Universität Augsburg

Institut für Mathematik

Sebastian Maier, Friedrich Pukelsheim

Bazi: A Free Computer Program for Proportional Representation Apportionment

Preprint Nr. 042/2007 — 11. Dezember 2007

Institut für Mathematik, Universitätsstraße, D-86 135 Augsburg http://www.math.uni-augsburg.de/

Impressum:

Herausgeber:

Institut für Mathematik
Universität Augsburg
86135 Augsburg
`http://www.math.uni-augsburg.de/forschung/preprint/`

ViSdP:

Sebastian Maier
Institut für Mathematik
Universität Augsburg
86135 Augsburg

Bazi: A Free Computer Program for Proportional Representation Apportionment

Sebastian Maier, Friedrich Pukelsheim[*]

Bazi version 2007.10

Contents

[*] Maier@Math.Uni-Augsburg.De, Pukelsheim@Math.Uni-Augsburg.De

1 Introduction

The name Bazi is derived from “Berechnung von Anzahlen mit Zuteilungsmethoden im Internet”, which translates into “Calculation of allocations by apportionment methods in the Internet”. A “Bazi” is also local Augsburg dialect for a very good friend. Bazi is generally pronounced ′batzee, with the stress on the first syllable.

Bazi is a program to convert vote counts into seat numbers using one of the many apportionment methods that have been devised by practitioners and theoreticians, during more than a century of proportional representation history. The user may key-in his or her own data, but Bazi also features an extensive empirical database. Hence the apportionment methods can be studied by evaluating real data, as is illustrated in Section 2. Bazi’s graphical user interface is described in Section 3.

In the sequel we use the language of one of the prime fields of application, of converting votes into seats at the end of a popular election. With a typical electorate going into the millions, the vote proportions are virtually continuous quantities. The problem is to proportionally map them into decidedly discrete seat proportions, namely integer multiples of $1/h$, where h is a prespecified house size.

There are further instances where the numerical problem turns out to be the same even though the subject matter is different. In the field of politics, another important group of examples arises from apportioning parliamentary seats to electoral districts, or other geographical units, on the basis of population data.

Outside politics, for census data or publications of statistical offices, the apportionment problem occurs when turning proportions into percentages. In this application, the house size — or more general: the number of units to be dealt out — are $h = 100$ percentage points. This does away with the frequent proviso in statistical tables that “Due to rounding errors, percentages may not sum to 100”. Using apportionment methods, percentages do sum to the correct total, 100 percent. If the goal is to round to the tenth of a percent, we set $h = 1000$ and thus guarantee that, after inserting the decimal point, the rounded quantities sum to the correct total, 100.0 percent.

The topic of proportional representation has generated an enormous body of literature. On the Bazi homepage we provide a literature list that to date has grown to some 700 items. Among these, the monograph Balinski and Young (2001) is seminal, and also points towards the operational tasks that are solved by the Bazi program. Of the German literature, Kopfermann (1991) provides a comprehensive treatment including a meticulously documented review of the historical records of the theory.

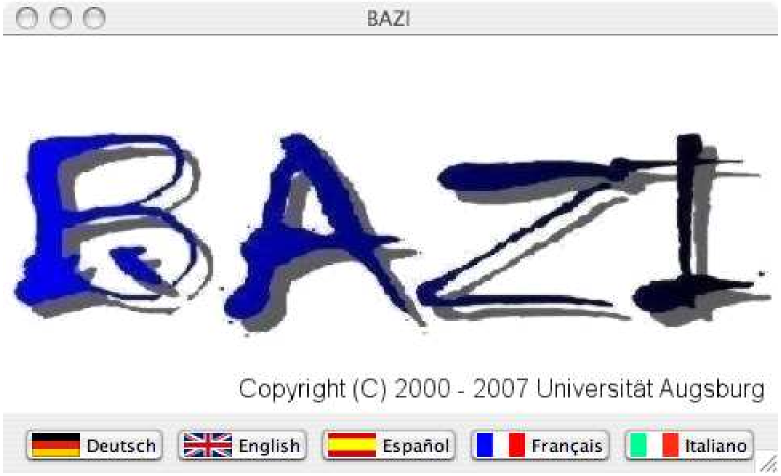


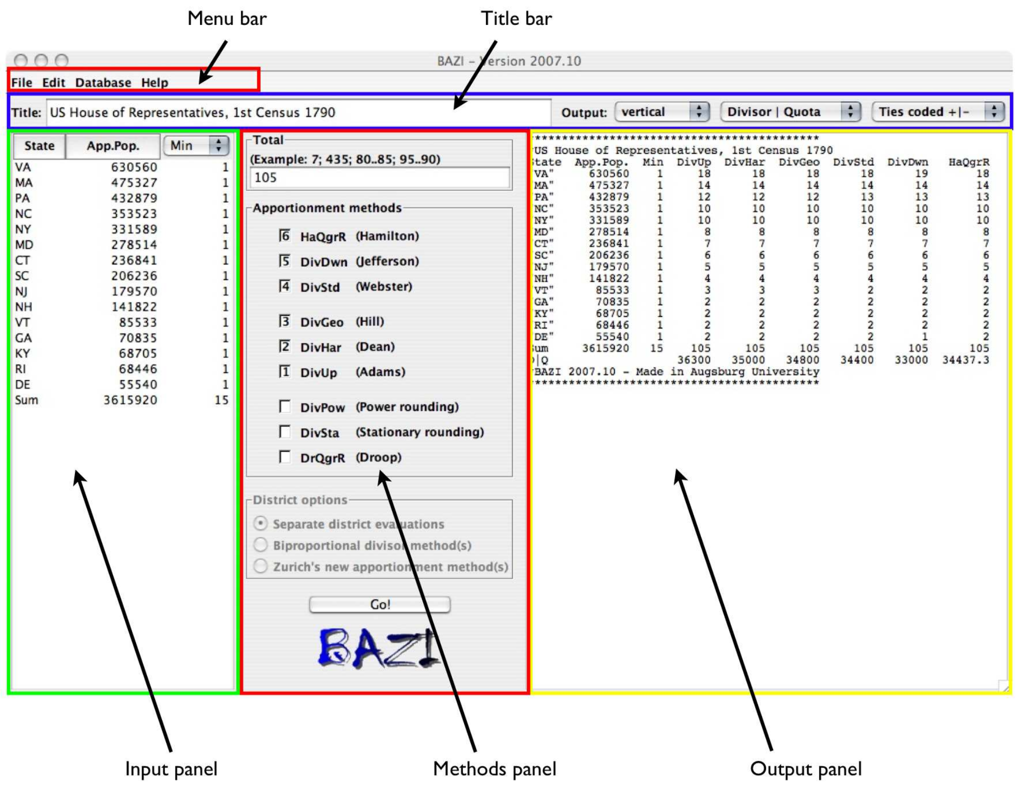
Fig. 1: Bazi's welcome screen. When invoked through the command `java -jar bazi.jar`, Bazi prompts the user for language selection. With an appropriate switch appended to the calling command, the prompt is shortcut.

