Rounding Considered Harmful

Carsten Schürmann[⋆]

DemTech
IT University of Copenhagen
Denmark
carsten@itu.dk

Abstract. Party-list proportional representation methods aim to allocate seats proportionally to the votes cast for each party. In general, exact proportionality is not possible as it would require a fractional allocation of seats. Therefore several methods have been devised to compute seat allocations that differ in the way they try to achieve proportionality. Examples of such methods include the d'Hondt and Sainte-Laguë methods that allocate seats according to fractional values. These methods are used in many countries. Numerically, these fractions appear harmless, however they are not. Computers do not work with infinite precision floating point numbers, implementations tend to round the fractions to several digits, which can, with a certain probability, lead to incorrect seat allocations.

1 Introduction

Denmark's electoral law requires a combination of d'Hondt and Sainte-Laguë methods to compute the seat allocation of parliament after a parliamentary election. These methods aim at producing a seat allocation that is proportional and reflects the vote. The methods are deceptively simple as they consist of computing a table of quotients and the selection of the largest those quotients, each of which corresponds to a seat.

However, whenever quotients are computed, one has to be careful with bad numeric effects. Floating point numbers are not represented with infinite precision, and often the digits after the comma are rounded off. In this paper we ask the question if it safe to round quotients, and if rounding can have an effect in real elections. The answer is that it is not safe to round and that rounding can change the outcome of the election (with a certain probability).

For conducting this work, we have been granted access the source code of Denmark's Seat allocation System (DSAS). While inspecting the software, we observed that DSAS rounds quotients to the next whole number before storing

[⋆] This paper was made possible by grant NPRP 097-988-1-178, Automated verification of properties of concurrent, distributed and parallel specifications with applications to computer security, from the Qatar National Research Fund (a member of the Qatar Foundation). The statements made herein are solely the responsibility of the author.

then in the table, and then uses randomness to draw lots to break ties. In the analysis described in this paper, we show that this is highly problematic and that situation can arise where DSAS computes the wrong result. We also conduct a Monte-Carlo experiment to provide some statistical evidence, how likely the error scenario actually is in practice, at least for the Danish case. We estimate that one out of every 66 Danish elections is effected and that DSAS computes the wrong seat allocation in average for one out of 132 elections. The version of DSAS under consideration in this paper was used only since 2007. Seat allocation for elections before 2007 were computed with another system. We have not conducted any further statistical analysis for other countries.

We have written this paper out of concern that similar programming mistakes may be hiding in seat allocations programs used by other countries.

This paper is organized as follows. In Section 2 we describe abstractly the d'Hondt method, variants of which are used in dozens countries around the world. In Section 3, we describe the effects of rounding and define the probability of how prematurely rounding quotients can effect the result of the seat allocation algorithm. In Section 4 we then look at real election data from the Danish 2007 parliament election, and argue, that the effects of rounding are likely and serious. We also provide evidence that the Danish Seat Allocation System, at least in the version examined, rounds quotients to the next whole number and breaks ties by drawing lots. In Section 5 we describe briefly how we disclosed the findings to the Ministry, before we we assess results and describe how a fix for the rounding problem in Section 6.

Related Work: The subject of this paper touches on several different areas: Party-list proportional representation methods are social choice functions, and their properties have been studied in political science and social choice [1, 5]. The d'Hondt was originally invented by Thomas Jefferson in 1791, and then introduced to Europe by Victor d'Hondt in 1878. The method was designed to be executed by hand. Nowadays, social choice functions such as d'Hondt, single transferable vote (STV) and others are implemented as computer programs. Programming mistakes are common place unless one uses formal methods to verify the implementation of tie breaking rules, which not many implementors do. One of the few works in this area is by Goré and Lebedeva [4], which focuses on verifying implementations of STV and the respective tie-breaking rules. Their reasoning techniques applied to d'Hondt (assuming it is correctly specified) would then automatically recognize programming mistakes such as those that we discuss in this paper. Lastly, even if premature rounding is used in the implementation of d'Hondt and Sainte-Laguë methods, variants of risk-limiting auditing tailored for d'Hondt elections [7] can be used to identify statistically, if seats have been erroneously assigned to the wrong party.

2 The d'Hondt Method

The d'Hondt and Sainte-Laguë methods are a part-list proportional representation methods used for seat allocation in more than 50 countries including

Denmark, Germany, and Switzerland. In the following, we focus on d'Hondt. It is defined as follows. Let $t_1 \dots t_n$ represent the vote totals of an election with n parties. The total number of votes cast is therefore $\sum_{1 \le i \le n} t_i$, and the goal of the d'Hondt method is to assign the m seats in such a way that they are proportionally allocated in the the number of votes obtained by each party: As there are no fractional seats, we cannot expect d'Hondt or any other voting rule for this matter to produce the perfectly proportional seat allocation. The underlying idea of d'Hondt is this: If a party were to pay for a seat with votes then then d'Hondt rule allocates the number of seats to each party to maximize the highest (average) price per seat.

