Political Network Electoral System

Jorge Jambeiro Filho

jorge.filho@jambeiro.com.br

Abstract

We present the political network electoral system, a new semi-proportional representation electoral system for members of parliament and councilors that replaces party-lists by a social network where each candidate chooses individually which other candidates can benefit from his or her votes. This allows the full liability of the former candidate for the latter and provides to the elector intuitive and difficult to disguise information about them. We show that the flexibility of the network can be restricted by rules created by each party and that, in an extreme case, the political network system can be reduced to a most open party-list system. This way, the system can be calibrated to comply with the preferences of candidates, parties and voters and has a good a chance of being useful in different contexts. The development of the proposed system required the creation of algorithms to carry proportional representation from political elections to social networks and, as a collateral effect, a new method for selecting sets of key players and simplifying networks emerged.

1. Letter to the reader

This paper goes till page 31. After that, there is only an appendix with mathematical proofs.

The paper was not peer reviewed so read it with care. It was submitted to five journals and they declined to review it. The most common reason for that was that the paper was out of scope. This is a contribution to politics from a computer scientist who is also a Brazil's Federal Revenue Fiscal Auditor. My main goal is to put an important fraud detection tool, which is social network analysis, at the service of the voters and make them more capable of separating candidates that are lying from those who are not. Apparently this does not fit well in political science.

To help you to decide if it is worth to read the paper, all rejection letters are copied bellow. Some editors wrote that they found the paper interesting. This does not mean they endorse its contents since none of them properly reviewed the paper.

In this version of the paper, a section about finding key players and simplifying social networks was included. This section was only present in the version that was submitted to Social Networks and not in the versions sent to the other journals which are more restricted to political science.

Some corrections were made based on the comments that came from the Journal of Theoretical Politics and a section explaining how the network can be constructed in steps was included.

Journal: ***Administrative Science Quarterly***

20-Mar-2017

Dear Dr. Jambeiro:

We have received and read your manuscript #ASQ-17-0098, entitled "Political Network Election System," which you submitted to the Administrative Science Quarterly. We appreciate your interest in the Administrative Science Quarterly. In the present instance, however, we feel that it is best to decline the opportunity to review your paper rather than sending it out to reviewers for comments and evaluation. At ASQ, we screen all manuscripts before initiating a review to ensure that each paper conforms to our editorial goals. We think that it is better to truncate the review process and spare authors lengthy delays when acceptance appears extremely unlikely. As detailed in our "Invitation to Contributors," the Administrative Science Quarterly publishes theoretical and empirical work on organizations that is grounded in the current literature on organizations and makes a significant addition to the literature. ***Your paper constructs and solves a mathematical model of a social network electoral system with properties that exceed, but can encompass, those of traditional party electoral systems. Although this is an innovative and rigorous study, the topic does not fall within the editorial scope of ASQ****.*

Submissions to ASQ should engage the theoretical and empirical literature on organizations and make significant contributions to it. This includes organizations that are part of the state, especially service-providing organizations such as health care systems and school systems, as well as state administrative systems. We could also include studies on political parties and how they function as organizations, and we do in fact have work on social movements, but our editorial domain does not include electoral systems and their properties. This delineation is because we see organizations as composed of people interacting in goal-driven activities, and electoral systems are designed to aggregate individual goals without direct individual interaction. In some ways they are the opposite of organizations because in electoral systems, individual goals are prior to collective goals (at least by design), and aggregation of individual goals happens without interpersonal influence.

Please do not interpret the decision to decline to review your manuscript as a comment on the merits of your work. We intend no such evaluation. This decision only reflects our judgment that your paper's objectives fall outside of ASQ's editorial domain. This paper clearly has potential in other journals closer to its goals. We appreciate your considering ASQ as an outlet for your work.

Best wishes,

Henrich

Dr. Henrich Greve

Editor, Administrative Science Quarterly

henrich.greve@insead.edu

Journal: ***Political Analysis***

02-May-2017

Dear Dr. Jambeiro Filho,

I write you in regards to manuscript # PA-2017-045 entitled "Political Network Election System" which you submitted to Political Analysis.

The editors have reviewed your manuscript. Based on this evaluation, at this point in time I must decline it for publication in Political Analysis and close the file on it for further consideration. ***While I found your paper interesting, it does not really break novel methodological ground (e.g., novel statistical tools). You paper, instead, develops a new voting system. I would suggest you send it to a journal such as the Journal of Theoretical Politics****.*

Thank you for considering Political Analysis for the publication of your research. I hope the outcome of this specific submission will not discourage you from the submission of future manuscripts.

Sincerely,

Jonathan N Katz

Co-Editor, Political Analysis

jkatz@caltech.edu

Journal: ***Electoral Studies***

13-May-2017

Title: Political Network Election System

Dear Dr. Jambeiro Filho,

Thank you for submitting your manuscript to Electoral Studies. Unfortunately, after reviewing your paper ***I feel that it is not suitable for publication in the journal*** *and is unlikely to be favorably reviewed by the referees. Accordingly, the manuscript is being returned without review.*

Thank you for giving us the opportunity to consider your work.

Kind regards,

Professor Clarke

Editor

Electoral Studies

Journal: Social Networks

10-jun-2017

Title: Political Network Electoral System

Dear Dr. Jambeiro Filho,

Thank you for submitting your manuscript to Social Networks. Unfortunately, after reviewing your paper ***I feel that it is not suitable for publication in the journal as it falls outside the scope of the journal****. Accordingly, the manuscript is being returned without review.*

Thank you for giving us the opportunity to consider your work.

Kind regards,

Professor Everett

Co-Editor

Social Networks

Journal: Journal of Theoretical Politics

19-Sep-2017

Dear Dr. Jambeiro Filho

I write you in regards to manuscript # JTP-2017-0953 entitled "Political Network Electoral System" which you submitted to the Journal of Theoretical Politics.

Unfortunately, we have been unable to procure any useful reviews on your manuscript*, so we have considered whether it is best to continue to seek reviewers or render a decision on the basis of our own reading.*

While we find your manuscript very interesting*, we are sufficiently pessimistic about the prospects for external review that we are declining it for publication at this stage. First and foremost, the manuscript is very hard to read due to grammar and syntax issues (for example, when you say "insensibility," you mean "insensitivity"). Second, based on our reading, many of the results are not very precisely stated (and accordingly, the proofs are impossible to verify). The literature on electoral systems is, as you know, vast, and* ***it is not clear how to properly situate this system in the literature*** *without more specification (for example, considering the process as simultaneous-move or sequential-move noncooperative game between the candidates or parties) of how the neighbors would actually be chosen (and, arguably, how voters would respond to them).*

I am very sorry that we could not procure a more detailed set of reports on the manuscript, but we suspect that our difficulty is at least in part due to these issues. We hope this feedback, as abbreviated as it is, is helpful as you pursue this research further.