1.1 Getting started

Bazi is made available by the University of Augsburg on its homepage `www.uni-augsburg.de/bazi`. Since Bazi is a Java program, it requires the Java run-time environment to be installed prior to running Bazi. Two downloads are offered. The first is preferably for Windows operating systems, the second, for all Unix dialects. These are packed files which, after saving the appropriate version to the user's machine, need to be unzipped. Bazi is meant to run under Windows, and under all Unix dialects. We would appreciate any feedback from users who find that this hope does not materialize.

Usually a double-click on the file named `Bazi.jar` will start the Bazi program. It may be convenient to place a link to this file to the desktop; the package includes an appropriate icon, featuring a checked ballot box. The Bazi program may also be started by executing the command `java -jar bazi.jar`. Bazi comes up with the welcome screen shown in Fig. 1.

The welcome screen prompts the user to select one of the five languages Deutsch, English, Español, Français, or Italiano. The prompt may be shortcut by appending to the call of Bazi one of the switches `de`, `en`, `es`, `fr`, or `it`. Another switch that may be added is `max`, telling the computer environment to start Bazi in a maximized window.



Title: US House of Representatives, 1st Census 1790

Output: vertical | Divisor | Quota | Ties coded +|-

State	App.Pop.	Min
VA	630560	1
MA	475327	1
PA	432879	1
NC	353523	1
NY	331589	1
MD	278514	1
CT	236841	1
SC	206236	1
NJ	179570	1
NH	141822	1
VT	85533	1
GA	70835	1
KY	68705	1
RI	68446	1
DE	55540	1
Sum	3615920	15

Total
(Example: 7; 435; 80..85; 95..90)
105

Apportionment methods
6 HaQgrR (Hamilton)
5 DivDwn (Jefferson)
4 DivStd (Webster)
3 DivGeo (Hill)
2 DivHar (Dean)
1 DivUp (Adams)
□ DivPow (Power rounding)
□ DivSta (Stationary rounding)
□ DrQgrR (Droop)

District options
● Separate district evaluations
○ Biproportional divisor method(s)
○ Zurich's new apportionment method(s)

```
*****************************************
US House of Representatives, 1st Census 1790
State  App.Pop.  Min  DivUp  DivHar  DivGeo  DivStd  DivDwn  HaQgrR
VA"     630560    1     18      18      18      18      19      18
MA"     475327    1     14      14      14      14      14      14
PA"     432879    1     12      12      12      13      13      13
NC"     353523    1     10      10      10      10      10      10
NY"     331589    1     10      10      10      10      10      10
MD"     278514    1      8       8       8       8       8       8
CT"     236841    1      7       7       7       7       7       7
SC"     206236    1      6       6       6       6       6       6
NJ"     179570    1      5       5       5       5       5       5
NH"     141822    1      4       4       4       4       4       4
VT"      85533    1      3       3       3       2       2       2
GA"      70835    1      2       2       2       2       2       2
KY"      68705    1      2       2       2       2       2       2
RI"      68446    1      2       2       2       2       2       2
DE"      55540    1      2       2       2       2       1       2
Sum    3615920   15    105     105     105     105     105     105
D|Q                  36300   35000   34800   34400   33000 34437.3
BAZI 2007.10 - Made in Augsburg University
*****************************************
```

Fig. 2: Bazi's graphical user interface. The example, from Subsection 2.1, shows the apportionment of 105 seats, using six apportionment methods. The output prints the apportionment results side by side, to facilitate comparison.

1.2 Command-line mode

Bazi may also be run in a command-line batch mode circumventing the graphical user interface, particularly useful for advanced users carrying out a simulation study, say. Detailed instructions for installation and the command-line mode are provided in the `readme.txt` file that comes with the download.

1.3 Bazi's graphical user interface

Bazi's graphical user interface is subdivided into two bars at the top, the menu bar and the title bar, and three panels in the main body of the window, the input panel, the methods panel, and the output panel. See Fig. 2.

Rather than right away delving into Bazi's program structure, we follow the learning-by-doing principle and start out by demonstrating Bazi's potentials in a series of examples, in Section 2.

2 Apportionment methods and examples

The following subsections take the reader on a guided tour illustrating the apportionment methods implemented in Bazi, and the ramifications offered. All examples are drawn from the database that comes with the program.

Bazi contains an extensive database with empirical data, with all the information needed to carry out an apportionment problem: vote counts (or population data), house size, and apportionment methods appropriate for the particular instance. The files may also include some additional information on the data which, when extant, is displayed in another window. An example of an input file in the `.bazi`-format is reproduced in Subsection 3.6.

Subsections 2.1 to 2.3 contain basic examples, whence our commentary turns out to be somewhat more verbose. Subsections 2.4 to 2.12 comprise advanced examples and, assuming that by then the user is familiar with the Bazi program, our commentary is more compact.

With Bazi up and running, the user is expected to click into the database menu that is located in the menu bar, and then follow the path given below in each of the subsections. The path ends in a data file that will then be automatically loaded into Bazi's input panel on the left part of the screen. A click on the *Go!* button, located at the bottom of the grey methods panel in the middle of the screen, makes Bazi carry out its calculations and print the results into the output panel on the right side of the screen.

Before proceeding from one example to the next, the user may want to refresh Bazi by going to the file menu (or edit menu) located in the menu bar, and select the *Restart* option to wipe out the input, methods, and output panels, and start afresh.

2.1 Apportionment tables

The database file *USA>US House of Representatives>1st Census 1790* is loaded with actually six apportionment methods checked. See Fig. 3.

A click on the *Go!* button prints the apportionment results into the output panel on the right part of the screen. This example reproduces the table on page 158 in Balinski and Young (2001), and permits to study the effects of the six apportionment methods checked. See Fig. 2.

The numbering of the methods indicates the sequence in which they are executed. A mouse click on one of the numbers makes it disappear, with the sequencing instantly adjusted. When all methods are deleted until a single method is left, only this one method is evaluated. A click on the *Go!* button causes Bazi to calculate the apportionment and to print the result into the output panel. With a single method, the output more transparent.

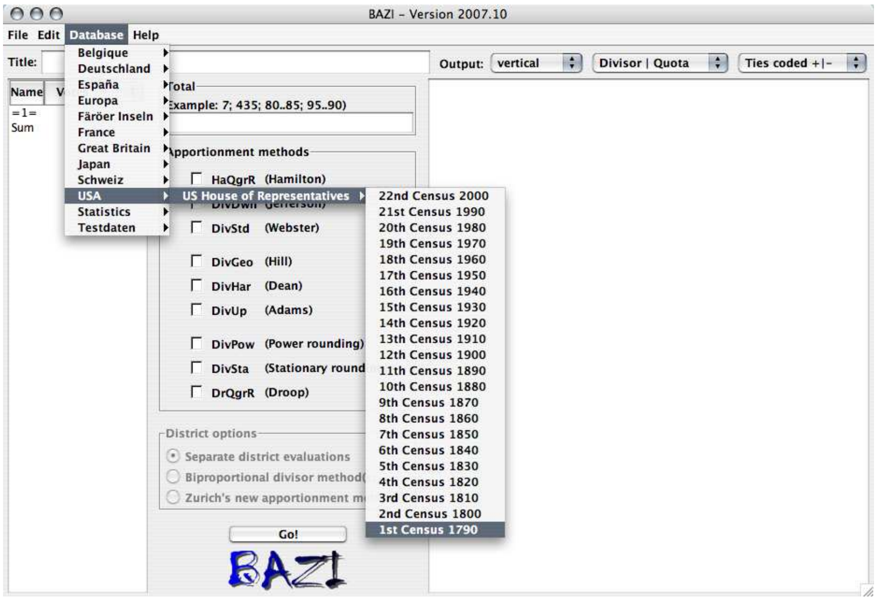


Fig. 3: Bazi's database menu. A click on the database menu opens the database directory. Here the path *USA>US House of Representatives> 1st Census 1790* is shown, see Subsection 2.1.

