All correspondence regarding *Voting matters* should be addressed to:

B A Wichmann, The Editor, *Voting matters*, The Electoral Reform Society, 6 Chancel Street, London, SE1 0UU.

or using e-mail to Brian.Wichmann@freenet.co.uk.

To aid production, the Editor would welcome contributions on IBM-PC discs (with a printed copy as well) or to Brian.Wichmann@freenet.co.uk.

Voting matters

for the technical issues of STV

The Electoral Reform Society

Issue 13 April 2001

Editorial

As the political debate intensifies prior to a General Election, the search for a better technical understanding continues here.

Hugh Warren responds to remarks made about his suggestion for merging X voting with STV.

The issue of undertaking recounts with STV is very unclear. Even with computer counts, ensuring that there are no errors whatsoever in the data input is unrealistic. My own paper provides details of a computer system designed to detect if an STV election is close enough to justify a recount.

Joe Otten provides details of an algorithm for handling STV elections with complex constraints. Even though such constraints override the voters' intentions, it seems that several elections are of this type and hence there is a demand for such an implementation.

David Hill provides an analysis of STV when equality of preference is permitted. It seems that there are problems in this area, so the fact that conventional STV does not provide equality is not necessarily a disadvantage.

Lastly, I provide a paper concerned with the *transparency* of STV. The conclusion is to call for the partial disclosure of the votes so that anybody can perform an effective check on the counting process. Comments on this and all the other papers are welcome!

CD-ROM Publication

With the support of the McDougall Trust, I am collecting electoral material with the aim of publishing it in CD-ROM format. It is intended, for example, that the publication will include all of *Voting matters*. (As a separate exercise, back issues of the journal *Representation* may be made available on CD ROM as well.) The main emphasis will be on the collation of election data, especially that involving STV or preferential voting. If you have or know of material which you think could be suitable, please contact me. A key advatange of the CD ROM media is that well over 5,000 pages can be placed on one disc.

Brian Wichmann

The principal objects of the McDougall Trust (The Arthur McDougall Fund) are to advance knowledge of and research into representative democracy, its forms, functions and development and associated institutions. The Trust is governed by a High Court Scheme issued in 1959 which states its charitable purposes as being ‘to advance knowledge of and encourage the study of and research in: political or economic science and functions of government and the services provided to the community by public and voluntary organisations: and methods of election of and the selection of and government of representative organisations whether national, civic, commercial, industrial or social.’

Mixing X-Voting and Preference Voting

C H E Warren

In my paper on incorporating X-voting into preference voting by STV[1], without saying so I had treated it as axiomatic that a method of mixing X-voting and preference voting should reduce to either X-voting or preference voting by STV should all the voters be of one sort.

In a comment at the end of my paper, the Editor suggested an alternative formulation which, sadly, would not reduce to X-voting as it is always practised should all the voters be X-voters. The Editor's formulation would not therefore satisfy the axiom mentioned above.

The answer to the question at the end of David Hill's paper[2] “Is there a way of doing it that everyone would think fair in all cases?” is surely “No”.

There are the hardliners on both sides — those who think that anything other than X-voting is not fair, and those who think that anything other than preference voting by STV, which I imagine includes David Hill, is not fair.

The most that one can hope for, then, is not a way of doing it that everyone would think fair, but a way that a majority of considered opinion would think fair.

The major response that I have had to my paper[1] so far is that “it is a good idea”.

References

1. C H E Warren. Incorporating X-voting into Preference voting by STV. *Voting matters*, Issue 11, p2, 2000.

2. I D Hill. Mixing X-voting and preference voting. *Voting matters*, Issue 12, p6 , 2000.

Recounts with STV

B A Wichmann

Introduction

With Westminster elections, if a result is sufficiently close, a recount is undertaken to reduce the risk of an incorrect result being declared. Of course, with First Past The Post, a simple measure of the closeness of the result is possible, so that the criteria for a recount can be easily given. (A virtually identical problem has arisen with the US elections in Florida in which obsolete technology is employed!)

With STV, recounts are very rarely undertaken due to the problems that this would give. In Newland and Britton rules[1], both first and second edition, there was an instruction, at the end of each stage “Ascertain that candidates and/or their agents are content” and a recount of the stage could be called for if not. The difficulty with this is that it may not become evident that an early stage needs checking until a later one has occurred, and the only sure strategy for candidates was always to ask for a recount after every stage. In the latest edition of the rules, those words have, in any case, been omitted.

However, when the count is conducted by computer, the computer itself can be used to assess the need for a ‘recount’. The article is not concerned with the actual process of undertaking a recount (merely running the counting program again would be pointless), but with providing a tool to assess the risks of an incorrect result being obtained due to a typing error when the papers are entered manually.

This article describes a set of computer programs, developed for Electoral Reform Ballot Services, which assesses the need for a recount.