Example 1. Let A and B be two parties with 10000 and 15000 votes, respectively, and five seats to be allocated.

	A's bid	B's bid	Allocation
(1)	$10,000$ for 1 seat	$15,000$ for 1 seat	B
(2)	$10,000$ for 1 seat	$15,000$ for 1 seat	B, A
(3)	$10,000$ for 1 seat	$7,500$ for 2 seats	B, A, B
(4)	$5,000$ for 2 seats	$7,500$ for 2 seats	B, A, B, A
(5)	$5,000$ for 2 seats	$5,000$ for 3 seats	B, A, B, A, B

The algorithm starts with highest price, here $15,000$ votes, and reduces the price until all seats are sold! In line (1), the first seat goes to party B, because B is bidding $15,000$. In line (2), A obtains the second seat, because at this stage the price is $10,000$. B cannot bid, because B has spent all of its money. In line (3), the average price for a seat has dropped to $7,500$, which allows B to argue that it should be entitled to a second seat. (B has already spent $15,000$ and $2 \times 7,500 = 15,000$). After 5 rounds, all seats are sold.

The d'Hondt rule results in a simple algorithm that consists of two steps.

1. Construct a table, one column per party, where the first row are initialized with the vote totals $t_1 \dots t_n$. All other rows are identified by a divisor, and the row is computed from the first row by dividing t_i by this divisor. The entries in the table are also called *quotients*. In the simplest case, the divisors range over $1, 2, 3, \dots$, but other choices of divisors are used in practice as well, as we will see in Section 4.
2. To allocate s seats, traverse the table and mark the s highest quotients. The number of markings in each column correspond to the seats assigned for the respective party.

The intended meaning of the table is that the field located at row i and row j is the bidding price for party i for j seats. Note, that if a quotient is marked in a table, all the quotients above (in the same column) are also marked.

Example 2. Back to the example above. The table in this case has the following form. The markings are indicated as check marks.

Divisor	Party A	Party B
1	10000 ✓	15000 ✓
2	5000 ✓	7500 ✓
3	$3333.\bar{3}$	5000 ✓
4	2500	3750
5	2000	3000

3 To Round or Not to Round?

Although computing the table is not difficult there are some decision designs that have to be taken when implementing it. The most important perhaps is if and how to round quotients that are not whole numbers. The quotient in Row 3, Party A, for example, is a number with infinitely many digits after the comma. Would it be ok to round these quotients to the nearest whole number? By rounding to the next whole number we mean that if the digits after the "," < 0.5 the number will be rounded down and if ≥ 0.5 it will be rounded up. One may expect the answer is yes, after all, the differences between the tallies of a typical national election are usually quite large, so what could go wrong? Rounding must be considered problematic, if it affects the result of seat allocation, which means that the margin between two quotients in the d'Hondt table are sufficiently close, and this means introducing a tie that has to be resolved by a tie-breaking rule.

Example 3. Consider, for example, a multi-member constituency where three parties A, B and C with tallies 999, 500, 1501, respectively, compete for four constituency seats. Selecting the four highest quotients from the table below

	A	B	C
Tallies	999	500	1501
Divisor 1	999	500	1501
Divisor 2	499.5	250	750.5
Divisor 3	333	$166.\bar{6}$	$500.\bar{3}$

results in the following correct election result: one seat is allocated to A, three seats are allocated to C. In the case with rounding to the next whole number, the table has the following form:

	A	B	C
Tallies	999	500	1501
Divisor 1	999	500	1501
Divisor 2	500	250	750
Divisor 3	333	167	500

The algorithm allocates the first three seats to C, A, and C, and the last seat is drawn by lot, and as each party has a quotient of 500, the probability that the correct result is drawn is only 1/3.

If we were to allocate five seats instead of four, the correct result implies that the fifth seat belongs to B, because $500 > 499.5$. In the rounded version, however, we would have to draw two seats of the set of three, which means here again, the probability that the correct result is only $1/3$.

In real elections there are several factors that impact the margins of the quotient registered in the table: (1) In some countries, d'Hondt is applied not only to the nation-wide totals, but often also on the constituency level, where the tallies are much smaller, which means the likelihood that two quotient are close increases. (2) The choice of divisors varies form country to country. In some countries, for example in Denmark, under certain but rare circumstances, the divisors $1, 4, 7\ldots$ are being used to construct the d'Hondt table, which means that for large divisors, the quotients, can become quiet close. (3) The size of the elected body plays a big role in how small the margins are between any to quotients in the table.