Note that the third column in the input panel is active, labeled `Min` in its header, with a 1 in each cell. This means that every State of the Union is guaranteed one seat, as is required by the US Constitution. The third column may be deactivated by a click on the little downward arrow in its header, and then selecting the three dashes `---`. A click on the *Go!* button makes Bazi do its job, resulting in the identical seat apportionments as before. The reason is that, with these data, each State is allocated one seat or more, whence with or without the minimum restriction the results remain the same. This is peculiar to the 1790 Census. With other data, the apportionment results may differ depending on whether the minimum restriction is active or not.

2.2 Divisor methods

An apportionment method with a long history is the divisor method with rounding down (Jefferson/D'Hondt/Hagenbach-Bischoff). An example is provided by the database file *Deutschland>Bundestag>Einzelauswertungen> 10.Dt.BT>Oberzuteilung.*

Note that the third column in the left input panel is not active, as is indicated by `---` and the fact that, in this column, numbers are shown in grey rather than black color. A click on the *Go!* button prints the corresponding apportionment result into the output panel on the right part on the screen.

By going to the very top of the third column, a click on the arrow displays the options offered. When selecting `Dir`, the third column features the direct seat winners, that is, the number of constituencies won by a candidate of that party. After a click on the *Go!* button, the output now contains an additional "overhang" line with zeros throughout. This is to say that the proportionally justified seats are sufficient to support all direct seat winners.

The apportionment method used in this example is the divisor method with rounding down (Jefferson/D'Hondt/Hagenbach-Bischoff), *DivDwn*. In order to compare the effect of this method with another method, a click into the little square box in front of *DivStd* activates this method, the divisor method with standard rounding (Webster/Sainte-Laguë/Schepers).

Another click on the *Go!* button produces the apportionment results for the two methods checked, the two largest parties losing one seat each and the two smallest parties gaining one. The reason is that the divisor method with rounding down (Jefferson/D'Hondt/Hagenbach-Bischoff) is biased in favor of larger parties and at the expense of smaller parties, while the divisor method with standard rounding (Webster/Sainte-Laguë/Schepers) is unbiased.

The divisor method with standard rounding (Webster/Sainte-Laguë/Schepers) is utilized by the German Bundestag to allocate committee seats on the basis of the parties's representation in the Bundestag. For an example, see the database file *Deutschland>Bundestag>Einzelauswertungen>16.Dt.BT>Ausschusssitze*. A click on the *Go!* button prints the apportionment results into the output panel on the right part of the window.

A third example from the divisor method family is the divisor method with rounding up (Adams), *DivUp*. This method favors smaller participants at the expense of larger participants, which makes it an attractive method to represent geographical units where often the political intention is to over-represent sparsely populated (rural) areas. It is used in France to apportion the 570 seats of the Assemblée nationale among the 100 Départements, see the database file *France>Répartitions à l'Assemblée nationale 1999*. A click on the *Go!* button produces the apportionment output. The example again features a minimum restriction of one seat for each Département, and is set up to fully reproduce the table on pages 316–318 in Balinski (2004).

Before proceeding to the quota methods examples the user may want to select the *Restart* option in the file menu (or edit menu), in order to clean the input panel, the method panel and the output panel.

It is quite easy to understand how divisor methods operate. For every divisor method Bazi prints into the output an appropriate divisor, either at the very right as the final output column, or else at the very bottom as the final output line. The divisor may be used to very easily double-check the seat numbers of the participants. All we have to do is to divide the vote counts of a party by the divisor offered by Bazi, and round the resulting quotient using the rounding rule that comes with the method.

In other words, using the divisor method with rounding down (Jefferson/D'Hondt/Hagenbach-Bischoff) all quotients are rounded down to their integer part. For the divisor method with standard rounding, quotients are rounded up or down according to the fractional part of the quotient being larger or smaller than one half. For the divisor method with rounding up (Adams), all quotients are rounded up to the next integer. In any case the divisors guarantee that the rounded numbers sum to the prespecified house size.

Usually there is a whole range of numbers that may serve as divisors, from which Bazi picks an easy-to-communicate intermediate value. Because of this feature, divisors are also called "flexible electoral keys" (bewegliche Wahlschlüssel). In contrast, quota methods use "fixed electoral keys" (feste Wahlschlüssel), to be discussed next.

2.3 Quota methods

A widely used quota method used is the Hare-quota method with residual fit by greatest remainders (Hamilton/Hare/Niemeyer). It is based on the "Hare-quota", the quotient of the vote total and the house size. In other words, the Hare-quota signifies the average (fractional) number of votes needed to obtain one seat. Many authors just refer to the "greatest remainders" method, or the "largest remainders" (LR) method.

An example is provided by the database file *Deutschland>Bundestag> Einzelauswertungen>16.Dt.BT>Oberzuteilung*, with the 2005 nationwide results for the German Bundestag electoral system. The third-column option `Dir`, for direct seats, is active. After a click on the *Go!* button, the proportionally justified seats are seen to suffice to support all direct seat winners.

With the chosen output format *vertical*, the output comes in columns. The first column features the names of the parties, the second column reproduces the input votes. The third column gives the seat apportionment of the 598 seats, and the last column, labeled *overhang*, contains zeros throughout since all direct seats are taken care of by the proportional seat apportionment.

A political entity that experimented with various quotas, each closely related to the Droop-quota, is the Swiss Canton Solothurn. The database files *Schweiz> Solothurn>...* are organized into four periods, the oldest three using methods that are closely related to the Droop-quota method with residual fit by greatest remainders, *DrQgrR*. More details are provided in the *Information* window, explaining the differences of the method used in the Canton Solothurn. In order to exhibit the differences between the seat assignments according to the Solothurn electoral law, and the seat apportionment calculauted by the Bazi program, the third-column, containing the actually allocated seats, has option `Sub` activated. Hence Bazi prints out the difference between the actual apportionment, and the seat numbers that are calculated under the checked method, *DrQgrR*. In most cases the output shows that the discrepancy is zero. For these cases, the method that was defined in the Solothurn electoral law produced the same apportionment results as the Droop-quota method with residual fit by greatest remainders, *DrQgrR*. See Pólya (1918) for more details on the Solothurn electoral systems.