The concept

At first, I thought that the problem was too difficult to undertake, since if a change is made to even one ballot paper, it is hard (in general) to predict any change of result. However, given a computer program that can undertake a count in a matter of minutes (if not seconds) then an alternative method is available which does not require any analysis of the result of changes in specific papers.

The stages are as follows:

1. A simple model is produced of the manual data entry process, together with the likely data entry errors.

2. From the data entry error analysis, a computer program is produced which simulates such errors.

3. The above computer program is used to construct a hundred (or more) copies of the original election data with simulated errors.

4. The simulated elections are counted by program and the results compared with the original results to see if an incorrect result is likely.

This process can be made effective since the speed of modern computers allows a hundred of more copies of an election to be counted in a reasonable time. (It is surely sufficient for an overnight batch computer run to produce the result — although for smaller elections, a result should be obtained in a few minutes. Examples so far have only taken about an hour to run.)

The system

The system consists of two programs: one which produces copies of the original election with data errors added, and another which analyses the results from all the elections. Provision is made to handle the Meek rules[2] or the ERS97 rules[1]. In addition, a batch execution run is produced to call the relevant election counting program on all the simulated elections.

The data entry model is essentially one of key depressions using ballot papers in which the voter adds preference numbers. Since typing errors have known patterns, a reasonable guess can be made of the potential errors in terms of those errors. However, it is difficult to accurately calibrate the rate of errors. Such errors are naturally rare, say 1 in 5,000 characters, but at this rate one would need to double-check many thousands of characters to obtain a good estimate of the error rate. In addition, the computer entry programs used for ballot entry already include some checks and hence the error simulation program ensures that these checks will be passed. Also, the staff of ERBS are naturally familiar with the requirements and appear to take special care with the first preference (not actually allowed for in the current program). There is some evidence that the staff at ERBS may realise at the end of the ballot paper that they are ‘out-of-step’ and hence go back to correct an error. In view of the above, there is clearly some doubt as to the accuracy of the model of data errors, but the statistical nature of the problem makes some doubt inevitable.

After some experimentation, the data error rate was set at one key depression per 6,000 characters. However, if the error would then be detected by the STV program, such as arising from a repeated preference, the corresponding change is not made.

Results

This can be illustrated by an example taken from a real election (which has been made anonymous).

```
Data error analysis program, version 1.01
Basic data of original election:
 Title: R048: STV Selection Example 1
 To elect 10 from 29 candidates.
 Number of valid votes: 944
 Count according to Meek rules

Data used to simulate input errors to count:
 Key errors taken as 1 in 6000 key depressions.
 Duplication and removal of papers taken
                    as 1 in 6000 papers.
 Number of simulated elections produced: 100
 Seeds were initially:  16215,  15062 and   7213
          and finally:  17693,  15003 and  25920
```

```
Some statistics from the generated election data:
 Average number of commas added for each election: 1
 Average number of commas deleted for each
                                         election: 1
 Average number of interchanges for each election: 2
 Average number of papers deleted for each
                                         election: 0
 Average number of papers duplicated for
                             each election: 0
 Average number of papers changed for each
                                  election: 4
 Average number of papers changed at
                            preference: 1 is 1

Candidates elected in the original election and all
simulated ones:
 Jane BENNETT
 Robert BROWNING
 Joan CRAWFORD
 Francis DRAKE
 Mary-Ann EVANS
 Kate GREENAWAY
 John MASEFIELD
 Alfred TENNYSON
 Sybil THORNDIKE

Candidates not elected in the original election or
any of the simulated ones:
 James BOSWELL
 Emily BRONTE
 George BYRON
 Eric COATES
 Ella FITZGERALD
 Stella GIBBONS
 Graham GREENE
 Sherlock HOLMES
 Samuel JOHNSON
 John KEATS
 Alice LIDDELL
 Harold PINTER
 Walter RALEIGH
 Margaret RUTHERFORD
 Will SHAKESPEARE
 Percy SHELLEY
 John WESLEY
 Virginia WOOLF
```

The program records the known details of the election which includes the type of count used: Meek in this case. Then the statistics are recorded on the simulated elections. Firstly, there is the key depression error rate used, then the seeds used for the pseudo-random generator so that the process can be re-run if required. Then a summary is produced of the changes made to the papers. Note that one of the changes is that of repeating *and* duplicating a paper (both changes are needed to reflect the checks made on the total number of papers). The commas indicate moving onto the next preference. Note that of nearly 1,000 papers, typically one change is made to the first preference position.

Of course, the changes that will be of most interest are those relating to the election of the candidates. The first two lists are the candidates which are always elected or always excluded — there should be no doubt about the status of these.

```
Number of other candidates: 2
 Original Result  Simulated Result(95% conf. limits) Name
     Elected          Elected  98% ( 93% to 100%)     Clara BOW
  Not  Elected    Not Elected  98% ( 93% to 100%)     Benjamin FRANKLIN

End of report
```

The last table indicates the position with those candidates whose status varied in the 101 elections performed (1 original and 100 simulated).

