Validation of Implementation of the Meek Algorithm for STV

B A Wichmann

28th April, 2000

1 Introduction

ERBS, as part of their plan to validate all the software used to count elections, have requested that the Meek algorithm be validated.

This request has been interpreted as requesting the 'validation' of two implementations of the Meek Algorithm, against the definition given in the original *Computer Journal* publication [3], and denoted by CJ.

This work has a number of facets as follows:

1. A comparison between the two implementations by David Hill (DH) and the author (BW), written by us both [4].
2. An analysis of the accuracy requirements for the Meek Algorithm [5].
3. A report on testing both algorithms against a set of over 300 test elections, which is reported below.

All these three facets are brought together in the conclusions to this paper.

2 Testing two implementations of Meek

Although the comparison paper [4] provides a detailed account of the differences between the two main implementations, it seems wise to confirm the differences by detailed testing. Since some classes of tests are designed to mirror actual elections, one can estimate the occurrence of various aspects of the algorithm.

Both implementations have restrictions which reduce the number of tests which can be successfully executed. The table below indicates the number of those tests outside the specification of each implementation for each directory. The directory CAPT1 contains mainly large elections which therefore are outside the capacity limits of DH.

An exact comparison cannot be expected due both to differences in the specification and due to the breaking of ties. The basic test was to note if exclusions were in the same order. Given this (which implies the same candidates are elected), then the test was 'passed'. If exclusions were in a different order, then the test was investigated further. If both implementations reported a tie-break, then this was usually checked to show that a different tie-break caused the difference in the exclusion order. As there were a lot in this class, no analysis was undertaken unless the different exclusion order caused different candidates to be elected. If either no tie-break was observed or different candidates were elected, then clearly an investigation is needed, and these are shown under the 'Special case' column in the table below. Several cases arose in which exclusions were in the same order, but BW reported a tie-break, but DH did not — these were not always investigated due to the effort involved.

Directory	Cases	Outside specification		Special
		BW	DH	cases
CAPT1	8	0	7	-
MOCK1	50	0	0	-
MOCK2	41	0	0	-
MOCK3	51	0	0	M112,113,115
				M135,142,143
REAL2	4	0	0	R036,052
SEMI1	36	0	1	-
TEST1	50	3	3	T008,017,014
				T034,046
TEST2	50	8	6	T073,
TEST3	50	1	1	T134,139,141
				T145,149
TEST4	8	0	0	T153,157,158
Total	398	12	18	22

2.1 Test Analysis

This validation work and the testing of the two main implementations has revealed some issues which need some consideration as detailed below. However, only one minor defect was found in DH, see Appendix B.1, and one significant error was found in BW, see Appendix B.2, both of which have been corrected.

The issues requiring consideration are as follows (the most important first):

Quota $+\epsilon$. DH adds a small amount to the quota as computed by the published algorithm, following the original concept of Droop. However, this results in one case in which CJ/BW gives a different result from DH which is really a tie (see Appendix A.16).

Tie-breaking by an additional rule. DH has introduced a rule like that in many of the hand-counting rules for breaking a tie based upon the votes for the candidates at previous stages. To clarify this, David Hill has agreed to produce a message on the printed output indicating that this rule has been applied (and therefore that a tie has been broken). One problem with the introduction of this rule is that the removal of a candidate that has been excluded does not necessarily produce the same result. For an example of this issue, see Appendix A.10. Also, to aid the testing, some tests were re-run with the tie-breaks taken by BW using DH. This is not always possible due to this rule, see Appendix A.7.

Tie-breaking by rounding. Several test cases were noted in which the mathematical result was a tie, but one of both the implementations did not report a tie due to the rounding errors in the computation. For an example of this, see Appendix A.4.

Excessive accuracy. A separate report notes that in artificial cases, the mathematically correct result requires the algorithm to use excessive computational accuracy. No instance of this problem arose during testing and it is not thought to be anything other that a theoretical possibility that the 'wrong' result could be obtained this way. Additional tests were added to illustrate the issue, see Appendix A.20.

Slow convergence. Under certain circumstances, the iterative process used to compute the new quota and votes for each candidate can converge very slowly. This slow convergence is worse when there are more candidates and can arise in real elections. However, with the speed of modern computers, the computational time is not excessive. (An attempt was made to improve the convergence of the algorithm, but the improvements were too small to be worth implementing, see Appendix A.15. In any case, the time taken by computer is small compared with any manual method of conducting an STV count.)

3 Conclusions

Both implementations conform to the principles of the Meek Algorithm. All the issues identified during this work are very minor (and all the visible ones involve ties).