Before proceeding to the next example the user may want to execute the *Restart* cammand in the file or edit menus. From now on we trust that the user is familiar with the functions of the *Go!* button and the Restart option, whence we will no longer make any reference to these actions.

The seat apportionment that is obtained from the Hare-quota with residual fit by greatest remainders (Hamilton/Hare/Niemeyer) is found as follows. The vote counts are divided by the Hare-quota and then the resulting quotients are evaluated in two steps. The first step, the "main apportionment", assigns to each party the integer parts of its quotients. Since in this first step the fractional parts of the quotients are neglected, they amount to a few residual seats. In the second step, the "residual apportionment", these residual seats are allocated, one by one, to the parties that feature the greatest remainders. This procedure explains the acronym that Bazi is using in the methods panel and that is shown at the top of the output column, *HaQgrR*, the Hare-quota method with residual fit by greatest remainders (Hamilton/Hare/Niemeyer).

Most quota methods use the greatest remainders strategy to allocate the residual seats, though other strategies have also been discussed. The bigger challenge is the decision of which quota to use. Originally Thomas Hare did not advertise what we nowadays call the Hare-quota, but recommended to round it down to its integer part. Of course, the reason is that in his days calculations were carried out with pencil and paper, rather than with machines, and agreeing to start calculations with whole numbers sounded more persuasive.

Nevertheless, other proposals emerged, with the goal to increase the number of seats to be allocated in the first, main apportionment step, and decrease the number of residual seats. Thus Henry Droop proposed what nowadays is called the Droop-quota, that is, one plus the integer part, of the quotient of the vote total divided by one plus the house size.

2.4 Combined lists, overhang seats, NA

The database file *Deutschland>Bundestag>16.Dt.BT* produces a full apportionment for the 2005 Bundestag election, illustrating two features. The first is list combinations. All other parties run in all 16 Länder, except for the CDU and CSU. The CDU runs in 15 Länder except Bavaria, the CSU campaigns only in Bavaria and no other Land. The prepended plus-operator indicates that, for the nationwide seat apportionment calculations, each party pools the vote counts in the Länder where campaigning. Then, for each party, the overall seats from the national level are sub-apportioned among the participating Länder lists.

The second feature is that the sub-apportionments of the overall SPD and CDU seats to their combined lists falls short of supporting all direct seat winners. Overhang seats emerge, nine for the SPD, and seven for the CDU.

If the third-column option `Min` is activated then, due to the shortcomings of the Hare-quota method with residual fit by greatest remainders (Hamilton/Hare/Niemeyer), the SPD sub-apportionment cannot be completed. Whenever Bazi is frustrated, it prints a `NA: Not available` into the output.

If, in addition, the apportionment method is changed to the divisor method with standard rounding (Webster/Sainte-Laguë/Schepers), calculations complete successfully. In each Land the SPD gets at least as many seats as are needed for the direct seat winners, as does the CDU. The active minimum restriction evades the occurrence of overhang seats.

2.5 Restrictions: Min, Max, Min-to-Max

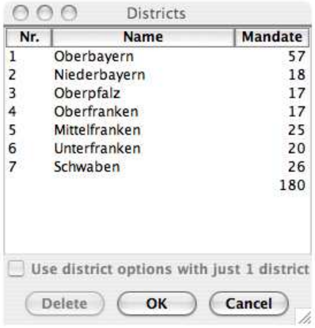
The database file *Great Britain>Scotland>2007* is an example where the divisor method with rounding down (Jefferson/D'Hondt/Hagenbach-Bischoff) is employed in the presence of minimum restrictions defined through the direct seat winners. The Scottish system evades the overhang defects of the German system, see Siaroff (2000). Changing the third-column option to `Dir`, as in the German Bundestag system, three overhang seats would emerge.

The database file *Europa>Wahlberechtigte>EU27 2004, Variante A* is a possible manifestation of what is envisioned in the Treaty establishing a Constitution for Europe. Each member state gets at least 6 and at most 96 seats in the European Parliament. Hence the Min-to-Max range is set to be 6..96 (Pukelsheim, 2007).

2.6 A split into two groups

The database file *Europa>Wahlberechtigte>EU27 2004, Variante B* features a proposal of how to protect the smaller EU member states from being overwhelmed by the larger countries. The 27 member states are split into two groups, in Bazi parlance groups are termed districts. The idea of splitting the whole into two parts, in order to honor the identity of each part, goes back to the Treaty of Westphalia 1648 (Pukelsheim and Maier, 2006).

The first group comprises the seven larger member states, to which 500 seats are allocated. The second group is made up of the 20 remaining smaller member states, to share 250 seats. In order to prevent a smaller member state to receive more seats than a larger member state, the Min-to-Max seat range for the smaller group is set to 6..30, for the larger group, to 30..96 (Pukelsheim, 2007).



Districts

Nr.	Name	Mandate
1	Oberbayern	57
2	Niederbayern	18
3	Oberpfalz	17
4	Oberfranken	17
5	Mittelfranken	25
6	Unterfranken	20
7	Schwaben	26
		180

□ Use district options with just 1 district

Fig. 4: Bazi's district dialogue. The example shows the district names and the district magnitudes which are editable, for the Bavarian state elections of Subsection 2.8.

2.7 A merger of two groups

The database file *Europa>Wahlberechtigte>EU27 2004, Variante C* offers yet another possibility. Each of the 27 member states is guaranteed 6 fixed seats. Bazi only deals with the remaining 588 seats, to apportion them proportionally to the member states' populations. Now, in the third-column, the options `Max` becomes relevant, and is set at 90. Hence no member state can exceed the overall maximum restriction 96 (= 6 + 90) from the Treaty Constitution (Pukelsheim, 2007).

2.8 Separate district evaluations

In many instances the whole electoral region is subdivided into various electoral districts. We have met such an instance already in Subsection 2.3, while discussing the situation in the Swiss Canton Solothurn. Another such instance occurs in the German State of Bavaria, see the database file *Deutschland>Landtage>Bayern>15.Bay.LT.*

The Bavarian State Legislature has 180 seats which, in the middle of the legislative period, are apportioned among the seven electoral districts proportional to population counts. Hence the total house size of 180 is broken down into seven district magnitudes, which function as the subtotal for the separate, within-district apportionment calculations. See Fig. 4.

At the end of the output, Bazi prints a summary table. Whether parties are displayed horizontally, across the output panel, or vertically, downwards, is determined through the *Output* switches locted in the Title bar. In some applications one format is more easily communicated than the other.

2.9 Biproportional apportionment methods

The previous example from Bavaria is used to also illustrate biproportional divisor methods. When that database file was loaded, in the *District options box* the button *Separate district evaluations* was activated. For biproportional methods, the user needs to instead push the *Biproportional divisor methods* button. However, biproportional methods only necessitate divisor methods, hence pushing the biproportionality button extinguishes the default quota method. For the Bavarian data, we recommend to check-mark the divisor method with standard rounding (Webster/Sainte-Laguë/Schepers).