Thank you for considering the Journal of Theoretical Politics for the publication of your research.

We hope the outcome of this specific submission will not discourage you from the submission of future manuscripts.

Sincerely,

John Patty

Coeditor, Journal of Theoretical Politics

jwpatty@uchicago.edu

2. Introduction

We present the political network electoral system, a new semi-proportional representation electoral system for members of parliament and councilors that replaces party-lists by a social network called the political network.

Proportional representation party-list systems are voting systems where the number of elected candidates in each party-list is approximately proportional to the total number of votes obtained by the party. There are two types of them: the open party-list systems and the closed party-list systems (Norris 1997).

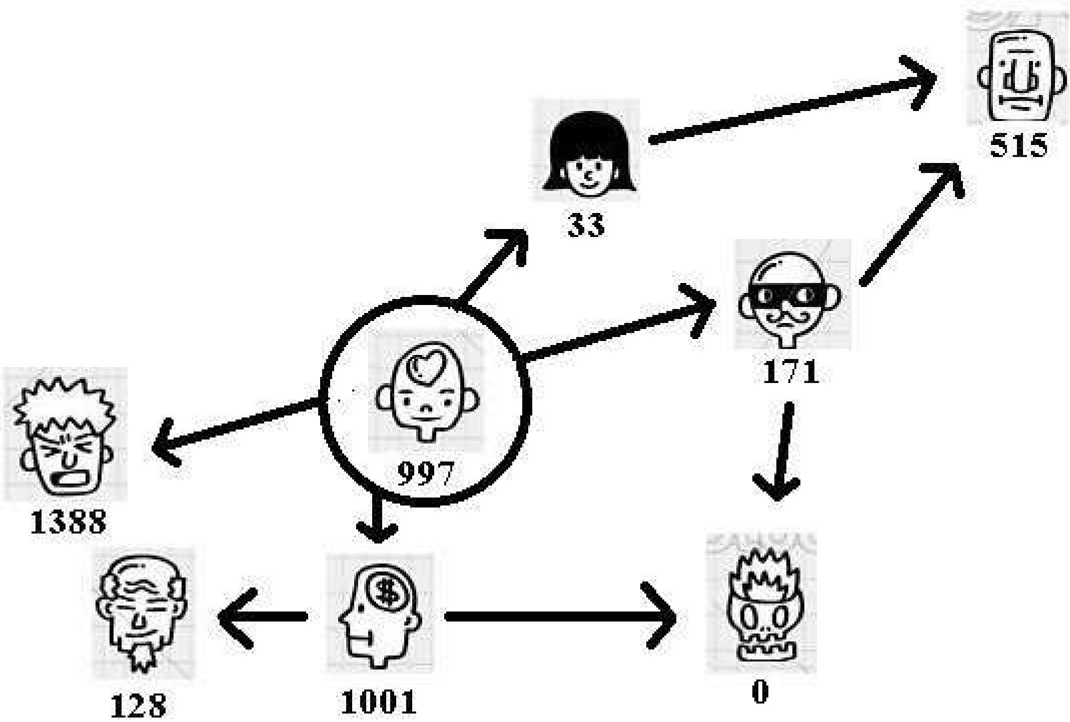
With closed party-lists, the order of election of individual candidates within each party-list is supplied by the party itself without any influence from the voters. With open party-list systems, such influence exists in different degrees. In the Netherlands, for example, the list is ordered by the party, but individual candidates that receive more votes than a certain threshold are ensured to get a seat, regardless of the original list order (Leenknegt and Schyff 2007). In Brazil (Brasil 1965), Finland (Ministry of Justice of Finland 2016) and Latvia (Hardman and Renwick 2012), the number of preference votes received by each individual candidate fully determines which candidates are elected within each party. We refer to these systems by the name of "most open party-list systems" as they are already referred in the internet (Wikipedia 2016) .

In the political network system, each candidate chooses a set of neighbors who are the other candidates for which he or she wants his or her votes to be transferred if he or she is eliminated from the election or has more votes than needed to be elected. He or she also indicates the percentage of votes that should be passed to each of them. Together, the choices made by all candidates build the political network.

We will show ahead, that a particular set of connection choices can reduce the political network system to a most open party-list system, characterizing the former as a generalization of the latter.

The faces in the neighborhood of a candidate bring more information to the voters than simple party lists, making it safer for them to vote for the candidate of their preference if they notice that, in case the candidate looses the election, the vote will be transferred to a few other reasonably good candidates than if they could only count on the information that any one among a hundred other candidates of the same party could benefit from the vote. On the other hand, a candidate that adopts a pleasant discourse but chooses to transfer votes to suspect candidates is also suspect as illustrated in Figure 1.

Figure 1: view of the neighborhood of candidate 997



In contrast to a candidate's discourse, that may be forged with impunity, his or her choices of neighbors have strong practical consequences in the election, what favors transparency and puts the candidates in a tight spot. They will have to confess their true relationships or pass votes to whom they don't want to.

A lot of information about the candidates like positions occupied in the past, amount of votes received in previous elections, parties to which they belong or belonged, the lawsuits to which they respond, heritage, heritage recent variation and votes against or in favor of any project discussed in prior incumbencies become much more remarkable if observed from the network's point of view.

In Figure 2, we show that the voters can have a bird's-eye view of the network with remarks of their own choice. In this view, most candidates are represented by light gray squares, but candidates who satisfied some positive criterion specified by a voter are displayed in a darker shade and candidates who satisfied some negative criterion are displayed in black. With such view, the electors can choose a candidate that is near a darker shade zone and far from black zones as an starting point to decide for whom they will vote.

Reasoning about the network using different points of views, as exemplified in Figure 1 and Figure 2, is a simple kind of social network analysis (Scott 2012). This kind of analysis is currently used to detect frauds both in private (Subelj, Furlan and Bajec 2011) and public sectors (Castro 2016) and, if the underlying structure of a voting system is a social network, can, very well, help voters to spot deceiving candidates and highlight interesting one's. Tools for social network analysis are, nowadays, pretty common (Wikipedia 2017) and voters don't really need to use them themselves, the candidates can do that for them and announce the results. Opponent candidates and political commentators can also help.