Due to the wide usage that David Hill's system has had, its use is to be recommended. If the limitations of that program are too severe for a specific election, then my program can be used. If the CSV output is required (bearing in mind its limitations for Meek), then my program can be used just for that purpose.

References

[1] B. L. Meek. Une nouvelle approche du scrutin transférable. *Mathématiques et sciences humaines* **25**, 13-23 (1969).

[2] B. L. Meek. Une nouvelle approche du scrutin transférable (fin). *Mathématiques et sciences humaines* **29**, 33-39 (1970).

[3] I. D. Hill, B. A. Wichmann and D. R. Woodall. Algorithm 123 — Single Transferable Vote by Meek's method. Computer Journal. 1986.

[4] I. D. Hill and B. A. Wichmann. A comparison of two implementations of the Meek algorithm. April 1999.

[5] B. A. Wichmann. The computational accuracy using the Meek algorithm. April 1999.

A Test case analysis

Each test case requiring study is listed here in alphabetical order.

A.1 M112

In this case, C1 is elected, and a different exclusion occurs with either C20 or C23. In fact, both C20 and C23 have the same number of first-preference votes *and* the same number of transfers from C1. In consequence, the tie is independent of the value of the weight (k in M135) for C1.

A.2 M113

This case is rather different. Both versions initially elect C1 and C2. However, then David Hill uses his logic to eliminate C11 early. My version first elects C3, C8 and C6, causing transfers to C11, so that C9 is eliminated first.

A.3 M115

This is essentially the same as M112. The choice between the exclusion of C15 and C18 was different, but both have no first-preference votes and 4 transfers from the elected candidate C1.

A.4 M135 (by David Hill)

As noted above, this test produced different results with the two Meek algorithms, although neither indicated a tie.

The test example was generated by program, but designed to reflect real election statistics. Twenty-five candidates competed for 7 seats, with 679 votes. After the election of C1, and the exclusion of a number of candidates, David Hill's program excludes C13, while the other program excludes C16. Given a different exclusion at this point, the two implementations naturally diverge due to the effect of the subsequent transfers.

The relevant votes at the critical point of excluding C13 or C16 can be found to be:

C1	$125k$ = quota
C13	$32 + 4(1 - k)$
C16	$33 + (1 - k)$
non-transferable	$10 + 7(1 - k)$
quota	$(679 - 10 - 7(1 - k))/8$

Equating the two expressions for the quota leads to $k = 2/3$. Thus whether to exclude C13 or C16 depends on comparing $32 + 4 \times 1/3$ and $33 + 1/3$, and so it is really a tie, but for the inevitable rounding errors. Indeed, since 1/3 cannot be exactly represented in binary, it is clear than equality cannot be expected in this case.

Logically, it might be preferable to detect the equality, but some means is still required to break the tie. Allowing the rounding errors to break the tie is therefore surely acceptable. However, this does imply that different, logically correct, implementations can produce different results in cases like this.

A.5 M142

The same situation as M112. C1 is elected, then a different exclusion of C19 and C22, in which my version makes a random choice. Here, C19 and C22 have the same number of first-preference votes (11) and one transfer from C1 each. (Mistake in data file M143, same as M142!)

A.6 R036

Simple example of a random choice due to equality.

A.7 R052

In this case, DH reports 15 tie-breaks and BW 26. Due to this large number, it is not possible to be sure that the result of electing different candidates is caused by the tie-breaks alone. Hence the DH implementation was re-run with manual selection at each tie-break. However, the result obtained by BW was not then produced, since the change in the tie-breaking logic introduced by David Hill did not permit that. In consequence, it has not been possible to resolve this test, although the differences are surely due to the tie-breaks.

A.8 T008

Simple example of a random choice due to equality.

A.9 T017

Simple example of a random choice due to equality.

A.10 T014

Here, a different candidate is elected, even though DH does not indicate a tie-break. The reason for this is that there is a tie at the last stage, but the tie-breaking rule which David Hill has added to his program chooses a different candidate to elect than the random choice made by BW.

A.11 T034

Here, a different exclusion order appears, even though DH does not indicate a tie-break. The reason for this is that there is a tie at an exclusion, but the tie-breaking rule which

David Hill has added to his program chooses a different candidate to exclude than the random choice made by BW.

A.12 T046

Simple example of a random choice due to equality.

A.13 T073

Simple example of a random choice due to equality.

A.14 T134

Simple example of a random choice due to equality on exclusion, resulting in a different election result.