Bazi then carries out two apportionments. First comes the super-apportionment where votes are pooled across the entire electoral region, to apportion the 180 seats among the three parties proportional to their overall vote totals. The second step is a sub-apportionment that respects both, the party seats from the super-apportionment, as well as the district magnitudes that were determined prior to the election.

In order to verify the seat assignment that is shown in the district-by-party matrix, the vote count of a party in a district is divided by the corresponding district divisor as well as by the corresponding party divisor, and then the resulting quotient is rounded in a standard fashion.

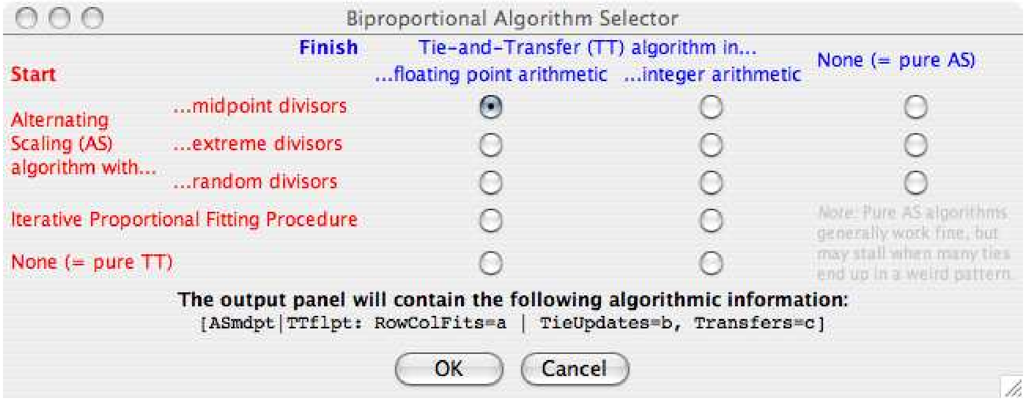
Thus the Grüne in Schwaben win 105 256 votes. The district divisor for Schwaben is 47 000, the party divisor for the Grüne is 0.964. A double division results in the quotient 2.3. Hence standard rounding assigns 2 seats, to the Grüne party in the Schwaben district. A comparison with the summary table from Subsection 2.8 shows that the apportionment by separate district evaluations differs from the biproportional apportionment only through the transfer of one seat, from the SPD to the Grüne in the Oberbayern district.

Bazi provides thirteen algorithms to calculate the sets of district divisors and party divisors. The Edit menu, in the Title bar, offers the item *Biproportional algorithms ...* where the user may select an algorithm of his or her choice. See Fig. 5.

2.10 Biproportionality with global minima

The database file *Testdaten>Biproportional problems>Diverse>Biproportionale Testdaten Italien* is an example for minimum restrictions, in a biproportional setting (Pennisi, 2006). Since a biproportional calculation actually has two steps, the super-apportionment and the sub-apportionment, Bazi provides two third-column options, `MIN(global)` and `min(local)`.

The option `MIN(global)` sets a global minimum per party, to be applied in the super-apportionment calculations. In the present example party A, the overall winner, gets 340 of the 617 seats. It is sufficient to enter the global minimum 340 in just one district; here it is district 1. If a user enters global minima in several districts, Bazi takes the largest of such entries. The



Biproportional Algorithm Selector

Start	Finish	Tie-and-Transfer (TT) algorithm in... ...floating point arithmetic	...integer arithmetic	None (= pure AS)
Alternating Scaling (AS) algorithm with...	...midpoint divisors	●	○	○
	...extreme divisors	○	○	○
	...random divisors	○	○	○
Iterative Proportional Fitting Procedure		○	○	
None (= pure TT)		○	○	

Note: Pure AS algorithms generally work fine, but may stall when many ties end up in a weird pattern.

The output panel will contain the following algorithmic information:

```
[ASmdpt|TTflpt: RowColFits=a | TieUpdates=b, Transfers=c]
```

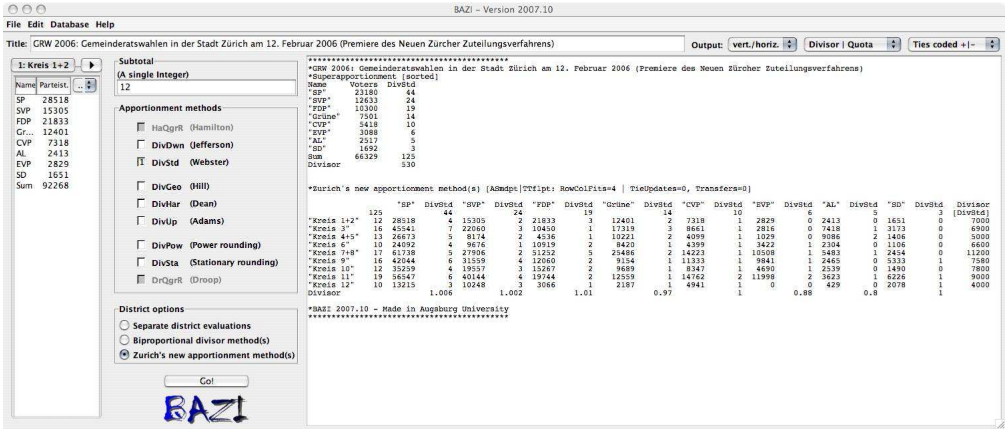
Fig. 5: Biproportional algorithms. Bazi offers three pure Alternating Scaling (AS) algorithms (far right), two pure Tie-and-Transfer (TT) algorithms (bottom), and eight hybrid algorithms with an AS start and a TT finish.

special effect of advantaging party A can also be read-off from its small divisor (0.3822) in the sub-apportionment.

When the global minima sum up to the prespecified house size, they actually predetermine the overall party seats. This is of no interest for electoral systems, but it is of definite interest for statistics. Often a contingency table must be proportionally fitted with both marginals being prespecified. The database file *Statistics>Contingency tables>BFH Table 3.6-2* provides an example. The same operational problem emerges when a given matrix is proportionally fitted to become doubly stochastic, see the database file *Statistics>Double stochastic matrices>MO A, 3 digit accuracy*.

2.11 Biproportionality with local per-district minima

The database file *Färöer Inseln>Biproportional spekulativ>2004 (ohne Partei G, mit Min-Restr.)* is an example where minimum restrictions are applied locally, in every district. The minimum restrictions shown ensure that, in every district, the party with the most votes is allocated at least one seat. When in the third column the local minimum restriction option is deactivated, none of the two seats in district NS goes to party B which, with 663 votes, is strongest in this district.