We distinguish to kinds of errors due to rounding, depending on if they affect allocated seats alone, or allocated and non-allocated seats. The former is called *Incorrect Allocation Order* is relatively harmless, because it does not affect the overall election result, but only the evidence of how the election result was determined. In some countries already this might be considered an infringement of the law. Things become much more worrisome, when rounding creates an artificial tie situation affecting the last seat(s) to be awarded. In this case, the drawing of the lots may in fact the overall election result. This error is called *Incorrect Seat Allocation.*

To make our analysis precise, we need to distinguish between two n-way tie situations: We say an n-way tie between n quotients is *genuine*, if the quotients (before rounding) are all equal. In the case that the quotients (before rounding) are not equal, but they are equal after rounding, we speak of a *false* n-way tie situation. A genuine tie must be broken by drawing lots, whereas a false tie must not. For the following two definitions, we assume a false tie situation.

Error 1: Incorrect Allocation Order In the case that the number of seats to be allocated exceeds the number of k quotients rounded to the same number, drawing lots may affect the order in which the seats are allocated, but it does not effect the overall election result. For a false tie, the probability p of allocating seats in the wrong order is

$$p = \frac{1}{k!}.$$

Error 2: Incorrect Seat Allocation In the case that the number of quotients rounded to the same number exceeds the number of seats to be allocated, lots will have to be drawn, which means that seats may be allocated in error with non-negligible probability. More precisely, if there is a false tie between n rounded quotients and only $k(< n)$ seats are left to be allocated, the probability p that the result of seat allocation is correct result is precisely

$$p = \frac{1}{\binom{n}{k}}.$$

4 Case Study Denmark

In order to learn if this a real or just an academic problem, we describe a case study that we conducted in Denmark. We discuss the Danish legal framework in Section 4.1, empirical evidence that Errors 1 and 2 could have actually been encountered in Section 4.2, and a discussion of the implementation of Denmark's seat allocation system, in Section 4.3 where we demonstrate that the system actually rounds quotients.

4.1 Legal Framework

Danish election law[1] defines the rules for how mandates are to be distributed. The law distinguishes two kinds of seats, constituency seats *kredsmandater* and compensatory seats *tillægsmandater*. In this report, we focus mostly only compensatory seats, but our findings also apply to the calculation of constituency seats. We quote the relevant sections from the Danish Parliamentary Elections Act [3].

Allocation of Constituency Seats

> §76. (1) The votes cast for each party in all nomination districts in a multi-member constituency shall be summed up. The votes cast for each individual candidate shall equally be summed up.
>
> (2) Each number of votes appearing as a result of the summation, cf. subsection (1), shall be divided by 1 - 2 - 3 and so on until such number of divisions equivalent to the maximum number of seats expected to be allocated to the party or to the independent candidate has been performed. The party or the independent candidate having the highest resulting quotients shall be given the first seat in the multi-member constituency. The second highest quotient entails the second seat and so on and so forth, until all constituency seats in the multi-member constituency have been distributed among the parties and the independent candidates. If two or more quotients are of equal size, lots shall be drawn.

Allocation of Compensatory Seats to Parties by Region

> §78.(1) For each of the parties which are allocated compensatory seats according to section 77, the number of votes cast for the party in each of the three regions shall be computed.
>
> (2) Each of these votes shall be divided by the figures 1 - 3 - 5 - 7 and so on. Next, a number of the largest quotients equivalent to the number of constituency seats obtained by the party in the region according to section 76 shall be omitted.

[1] LBK nr 416 af 12/05/2016

(3) The region and the party which subsequently has the largest quotient shall have the first compensatory seat. The region and the party which has the second largest quotient shall have the next compensatory seat and so on and so forth. Where a region or a party has obtained the number of compensatory seats it should have, cf. sections 10 and 77, the region or the party shall not be considered any further. The allocation continues for the other regions and the other parties until all compensatory seats have been distributed. If a party which has not received votes in all three regions cannot be allocated the compensatory seats to which the party is entitled by this distribution, these seats shall be allocated in advance to the party in the regions where votes have been cast in its favor.

This law text describes the social choice function to compute the seat allocation for the 135 constituency seats and 40 compensatory seats in the Danish Parliament, called *Folketinget*. We focus our attention on the allocation of compensatory seats, because the divisors become larger and the quotients smaller than those when allocating constituency seats. Denmark is divided into three regions, *Hovedstaden*, *Sjælland-Syddanmark*, and *Midtjylland-Nordjylland*.

The technique described in the law is a variant of d'Hondt, as described in Section 2 (see §10 (2), *den største brøks metode*). Constituency seats are assigned using divisors $1, 2, 3, \ldots$, and compensatory seats are allocated in a second step using divisors $1, 3, 5 \ldots$ In the case of a tie, lots shall be drawn. We want to emphasize, that neither the published versions of the largest remainder methods, nor the Danish election law permits that rounded versions of the quotients may be considered. This interpretations has been confirmed by the Danish Ministry.

