LEGISLATIVE ASSEMBLY
FOR THE AUSTRALIAN CAPITAL TERRITORY

QTON No. 01

STANDING COMMITTEE ON JUSTICE AND COMMUNITY SAFETY
JEREMY HANSON CSC MLA (CHAIR), MARISA PATERSON MLA (DEPUTY CHAIR), JO CLAY MLA

Inquiry into referred 2019–20 Annual and Financial Reports and Budget Estimates 2020-21
ANSWER TO QUESTION TAKEN ON NOTICE
19 February 2021

Asked by Mr Hanson on 19 February 2021: Electoral Commissioner, Damian Cantwell took on notice the following question:

[Ref: Hansard Transcript [19 February 2021] [PAGE #49]

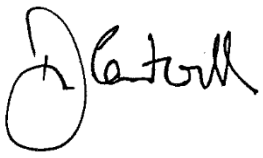
In relation to:

Can you provide us with a copy of her [Associate Professor Vanessa Teague] initial report that you said is in the public domain?

Mr Cantwell's answer:

A copy of the report by Vanessa Teague and Andrew Conway is attached to this response.

Approved for circulation to the Standing Committee on Justice and Community Safety

Signature:  Date: 25 February 2021

By the Electoral Commissioner, Damian Cantwell AM CSC

Errors in the ACT's electronic counting code

Andrew Conway and Vanessa Teague*

November 24, 2020

Abstract

This report describes several coding errors in the 2020 version of the Australian Capital Territory's EVACS vote-counting system. The code is not openly available, but the errors are evident from the official distribution of preferences given the published votes. Although we do not believe any of these errors caused the wrong candidates to be elected this time, they certainly caused tallying mistakes of more than 20 votes. It is only by good luck that no outcomes were close enough for these errors to matter, and there is a realistic possibility that they could cause wrong election results in future years if they are not corrected.

1 Background on EVACS

The Australian Capital Territory (ACT) has used electronic counting for decades, on a system that until recently was openly available for public scrutiny. Over the years from 2001 to 2016, independent examination has identified, and generally led to the correction of, errors in the counting code ([3] and unpublished work by Tiu and Wilson-Brown). The ANU logic group have written both detailed general analyses of the ACT's legislated counting algorithm [2] and methods for formal verification of STV code [1].

Despite this, in 2020 Elections ACT decided to reimplement the code from scratch, not to ask any of the experts who had identified problems or published analyses of it in the past, and not to make the code openly available for public scrutiny in advance of the election.[1]

A separate system from the same supplier is also used for pollsite e-voting. (T Wilson-Brown identified privacy problems in this system in 2018 [4].) Some votes are received over the Internet from a system designed by a different supplier. These two systems are outside the scope of this report, but we note that their assurance and quality control measures are similar to those of the counting system, in which we have discovered multiple serious errors.

*vanessa.teague@anu.edu.au

[1] It was available only under a non-disclosure agreement, and only after voting had begun.

2 Summary of observed errors

We found three errors that could potentially change the results of an election, though this year by good luck they do not seem to have changed the winners.

1. EVACS incorrectly groups votes by transfer value, failing to recognise when votes deserve to be grouped because they acquired the same transfer value in different ways. In 2020 this caused some tallies to be wrong by more than 20 votes; in general, it could cause much larger divergences. This is described in Section 3.

2. The ACT Electoral Act explicitly requires rounding *down* (to 6 decimal places), but EVACS rounds to the *nearest* 6 decimal places. See Section 4.

3. EVACS has some other inaccuracies that are consistent with rounding transfer values, despite this not being specified in the legislation. This is important because a transfer value's effect may be multiplied by thousands of votes. See Section 5.

Sometimes exclusions continue even though the distribution of preferences should be finished. This makes no difference to the winners, but is against the explicit instructions of the legislation, and could be confusing. See Section 6.

There are also a number of accounting choices and labels which, though not strictly incorrect, are sufficiently confusing that they ought to be changed. These are described in Section 7.