Figure 2 : bird's-eye view of an illustrative political network with voter highlights

There are no radical barriers among people ideas, but in the party-list systems two candidates either belong to the same party or do not. There are no degrees of proximity. In the political network system, the closer two candidates are in the network structure the more similar they are. Any candidates with ideas in common may transfer part of their votes to each other. This way, if a candidate that, for example, defends animals rights is eliminated from the election, the probability that another animal defender is elected will rise as a compensation.

In the political network system, when candidates are eliminated or have surplus votes, votes are reused by a process that relies on a network structure that was built by the candidates themselves. At first, giving electors more detailed control over their votes like in single transferrable voting systems (Tideman 1995) or cumulative voting systems (Pildes and Donoghue 1995) may look better. However, benefiting from such enhanced control requires acquiring more knowledge. Many voters don't think it is worth to do it and are, sometimes, described as rationally ignorant (Downs 1957). In practice, with single transferrable voting systems voters may fail to rank more than a few candidates and have their votes eventually wasted after they become non transferable (Tideman 1995). Even if they do rank many candidates, it is unlikely that they are well-informed in respect to the whole list, what can even result in "donkey voting" (Orr 2002) . Cumulative voting systems put an even heavier burden on the voters.

On the other hand, candidates are usually highly committed to an election and much more involved with politics. Thus, they can be expected to know much more about each other than the voters. It is true, that candidates cannot be trusted to defend the interests of the voters before they defend their own. However, in the political network system, candidates will be judged by the electors after they choose their neighbors and it will be dangerous for them to choose badly. It can be easier for the electors to approve or disprove candidates choices than making the choices themselves.

Voters misperceive where candidates stand on important issues, fail to recall relevant facts related to prior administrations and badly estimate the support of a candidate within particular social groups (Bartels 2008). Instead of accurate knowledge about politics they use information shortcuts to decide their votes. Some authors argue that, in the end, people vote rationally using such shortcuts (Popkin 1994), while others disagree (Shenkman 2009). The relations among candidates provided by the political network can become informative and reliable shortcuts to judge a candidate and help rationally low-informed electors to vote well.

The idea of allowing candidates to delegate their votes to other candidates in order to obtain transparency and simplicity for the electors without limiting their choices like closed party-list systems or single member district voting systems (Birch 2005) and in order to exploit the fact that candidates know more about each other than the voters is not entirely new and has lead to the conception of multiple-seat delegative elections (Ford 2002). However, this electoral system does not possess two essential mathematical properties of the political network system that will be presented ahead: the insensitivity to the order of eliminations and elections, that neutralizes an important source of randomness and unfairness in the election process and the guarantee of the minimum number of elected candidates for network solid coalitions, that is essential for a proportional representation system.

It is worth to mention a voting system that has been proposed for communities of users of electronic social networks (Boldi, et al. 2009). This specialized system does offer a mechanism to elect multiple representatives and, in principle, was a candidate to serve as a base for the construction of a political electoral system whose underlying structure is a social network. However, this system does not lead in a natural way to proportional representation. To achieve it, its authors suggest to detect the isolated components of the network or to build components using clustering methods. The components are then treated as parties or coalitions and traditional methods to obtain proportionality like the D'Hondt method (Gallagher 1991) are applied. In contrast, the political network system leads to proportionality naturally and a lower bound to the number of elected candidates for coalitions is established without the system having to treat isolated components specially or form clusters at all, indicating that this lower bound emerges from a general tendency to proportionality.

The system developed for electronic social networks cited above is partially based on PageRank (Page, et al. 1999), whose foundation is the convergence of an infinite process called random walk, which is, in its turn, an application of a Markov Chain (Kemeny and Snell 1976) . PageRank has also been employed to rank members of social networks (Heidemann, Klier and Probst 2010, Pedroche 2010) and was, itself, a candidate to be the base of the political network electoral system. However, PageRank was conceived to rank web pages, not to assign a fixed number of seats to politicians. Though there are similarities between the two process, there is at least one aspect that is important for political elections that would be completely ignored by a PageRank approach: in a proportional representation system, the eliminated candidates can transfer all their votes, but the elected candidates must retain a certain number of votes to back their own elections and can only transfer their surplus votes to other candidates. In single transferrable voting systems, this essential aspect is captured in the concept of an electoral quota. Having proportionality as a goal, the political network system also needed to have the notion of a quota inserted in its core and could not be grounded simply on a random walk process. Thus, we conceived another type of infinite processes that act over free networks and proved that they achieve proportionality.

3. Definition, basic properties and feasibility

We describe the political network electoral system through the following statements:

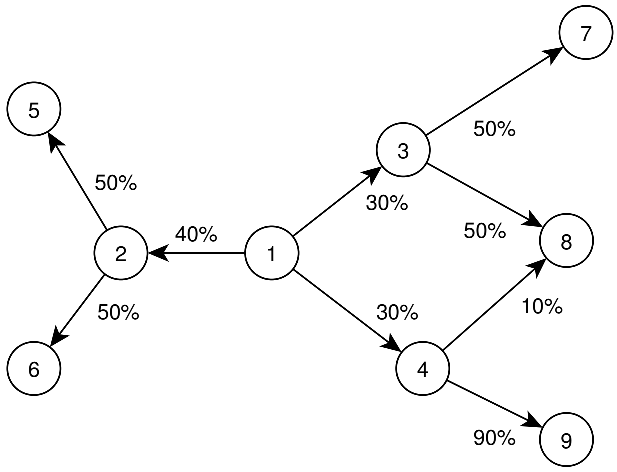
- before the election begins, each candidate chooses a set of other candidates as his or her neighbors and specifies the percentage of votes to be transferred to each of them in case he or she happens to have any transferable votes (note that neighborhood relations don't need to be symmetric);
- if a neighbor set is not empty, the percentages of transfer to each of its elements sum one;
- any votes in an eliminated candidate, received directly from voters or through transfers from other candidates are transferable;
- any votes beyond the current quota in an elected candidate, received directly from voters or through transfers from other candidates are transferable;
- any votes received by a remaining candidate, who is a candidate that has not yet been elected or eliminated, directly from voters or through transfers from other candidates are not transferable;
- the current quota is the current number of valid votes divided by the number of seats in dispute;
- the current number of valid votes is the number of valid votes minus the number of votes that already had to be discarded by the election procedure;
- a direct vote transfer is the division of a package of transferrable votes belonging to one candidate among his or her neighbors according to the percentages chosen by he or she;
- electors vote for a single candidate or for a party if they don't want to specify a particular candidate;
- votes for a party are treated as if the parties were virtual candidates who are eliminated in the beginning of the election and whose neighbors are their members, all of them associated to the same percentage of transfer;
- the election procedure is as follows:
    - eliminate party virtual candidates
    - transfer and discard votes
    - while there are remaining candidates
        - while there are remaining candidates with at least as many votes as the current quota

