Short Description of the Seating Algorithm

Raul Persa, Lukas Vogel

November 15, 2015

To guarantee the best possible performance we decided against an iterative method for seat assignment and used the method illustrated in http://www.wahlrecht.de/bundestag/index.htm.

The algorithm is fully implemented in SQL and does not depend on any application logic. Parts that cannot be expressed by pure SQL-Queries are implemented as plpgsql-functions. This guarantees a clean interface between the database and the application layer.

First the algorithm generates views for a variation of sub-tasks like determining:

- The winners of a direct mandate `directmandate_winners`
- The parties eligible for seats in the Bundestag `parties_in_bundestag`
- A Bundesdivisor, specifying the amount of votes needed to gain a seat in the Bundestag `bundesdivisor`
- The amount of seats for each party in the Bundestag `total_num_seats` calculated with the Sainte-Lague method and the Bundesdivisor.

A plpgsql-function `find_partydivisor` algorithmically determines a party divisor for every party to distribute the seats back to each bundesland and the correspondent landesliste. A binary search is used for performance reasons.

The view `members_of_bundestag` finally specifies all members of the Bundestag. Those are:

- All `directmandate_winners`
- All candidates (that have not already won via direct mandate) specified by `remaining_cand_on_ll`. They are a member if they are on a landesliste with a listenplatz lower than the number of remaining seats for a party in a bundesland.

The algorithm needs about 200ms to calculate the whole Bundstag-composition on commodity hardware (Laptop with Intel i5-4210U).