3 Error 1: incorrect grouping by transfer value

3.1 What the law specifies

When a candidate is excluded, their votes are redistributed. They may have different transfer values because they may have arrived at that candidate via different routes. The legislation states:

> If a candidate is excluded ..., the ballot papers counted for the candidate shall be sorted into groups according to their transfer values when counted for him or her.[2]

This is why it can take multiple counts to exclude a candidate. Separating them by transfer value can make a small difference to rounding, but the largest effect occurs if some other candidate is elected as a result of some of these transfers. When candidate B is excluded, if candidate A is elected part way through the counts in which B's votes are being distributed, A ceases to be a continuing candidate and does not receive any more votes from later counts in B's distribution.

[2] ACT Electoral Act 1992, Schedule 4, Part 4.2, Clause 9(1)

3.2 What EVACS does

The ACT's counting code does not reliably combine votes with the same transfer value, particularly if they arrived at that transfer value in different ways.

Consider again the example from the previous section, in which A reaches a quota during the distribution of votes from excluded candidate B. If votes are not grouped correctly, parcels of votes that should be transferred to A in a single step can instead be split up so that only some are transferred, with the rest going elsewhere after A reaches a quota. This can have significant effects, particularly given the importance of the last parcel in ACT elections.

There are several examples in the 2020 count in which votes with the same transfer value are not transferred in the same count when a candidate is excluded. In each case, this results in candidates getting the wrong tally, though by good luck these errors did not change the outcome this year.

For example, in Murumbidgee, count 28, Bec Cody is up for exclusion. She has received votes with Transfer Value 1 from counts 1 to 21 and $23, 25, 27$, along with votes of transfer value 0.374100... from counts 22, 24 and 26. So there should be two counts for Cody to be excluded: one transferring all the votes of transfer value 1, and a second transferring all the votes of transfer value 0.374100... However, in the official tally, there are three.

Count 28: distributes the 5421 votes, with transfer value 1, which Cody received from counts 1 to 21, 23 and 25.

Count 29: distributes the 80 votes, with transfer value 1, which Cody received from count 27.

Count 30: distributes the 140.287769 votes, with transfer value 0.374100..., which she received in counts 22, 24 and 26.

The distributions in counts 28 and 29 should be done as one batch because they have the same transfer value, but they are mistakenly split into two.

A more extreme case happened in Bridabella, when Andrew Wall got eliminated. All his votes had transfer value 1, but in the official distribution of preferences, his votes were distributed in three separate counts. Furthermore, two candidates were elected in the first of these three counts, meaning they were not continuing candidates for the rest of the preferences. Nicole Lawder thus missed out on 21 votes. Mark Parton missed out on 13. By good luck, this did not change who was elected as there was only one continuing candidate left at this point.

This is a very serious bug that could easily change the election result.

3.3 What causes this error?

We do not have the source code, so we cannot be certain exactly what causes this error or how to correct it, but we speculate that the implementors of EVACS didn't notice that votes can reach the same transfer value in different ways. This is easily illustrated for transfer value 1, which can be reached

1. by ballots that have never been part of an elected candidate's surplus;
2. by ballots that have been part of an elected candidate's surplus, when the number of transferable votes exactly matched the excess (an unlikely but possible scenario); or
3. by ballots that had a value of 1, and became part of an elected candidate's surplus, when the number of transferable votes was less than the excess.

In the third case, rule 1C(4) from schedule 4 part 4.1 applies:

> However, if the transfer value of a ballot paper worked out in accordance with subclause (2) would be greater than the transfer value of the ballot paper when counted for the successful candidate, the transfer value of that ballot paper is the transfer value of the ballot paper when counted for the successful candidate.

Returning to the Murumbidgee example, the votes of transfer value 1 distributed in count 28 have never been part of an excess (case 1). The votes mistakenly separated into count 29, despite also having transfer value 1, got that transfer value as a result of the application of rule 1C(4), (case 3).

We speculate that the coding error is a failure to check for votes that received the same transfer value in these two different ways.