        - declare such candidates elected;
        - if there are no remaining candidates with more than zero votes and none of them can receive votes from any elected candidate or there are no remaining candidates with more than zero votes and all seats are filled, fill the remaining seats declaring some candidates with zero votes elected using any arbitrary criterion, eliminate all remaining candidates, terminate the election and exit
        - transfer and discard votes
      - if there are remaining candidates
        - eliminate the candidate with least votes
        - transfer and discard votes
- the procedure to transfer and discard votes is any procedure that, in a finite number of steps, results in the same number of votes in any candidate as a, possibly infinite, process that
  - never transfer or discard non transferrable votes;
  - changes the number of votes in any candidate only through direct vote transfers or vote discards;
  - converges to a state where
    - all transferable votes have reached remaining candidates except for the votes that cannot reach a remaining candidate through any number of direct vote transfers;
    - all transferrable votes that cannot reach a remaining candidate through any number of direct vote transfers have been discarded.

To visualize a vote transfer process, suppose that, in Figure 3, candidate 1 received 1000 votes from the electors, but latter was eliminated from the election. In this case, candidate 2 would receive 400 votes, while candidates 3 and 4 would receive 300 votes each. If division is not exact, fractions are used normally.

Imagine now that candidate 3, who chose candidates 7 and 8 as neighbors with percentages of 50% for each, received 2000 votes from voters. With the 300 that he or she received from candidate 1, he or she now has 2300 votes. If he or she is then eliminated candidates 7 and 8 will receive each one 1150 votes. Note that, of these, 150 votes came actually from candidate 1. So the votes of a candidate can help the neighbors of his or her neighbors, but they tend to do that with less intensity than they help the immediate neighbors.

Figure 3: structure for vote transfer around candidate 1



In the political network system, just like in single transferrable voting systems, the process of definition of the elected candidates involve dividing the sum of all valid votes by the number of available seats getting what is called the quota.

Candidates who have more votes than this quota are already elected and their surplus votes are transferred through the network.

In the example in Figure 1, if, rather than having been eliminated, candidate 3 had been elected and if the quota was of, for example, 1800 votes, candidate 3 would transfer 500 votes to his or her neighbors, instead of 2300. In this case, candidates 7 and 8 would receive 250 votes each.

After the transfers of all initial surplus votes, some other candidates may reach the quota and be elected immediately what causes their surplus votes to be transferred, possibly causing others to be elected too.

If there is no direct or indirect path going from a certain candidate to any remaining candidates, any votes in this candidate can never reach one of them. This happens, for example, because a candidate didn't choose any neighbors or because there is a closed sub network without any remaining candidates. In this case, the votes are discarded, that is, they are subtracted from the total number of valid votes and the current quota is adjusted accordingly, what also may cause other candidates to be elected.

Reductions in the quota when some votes cannot be transferred are also employed in single transferrable voting systems, for example, when Meek's method is applied (Hill, Wichmann and Woodall 1987). Surplus votes of already elected candidates that result from the reduction of the quota are transferred through the network normally.

When no more candidates can be elected this way, all remaining candidates are compared and the one with least votes is eliminated (if there is a tie, some arbitrary criterion should be used to break it). This candidate has his or her votes transferred to remaining candidates according to his or her neighbor set. The elimination of the candidate with least votes and transfer of his or her votes is also borrowed from single transferable voting systems.

The process continues always declaring those who have reached the quota as elected, eliminating the candidate with least votes and making the necessary vote transfers and adjustments in the current quota till all candidates have been either elected or eliminated.

In each step of the process, every time there are transfers for candidates that are already elected or eliminated they are transferred again till all the votes have been distributed to remaining candidates, except if that is impossible.

In the appendix, we show that the number of elected candidates indicated by the election procedure always match the number of available seats (Theorem 1).

The reader may worry about the fact that the definition of the system does not specify the exact order of vote transfers and some other implementation details. Indeed, if such details could affect the election result, further arbitrations would be required and they would hinder trust in the system. However, in the appendix lies the proof that, regardless of the order of vote transfers and other details, any implementation that respects what is written in the present section produces exactly the same set of elected candidates (Theorem 2).

Small differences in amounts of votes can affect the order in which candidates are claimed to be elected or eliminated. There can be concerns that this order could have big consequences to the final results introducing randomness and unfairness in the election. This could also lead candidates and voters to try to predict the order of elections and eliminations before, respectively, choosing neighbors and deciding votes, what would make their behavior more tactical than ideological. However, the order of elections and eliminations, by itself, has no consequences at all. At any point in the election process, after all required transfers have occurred, given a set of already eliminated candidates and a set of already elected candidates the number of votes in any candidate is guaranteed to be the same regardless of the order in which the elections and eliminations took place (Theorem 9).

The reader may also be worried about the practical feasibility of the system, since the presence of cycles in the network can lead to infinite sequences of vote transfers. We cannot work around this problem simply inserting barriers in the vote transfer process, like disallowing transfers to non remaining candidates or arbitrating a maximum number of transfers for each vote. Such barriers would lead to waste of votes and would break important properties like the insensitivity to the order of eliminations and elections and the guarantee of the minimum number of elected candidates for network solid coalitions that will be presented ahead. Fortunately, the result of the infinite processes of vote transferring can be calculated exactly, in a number of steps that is finite (Theorem 3) and acceptable for typical numbers of candidates, which are bellow 2000 (Theorem 6). The amount of memory required for that is not a problem also (Theorem 7). Using a personal computer with 4GB of RAM and an Intel Quad core, Q9550, we were able to run a simulated election with 2250 candidates in 25 minutes. We employed a network involving connections among all candidates what is the worst case, both for memory storage and execution time.