The number of such candidates is two. In the case of Clara Bow, she was elected in the original election and also in 98% of the simulated ones, ie in two cases she was not elected. The case with Benjamin Franklin is exactly the opposite. However, merely knowing that percentage is not what is required. We need an estimate of the probability of an incorrect result, which is the likely value of the percentage in the long run, that is if infinitely many simulated elections were used. This long-term value is estimated to lie between 93% and 100% (to a 95% probability).

In this particular case the result is not seriously in doubt. However if the percentage range included the 50% figure, then it is proposed that this would be sufficient to require a recount.

Conclusions

The method proposed here appears to be an effective means of determining if a recount should be undertaken for an STV election. However, the technique does depend upon a statistical model of the nature of the data preparation errors which is always going to be hard to produce.

The method can be applied to assess the impact of data errors arising from mechanically produced data, assuming the data error rate is high enough to warrant its use.

I am grateful to David Hill who provided some Pascal code which gives the 95% probability ranges — a vital part of the system.

References

1. R A Newland and F S Britton. How to conduct an election by Single Transferable Vote. ERS, 1973, 1976 and 1997.

2. I D Hill, B A Wichmann and D R Woodall. Algorithm 123 — Single Transferable Vote by Meek's method. *Computer Journal*. 1987.

STV with multiple constraints

J Otten

Joe Otten is the author of an STV program for Windows which is being extended to handle constraints

The problem

David Hill writes in *Voting matters*[1] that the handling of constraints should be undertaken by marking as *doomed* candidates who cannot be elected if a conformant result is to be obtained, and marking as *guarded* those candidates who must be elected for a conformant result. A doomed candidate is eliminated immediately so that the next preference can be taken into account, while guarded candidates await attaining a quota (if that is possible). However, where multiple constraints are to be applied, then Hill states we should list all the possible ways that the constraints might be met, so that we can tell when it is necessary to guard or doom continuing candidates. If you are unfamiliar with these details, I recommend reading Hill's article first.

In this paper we consider the situation with two independent sets of constraints, such as nationality and gender. A group of candidates are those sharing the same constraining characteristics. While I agree that Hill's method works, and that simpler methods do not, there is a problem when the numbers of candidates and groups of candidates become large. For instance, suppose there are 20 candidates to be elected from 30 groups, with 2 candidates in each group, there would be astronomic number of cases ($\approx 3^{30}$), of which maybe only half can be ruled out by the constraints. Such a list of possibilities would take far too long to calculate on a fast computer with efficient code, and occupy an excessive amount of storage. This is clearly not feasible. It might appear that such complexity of constraints should not arise in practice — unfortunately it has arisen which has prompted the approach given here.

A worked Example

We re-work Hill's example which is that of 14 to be elected, where must be 7 English, 6 Scottish and 1 Welsh, and additionally 7 Men and 7 Women. We refer to each of these by the initial letter with the nationality first. In this example, there are 8 possibilities listed:

```
EM  EW  SM  SW  WM  WW
 0   7   6   0   1   0
 1   6   5   1   1   0
 1   6   6   0   0   1
 2   5   4   2   1   0
 2   5   5   1   0   1
 3   4   3   3   1   0
 3   4   4   2   0   1
 4   3   3   3   0   1
```

Each time an election or exclusion causes one or more of these results to become impossible, we cross it out. We can then see when it is necessary to guard or doom candidates.

This problem requires a solution that does not involve listing every combination since the size of the list rises exponentially with the number of groups. I believe this is possible if we deduce and keep track of every constraint as it applies to every group. In Hill's example this is possible. At the crucial point he argues that "...only 2 Scottish women remain, we have to elect 6 Scottish altogether and have elected none as yet. Therefore we must elect at least 4 Scottish men. But we are restricted to 7 men in total and we have already elected 3. It follows that we must elect exactly 4 Scottish men, and that means that the remaining 2 Scottish women must be guarded, and that the 2 English men must be excluded as soon as possible,..."

This argument is sound, and does not itself rely on an exhaustive listing of all the possible combinations. I propose a procedure which implements this sort of logic in a way that can be automated and performed at the start of the count and after every election and exclusion.

The way I propose to represent this is as in the following grid.

		English	Scottish	Welsh	Total
Men	*Elected*	0	0	0	0
	Min	0	0	0	7
	Max	7	6	1	7
	Cands	4	11	2	17
Women	*Elected*	0	0	0	0
	Min	0	0	0	7
	Max	7	6	1	7
	Cands	7	3	1	11
Total	*Elected*	0	0	0	0
	Min	7	6	1	14
	Max	7	6	1	14
	Cands	11	14	3	28

A row (of 4 lines) corresponds to each gender constraint and a column to each nationality constraint. A cell, with 4 entries, *Elected, Min, Max, Cands*, corresponds to a candidate group or to a row or column total or to the grand total. The grid has been initialized with the numbers of candidates in each group, and the various totals required by the constraints (as from Hill's example). Of course, we have none elected in this initial table, the constraints are as given before, and the new information is that concerning the candidates.