3.4 A simple example to show why this could get the wrong people elected

Clearly an error that can cause mistaken tallies can also cause the wrong candidates to be elected. We show here a simple example of how that could happen.

Suppose there are five candidates A, B, C, D, E, three seats available, and 39,996 votes, so the quota is $q = 39,996/4 + 1 = 10,000$. Suppose also that the votes are:

Preferences	Count
A	10,001
A, B, C, D	3,000
B, C, E	3,000
C	7,000
D	9,995
E	7,000

The distribution of preferences should be:

Count	A	B	C	D	E	Description
1	13,001	3,000	7,000	9,995	7,000	A elected
2	10,000	6,000	7,000	9,995	7,000	B excluded
3	10,000		13,000	9,995	7,000	C elected
4	10,000		10,000	11,495	8,500	D elected

When A's excess is distributed, there are fewer transferable votes than the excess, so Schedule 4, Part 4.1 Clause 1C(4) applies: "(4) However, if the transfer value of a ballot paper worked out in accordance with subclause (2) would be greater than the transfer value of the ballot paper when counted for the successful candid ate, the transfer value of that ballot paper is the transfer value of the ballot paper when counted for the successful candidate." Thus the transfer value of the 3,000 votes transferred to B is exactly 1.

When B's votes are distributed at count 3, they should all be transferred together because they all have the same transfer value: 1. They all go to C, who thus gets a total of 13,000. C's excess of 3,000 is evenly split between D and E, who each get 1,500. This causes D to be elected in the next count.

However, based on our observations of EVACS's output for the 2020 election, we believe that Elections ACT's official count would incorrectly separate the distribution of B's votes into two separate steps, probably (based on the example of Bec Cody's exclusion) the votes giving B as a first preference would be distributed first, then, in a subsequent count, the votes of transfer value 1 that had been transferred from A.

We believe the distribution of preferences would be:

Count	A	B	C	D	E	Description
1	13,001	3,000	7,000	9,995	7,000	A elected
2	10,000	6,000	7,000	9,995	7,000	B excluded
3	10,000	3,000	10,000	9,995	7,000	C elected
4	10,000		10,000	9,995	10,000	E elected

In Count 3, the first part of B's votes are distributed, thus giving C exactly a quota. In Count 4, the second part of B's votes are distributed—these have preferences B,C,E, but because C already has a quota, they are not distributed to C. Instead, they pass straight to E with transfer value 1. Thus E is incorrectly elected instead of D.

This causes a wrong result in an election that is not even close: D should have beaten E by nearly 3,000 votes, but instead loses because of the error.

3.5 Summary and implications

This is a serious error with real potential to seat the wrong people. It could cause tallies to be off by thousands of votes in realistic examples.

We are fairly certain this error was introduced in 2020 and was not present in 2016.[3] This suggests that the auditors certified the system without even checking that the new system's count of the 2016 votes was approximately similar to the official 2016 count.

[3]For example, in 2016 Yerrabi, on count 27 Michael Pettersson's 766 excess votes were distributed with a transfer value of 1 because of exhausted votes/last parcel. 158 went to Jacob Badakkadethu, and then in count 29, Jacob's votes were partially distributed, those with transfer value 1, which included the original transfer value 1 and the ones from count 27 as well, as they should.

4 Error 2: Rounding to the nearest 6 decimal places instead of rounding down

4.1 What the law specifies

The legislation states, in schedule 4 part 4.1 1A:

> 1A Meaning of count votes—sch 4
>
> (1) For this schedule, count votes, in relation to a candidate, means the number of votes worked out as follows:
>
> $$BP \times TV$$
>
> (2) *Any fraction must be rounded down to 6 decimal places.*
>
> (3) In this clause:
>
> BP means the number of ballot papers to be dealt with at a count that record the next available preference for the candidate.
>
> TV means the transfer value of those ballot papers.

Note that (2) explicitly specifies rounding *down*.

4.2 What EVACS does

EVACS seems to round to the nearest value with 6 decimal places.