BAZI – Version 2007.10
File Edit Database Help
Title: GRW 2006: Gemeinderatswahlen in der Stadt Zürich am 12. Februar 2006 (Premiere des Neuen Zürcher Zuteilungsverfahrens)
Output: vert./horiz.
Divisor | Quota
Ties coded +|-
1: Kreis 1+2
Go!
BAZI

Name	Parteist.
SP	28518
SVP	15305
FDP	21833
Gr...	12401
CVP	7318
AL	2413
EVP	2829
SD	1651
Sum	92268

Subtotal
(A single Integer)
12

Apportionment methods
☐ HaQgrR (Hamilton)
☐ DivDwn (Jefferson)
☑ DivStd (Webster)
☐ DivGeo (Hill)
☐ DivHar (Dean)
☐ DivUp (Adams)
☐ DivPow (Power rounding)
☐ DivSta (Stationary rounding)
☐ DrQgrR (Droop)

District options
○ Separate district evaluations
○ Biproportional divisor method(s)
● Zurich's new apportionment method(s)

```
******************************************
*GRW 2006: Gemeinderatswahlen in der Stadt Zürich am 12. Februar 2006 (Premiere des Neuen Zürcher Zuteilungsverfahrens)
*Superapportionment [sorted]
Name       Voters  DivStd
"SP"        23180      44
"SVP"       12633      24
"FDP"       10300      19
"Grüne"      7501      14
"CVP"        5418      10
"EVP"        3088       6
"AL"         2517       5
"SD"         1692       3
Sum         66329     125
Divisor               530


*Zurich's new apportionment method(s) [ASmdpt|TTflpt: RowColFits=4 | TieUpdates=0, Transfers=0]

                    "SP"  DivStd   "SVP"  DivStd   "FDP"  DivStd "Grüne"  DivStd   "CVP"  DivStd   "EVP"  DivStd    "AL"  DivStd    "SD"  DivStd    Divisor
             125                44              24              19              14              10               6               5               3   [DivStd]
"Kreis 1+2"   12   28518       4   15305       2   21833       3   12401       2    7318       1    2829       0    2413       0    1651       0       7000
"Kreis 3"     16   45541       7   22060       3   10450       1   17319       3    8661       1    2816       0    7418       1    3173       0       6900
"Kreis 4+5"   13   26673       5    8174       2    4536       1   10221       2    4099       1    1029       0    9086       2    1406       0       5000
"Kreis 6"     10   24092       4    9676       1   10919       2    8420       1    4399       1    3422       1    2304       0    1106       0       6600
"Kreis 7+8"   17   61738       5   27906       2   51252       5   25486       2   14223       1   10508       1    5483       1    2454       0      11200
"Kreis 9"     16   42044       6   31559       4   12060       2    9154       1   11333       1    9841       1    2465       0    5333       1       7580
"Kreis 10"    12   35259       4   19557       3   15267       2    9689       1    8347       1    4690       1    2539       0    1490       0       7800
"Kreis 11"    19   56547       6   40144       4   19744       2   12559       1   14762       2   11998       2    3623       1    6226       1       9000
"Kreis 12"    10   13215       3   10248       3    3066       1    2187       1    4941       1       0       0     429       0    2078       1       4000
Divisor                  1.006           1.002            1.01            0.97               1            0.88             0.8               1

*BAZI 2007.10 - Made in Augsburg University
******************************************
```

Fig. 6: Zürich's new apportionment procedure. The example shows the seat apportionment for the 2006 Zürich City Council, with the super-apportionment based on voter numbers, the sub-apportionment, on ballot counts.

2.12 Zürich's new apportionment procedure

The database file *Schweiz>Zürich>Gemeinderatswahlen>2006* is an instance of Zürich's new apportionment procedure. In fact, this is the worldwide first use of a biproportional system. See Fig. 6.

In Zürich, voters are granted as many ballots as there are seats to fill in their districts. Hence the number of ballots of a voter varies between districts. If the apportionment calculations were based on ballot numbers, then they would grant unequal weights to voters from different districts, thus violating the electoral equality principle. Therefore ballots are first related to voters by calculating voter numbers, defined by dividing the district magnitude into the ballot counts and rounding the resulting quotient to the nearest integer. The district voter numbers are the appropriate quantities to be pooled across the whole Canton, thus generating the input data for the super-apportionment.

Then, in a second calculation, the sub-apportionment is carried out in such a way that each district exhausts its district seats, and each party exhausts its overall party seats. The key numbers that Bazi must calculate are the party divisors and the district divisors. With these divisors being printed into the margins of the output table, the apportionment is easily verified using paper and pencil. The vote counts are divided by the corresponding party divisor and by the corresponding district divisor, and the resulting quotient is rounded to the nearest integer. For further details see Pukelsheim and Schuhmacher (2004); Balinski and Pukelsheim (2006); Pukelsheim (2006).

3 Bazi's graphical user interface

In this section we review Bazi's functionality in some greater detail. Bazi's graphical user interface features a menu bar and a title bar, and three panels: the input panel, the methods panel, and the output panel. See Fig. 2.

3.1 Menu bar

The menu bar contains the File menu, the Edit menu, the Database, and Help, which we discuss one after the other.

3.1.1 File menu

Open input allows the user to import a file in the `.bazi`-format. A simple way to acquaint oneself with the format is to read-in a file from the data base, or to key-in some data that one is interested in, and then export the input using the next option *Save input*. This will generate a primitive text file in the `.bazi`-format, which the user then can edit, or use as a template. A sample input file is provided in Subsection 3.6.

Save input exports the data from the input panel and the methods panel, and writes it into a file using the `.bazi`-format.

Save output exports the current state of the output panel into a file. The file is created in the primitive text format, which can be read and edited with any Ascii capable editor. Since users usually have their individual formating standards and preferences, we have decided to keep the formating features in the output panel to an absolute minimum.

Restart clears all panels and registers (except for the biproportional algorithm that may be selected in the *Edit* menu). The *Restart* command is repeated in the edit menu.

Exit terminates all Bazi operations.

3.1.2 Edit menu

Delete marked input line is a command that operates in the input panel. When data have been keyed-in into the input panel, individual lines may be erased by first marking a line, and then executing the present command.

Delete input erases everything in the input panel and in the methods panel, and puts the two panels back into the initial state.

Delete output erases what has been written into the output panel, leaving a clean output field.

Districts... is for use in situations where the electoral region is subdivided into various districts, and seats are assigned to districts prior to carrying out the apportionment calculation. The command opens a new window where the names of the districts may be entered, together with the district magnitudes. See Fig. 4.

Districts are assigned a running number that cannot be edited by the user. Districts also have a district name that is editable, to overwrite the default. In the third column, the user must enter the district magnitudes, that is, the number of seats to be apportioned in each district. When the window is left by pushing the OK button on the bottom, the input panel sets up as many cards as districts have been defined.

Biproportional algorithms... offers a choice of thirteen algorithms to calculate biproportional apportionment solutions. See Fig. 5.

3.1.3 Database

The database directory comprises a considerable collection of empirical data provided in the `.bazi`-format, ready for input into Bazi. When one of these data sets is imported into Bazi, an associated *Information* window is opened containing some pertinent information for that particular data set.

Most empirical data sets are such that the apportionment method that has actually been used is activated. However, for the purpose of comparison there may be other apportionment methods activated, too. For instance, in Subsection 2.1 six methods are checked, with the intention to reproduce the tables on pages 158–180 in Balinski and Young (2001).