The basic method is to repeatedly apply five rules to a table until a stable condition is produced which essentially provides a bounding box which must enclose any conformant solution. We need to apply these rules initially (to confirm that a solution is possible) and at each election and elimination. Each rule is triggered by a condition which should be satisfied by a conformant solution.

1. In each group we require: *Elected* ≤ *Min* ≤ *Max* ≤ *Cands*. **Rule** — increase *Min* or decrease *Max*. If as a result of applying the rules *Min* > *Max* then no conformant result is possible (there is no bounding box) and we do not regard this as a settled state.

2. In each group, the *Min* must be possible — i.e. it must be possible for this few to be elected, even if the current minimum is elected from the row/column, and the maxima elected from each other group in that row/column. **Rule** — increase *Min*.

3. Like 2, for maxima — in each group, it must be possible for this many to be elected, even if the current maximum is elected from the row/column, and the minimum elected from each other group in the row/column. **Rule** — decrease *Max*.

4. The row/column minimum must be at least the sum of the minima of the items in the row/column. **Rule** — increase *Min*.

5. The row/column maximum must be no more than the sum of the maxima of the items in the row/column. **Rule** — decrease *Max*.

Hence if any of the conditions required is violated, we apply the associated rule until a settled state is reached.

Once the grid is in a settled state, and if in any cell *Elected* = *Max* then continuing candidates in that cell are doomed. If in any cell *Min* = *Cands* then all continuing candidates in that cell are guarded.

I hope it is clear that each of these rules is a logical necessity, as is its **Rule** when it applies. What is not so clear is that following these rules is sufficient to ensure that candidates are always doomed or guarded as necessary.

To see what is going on, let us apply the above now before we start counting the votes, as we need to in order to ensure that there is a conformant result and to identify any candidates which may be initially guarded or doomed.

		English	Scottish	Welsh	Total
Men	*Elected*	0	0	0	0
	Min	0	3	0	7
	Max	4	6	1	7
	Cands	4	11	2	17
Women	*Elected*	0	0	0	0
	Min	3	0	0	7
	Max	7	3	1	7
	Cands	7	3	1	11
Total	*Elected*	0	0	0	0
	Min	7	6	1	14
	Max	7	6	1	14
	Cands	11	14	3	28

a) By 1, *Max* English Men must be reduced from 7 to 4 because there are not enough candidates. Similarly, *Max* Scottish Women must be reduced from 6 to 3.

b) By 2, *Min* Scottish Men = 2. There are at most 5 non-Scottish men, and we need 7 men altogether.

c) Similarly by 2, *Min* English Women = 3. Since *Min* English + *Max* English Men = 7.

d) By 2, *Min* Scottish Men = 3. Since *Min* Scottish Men + *Max* Scottish Women = 6.

This is a settled state, so we conclude that a conformant result is possible, and we can start counting the votes. The first event is the election of a Welsh man, which we mark as a 1 in the space referring to the number of Welsh men elected. This requires the following alterations:

a) By 1, *Min* Welsh Men = 1.

b) By 3, *Max* Welsh Women = 0.

c) By 2, *Min* English Women = 4.

d) By 3, *Max* English Men = 3.

This is a settled state. We now have 2 cells where *Elected* = *Max*, so the continuing candidates in those cells, a Welsh Man and the Welsh Woman are doomed. The doomed

		English	Scottish	Welsh	Total
Men	*Elected*	0	0	1	0
	Min	0	3	1	7
	Max	3	6	1	7
	Cands	4	11	2	17
Women	*Elected*	0	0	0	0
	Min	4	0	0	7
	Max	7	3	0	7
	Cands	7	3	1	11
Total	*Elected*	0	0	1	1
	Min	7	6	1	14
	Max	7	6	1	14
	Cands	11	14	3	28

candidates are removed from the grid by reducing the *Cands* entry.

The next events are — the election of 2 English Men and 2 English Women, and the exclusion of a Scottish Woman. We would in practice update the grid after each of these 5 events, but for the purpose of this example, we will do it in one go.

a) By 1, *Min* English Men = 2, due to the election.

b) By 1, *Max* Scottish Women = 2.

		English	Scottish	Welsh	Total
Men	*Elected*	2	0	1	3
	Min	2	3	1	7
	Max	3	6	1	7
	Cands	4	11	1	17
Women	*Elected*	2	0	0	2
	Min	4	0	0	7
	Max	7	2	0	7
	Cands	7	2	0	9
Total	*Elected*	4	0	1	5
	Min	7	6	1	14
	Max	7	6	1	14
	Cands	11	13	1	25

This completes the actions directly as a result of the elections, but now we must continue to give a settled state

c) By 2, *Max* Scottish Men = 4.

d) By 2, *Min* English Women = 5.

e) By 3, *Max* English Men = 2.

f) By 2, *Min* Scottish Men = 4.

g) By 2, *Min* Scottish Women = 2.

h) By 3, *Max* English Women = 5.