The calculation of the results of the infinite process is complex enough to require a computer, but with the use of an open source program, after the raw number of votes of each candidate is collect and publicized the algorithm to determine the elected candidates can be run at home, eliminating any suspicions that could be raised over closed systems.

The vote transfer process involves fractions, therefore we should worry about floating point rounding errors. For a small number of candidates, like, for example, 200, we can run the election process using infinite precision numbers and eliminate the problem. For elections with a larger number of candidates, finite precision numbers may need to be employed and there is the possibility that, in the moment of an elimination, the number of votes in the two candidates with least votes are identical till the last significant digit considered, but differ in the next digit. If this happens, a rounding error may cause the eliminated candidate not to be the most adequate. This condition is extremely unlikely since sixteen significant digits are typically used in computer systems and many more digits can be employed if desired. Moreover, if the two candidates are so close in their number of votes, none of them is really better from the point of view of the constituency and we can disregard this lack of precision and pick any of them.

In the appendix, we prove that rounding errors cannot break important properties of the system like the total number of elected candidates (Corollary 6) and the minimum number of elected candidates in a network solid coalition that we will discuss in the next section (Corollary 9).
To help the analysis of the network, also in the appendix, we show that it is possible to generate views where some candidates are removed from the role of middle men in any vote transfers (Theorem 8), making clear the relations among other candidates. A view that shows the relationships between any candidate and all incumbent candidates, for example, would eliminate inexpressive candidates and could be quite insightful.

4. Steps in the formation of a political network

To choose their neighbors the candidates will want to know who will be the neighbors of such neighbors. To avoid misunderstandings or that one candidate can cheat on the others, we propose that the neighborhood definitions are made in steps.
Bellow, we present an example sequence of steps that could be used:

1. definition of who will be the candidates;
2. first free choice of neighbors;
3. second free choice of neighbors: in this step it is already possible to consider the neighbors of the neighbors;
4. first neighborhood adjustment: in this step it is not possible to add neighbors, just to eliminate them or to change percentages of transfer;
5. second neighborhood adjustment: in this step percentages can only be changed in 50% of their original values;
6. third neighborhood adjustment: in this step percentages can only be changed in 25% of their original values;
7. forth neighborhood adjustment: in this step percentages can only be changed in 10% of their original values.

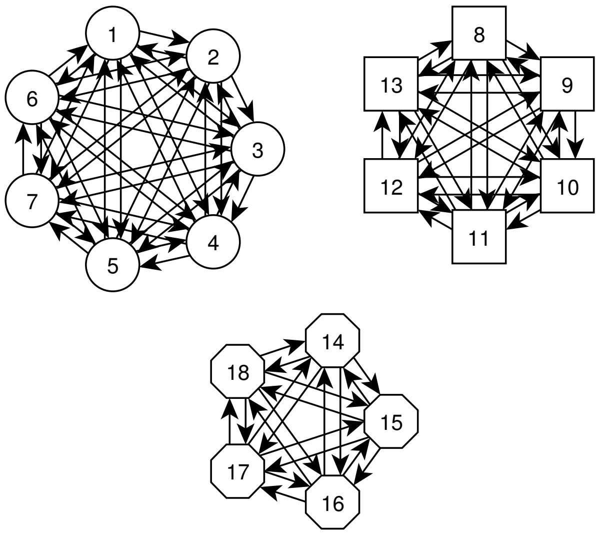
We don’t think that the details of the sequence presented above are important as long as the network is formed gradually and each candidate has the opportunity to react to the choices of the others.

5. Relation to most open party-list systems

It is convenient to notice that the political network can be viewed as a generalization of a most open party-list system, that is, there is a particular political network structure that presents the same key properties as a system of this type. We call this particular structure, the party-list network structure.

The party-list political network structure is a structure where each party corresponds to a party-list coalition containing all its candidates. A party-list coalition is a subnetwork where all candidates of the corresponding party are the only neighbors of each other and the percentages of transfer are all identical. In Figure 4, we show an example of such structure. For simplicity we don't indicate percentages of transfer.

Figure 4: political network restricted to party-list coalitions



The main property of proportional representation party-list systems is that the number of elected candidates in each party-list is proportional to the sum of the number of votes obtained by the party as a whole. The distinguishing property of most open party-list systems is that the elected candidates of any party are those that received more votes individually.

In general, party-list systems cannot guarantee perfect proportionality and they vary in how they approach such goal (Gallagher 1991). The party-list network structure does not mimic any established party-list system. Instead it guarantees that:

1. the number of elected candidates of each party is at least equal to the floor of $\frac{V_p \cdot M}{V}$, where V_p is the number of votes obtained by the party, V is the total number of valid votes and M is the number of seats available, as long as the

party has enough candidates for that (see appendix, Theorem 4 and Corollary 10);

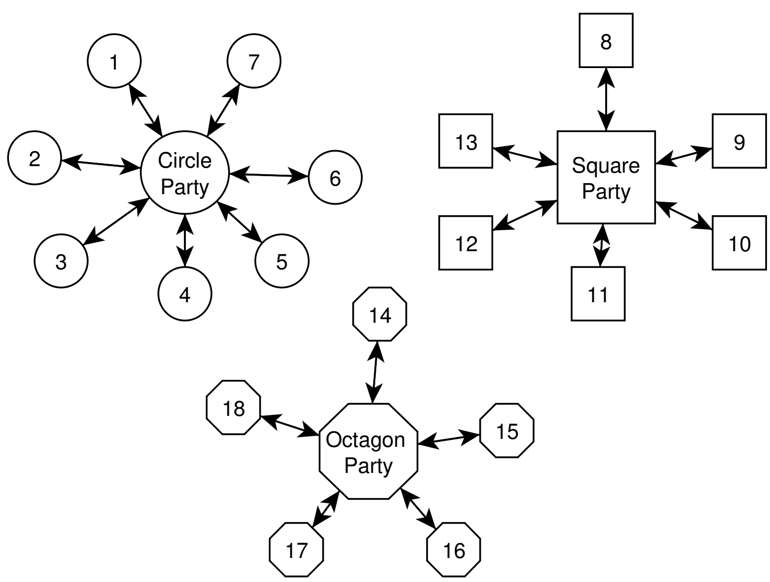
2. the elected candidates of each party are exactly those who received more votes individually (see appendix, Theorem 5).