The last entry of the database menu is *Testdaten*, with artificial data that have been constructed to test Bazi, to scrutinize apportionment methods, or to respond to other academic questions.

3.1.4 Help

Direct help grabs a question-mark that may be dropped anywhere on the Bazi window to provide the user with some informative tool tips. As of now, the feature has been implemented only partially.

About Bazi acknowledges the contributions of various student generations and colleagues who have contributed to the development of Bazi. The GNU General Public License, under which Bazi is made available to the public, is also included.

3.2 Title bar

The second line of the header field is the *Title bar* into which the user may enter any text deemed appropriate to serve as a title.

The *Output* group in the right half of the title bar offers a limited choice to specify how Bazi is supposed to write the results into the output panel. The first selector (*horizontal*, etc.) refers to the display of the parties. The second selector (*Divisor*, etc.) stipulates the amount of technical information concerning divisors, or divisor intervals. Sometimes inverse quantities are easier to deal with, whence the user may alternatively ask for information on multipliers, or multiplier intervals. The third selector (*Ties coded* +−, etc.) defines the display of ties in the output panel.

3.3 Input panel

The input panel consists of three columns, of which the first is initially labeled `Name`, the second, `Votes`. The user can enter his or her own labels, by double-clicking into the corresponding cells.

The third column offers a selection of restrictions that the user may want to impose on the apportionment method. When clicking on the drop-down arrow in the header of the third column, the user is offered the options `---, Dir, Min, Max, Min..Max, Sub`. The feasible restrictions offered in the third column depend on the context. When multiple districts are present, some of the options are suppressed. The three dashes `---` indicate that the column is inactive.

Whenever the third column is active, the output will contain an additional column to monitor the activated restriction. The option `Dir` keeps track of direct seats in two-ballots systems, such as the one for the German Bundestag.

Option `Min` imposes minimum restrictions on the seat apportionment, option `Max` is for maximum restrictions, and option `Min..Max` specifies ranges of seats. The last option `Sub` subtracts from the numbers in the third column the apportionment result; the difference, sometimes called “discrepancy”, is occasionally useful for comparative investigations.

The first column, labeled `Name`, generally starts out with default party names, such as `=1=` etc., for each row. Any party name may be preceded by a plus-operator, to generate list combinations. The members that form a list combination must be entered in lines next to each other, and each line must start with a plus-operator, except for the very first line of the group. For instance, a list combination of parties `=1=` and `=2=` and `=3=` must be entered as `=1=`, followed by `+=2=` and `+=3=`.

3.4 Methods panel

The middle column constitutes the methods panel. Three boxes are arranged below one another, the house size box, the apportionment methods box, and the district options box.

3.4.1 House size box

The top box features the house size and is labeled *Total*. When the district option is active, the label is *Subtotal*, and is usually called the district magnitude. Either way, these are the number of seats (more generally: discrete units, accuracy) to be apportioned.

For entering a list of house sizes, use a semi-colon ";" as list separator. Moreover, upward ranges are allowed, such as 80..85. Downward ranges are also allowed, such as 95..90. Downward ranges are occasionally convenient to study the behavior of the divisor, which will then increase.

When the user asks for ranges, Bazi prints out a table with one line (or column) for every house size in the range specified.

3.4.2 Apportionment methods box

The middle box lists the apportionment methods that Bazi offers. The methods are numbered and executed in the sequence in which they are checkmarked by the user. The majority of the methods are divisor methods, except for the first and the last, which are quota methods. It is one of the key messages of the Balinski and Young (2001) monograph that divisor methods are vastly superior to quota methods.

All divisor methods comprise two steps, the scaling step, and the rounding step. In the scaling step, the input numbers are scaled by a common divisor. A neat, appealing choice for such a divisor is printed into the output panel. Different divisor methods use different rounding rules. Contingent on the rounding rule, the divisor is adjusted to exhaust the prespecified house size.

Balinski and Young (2001) emphasize five "traditional" divisor methods, which they refer to through the names of the authoritative personalities who originally contributed to the method, see Pukelsheim (2002). In order to save space, Bazi uses acronyms:

- *DivDwn:* The "divisor method with rounding down", associated with the names of Jefferson, D'Hondt, Hagenbach-Bischoff.
- *DivStd:* The "divisor method with standard rounding", associated with the names of Webster, Sainte-Laguë, Schepers.

- *DivGeo:* The "divisor method with geometric rounding", named after Hill and Huntington, and also called the Equal Proportions method.
- *DivHar:* The "divisor method with harmonic rounding", named after Dean.
- *DivUp:* The "divisor method with rounding up", named after Adams.

Furthermore, there are two important one-parameter families of divisor methods. The first is the power-mean family. When the box *DivPow* is checked, the user is prompted to key-in the power-mean parameter, that is, any value strictly between $-\infty$ and $+\infty$. Power-mean divisor methods with parameter $0 \neq p \in \mathbb{R}$ are based on the signposts (Sprungstellen, seuils)

$$s_p(n) = \left(\frac{(n-1)^p + n^p}{2} \right)^{1/p} \in [n-1, n].$$

When the vote counts are divided by the divisor and the resulting quotient comes to lie in the interval $[n-1, n]$, the quotient is rounded down to $n-1$ when it falls below the signpost. It is rounded up to n when it comes to lie above the signpost.

The power-mean family comprises the five traditional divisor methods. For $p = 1$ we get the divisor method with standard rounding, *DivStd*, for $p = -1$, the divior method with harmonic rounding, *DivHar*. The limiting signposts, as $p \to 0$, are the geometric means of the interval endpoints and hence corresponds to the divisor method with geometric rounding, *DivGeo*. Under the name method of *equal proportions (EP)*, it is used for the apportionment of the seats of the US-House of Representatives to the States of the Union. The limits, as $p \to -\infty$, are the left endpoints, belonging to the divisor method with rounding up, *DivUp*. Similarly the limits $p \to +\infty$ generate the divisor method with rounding down, *DivDwn*.

The second family, *DivSta*, comprises the divisor methods with stationary rounding. Here the user is prompted to input a stationarity parameter, that is, a number between 0 and 1. Stationary divisor methods with parameter $q \in [0, 1]$ are based on the signposts $s_q(n) = n-1+q$. The choices $q = 0, 1/2, 1$ belong to the divisor method with rounding up, with standard rounding and with rounding down, respectively. A unorthodox choice is $q = 0.4$, proposed by Condorcet.

For divisor methods Bazi implements an algorithm that enjoys a minimum computational complexity, see Happacher and Pukelsheim (1996); Dorfleitner and Klein (1999).

The two quota methods supported by Bazi are the Hare-quota method (Hamilton/Hare/Niemeyer), *HaQgrR*, and the Droop-quota method(Droop), *DrQgrR*, both with residual fit by greatest remainders. The Hare-quota is the (unrounded) quotient of the vote total and the house size. The Droop-quota equals 1 plus the integer part, of the vote total divided by 1 plus house size. Hence the Droop-quota always is a positive integer. Bazi prints the value of the quota into the output panel.