A.15 T139

This test has been chosen to represent the problem of slow convergence. The convergence logic is the same in DH and BW, but DH uses fewer iterations due to the change documented in [4]. The eight cases with slowest convergence (slowest first) are: R051, R049, R007, R040, T139, R044, R009, and R050. T139 is taken since it only has 9 candidates, but it is the only artificial test case since all the others are real elections.

The problem with T139 is clear. If a candidate C is elected and a significant number of votes involving C only have preferences for other elected candidates, then the iterative process reduces the quota. This then implies that the weights must be reduced, giving a further reduction in the quota, etc. This process converges only very slowly. In practical elections, it occurs at the later stages when a large fraction of the candidates can be elected.

Attempts were made to estimate the reduction in the quota to improve the convergence. Unfortunately, the improvements were not good enough to be worth implementing, bearing in mind that the original algorithm is known to be robust (it will always eventually converge to the solution).

A.16 T141

In this example, no random choices are reported, but the two versions elect different candidates. After the transfer of the surplus from the initially elected candidate, two candidates have the ‘quota’, and since there are two places, it would seem there is no issue. BW performs the expected action, but with DH, the quota is increased by a very small amount, so that the second candidate is a very small amount less than the quota. This causes DH to eliminate other candidates to then have three candidates with the quota, ignoring rounding. The final decision is then determining by rounding, and produces a different result.

The two results are as follows:

```
Licence no: 20   Licensee: Brian Wichmann  (24 Oct 1999)
Version 6.4.1 -- RSS rules

Election result sheet for T141: Lichfield anomaly
   Number to be elected = 2
       Total valid vote = 48

         Count 1          Count 2          Count 3          Count 4
        To elect       To exclude       To exclude       To exclude
               A                E                D                C
                                                           To elect
                                                                  B

Quota     16.000           16.000           16.000           16.000

A         20.000    80.0%  16.000    80.0%  16.000    80.0%  16.000 Elected
B         13.000   100.0%  13.000   100.0%  14.000   100.0%  16.000 Elected
C         12.000   100.0%  16.000   100.0%  16.000   100.0%  16.000
D          2.000   100.0%   2.000   100.0%   2.000     0.0%   0.000
E          1.000   100.0%   1.000     0.0%   0.000               -


N/T        0.000            0.000            0.000            0.000

Total     48.000           48.000           48.000           48.000
```

```
Election for: T141: Lichfield anomaly
Date: Fri,31 Mar 2000.07:56:36
Number to be elected: 2
Valid votes: 48
Invalid votes: 0
Quota: 16.00
BAW - Meek-Arch : version 1.23
Election rules: Meek
                             Stage 2     Stage 3       Stage 4
                    First    Surplus of  Surplus of
Candidates       Preferences      A           C
----------------+-----------+-----------+-----------+--------+
A                         20 -4.00 16.00       16.00 Elected
B                         13       13.00       13.00
C                         12 +4.00 16.00       16.00 Elected
D                          2        2.00        2.00
E                          1        1.00        1.00
----------------+-----------+-----------+-----------+--------+
Non-transferable                    0.00        0.00
Totals                    48       48.00       48.00
----------------+-----------+-----------+-----------+--------+
```

A.17 T145

In this test case, exclusions are in a different order, but DH reports no tie-break while BW does. The reason for this is the change in the tie-breaking logic in DH.

A.18 T149

In this test case, the exclusions are in a different order, obtaining a different result, although both versions report a tie-break. This is an artifical test case with most of the candidates with the same first preference votes. The differences between the two implementations are caused by simple tie-breaks.

A.19 T153

Simple example of a random choice due to equality.

A.20 T157

This example is based upon that give in [5], but requires an accuracy of roughly 1 part in 10^{11}. The internal accuracy of DH implies that this test fails, but BW passes. Hence DH makes a random choice for exclusion which could imply that the voter with the algebraically smallest number of votes is not excluded.

A.21 T158

This example is based upon that give in [5], but requires an accuracy of roughly 1 part in 10^{15}. The internal accuracy of both DH and BW implies that this test fails in each case.

B A defect and an error

Since both these problems have now been corrected, their importance is mainly in indicating the potential risks of further undetected problems.

B.1 A defect

When the number of votes exceeded the limit which invokes a warning message (set at 99,999 votes), David Hill's program sometimes looped (tests T005 and T156). This defect has since been corrected.

B.2 An error

There was a simple coding error in my version of Meek which did not allow for the invalid votes when computing the quota. This was not noticed immediately, since so few of the tests in the database contain invalid votes.

C Document details

First complete draft produced 28th April 2000, revised 11th May and 21st May 2000.