4.2 Empirical Analysis

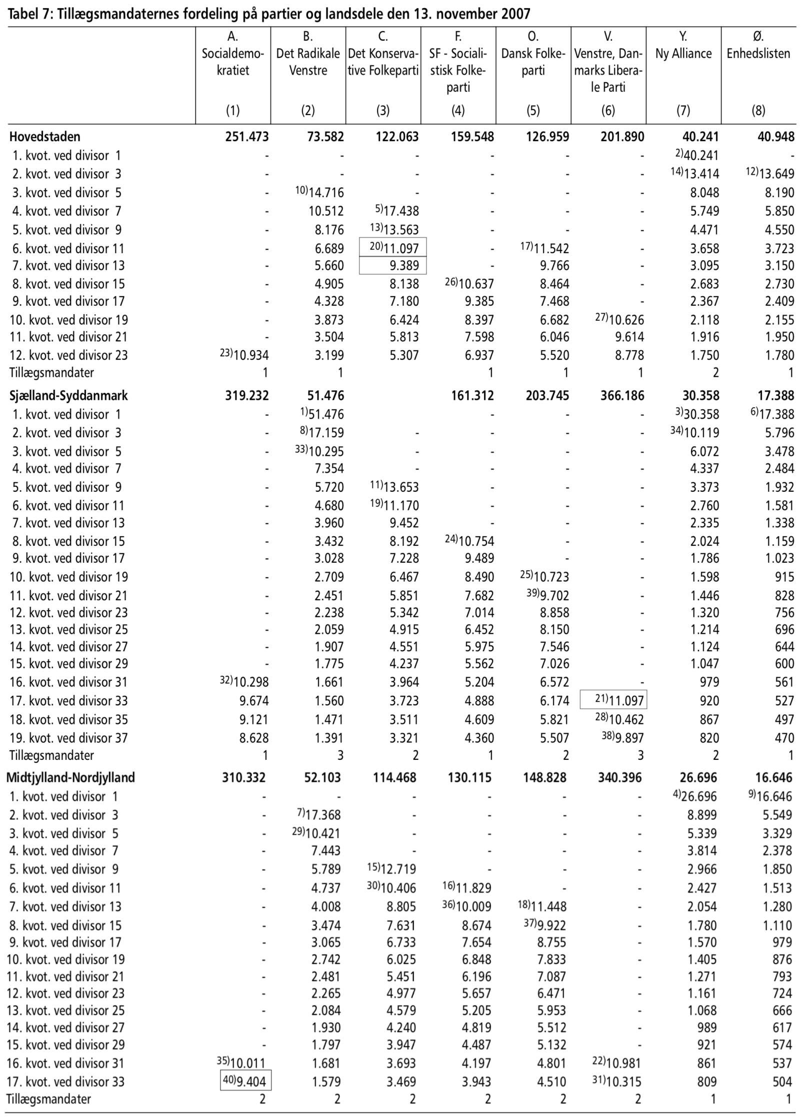
For our empirical analysis, the interesting step is 3. where the quotients are computed. Consider the seat allocation of compensatory seats 20 and 21 during Denmark's parliamentary election in 2007, as depicted in Figure 1, which has been taken from [2, page 17]. This election is relatively recent and it was chosen, because it demonstrates that Error 1 and 2 do actually arise in real elections. Both seats were awarded with a rounded quotient of $11,097$. The seat 20 was awarded to *Det Konservative Folkeparti*, because

$$\frac{122063}{11} = 11,096.6363636$$

and seat 21 was awarded to *Venstre* because

$$\frac{366186}{33} = 11,096.5454545.$$

In Figure 1, we have marked the important entries using boxes to help the reader identify these more easily.



Tabel 7: Tillægsmandaternes fordeling på partier og landsdele den 13. november 2007

