints (26)–(28) from the allocation model. Finally, if the solution is still feasible, we remove all front row seats that do not belong to the PP. Provided each party is still allocated at least as many seats as they require, we accept the solution. We assume that parties that are still assigned too many seats can choose themselves which ones they leave unoccupied.

After running the GCH one thousand times with this procedure, eight of the solutions remained feasible. The best and worst of these eight solutions under the MPM objective function are shown in Fig. 13 on the left and right, respectively. We see that

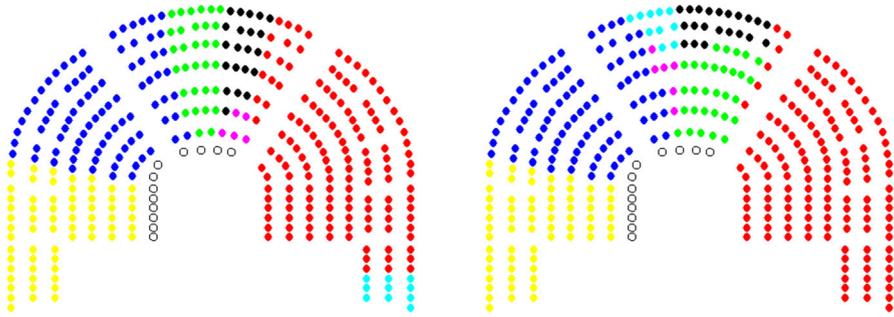


Fig. 13 Best and worst solutions produced by the GCH under the MPM objective function. The left solution has an MPM objective value of 42 and an FLM objective value of 1620. The right solution has an MPM objective of 49 and an FLM objective value of 1641 (color figure online)

the solutions are visually very good, with clearly defined borders between blocks, and the only notably non-compact set of seats is the pink seats in the right solution.

If we were to include more additional requirements, as we did with the LAH, it will become increasingly unlikely that any of the solutions generated by the GCH will remain feasible. Therefore, at least in its current form, the GCH is unsuitable when there is a large number of specific requirements that must be satisfied. However, we have seen that the solutions produced by the GCH tend to be very good, and so if it is possible to obtain feasible solutions then they are likely to be of a high quality.

9 Concluding remarks

We conclude with a brief summary of our findings, which are then discussed together with possible directions for further study.

9.1 Summary of findings

The original aim was to investigate ways in which mathematical optimization can be used to generate congress seating plans. We began with a discussion of possible approaches to the problem, noting the similarities with three other types of problems: political districting, facility location and graph partitioning.

Next, two optimization models were developed. The first was based on a facility location approach, while the second was a minimum k -partitioning model. Due to the number of binary variables, and the non-linearity in the case of the second model, optimal solutions could not be found in reasonable time by optimization software except for very small examples, and so we proposed two heuristics to generate ‘good’ solutions.

The location-allocation heuristic (LAH) separated the location and allocation components of the facility location model, while the geometric cutting heuristic (GCH) used straight-line cuts to partition the graph, with line directions chosen to minimize the number of edges that are cut.

A computational study was undertaken to assess the performance of the heuristics, with the solutions evaluated both numerically and visually. The results showed that under the central seats compactness measure both heuristics performed similarly on average, although the best and worst solutions found by the LAH tended to be better and worse, respectively, than the best and worst solutions found by the GCH. On the other hand, the GCH performed significantly better under the cut-edges measure than the LAH. Visual examination of the solutions concurred with the numerical results; the best solutions both looked of a similar quality, but the worst numerical solutions from the LAH looked significantly worse than those from the GCH. Therefore, for solving the basic problem of partitioning the seats into compact blocks, it is fair to say that the GCH performs better than the LAH. However, the GCH has the disadvantage of not being able to incorporate additional constraints easily, whereas in the LAH they can be included in the allocation phase.

We concluded with a case study of the Spanish Congress with additional constraints. When these additional constraints were included in the LAH we found that they can result in the seats forming irregular shapes, although this effect can be partly mitigated by scaling the objective function. The possibility of filtering solutions generated by the GCH was also considered and it was found that if any solutions are found then they are likely to be very good, but as the number of additional constraints increases it becomes increasingly unlikely that any solutions will remain feasible.