It is worth asserting that the cited theorems also show that if all members of a party, ***P***, form a party-list coalition completely isolated from the rest of the network while candidates of other parties do not, the party ***P*** can still count on the two guarantees above.

Guarantee 1, the guarantee of proportionality or guarantee of the minimum number of elected candidates throws away the fractional part of the party's vote share and is, thus, imperfect. However it is the same guarantee that exists for solid coalitions in single transferable voting systems (Dummett 1984) and that has been pointed out to be the justification to describe single transferable voting systems as proportional systems (Tideman 1995).

The political network system can, thus, be used as a proportional representation electoral system. However, if, for example, no candidate chooses any neighbors at all, there will be no proportionally and the winning candidates will simply be the most voted one's. Single transferrable voting systems also can be used in proportional or non-proportional ways and are called semi-proportional (Norris 1997). Therefore we also say that the political network system is a semi-proportional representation system.

Figure 5: party-list structure with party virtual candidates

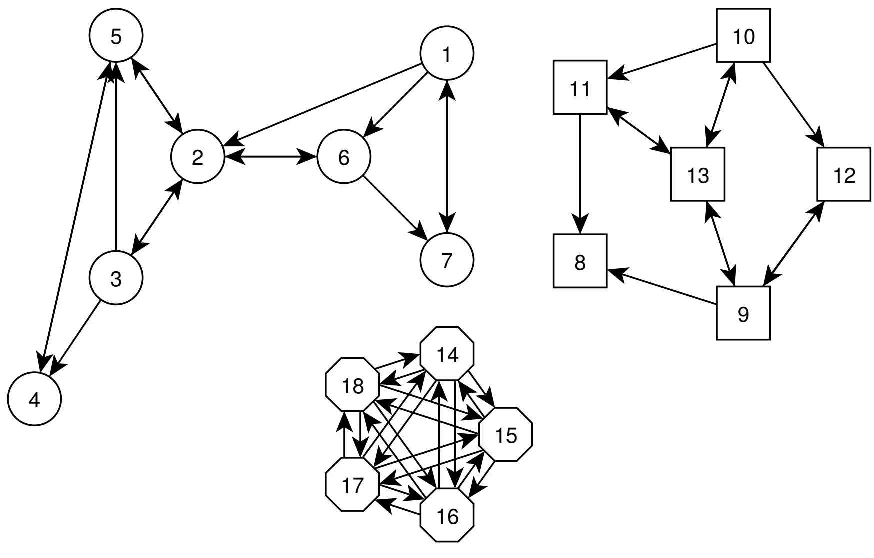


Parties are treated as virtual candidates that can be voted, but not elected. The party virtual candidates' neighbors are simply their members and the percentages of transfer to them are all identical. These virtual candidates are eliminated in the beginning of the election, what causes their votes to be divided among the neighbors. This allows our generic algorithm for vote transferring to handle non preferential votes unchanged and also simplifies the party-list structure, like in Figure 5.

The party-list structure mimics most open party-list systems, however, once the political network system has been adopted, the network structures may evolve in ways that will depend on the candidates, on the parties and on the voters. Candidates can potentially choose their neighbors individually, but each party can create its own rules restricting what their members can do. Electors, of course, can consider the choices made by parties and candidates before deciding their votes.

At least, a candidate could avoid transferring votes to party colleagues that were involved in any scandals. They could also easily point out which members of their own parties they really trust, like in Figure 6.

Figure 6: political network with free connections within parties



Note that, the party represented by octagons chose not to use the extra freedom offered by the political network and formed a party-list coalition. As stated, this party still counts on guarantees 1 and 2 of party-list structures.

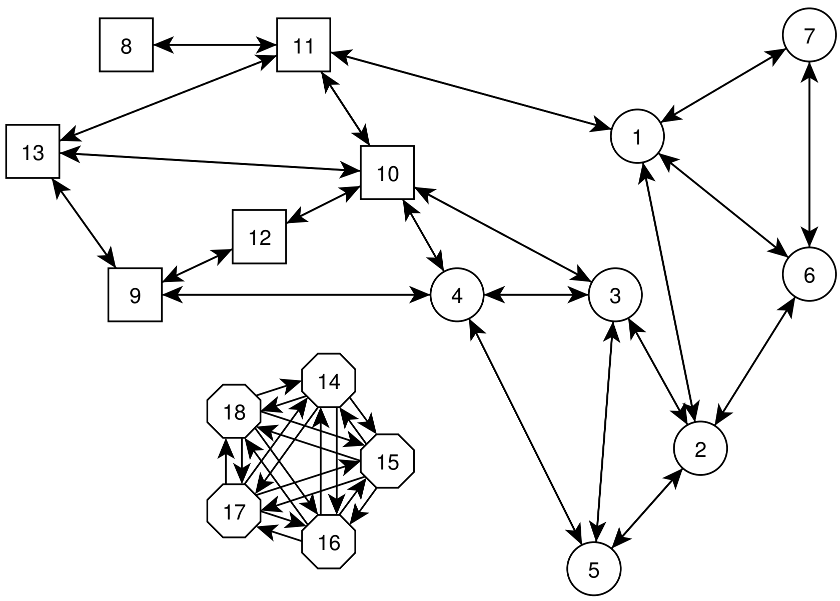
The candidates of the party represented by circles do not form a party-list coalition, but they form what we call a network solid coalition. In a network solid coalition, votes can go indirectly from any candidate to any other candidate in the coalition and no votes can go to candidates outside it. The guarantee of the minimum number of elected candidates holds for network solid coalitions, what is very much like what happens with solid coalitions in single transferable voting systems. On the other hand, because the circle party members are now expressing preferences in respect to other members, the exact candidates that will be elected are not anymore guaranteed to be the ones that received more votes individually.

If they want, parties may only allow its members to define a small part of their percentages of transfer to specific members. A percentage of, let's say, 90% could be enforced to go to the party (through a party virtual candidate) and thus be evenly shared by all other members. Even with reduced power, individual candidates choices could still bring useful information to the voters.

On the other hand, parties may do the opposite and allow members to freely decide how they will transfer almost 100% of their votes, but still force a non-zero percentage of votes, doesn't matter how small, to go to the virtual party candidate. In this case, the party will still form a network solid coalition and count on the guarantee of the minimum number of elected candidates.

Latter, some parties may colligate with other parties as seen in Figure 7.