	A. Socialdemokratiet (1)	B. Det Radikale Venstre (2)	C. Det Konservative Folkeparti (3)	F. SF - Socialistisk Folkeparti (4)	O. Dansk Folkeparti (5)	V. Venstre, Danmarks Liberale Parti (6)	Y. Ny Alliance (7)	Ø. Enhedslisten (8)
Hovedstaden	**251.473**	**73.582**	**122.063**	**159.548**	**126.959**	**201.890**	**40.241**	**40.948**
1. kvot. ved divisor 1	-	-	-	-	-	-	$^{2)}$40.241	-
2. kvot. ved divisor 3	-	-	-	-	-	-	$^{14)}$13.414	$^{12)}$13.649
3. kvot. ved divisor 5	-	$^{10)}$14.716	-	-	-	-	8.048	8.190
4. kvot. ved divisor 7	-	10.512	$^{5)}$17.438	-	-	-	5.749	5.850
5. kvot. ved divisor 9	-	8.176	$^{13)}$13.563	-	-	-	4.471	4.550
6. kvot. ved divisor 11	-	6.689	$\boxed{^{20)}11.097}$	-	$^{17)}$11.542	-	3.658	3.723
7. kvot. ved divisor 13	-	5.660	$\boxed{9.389}$	-	9.766	-	3.095	3.150
8. kvot. ved divisor 15	-	4.905	8.138	$^{26)}$10.637	8.464	-	2.683	2.730
9. kvot. ved divisor 17	-	4.328	7.180	9.385	7.468	-	2.367	2.409
10. kvot. ved divisor 19	-	3.873	6.424	8.397	6.682	$^{27)}$10.626	2.118	2.155
11. kvot. ved divisor 21	-	3.504	5.813	7.598	6.046	9.614	1.916	1.950
12. kvot. ved divisor 23	$^{23)}$10.934	3.199	5.307	6.937	5.520	8.778	1.750	1.780
Tillægsmandater	1	1		1	1	1	2	1
Sjælland-Syddanmark	**319.232**	**51.476**		**161.312**	**203.745**	**366.186**	**30.358**	**17.388**
1. kvot. ved divisor 1	-	$^{1)}$51.476		-	-	-	$^{3)}$30.358	$^{6)}$17.388
2. kvot. ved divisor 3	-	$^{8)}$17.159	-	-	-	-	$^{34)}$10.119	5.796
3. kvot. ved divisor 5	-	$^{33)}$10.295	-	-	-	-	6.072	3.478
4. kvot. ved divisor 7	-	7.354	-	-	-	-	4.337	2.484
5. kvot. ved divisor 9	-	5.720	$^{11)}$13.653	-	-	-	3.373	1.932
6. kvot. ved divisor 11	-	4.680	$^{19)}$11.170	-	-	-	2.760	1.581
7. kvot. ved divisor 13	-	3.960	9.452	-	-	-	2.335	1.338
8. kvot. ved divisor 15	-	3.432	8.192	$^{24)}$10.754	-	-	2.024	1.159
9. kvot. ved divisor 17	-	3.028	7.228	9.489	-	-	1.786	1.023
10. kvot. ved divisor 19	-	2.709	6.467	8.490	$^{25)}$10.723	-	1.598	915
11. kvot. ved divisor 21	-	2.451	5.851	7.682	$^{39)}$9.702	-	1.446	828
12. kvot. ved divisor 23	-	2.238	5.342	7.014	8.858	-	1.320	756
13. kvot. ved divisor 25	-	2.059	4.915	6.452	8.150	-	1.214	696
14. kvot. ved divisor 27	-	1.907	4.551	5.975	7.546	-	1.124	644
15. kvot. ved divisor 29	-	1.775	4.237	5.562	7.026	-	1.047	600
16. kvot. ved divisor 31	$^{32)}$10.298	1.661	3.964	5.204	6.572	-	979	561
17. kvot. ved divisor 33	9.674	1.560	3.723	4.888	6.174	$\boxed{^{21)}11.097}$	920	527
18. kvot. ved divisor 35	9.121	1.471	3.511	4.609	5.821	$^{28)}$10.462	867	497
19. kvot. ved divisor 37	8.628	1.391	3.321	4.360	5.507	$^{38)}$9.897	820	470
Tillægsmandater	1	3	2	1	2	3	2	1
Midtjylland-Nordjylland	**310.332**	**52.103**	**114.468**	**130.115**	**148.828**	**340.396**	**26.696**	**16.646**
1. kvot. ved divisor 1	-	-	-	-	-	-	$^{4)}$26.696	$^{9)}$16.646
2. kvot. ved divisor 3	-	$^{7)}$17.368	-	-	-	-	8.899	5.549
3. kvot. ved divisor 5	-	$^{29)}$10.421	-	-	-	-	5.339	3.329
4. kvot. ved divisor 7	-	7.443	-	-	-	-	3.814	2.378
5. kvot. ved divisor 9	-	5.789	$^{15)}$12.719	-	-	-	2.966	1.850
6. kvot. ved divisor 11	-	4.737	$^{30)}$10.406	$^{16)}$11.829	-	-	2.427	1.513
7. kvot. ved divisor 13	-	4.008	8.805	$^{36)}$10.009	$^{18)}$11.448	-	2.054	1.280
8. kvot. ved divisor 15	-	3.474	7.631	8.674	$^{37)}$9.922	-	1.780	1.110
9. kvot. ved divisor 17	-	3.065	6.733	7.654	8.755	-	1.570	979
10. kvot. ved divisor 19	-	2.742	6.025	6.848	7.833	-	1.405	876
11. kvot. ved divisor 21	-	2.481	5.451	6.196	7.087	-	1.271	793
12. kvot. ved divisor 23	-	2.265	4.977	5.657	6.471	-	1.161	724
13. kvot. ved divisor 25	-	2.084	4.579	5.205	5.953	-	1.068	666
14. kvot. ved divisor 27	-	1.930	4.240	4.819	5.512	-	989	617
15. kvot. ved divisor 29	-	1.797	3.947	4.487	5.132	-	921	574
16. kvot. ved divisor 31	$^{35)}$10.011	1.681	3.693	4.197	4.801	$^{22)}$10.981	861	537
17. kvot. ved divisor 33	$\boxed{^{40)}9.404}$	1.579	3.469	3.943	4.510	$^{31)}$10.315	809	504
Tillægsmandater	2	2	2	2	2	2	1	1