		English	Scottish	Welsh	Total
Men	*Elected*	2	0	1	3
	Min	2	4	1	7
	Max	2	4	1	7
	Cands	2	11	1	16
Women	*Elected*	2	0	0	2
	Min	5	2	0	7
	Max	5	2	0	7
	Cands	7	2	0	9
Total	*Elected*	4	0	1	5
	Min	7	6	1	14
	Max	7	6	1	14
	Cands	11	13	1	25

At this point, the grid is in a settled state, and we know precisely how many are in each group, so the constraints problem has been solved. *Elected* = *Max* for English Men, so the 2 continuing English Men must be doomed, and *Min* = *Cands* for the Scottish Women, so these must both be guarded. The count will continue to determine which of the English Women and which of the Scottish Men are elected.

All I have demonstrated here is that this method achieves the same result in this case as Hill's method. However, I hope that it is clear how it works and why it should therefore work for all 2-dimensional constraints problems.

Rules 4 and 5 were not needed as none of the Row total or Column total *Min* and *Max* could be altered. This was because the constraints were of the rigid “must equal 7” variety rather than the more flexible “must be between 5 and 9” variety.

Constraints and the STV rules

Given the logic above for handling constraints, then this must be integrated into an STV system which would use a specific rule set in the unconstrained case. We consider this with three sets of rules: The Church of England rules[2] (a hand-counting system which makes provision for constraints), the current ERS rules[3] (hand-counting with no provision for constraints) and Meek[4] (computer-counting with no provision for constraints).

The logic above, using *guarded* and *doomed*, depends upon electing and excluding candidates one at a time. None of the three sets satisfy this, and in consequence, the integration of these STV rules with the constraint logic is non-trivial. The addition is naturally simplest with the Church rules, since they have been written with that intent. However, the rules themselves are without constraints and a separate section gives a series of amendments to the rules which are to be applied in the case of constraints. The wording of the special section is reasonably straightforward since elections and exclusions take place one at a time.

Consider the following situations:

i) Suppose A is excluded, and this causes C and D to be doomed. The Church rules just exclude A at this stage, and then exclude C and D at the next stage. It seems possible to exclude all three together, but this surely makes no difference.

ii) Suppose A and B are to be excluded (with A having fewer votes than B), and the exclusion of A causes B to be guarded, and C and D to be doomed. This then is essentially the same case as above.

iii) Suppose A and B are to be excluded (with A having fewer votes than B), and the exclusion of A causes C and D to be doomed, but does not affect the status of B. It is clear that C and D should be excluded before B, since transfers from C and D could spare B from exclusion.

This last case shows the importance of exclusions being undertaken one at a time. This implies that the rules in ERS for multiple exclusions should be changed to handle constraints. Indeed, whatever method is used to handle constraints, the serialization of elections and exclusions is needed.

With Meek, the published algorithm only allows single exclusions, but the version implemented by I D Hill allows for a single exclusion and multiple elections at one stage. Both the elections and the exclusion need to be serialized to apply the constraints logic.

With all the rules, if two candidates achieve the quota at the same stage, then the election of one could cause the other to be doomed. Hence, if this is a tie, the tie-breaking logic would need to be applied to produce a result, even though this was not necessary without constraints.

Conclusions

The logic for handling constraints which was first specified by David Hill can be implemented in a manner that does not involve the use of large lists. This can be combined with the conventional STV rules, provided changes are made to elect and exclude candidates one by one.

Our illustration here was with an example having two independent types of constraint and therefore requiring two-dimensional tables. However, the same logic can be applied with higher dimensions if required.

With larger problems, the size and number of dimensions, and hence the computational requirements, will increase in proportion, not suffering the combinatorial explosion that the listing of all possible combinations does.

Software has been written to implement this procedure and successfully tested on a 4×16×9×3 hypercube.

References

1. I D Hill. STV with constraints. *Voting matters*, Issue 9 pp2-4. 1998.

2. GS1327: General Synod, Single Transferable Vote regulations 1990 and 1998. (Obtainable from Church House Bookshop, Great Smith Street, London SW1P 3BN.)

3. R A Newland and F S Britton. How to conduct an election by Single Transferable Vote. ERS, 1997.

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

Difficulties with equality of preference

I D Hill

One of the things that some people do not like about STV is the fact that voters have to give a strict order of preference of those candidates whom they mention, where they would sometimes prefer to be allowed to express equality. Even where they are clear about the ordering of their first few preferences, and their last few, they may well wish to separate out their middle candidates from their high ones and their low ones without ordering those middle ones.