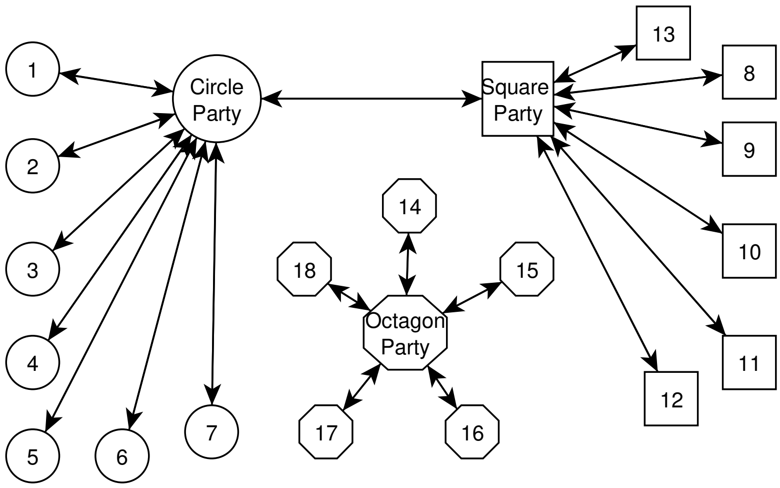
Figure 7: political network with free connections within some parties



If they prefer, parties can make weaker colligations, like in Figure 8. The connection between the square party and the circle party, can involve a very low percentage of transfer, in such way that almost 100% of the votes of each party would remain within it except if it had no more remaining candidates. In this case votes would be automatically inherited by the other party. Thus, a party could express preferences for other parties without really mixing with it.

If a political network system replaces a most open party-list system, it is either possible that all parties merge into a fully flexible political network or that, basically, nothing happens and the network stays still with a party-list network structure. However, the network may also evolve to semi flexible structures where different parties form coalitions with different characteristics. What will actually happen, depends on the actions of parties and candidates which, in turn, will be submitted to the surveillance of the voters.

Figure 8:limited party colligations



6. Vote transfer processes

The formal proof that there are vote transfer processes satisfying the definition of the political network system and that they all converge to the same result is left to the appendix. In this section we present only key ideas.

Cycles in the network generate an infinite sequence of vote transfers. While the votes circulate, some of them reach remaining candidates where they stop, but some reach non remaining candidates from which there is no path to any remaining candidates and they are, eventually, discarded. Vote discards push the quota down, creating a surplus of votes in all elected candidates and possibly causing more vote discards. While the number of votes in all elected candidates is not equal to the current quota, the process cannot stop. Moreover, we avoided the introduction of details in the definition of the system and the precise moment of any transfers or discards is free to vary. Besides that, nothing in the definition prevents vote packages from being split into smaller packages and it is optional to join packages that came from different paths, but currently belong to the same candidate and transfer them together from this point on or not to do so. Thus, vote transfer processes are not so regular and we could not prove that they all converge to the same state of equilibrium simply calculating limiting properties of a Markov chain, like eigenvectors or absorbing probabilities.

The reason why we allowed such a degree of freedom in the vote transfer processes is that if we had adopted any particular policy for the vote transfer sequence, we would have to defend that it was the best from a political point of view, what would be impossible. A better alternative was to face the complications derived from the granted freedom and prove that any processes that satisfy the minimum requirements for the vote transfer process in the definition of the political network system converge to the same state of equilibrium.

If all non remaining candidates only had remaining candidates as neighbors, there would be no infinite vote circulation. Votes would be transferred once and stop. With this simpler structure we would be able to algebraically correlate the quota in the end of the process to the number of discarded votes and calculate them both solving equations.

Our goal is, thus, to remove from the network all edges going to non remaining candidates, without modifying the result of the vote transfer process.

Once we know how to remove one edge without changing the result of interest, it becomes easy to remove them all. Procedure 9 of the appendix does this job, removing an edge going from a candidate C_i to a candidate C_k, that is, it removes C_k from the neighbor set of C_i, where C_k is a non remaining candidate. The candidate C_i can be any candidate, but since his or her outgoing edges would only be used after he or she has ceased to be a remaining candidate, C_i, can be treated as if he or she was a non remaining candidate too and Procedure 9 acts accordingly.

We show a copy of this procedure bellow, where $\widehat{\boldsymbol{Z}}_j$ is the set of current neighbors of a candidate C_j and $\hat{P}_{lj}$ is the current percentage of votes that go to a candidate C_j if a direct vote transfer is originated in a node C_l. If C_j is not a neighbor of C_l, $\hat{P}_{lj}$ is considered to be zero.

The virtual discard candidate C_0, was created with the role of receiving the votes that should be discarded. Before starting the election process, all candidates without any neighbors, whose transferrable votes, therefore, would always be discarded, have C_0

inserted in their neighbor set with a percentage of transfer equal to one. Procedure 9 counts on that.

If C_i and C_k were the only neighbors of each other, any votes going fromC_i to C_k would circulate forever. They cannot reach a remaining candidate and can be discarded immediately. Making the neighbor set of C_i equal to $\{C_0\}$, achieves exactly that.

Otherwise, votes that would go from C_i to C_k would latter go to the neighbors of C_k, who, after the edge removal, must receive the votes directly from C_i. Thus, C_k is replaced in the neighbor set of C_i by his or her neighbors.

Procedure 9: $\boldsymbol{RemoveFromNeighborSet}(\boldsymbol{C_k}, \boldsymbol{C_i})$

1. If $\widehat{\boldsymbol{Z}}_i = \{C_k\}$ and $\widehat{\boldsymbol{Z}}_k = \{C_i\}$
   a. $\widehat{\boldsymbol{Z}}_i \leftarrow \{C_0\}$
   b. $P_{i0} \leftarrow 1$
2. Else
   a. $\widehat{\boldsymbol{Z}}_i \leftarrow \widehat{\boldsymbol{Z}}_i - \{C_k\} + \widehat{\boldsymbol{Z}}_k - \{C_i\}$ // eliminates C_k from the neighbor set of C_i, $\widehat{\boldsymbol{Z}}_i$, but adds all neighbors of C_k to $\widehat{\boldsymbol{Z}}_i$, excepts for C_i itself.
   b. $\forall j | C_j \in \widehat{\boldsymbol{Z}}_i$
      i. $\hat{P}_{ij} \leftarrow (\hat{P}_{ij} + \hat{P}_{ik} \cdot \hat{P}_{kj})/(1 - P_{ki} \cdot P_{ik})$ // corrects the percentages of transfer to each neighbor

The percentages of transfer must, of course, be adjusted. If C_i is not a neighbor of C_k, it is easy to see that for any C_j, $\hat{P}_{ij} \leftarrow (\hat{P}_{ij} + \hat{P}_{ik} \cdot \hat{P}_{kj})$ does the job. The percentage to go from C_i to C_j is just incremented by the amount of votes that would go from C_i to C_k and then from C_k to C_j.