Fig. 1. Compensatory seat allocation.

If we were to use instead a seat allocation algorithm that rounds all quotients to the nearest whole number, these two quotients would round to $11,097$ and consequently seat 20 and seat 21 can only be allocated by drawing lots. In this situation, the following two outcomes are equally likely.

1. Seat 20 will be allocated to *Det Konservative Folkeparti* and seat 21 to *Venstre*.

2. Seat 20 will be allocated to *Venstre* and seat 21 to *Det Konservative Folkeparti*.

Both outcomes are correct, as we have discussed in Section 2, however Denmark's law does regard — strictly speaking — only the first outcome as correct and the second outcome as an instance of Error 1. Evidently, the order in which the seats were assigned was correct, perhaps because it was computed with an earlier version of seat allocation system and not the one we will discuss below.

Regarding Error 2, we notice that the last compensatory seat (Seat 40) was awarded with a quotient of exactly 9404 to the *Socialdemokratiet*. The next highest quotient in Figure 1 is is $9,389$ but what cannot be seen in the table is that there are actually two entries: One for region *Hovedstaden*, *Det Konservative Folkeparti*, divisor 13,

$$\frac{122063}{13} = 9389.46153846$$

and the other for region *Sjælland-Syddanmark*, *Venstre*, divisor 39

$$\frac{366186}{39} = 9389.38461538.$$

This means that in the hypothetical situation where the Danish Parliament had 41 compensatory seats, a rounding seat allocation system might have assigned the 41^{st} seat to the wrong party with a probability of 50%, i.e. to *Venstre* instead of *Det Konservative Folkeparti*.

The remaining question is, of course, how big of a problem Error 1 and Error 2 in practice really are. A statistical analysis proves difficult, in part because of the many random variables that need to be considered. Therefore we resort to a Monte-Carlo experiment and develop an election simulator to compute the probabilities of false n-way ties. In our experiment, where we work with 8 parties with tallies chosen at random between $20,000$ and $400,000$ votes, a situation, which pretty accurately describes the parameters of a Danish election. We then run the simulator $1,000,000$ times where we compute a d'Hondt table with 50 rows (which corresponds to a highest divisor of 101).

Error 1: The following table depicts the expected value of n-way ties occurring in a single d'Hondt table.

$n =$	1	2	3	4
E(false n-way tie)	173.17545	0.904989	0.0048	0.000022

This means that in 9 of of 10 d'Hondt tables, there will be in average one false 2-way tie that is decided by drawing lots. However, our experiment also shows that the risk of an Error 1 in the cases of false 3- and 4-way ties are extremely rare. Recall, that Error 1 will not change the election result but only the order in which seats are assigned.

Error 2: In the same statistical experiment, we look at the 175th seat being chosen. The following table depicts the number of false n-way ties observed.

$n =$	1	2	3	4
# observations	$984,089$	$15,766$	144	1

This means, that the probability of a false 2-way tie situation is roughly 1.5%, which means that in one out of 66 elections, the last seat will be awarded with drawing lots.

In summary, rounding while computing the d'Hondt and Sainte-Laguë tables should be considered harmful. We have demonstrated that there is a non-negligible chance that seats are being allocated in the wrong order and/or to the wrong party.

4.3 Implementation

The software for Denmark's Seat Allocation System (DSAS) has been implemented by Statistics Denmark (*Danmarks Statistik*), in a programming language called PL/SQL (part of the official ORACLE database distribution) used to program stored procedures. DSAS implements the problematic rounding version of the seat allocation algorithm outlined in the previous section. In a nutshell, the implementation rounds to the nearest number and then generate randomly the digits after the comma as a tie breaker. For the above example, instead of computing the quotients precisely, DSAS rounds and randomly generates after comma digits.

Fraction	Correct result	DSAS result
$\frac{122063}{11}$	$11,096.6363636$	$11,096 + r_1$
$\frac{366186}{33}$	$11,096.5454545$	$11,096 + r_2$

where $r_1, r_2 \in [0, 1)$ randomly chosen.