Instructions to voters

Difficulties arise in deciding how such equality is to be specified. Suppose candidate A is first preference, then B and C equal, followed by D. Some voters will wish to mark those first four candidates as 1, 2, 2, 3. Others will insist that logic requires 1, 2, 2, 4, while still others may want to use 1, 2½, 2½, 4. What is allowed has to be specified and made not too difficult to follow.

One way out of such difficulties is to say that any numbers the user may wish can be used, but only their order will be taken into account. But if such freedom is to be allowed to those who use equality, it must in fairness also be allowed to those not using equality. This disables some useful tests that can be made for correctness of data input to a computer file. Furthermore suppose someone uses 0; is this to be regarded as better than 1? Then suppose that there are 17 candidates in total and that one voter marks four candidates as 1, 2, 3, 4 while another marks four candidates as 1, 2, 3, 17. Did they both really mean the same thing? I doubt it.

Such difficulties are not fatal, but they need careful thought, and they may complicate the instructions to voters. If they lead to less secure input of data to the computer because of the checks that can no longer be made, that also matters.

Counting the votes

There are other difficulties though in how to count such votes. The basic idea is as set out by Brian Meek[1], that a vote for A(BC)D, where the brackets indicate equality of preference for B and C, should be treated as half a vote reading ABCD and half a vote reading ACBD, and similarly with equalities of more than two candidates. This needs careful handling to avoid a “combinatorial explosion” if equality of large numbers of candidates is allowed.

However there is a difficulty of principle, rather than merely of the mechanics of the operation, that arises if voters choose to mention all candidates and to put two or more of them in equal last place. Meek's paper mentioned this possibility with approval, as allowing voters the option of indicating all remaining candidates as equal, as an alternative to not mentioning them at all. It is the one point in Meek's STV papers where I have to disagree with him, for allowing that option would mean having to explain to voters how to choose which method to use and what their different effects could be; not a task that I would wish on anyone. Or alternatively, just not to mention it, leaving voters uninformed about what they are doing.

The trouble is that there are two important principles in counting votes that are here in conflict:

1. that a vote should be interpreted in accordance with what is actually written on it, and in no other way;

2. that votes of identical meaning should be treated identically.

Now, with five candidates, for example, if one voter marks ABC as the first three preferences and stops there, while another voter marks ABC(DE), the strict interpretation of how to handle the two votes, once the fate of A, B and C has been settled, is different, but their meaning, in terms of preferences, is identical. If voters had been asked to express *degrees* of preference in some way, perhaps those two things might not be thought identical, but all that they have been asked for is an *order* of preference, and I cannot see how those two orders could possibly be thought different. This difficulty does not arise where equality is not allowed, since it so happens that two votes ABCD and ABCDE are treated identically by STV in any case, if those five are the only candidates.

There are three options: (1) to treat them differently even though their meanings are identical; (2) to treat both votes as if they had been ABC(DE); (3) to treat both votes as if they had been ABC. Of these I believe the third option to be the most satisfactory, in that there are cases where an abstention gives a better result than an equality of all remaining candidates, but I know of no case where the opposite can be claimed. (See Woodall's discussion of “symmetrical completion”[2]). I have therefore adopted this approach in my STV computer program.

The difference comes out very clearly in the results of an actual election, that used my program and allowed equality. Some voters, believe it or not, put all the candidates (not merely enough to fill all seats) as equal first choice. The program did not blink an eyelid but put those votes at once into non-transferable, treating them merely as a new way of abstaining. Surely this is right, rather than the alternative of diluting the meaningful votes with this useless information.

Having decided on option (3) then, there arises yet another problem. One of the two fundamental principles on which the Meek system is based is “If a candidate is eliminated, all ballots are treated as if that candidate had never stood”.

Suppose then that we have 5 candidates and someone has voted AB(CD). The (CD) equality has to be included as these are not last places; it is important to the voter's wishes that C and D, though not differentiated from each other, are both preferred to the unmentioned E.

If E is now excluded, we must behave as if E had never been a candidate. With E gone, all four remaining candidates are mentioned and, in accordance with the option adopted above, the AB(CD) vote must now be treated as AB. Any part of the vote that was previously awarded equally to C and D now becomes non-transferable instead. This still treats them equally, of course, but it can have the odd effect that somebody's vote may go down in the course of the count, whereas normally votes can only go up until the candidate is elected or excluded. This is certainly an extra complication that one has to be ready to explain if it occurs.

Overall, my conclusion is that, although allowing equality has some advantages, and it can be implemented, the complications may be too many to be worth it. On the other hand, those bodies that have actually used it report no difficulties, and say that the facility is strongly valued by a significant number of electors.

References

1. B L Meek, A new approach to the single transferable vote, paper II. *Voting matters*, issue 1, 7-11.

2. D R Woodall, Properties of preferential election rules. *Voting matters*, issue 3, 8-15. (Elections 11 and 12 on page 14).

Is STV transparent?