For example, consider Murrumbidge, Count 22. Chris Steel went 312 votes over quota and his votes were redistributed. There were 875 votes in the last parcel. (These numbers can be seen from the official distribution of preferences). The complete data shows there were 41 of those not continuing, so $875-41 = 834$ continuing, so the transfer value is 312/834. Marissa Paterson got 354 of those papers, so she should get $354 * 312/834$ votes, which is 132.431654676...

This should be rounded down to 132.431654. EVACS rounded up to 132.431 655. There are numerous other similar examples.

4.3 Summary and implications

This is unlikely to make a large difference, and could change the outcome only in a very close race. Nevertheless it indicates some of the challenges of working with decimal rounding, which doesn't align naturally with the binary representation used in a computer. Although this particular error is reasonably easily corrected, there are likely to be numerous subtle bugs associated with trying to implement the requirement to round down to 6 decimal places.

5 Error 3: Rounding transfer values

5.1 What the law specifies

The ACT Electoral Act does not mention rounding transfer values.

5.2 What EVACS does

EVACS seems to round transfer values to the nearest value with 6 decimal places when applying the rule in schedule 4 part 4.1, 1C(4).

5.3 An example

Still in Mumrumbidgee, this time consider count 32, the last count. (This count should not be done at all—see Section 6—but it was done and it indicates an arithmetic error.)

In Count 32, Emma Davidson's votes were distributed. These came from the last parcel from Count 31. Davidson had an excess of 326.962675, with 331 continuing papers in the last parcel, all going to Fiona Carrick.

It might seem at first that all 326.962675 votes should go to Carrick. However, the transfer value would be 326.962675/331 which is greater than the transfer value of 1329.553957/1349 at which Davidson got these votes. The rule in schedule 4 part 4.1, 1C(4) comes into effect (see Section 3), and the votes should keep the transfer value at which they were received, 1329.553957/1349. When multiplied by 331 papers, this gives 326.228584.

This means that the remaining 0.734091 votes disappear. (EVACS accounts for these in a column entitled "loss by fraction," which is confusing but not strictly wrong—see Section 7.)

However, those are not the values Elections ACT computed. They got 326.228635 and 0.734040 respectively, which are similar but noticeably different.

Without access to the source code, we cannot be certain how these values were computed, but they do happen to be the values that one would get if one normally computed transfer values with high precision, but, when using rule 1C(4), decided to round the transfer value itself to six decimal places before multiplying it by the number of papers.

5.4 Summary and implications

This bug had no effect in this election, of course, since it came into play in a count that should not have been done in the first place. But in general, it could change tallies by a very small number of votes, which could change the outcome if the election was nearly tied. Still, it is significantly worse than rounding incorrectly *after* multiplying by the number of papers (Error 2, Section 4).

6 Error 4: Eliminations continue after the count has finished

6.1 What the law specifies

Schedule 4, 4.2, clause 4.1 states:

> If, after a calculation under clause 3 (3), 6 (4) or 9 (2) (d), the number of successful candidates is equal to the number of positions to be filled, the scrutiny shall cease.

6.2 What EVACS does

The final count of Murumbidgee occurs AFTER all five candidates were elected.

6.3 Summary and implications

This seems like a bug, but it is no more than a waste of time and space; the candidates are all elected and nothing can happen.

7 Confusing accounting

The legislation does not specify how to explain or account for ballots, particularly lost or exhausted ballots, as the preferences are distributed. However, EVACS has some choices which, while not strictly incorrect, are sufficiently confusing that we strongly recommend changing them.

7.1 Accounting for exhausted votes in an excess under "loss by fraction"

As explained in Section 5, the application of rule 1C(4) can cause votes to disappear when the number of transferable votes is smaller than the excess. This has nothing to do with rounding, but rather with the combination of

- transferring only the last parcel, and
- applying rule 1C(4) to prevent the value of a vote from increasing.

EVACS counts these disappearing votes correctly, but puts them in a column entitled "loss by fraction," which is unhelpful—"loss due to distributing only the last parcel" would be more accurate. Unlike everything else described in this report, this issue is also present in earlier versions of EVACS.