The problematic code can be found in the file `packages.sql`, dated 16. September 2013, 16:12. The three procedures that support our claim regarding the allocation of compensatory seats are depicted in Figure 2 that describes how quotients for compensatory seats are computed, Figure 3 that illustrates how the quotients are introduced into the main table, and Figure 4 that demonstrates how the quotients are ordered for further computation. The program for allocating the constituency seats is very similar, suffers from the same rounding problem, and can be found elsewhere in this file.

Tillaeg_landsdel_dankvot_ins In Figure 2, the loop starting in line 23158 ranges over all possible divisors, starting from 1 until the maximal number stored in variable `V_antal_divisioner`. The body of the loop consists of two steps. In the first step, a temporary table `TempTab` is defined that stores all possible quotients (in no particular order). In lines 23165–23174 it is determined what precisely is stored in `TempTab`. The two critical lines here are 23172 and 23173. In the former DSAS uses SQL's rounding function to compute `round (Antal_stemmer / V_divisor)`, which computes the quotient rounded to the nearest whole number. The string `Kvotient` tells the ORACLE database engine to name the column `Kvotient`. In the latter DSAS stores a random value using Oracle's random generator in a column called `Random_nr` as a tie breaker.

```
for I in 1 .. V_antal_divisioner loop
  if I = 1 then
    V_divisor := 1;
  else
    V_divisor := V_divisor + 2;
  end if;

  select Obj_tillaegsmandater_landsdele
              (Landsdel_id, Parti_id, Antal_stemmer,
               Antal_kredsmandater, Kvotient_nr, Kvotient, Random_nr)
    bulk collect into Temptab
    from (select Landsdel_id,
                 Parti_id,
                 Antal_stemmer,
                 Antal_kredsmandater,
                 V_divisor Kvotient_nr,
                 round (Antal_stemmer / V_divisor) Kvotient,
                 dbms_random.Normal Random_nr
            from table (P_col_tillaegsman_landsdele));

  Insert_tillaeg_landsdel_kvot (Temptab,
                                P_log_bruger_in,
                                P_term_bruger_in,
                                P_valg_id_in,
                                P_valgfase_in,
                                P_debug_in,
                                P_log_id_in);
end loop;
```

Fig. 2. Procedure `Tillaeg_landsdel_dankvot_ins`

How precisely Oracle's random generator was seeded, could not be determined. As we will see below, if two quotients are compared and if the `Kvotient` part is equal, the `Random_nr` will determine which of the two is ranked higher. In the second step, after computing all quotients, DSAS calls a function to copy the quotients from `TempTab` into the right table using a stored procedure called `Insert_tillaeg_landsdel_kvot` (see line 23176), which we discuss next. Note that the first argument to this method is `TempTab`.

```
23384  procedure Insert_tillaeg_landsdel_kvot (
...
23404    insert into Tillaeg_man_land_kvotienter (Landsdel_id,
23405                                            Parti_id,
23406                                            Antal_kredsmandater,
23407                                            Tmk_land_kvotientnr,
23408                                            Tmk_land_kvotient,
23409                                            Tmk_random_nr,
23410                                            Valgfase_kode,
23411                                            Tmk_koersel_id,
23412                                            Valg_id)
23413      select Landsdel_id,
23414             Parti_id,
23415             Antal_kredsmandater,
23416             Kvotient_nr,
23417             Kvotient,
23418             Random_nr,
23419             P_valgfase,
23420             P_log_id_in,
23421             P_valg_id_in
23422        from table (P_col_tillaegsman_landsdele);
```

Fig. 3. Procedure `Insert_tillaeg_landsdel_kvot`

Insert_tillaeg_landsdel_kvot Figure 3 depicts a procedure that simply reads all tuples from the table referred to by first argument, i.e. `TempTab`. The destination table is `Tillaeg_man_land_kvotienter`, where the attributes for the table are renamed to `Tmk_land_kvotient` (line 23408) and `Tmk_random_nr` (line 23409), respectively.

The tuples in the table `Tillaeg_man_land_kvotienter` are stored in no particular order.