B A Wichmann

'When I use a word' Humpty Dumpty said in a rather scornful tone, 'it means just what I choose it to mean, neither more nor less'. *Through the Looking-Glass, Lewis Carroll, 1832-1898.*

Introduction

The problem with the issue of transparency is to decide what it means. Even then, to be useful, we need something which can be measured, at the very least in an informal sense. Is transparency just a matter of assurance? In which case this can be assisted by auditing, such as is used in the ISO 9000 quality management standard. I think not, since we surely accept that we need to trust those performing the election count. Even with a witnessed count, such as in public elections, we still need to trust those handling the ballot as any conjurer can testify. Even given that trust, we expect *evidence* that the count has been conducted according to the relevant rules.

Use of computers

Even if the election rules are such as to permit a manual count, it is quite likely that an STV count will be conducted using a computer. Hence we now have to question the validity and evidence for such a computer-based operation.

The public perception of computers is mixed. Few check the arithmetic in their bank statements — so surely we should accept such arithmetic when it can be checked by hand. On the other hand, the very complex calculations in weather forecasting cannot be checked, and we all know that the results are far from perfect. Fortunately, an STV count is nearer to a bank statement than to weather forecasting and hence public trust is not unreasonable.

An interesting analogy to trusting a computer-based count is that of safety-critical software which *must* be trusted. The recent problems in the railway industry, specifically passing a signal at red, is being tackled by the automatic train protection system which uses computers to stop the train. Indeed, on the Docklands Light Railway, the problem has been solved by having no drivers! In other words, we trust computers to be *more* reliable than people, at least when the situations are well-defined.

Nevertheless, there is something comforting about seeing piles of ballot papers building up against each candidate which is lost when machine counting takes place. For those witnessing a manual count, it is comforting because it is easier to place trust in people you can see. The experience in Florida is a warning that machine counting can be flawed unless sufficient controls are exercised.

Complexity

It cannot be denied that the counting process of First Past The Post (FPTP) is simple. This, in itself, is a substantial aid to transparency. Hence the simpler the rules, the easier it is to demonstrate beyond reasonable doubt, that the rules have been applied. Indeed, *transparency* might be a euphemism for *to understand* rather than anything associated with verification and auditing.

All the different STV rules must be regarded as complex. The nature of the complexity is different in the hand-counting variety compared with the machine-based versions like Meek. If rules designed for manual counting are used, but implemented using a computer, then the issue of transparency is different — since one must be concerned with the correctness of the software.

Proponents of a specific rule are likely to claim it is simple — not unreasonable if they know it well. The fact is that we have no widely accepted measure of complexity and hence we cannot use complexity as a means of quantifying transparency.

Criteria for Transparency

The main approach is to demonstrate traceability from the ballot papers through to the election result. The actual papers themselves are of no concern (in this article) and hence it is assumed that they can be (or have been) transcribed without error. There is no doubt that FPTP is 100% transparent.

We now consider three examples of STV from the point of view of transparency.

ERS97, by hand

We are assuming that ERS97 is followed to the letter[2]. Hence we have a defined result sheet. Hence the question arises as to whether this information provides complete traceability. It does not since the following information is missing:

1. The transfers at substages are merged and just the total transferred listed.

2. The quota is listed only once, and hence if quota reduction takes place, one assumes that only the final quota is listed. Hence it will not be clear that quota reduction has taken place.

3. When a tie-break is required, there is no indication as to how this should be recorded (if at all).

4. In ERS97, a tie can be broken on the basis of (the first difference of) a substage result, but these results are not recorded on the result sheet.

Church of England, by computer

The Church of England regulations[3] do not specify in detail the form of the result sheet, but a pro-forma result sheet is provided by Church House. This is similar to the ERS97 result sheet and therefore does not list substages as above.

Items 2 and 4 of the previous case do not apply to the Church of England rules, and therefore the remaining issue is the manner for recording tie-breaks.

However, all the computer programs that conduct STV counts provide substantial detail on the actions performed — much greater than the typical result sheet. This includes the resolution of any tie-break. Hence one has a reasonable degree of transparency if the fullest form of computer output is available.

On the question of checking the computer software, the Church of England rules are relatively easy to program and the corresponding checking of the software is also manageable (at least without the facility of constraints which is not considered here).

Meek, by computer

The issues here are quite different from those with the two previous cases with hand-counting rules. The algorithm is defined[1], and hence the correctness of the software is relatively simple to address.

The problem is that at each stage, a computation is required which needs at least a Spreadsheet to handle with ease. Moreover, without any other information than the votes and keep values for each candidate at each stage, it is not possible (in general) to determine the preferences which gave the observed result. In other words, we have lost traceability to the actual ballot papers. (A similar situation arises with multiple exclusions with ERS97[2], but it is not so common.)