3.4.3 District options box

The bottom box of the methods panel is the district options box. The box is activated either when the Bazi input file has its data arranged in districts, or else through the *Districts...* dialogue in the edit menu.

Separate district evaluations means that each district is considered on its own. For instance, when there are twelve districts, Bazi performs twelve separate seat apportionments. As a courtesy to the user, Bazi prints a summary table at the end of the output.

Biproportional divisor methods treat the input data in its entirety, as a matrix with districts as rows and parties as columns. For further details see Balinski and Demange (1989); Balinski and Pukelsheim (2006); Maier (2006); Zachariasen (2006); Rote and Zachariasen (2007).

Zürich's new apportionment preocedure first transforms the ballot counts into "voter numbers" (Wählerzahlen). Then the selected divisor method is applied to carry out a biproportional apportionment, see Pukelsheim and Schuhmacher (2004); Pukelsheim (2006).

3.4.4 The *Go!* button

The footer of the method panel offers the *Go!* button which tells Bazi to carry out its calculations. This is the only way to make Bazi do its job.

3.5 Output panel

Bazi writes its results into the output panel. From there, the output may be exported into a file by calling the command *Save output* in the File menu.

Alternatively, the user may mark the output by pressing control-A, or by using the mouse, then copy the marked area into the computer's clip board by pressing control-C, and finally retrieving the clip board material and pasting it into a file of the user's choice, often by pressing control-V or otherwise with the appropriate editor command. This functionality is available only when the user first clicks with the mouse into the output panel, so that it gets activated.

3.6 A sample input file

For repeated use of the Bazi program it is convenient to store the data in a `.bazi` file. The database file that generates the output in Fig. 2 may serve as a generic example.

```
=TITLE=     US House of Representatives, 1st Census 1790
=OUTPUT=    vertical, D|Q, codesonly
=SEATS=     105
=METHODS=   DivUp, DivHar, DivGeo, DivStd, DivDwn, HaQgrR
=INPUT=     State, Apportionment Population, Min
=DATA=      "VA" 630560 1
            "MA" 475327 1
            "PA" 432879 1
            "NC" 353523 1
            "NY" 331589 1
            "MD" 278514 1
            "CT" 236841 1
            "SC" 206236 1
            "NJ" 179570 1
            "NH" 141822 1
            "VT"  85533 1
            "GA"  70835 1
            "KY"  68705 1
            "RI"  68446 1
            "DE"  55540 1
=INFO=
Source:     Michel Balinski / Peyton Young:
            Fair Representation.  New Haven CT, 1982.
            Second edition:  Washington DC, 2001.
US Postal Service two-letter State abbreviations:
            1787 DE Delaware
            1787 NJ New Jersey
            1787 PA Pennylvania
            1788 CT Conneticut
            1788 GA Georgia
            1788 MA Massachusetts
            1788 MD Maryland
            1788 NH New Hampshire
            1788 NY New York
            1788 SC South Carolina
            1788 VA Virginia
            1789 NC North Carolina
            1790 RI Rhode Island
            1791 VT Vermont
            1792 KY Kentucky
=END=
```

Acknowledgement: Financial support of the Deutsche Forschungsgesellschaft is acknowledged. We are grateful for the valuable assistance of our chief programmer, Christian Brand. We highly appreciate the feedback on an earlier version of this paper that was provided by Daniel Bochsler, Norman R. Draper, Antonio Palomares, Christian Pukelsheim, Victoriano Ramírez González, Antony Unwin, and Martin Zachariasen.

References

Balinski, Michel and Young, Peyton (2001). *Fair Representation. Meeting the Ideal of One Man, One Vote. Second Edition.* Brookings Institution Press: Washington, DC.

Balinski, Michel (2004). *Le suffrage universel inachevé.* Belin: Paris.

Balinski, Michel and Demange, Gabrielle (1989). Algorithms for proportional matrices in reals and integers. *Mathematical Programming*, 45:193–210.

Balinski, Michel and Pukelsheim, Friedrich (2006). Matrices and politics. In Liski, E., J. Isotalo, S. Puntanen, and Styan, G. P. H., editors, *Festschrift for Tarmo Pukkila on His 60th Birthday.* Department of Mathematics, Statistics, and Philosophy: Tampere.

Dorfleitner, Gregor and Klein, Thomas (1999). Rounding with multiplier methods: An efficient algorithm and applications in statistics. *Statistical Papers*, 20:143–157.

Happacher, Max and Pukelsheim, Friedrich (1996). Rounding probabilities: Unbiased multipliers. *Statistics & Decision*, 14:373–382.

Kopfermann, Klaus (1991). *Mathematische Aspekte der Wahlverfahren. Mandatszuteilung bei Abstimmungen.* Wissenschaftsverlag: Mannheim.

Maier, Sebastian (2006). Algorithms for biproportional apportionment. In Simeone and Pukelsheim (2006), pages 105–116.

Pennisi, Aline (2006). The Italian bug: A flawed procedure for biproportional seat allocation. In Simeone and Pukelsheim (2006), pages 151–164.

Pólya, Georg (1918). Über die Verteilungssysteme der Proportionalwahl. *Zeitschrift für die schweizerische Statistik und Volkswirtschaft*, 54:363–387.

Pukelsheim, Friedrich (2002). Die drei in Deutschland verwendeten Mandatszuteilungsmethoden und ihre Namenspatrone. `www.uni-augsburg.de/pukelsheim/2002g.html`.

Pukelsheim, Friedrich (2006). Current issues of apportionment methods. In Simeone and Pukelsheim (2006), pages 167–176.

Pukelsheim, Friedrich (2007). A parliament of degressive representativeness? Technical Report Preprint Nr. 15/2007, Institut für Mathematik, Universität Augsburg. `www.opus-bayern.de/uni-augsburg/volltexte/2007/624`.

Pukelsheim, Friedrich and Maier, Sebastian (2006). A gentle majority clause for the apportionment of committee seats. In Simeone and Pukelsheim (2006), pages 177–188.

Pukelsheim, Friedrich and Schuhmacher, Christian (2004). Das neue Zürcher Zuteilungsverfahren bei Parlamentswahlen. *Aktuelle Juristische Praxis - Pratique Juridique Actuelle*, 13:505–522.

Rote, Günter and Zachariasen, Martin (2007). Matrix scaling by network flow. In *Proceedings of the 18th Annual ACM-SIAM Symposium on Disc rete Altorithms (SODA), New Orelans, January 2007*, pages 848–854.

Siaroff, Alan (2000). British AMS versus German Personalized PR: Not so different. *Representation*, 37:19–28.

Simeone, Bruno and Pukelsheim, Friedrich, editors (2006). *Mathematics and Democracy. Recent Advances in Voting Systems and Collective Choice.* Springer: New York.

Zachariasen, Martin (2006). Algorithmic aspects of divisor-based biproportional rounding. Technical Report 06-05, DIKU, Department of Computer Science, University of Copenhagen. `www.diku.dk/publikationer/tekniske.rapporter/rapporter/06-05.pdf`.