Tillaeg_landsdel_hentpotkvot A fragment of the procedure that accesses and sorts the table `Tillaeg_man_land_kvotienter` is depicted in Figure 4. It illustrate how the quotients and random numbers are used for further computation (which we will not discuss here).

```
select Obj_tillaeg_ldel_til_kvvalg (Tmk_land_id, Landsdel_id,
                      Parti_id, Antal_kredsmandater, Kvotient, Random_nr)
  bulk collect into P_col_tillaeg_ldel_til_kvvalg
  from (select   Tmk_land_id,
                 Landsdel_id,
                 Parti_id,
                 Antal_kredsmandater,
                 Tmk_land_kvotient Kvotient,
                 Tmk_random_nr Random_nr
            from (select Tmk_land_id,
                         Landsdel_id,
                         Parti_id,
                         Antal_kredsmandater,
                         Tmk_land_kvotient,
                         Tmk_random_nr,
                         row_number ()
                         over (partition by Landsdel_id, Parti_id
                               order by Tmk_land_kvotient desc)
                           Nr_starttillaeg
                    from (select Tmk_land_id,
                                 Landsdel_id,
                                 Parti_id,
                                 Antal_kredsmandater,
                                 Tmk_land_kvotient,
                                 Tmk_random_nr
                            from (select Tmk_land_id,
                                         Landsdel_id,
                                         Parti_id,
                                         Antal_kredsmandater,
                                         Tmk_land_kvotient,
                                         Tmk_random_nr
                                    from Tillaeg_man_land_kvotienter
                                   where     Valg_id = P_valg_id_in
                                         and Valgfase_kode = P_valgfase_in)))
           where Nr_starttillaeg > Antal_kredsmandater
        order by Tmk_land_kvotient desc,
                 Tmk_random_nr desc);
```

Fig. 4. Procedure `Tillaeg_landsdel_hentpotkvot`

All tuples from table `Tillaeg_man_land_kvotienter` are ordered lexicographically, first by the rounded quotient `Tmk_land_kvotient` (line 23617) and the random number `Tmk_random_nr` (line 23618), both in descending order.

5 Responsible Disclosure

Denmark has been using a computer program to compute the seat allocations of the Danish Parliament since for at least two decades [6]. The new version of DSAS (studied in this paper) was introduced only after 2007. We identified the rounding problem in DSAS in 2016 and informed the Ministry immediately about our findings. To the best of our knowledge the software as been updated, and the rounding problem has been addressed and fixed.

6 Conclusion

We have shown, that countries that use d'Hondt or Sainte-Laguë methods for computing the final seat allocation of parliament should be aware that rounding quotients in the table may lead to Error 1, *Incorrect Allocation Order*, or even worse, Error 2, *Incorrect Seat Allocation*. We have shown that erroneously rounding can impact an election outcome and that this observation is not just hypothetical, but can with a non-zero probability actually impact real elections. We have also shown that accidental rounding is difficult to detect, after all, on the face of it, how much damage could rounding actually do?

Social choice experts agree that rounding quotients when implementing d'Hondt or Sainte-Laguë methods is a mistake. However, it the election law and relevant election regulations that define the exact rules. The law defines the requirements for seat allocation systems, and if the law requires to round two digits then so be it. The Danish law does is not specific when it comes to rounding, and therefore, it should be the mathematical definition of the voting method that prevails.

Therefore, to implement a d'Hondt or Sainte-Laguë voting rule correctly is easy: One must not store the quotient but instead store both numerator (the `Antal_stemmer`) and denominator (the `V_divisor`) in two different fields. Using the following simple rule of arithmetic assuming $b, d \neq 0$

$$\frac{a}{b} < \frac{c}{d} \quad \text{if and only if} \quad a \cdot d < c \cdot b,$$

it is possible to implement the seat allocation for both constituency and compensatory seats without fractions and rounding guaranteeing that the correct seat allocation is computed.

Acknowledgments

The author would like to thank the Danish Ministry for Economic Affairs and the Interior and Statistics Denmark for their support and cooperation. The author would also like to thank Prof. Jørgen Elklit for comments on earlier drafts of this paper.

References

1. Kenneth Benoit. Which electoral formula is the most proportional? A new look with new evidence. *Political Analysis*, 8(4):381–388, 2000.
2. Danmarks Statistik. Folketingsvalget den 13. november 2007. Statistiske Efterretninger, 27. November 2007.
3. Folketing (parliamentary) elections act. Online Report, 2014. Consolidated Act No. 369 of 10 April 2014, translated.
4. Rajeev Goré and Ekaterina Lebedeva. Simulating STV hand-counting by computers considered harmful: A.C.T. In *Electronic Voting - First International Joint Conference, E-Vote-ID 2016, Bregenz, Austria, October 18-21, 2016, Proceedings*, pages 144–163, 2016.
5. Arend Lijphart and Bernard Grofman. Degrees of proportionality of proportional representation formulas. In *Electoral laws and their political consequences*, pages 170–179. Agathon Press, 2003.
6. Rune Pedersen. Unix-veteran kender valgresultatet før alle andre. Computerworld, 13. November 2007.
7. Philip B. Stark and Vanessa Teague. Verifiable european elections: Risk-limiting audits for D'Hondt and its relatives. *USENIX Journal of Election Technology and Systems (JETS)*, 1:18–39, 2014.