9.2 Discussion and possible further study

The primary objective of this paper was to investigate how mathematical modelling and optimization could be used to produce seating plans for congress chambers. We have discussed several approaches, and have shown that if a set of basic rules can be agreed upon regarding how the seats should be assigned, then it is possible to generate politically unbiased seating plans that satisfy these rules. In particular, the removal of any potential political bias should help to prevent complaints such as those from the Podemos spokesperson mentioned in the introduction. On the other hand, we found that the inclusion of certain rules can result in irregular looking seating plans, and future research should try to address this problem.

In this sense, it would be particularly interesting to include explicitly in the model constraints that enforce connectivity (remember that our models encourage connectivity, but do not guarantee it). As part of our future research, we would like to study the inclusion of connectivity constraints such as those described in Carvajal et al. (2013) and Wang et al. (2017). This has not been done in this paper because here we have tried to show that some simple models can already provide unbiased and meaningful solutions to arrange seats in a congress chamber. It is true that it is not guaranteed that the solution will be perfect, but we have shown that most of the solutions are very good starting points that require a minimal manual modification to provide satisfactory solutions. The constraints described in those two papers would increase the complexity of the models and some algorithms, much more specialized, would be necessary to solve those larger models. Although very interesting to us from the practical and

mathematical point of view, we feel that we would be losing from sight the simplicity aspect of this paper.

Another benefit of using a mathematical approach rather than devising seating plans by hand is that the heuristics are able to produce a large number of possible plans very quickly. Moreover, there is a lot of variety in these plans, as can be seen from the diagrams in Sects. 7 and 8, which presents the user with lots of flexibility, although of course this means restoring some of the decision-making to human hands, thereby reopening the potential for bias. However, as we mentioned in the introduction, the inherent subjectivity of the problem means that trying to obtain the ‘perfect’ seating plan will always be impractical.

There are many possible approaches to this problem, and these could be further explored by additional study. In particular, there are many ways to interpret what makes a seating plan visually appealing and how to express this mathematically. The success of the geometric cutting heuristic (GCH) when applied to the basic problem suggests that further investigations into geometric approaches would be worthwhile.

Finally, it would be interesting to combine the topic with a more political-focused study into what the politicians themselves want from a seating plan. This would serve to clarify and consolidate the objectives, which would in turn assist the development of possible optimization-based approaches.

Open Access This article is distributed under the terms of the Creative Commons Attribution 4.0 International License (<http://creativecommons.org/licenses/by/4.0/>), which permits unrestricted use, distribution, and reproduction in any medium, provided you give appropriate credit to the original author(s) and the source, provide a link to the Creative Commons license, and indicate if changes were made.

References

- Carvajal R, Constantino M, Goycoolea M, Vielma JP, Weintraub A (2013) Imposing connectivity constraints in forest planning models. *Oper Res* 61(4):824–836
- Chopra S, Rao MR (1993) The partitioning problem. *Math Program* 59:87–115
- García S, Cacchiani V, Vanhaverbeke L, Bischoff M (2014) The table placement problem: a research challenge at the EWI 2007. *TOP* 22(1):208–226
- Hess SW, Weaver JB, Siegfeldt JN, Whelan HJ, Zitlau PA (1965) Nonpartisan political redistricting by computer. *Oper Res* 13(6):998–1006
- Hojati M (1996) Optimal political districting. *Comput Oper Res* 23(12):1147–1161
- Horn DL, Hampton CR, Vandenberg AJ (1993) Practical application of district compactness. *Polit Geogr* 12(2):103–120
- Kalcsics J, Nickel S, Schröder M (2005) Towards a unified territorial design approach—applications, algorithms and GIS integration. *Top* 13(1):1–56
- Kernighan BW, Lin S (1970) An efficient heuristic procedure for partitioning graphs. *Bell Syst Tech J* 49:291–307
- Niemi RG, Grofman B, Carlucci C, Hofeller T (1990) Measuring compactness and the role of a compactness standard in a test for partisan and racial gerrymandering. *J Polit* 52(4):1155–1181
- Ricca F, Scozzari A, Simeone B (2011) Political districting: from classical models to recent approaches. *4OR Q J Oper Res* 9:223–254
- Serafini P (2012) Allocation of the EU parliament seats via integer linear programming and revised quotas. *Math Soc Sci* 63:107–113
- Wang Y, Buchanan A, Butenko S (2017) On imposing connectivity constraints in integer programs. *Math Program* 166(1–2):241–271

Wolkowicz H, Zhao Q (1999) Semidefinite programming relaxations for the graph partitioning problem. *Discret Appl Math* 96–97:461–479

Publisher's Note Springer Nature remains neutral with regard to jurisdictional claims in published maps and institutional affiliations.