Other issues

Two questions a voter could reasonably ask need consideration:

What happened to my vote? In the case of hand-counting rules, a detailed knowledge of the rules is required as well as the result sheet. The rules are devised so that relatively few of the preferences given are used — this is deliberate to minimise the actual work involved in a count. Hence, in most cases, it is simple to trace the position of the paper amongst the piles of papers within the count. For the Church of England rules which does not allow multiple exclusions, it is more straightforward to trace your vote. It is even simpler with Meek[5], since at each stage, all the papers are re-considered. The formula using the keep values for each candidate gives the fraction of the paper going to each candidate. If issues of security could be resolved, a voter could interrogate the voting system to validate and trace his/her vote.

What if I changed my vote? This is similar to the last question except that if the change was sufficient to alter the decisions on election and elimination, then the subsequent stages would be in doubt. The uncertainty arises because preferences may then be inspected which were never examined before — and hence cannot be determined from the result sheet.

The Data Protection Acts of 1984 and 1999 imply that the candidates have some rights of access to the information about them contained in the preferential ballots. The 1984 Act is reasonably straightforward to follow and my view was that the candidate should be told, if a request is made, of the number of votes he/she attained in each preference position, assuming the data was held on a computer.

The 1999 Act is much more complex and very hard for anybody other than a trained lawyer to interpret. It does cover manual as well as computer counts. As I understand it, ERBS has never been asked for information under the Act nor has concluded what information should be disclosed.

Conclusions

The transparency of STV is nowhere near that of FPTP, regardless of the voting rules in use. Currently, it is not really possible for a voter to obtain the same level of understanding for an STV ballot as for FPTP. This is a serious loss, since in many cases, the impact of a single vote with STV could change the result and the voter should be aware of this. (Of course, this loss is more than compensated by the additional information STV uses.)

I conclude that the above should be rectified by two changes to current practice:

Preferences should be published if they contribute to the count.

This is not complete publication of the ballot papers. My own experience suggests that complete publication might allow some individual papers to be identified which would be contrary to the overriding need for a secret ballot. For the last remaining candidate, say, only the initial preference is inspected, and hence all that would be stated would be the total number of first preferences attained. Similarly, many papers differing in some preferences would be grouped together, since the differences were not used in the ballot.

It has been suggested to me that full publication would be possible for large elections in which the identification of a single paper would be more difficult. I have rejected this since it would imply an arbitrary decision as to when an election is *large*. Moreover, for large ballots, the published summary of the papers would be small compared with the total, and hence would not be an excessive requirement.

Full publication would also allow candidates to try other STV rules which would not necessarily encourage acceptance of the declared results. For some (small) elections, the summary proposed here for publication would be the complete data from the ballot papers. However, in this situation, it may well be possible to derive that information directly from the result sheet anyway, so formal publication could not be regarded as sacrificing ballot secrecy.

In the case of the Meek rules, the removal of the unseen preferences is undertaken as follows (where KV is the Keep Value of a candidate):

1. Remove all preferences for withdrawn candidates
2. For each eliminated candidate A, compute at the point of elimination, the set X of candidates having KV=1.0 (must be continuing or elected candidates). Remove all preferences for A that appear after any candidate within X (in each paper).
3. For those candidates B for which KV=1.0 at the end of the count, eliminate all preferences after B. (Hence a first preference for B will have only a first preference.)

Similar logic can be produced to remove unseen preferences for the hand-counting rules.

Joe Otten made an interesting comment about a witnessed count. If you could not go in person, could you provide your own copy of a vote-count program to *observe* the count? I think not, since it would provide terrible problems if the results did not agree, and the returning officer could not be expected to ensure that the provided program only undertook appropriate actions. (David Hill[4] made a similar point that the data should be available for people to run their own program.)

Internet facilities should be available for voters to determine what happened to their vote.

This would be simple to provide and can be made secure by means of a Java applet that runs on the voter's computer.

Assuming that the used preferences are available in an electronic format, then anybody would be able to re-run the election count with suitable software. This is surely as transparent as possible. The Internet facility would allow voters to understand the impact of their vote without having to be an expert in the particular STV rules in use.

Acknowledgements

The author received very detailed comments on an earlier draft of this paper from: David Hill, Michael Hodge, Joe Otten, Steve Todd and Joe Wadsworth. I have tried to take these comments into account, while not necessarily accepting all of them.

References

1. I D Hill, B A Wichmann and D R Woodall. Algorithm 123 — Single Transferable Vote by Meek's method. *Computer Journal*. Vol 30, pp277, 1987.
2. R A Newland and F S Britton. How to conduct an election by the Single Transferable Vote. ERS 3rd Edition. 1997.
3. Church of England, General Synod. Regulations for the conduct of elections by the method of the single transferable vote. GS1327. 1990 and 1998.

4. I D Hill. Are better STV rules worthwhile? *Voting matters* 3, Page 8. December 1994.

5. I D Hill. How was my vote used? *Voting matters* 7, Page 7. September 1996.