It would be even better to change ACT electoral law to transfer all the votes rather than only the last parcel, when a candidate gets more than a quota. Then fewer (if any) votes would be lost in this way. This is also recommended by Goré and Lebedeva [2]. Most other STV rules, including those for the Australian Senate and the Victorian Legislative Council, already do this.

7.2 Splitting exhausted votes into two columns

When a vote needs to be distributed, but has no further preferences, it makes sense to add it to the "Votes exhausted at count" column. EVACS, confusingly, adds the integer part to the "Votes exhausted at count" column and the fractional part to the "loss by fraction" column. The legislation never mentions these columns, so it isn't wrong, but it is unnecessarily confusing.

Consider Murrumbidgee as an example again, this time Count 24. Ed Cocks is being eliminated. He has some votes with transfer value 1 distributed in Count 23, and some with transfer value 0.374... distributed in Count 24. There are 7 papers exhausted. So there are 7*TV=2.618... votes lost to exhaustion. EVACS takes the integer part (2), appends ".000000" and puts it in the "Votes exhausted at count" column, then puts the fractional part (0.618...) in the "Loss by fraction" column. This seems unhelpful to anyone trying to understand the count.

7.3 Marking candidates as fully excluded before they are fully excluded

The same count also states "Ed COCKS fully excluded," in Count 23 before he is fully exluded in Count 24. Generally, if someone takes n rounds to be excluded, the official distribution of preferences says they are fully excluded on (relative) count $n-1$, is blank on count n, and calls them partially excluded on count $n-2$. Again, there is no legislation that prohibits this, but it is confusing.

8 Our source code

Our complete implementation is available at
`https://github.com/SiliconEconometrics/PublicService`.

We have implemented what we believe to be a correct version (though corrections are always welcome), together with flags for adding each of the errors described in this report. If we introduce all these errors into our code, we find no more discrepancies with the official count.

9 Discussion and conclusion

This shows that the ACT's current audit and certification process does not protect ACT elections from serious software errors.

It also shows that keeping the code secret does not keep all its errors secret. (Note also that one year's worth of input and output data does not necessarily make all its errors evident.)

We could easily have helped Elections ACT detect and correct these problems if its source code and test data had been made openly available before the election. We do not understand why Elections ACT excluded all those who had

identified (and helped to correct) errors in earlier versions of EVACS from the opportunity to assess the 2020 code before it was used.

We are highly skeptical that either the pollsite e-voting code or the Internet voting code are free of equally serious errors. If anything, such errors would be more serious because they could affect the ballot records themselves in a way that could not be detected.

We recommend that ACT Electoral law be amended to ensure:

- that all relevant code, including the counting code, and e-voting code, be made openly available for public inspection at least six months before the election, in order to have some chance of detecting the most serious errors and vulnerabilities;
- that the pollsite e-voting system have a voter-verifiable paper record, so that an immutable record of the vote can be verified by the voter independently of the software;
- that Internet voting be discontinued. If they can't even count right, there is no chance Elections ACT have successfully solved one of the hardest open problems in online security.

References

[1] Milad K Ghale, Rajeev Goré, Dirk Pattinson, and Mukesh Tiwari. Modular formalisation and verification of STV algorithms. In *International Joint Conference on Electronic Voting*, pages 51–66. Springer, 2018.

[2] Rajeev Goré and Ekaterina Lebedeva. Simulating STV hand-counting by computers considered harmful: Act. In *International Joint Conference on Electronic Voting*, pages 144–163. Springer, 2016. `https://openresearch-repository.anu.edu.au/bitstream/1885/113153/2/01%20Gore%20and%20Lebedeva%20Simulating%20STV%20hand-counting%202017.pdf`.

[3] ANU Logic and Computation Group. Electronic voting - a review of the hare-clark model of eVACS, 2003. `http://users.cecs.anu.edu.au/~rpg/EVoting/evote_revacs.html`.

[4] T Wilson-Brown. Possible vote disclosure in ACT elections, 2018. `https://github.com/teor2345/Elections2018/blob/master/ElectionsACTDisclosure.md`.