If C_i is a neighbor of C_k, we can focus on the votes going back and forth between C_i and C_k and observe that, while this happens, some votes keep going to their other neighbors. The percentage of the first transfer from C_i to a neighbor C_j is $\hat{P}_{ij}$ and in every bounce the amount of votes that go from C_i to C_j, is smaller than in the previous bounce. It is not difficult to see that the rate of the decrease is given by $P_{ki} \cdot P_{ik}$. The percentage of the first transfer from C_k to C_j is $\hat{P}_{ik} \cdot \hat{P}_{kj}$ and in every bounce, the amount of votes that go from C_k to C_j, is also decreased at a rate given by $P_{ki} \cdot P_{ik}$. So the total amount of votes that go to C_j is the sum of two geometric series. Algebraic manipulations lead to the formula $\hat{P}_{ij} \leftarrow (\hat{P}_{ij} + \hat{P}_{ik} \cdot \hat{P}_{kj})/(1 - P_{ki} \cdot P_{ik})$.

Of course, the sequence of back and forth transfers between C_i and C_k would not occur with a clear separation from all other transfers and discards. Instead, it would be obfuscated by intercalated transfers between other candidates, by possible vote package splits and mainly by joins with other vote packages that could reach candidates C_i and C_k. In the appendix, we handle this details and in Lemma 9 we formally prove that the use of Procedure 9, indeed, cannot change the result of the vote transfer process.

Note that the edge removal process does not differentiate elected from eliminated candidates. They are both just intermediate nodes, while the remaining candidates are the ultimate targets of all the flow in the network. We can say that the network structure modified by Procedure 9 is equivalent to the original structure in respect to the flow of votes toward the remaining candidates.

Once we have removed all edges going from non remaining candidates to other non remaining candidates, all transfers become direct to remaining candidates and all vote discards become represented by a transfer to C_0. Thus, we can write:

$$\tilde{V} = \sum_{\forall C_i \in \boldsymbol{L}} (\hat{V}_i - \dot{Q}) \cdot P_{i0} + \sum_{\forall C_i \in \boldsymbol{E}} \hat{V}_i \cdot P_{i0},$$

$$\tilde{V} = \hat{V} - M \cdot \dot{Q},$$

where $\tilde{V}$ is the number of votes that will be discarded in the vote transfer process, $\hat{V}_i$ is the number of votes in candidate C_i before the vote transfer process and P_{i0} is the percentage of transfer from C_i to C_0, which represents the percentage of transferrable votes of C_i that should be discarded. At the same time, $\boldsymbol{L}$ is the set of all elected candidates and $\boldsymbol{E}$ is the set of all eliminated candidates. $\hat{V}$ is the number of valid votes before the vote transfer process, M is the number of seats in the dispute and $\dot{Q}$ is the current quota after the vote transfer process.

The equation system is linear and has only one solution, that is given by

$$\dot{Q} = \frac{\hat{V} - \sum_{\forall C_i \in \boldsymbol{L} \cup \boldsymbol{E}} \hat{V}_i \cdot P_{i0}}{\left(M - \sum_{\forall C_i \in \boldsymbol{L}} P_{i0}\right)}.$$

This way, we can obtain the same result that would be achieved by the infinite process in a finite and deterministic way. Since the obtained result does not depend on details about the original infinite vote transfer process, like the order of transfers and discards or the occurrence of joins and split of vote packages, that proves that any suitable infinite vote transfer process converges to the same state of equilibrium.

In practice, we don't need a full edge removal process for each vote transfer process. We only need to remove all edges going to candidates that have just been elected or eliminated and proceed using the modified structure from this point on.

Having only direct transfers to handle and counting on a formula to update the current quota, the proofs of the properties of the political network system like the insensitivity to the order of elections and eliminations and the minimum number of elected candidates for network solid coalitions become much easier, but are still laborious and are left to the appendix.

7. Selection of key players and social network simplification

In the political network electoral system, N candidates form a social network and the system picks M of them to fill available seats in some political house. The M candidates are chosen in order to be a proportional representation of the constituency that voted for them. Other electoral systems also target proportional representation, what recognizes it as an important goal, but they work over much more rigid structures like lists, not arbitrary graphs. So, the political network electoral system extends proportional representation to social networks.

The key players of a social network are the nodes that are considered "important" with regard to some criteria (Ortiz-Arroyo 2010). If we consider proportional representation to be this criteria, we will get very close to the problem that is solved by the political network electoral system. We only need a few adjustments to "elect" key players that offer a proportional representation of the network.

Let's consider every node in a social network to be a candidate. If there are measures of the strengths of the connections between nodes, as long as all outgoing edges of each node are normalized to sum one, they can be used to define the percentages of vote transfer. Otherwise, uniform percentages of transfer may be adopted.

If there are any weights associated to the network nodes, they should be used as the initial number of votes for each candidate. Otherwise, the simplest alternative is just to give one vote to each node. In this case, because every node starts exactly with the same number of votes, it is important use some centrality measure as a tie breaker.

If we want, we may not consider all nodes of the network to be candidates, but only those that satisfy some specified restrictions. In this case, we should eliminate all other nodes from the election in the way we usually eliminate virtual party candidates.

Let's compare this method of selecting key players to alternative strategies with an example. To make it simple, suppose that the network has 4 completely isolated components. The first has 60 nodes, being 3 of them connected to every other node in the component and all others connected only to this 3 central nodes. The second has 40 nodes, one them connected to every other node in the component and the others connected only to the central node. The third and forth components have 25 nodes each and also follow the pattern of having one central node connected to every other node and the others connected only to the central node. Let's suppose that we can afford only 3 key players.

Any selection method that is based on centrality measures (Freeman 1978) and that does not worry about redundancy will tend to either pick the 3 nodes of the big component or none of them. Since these 3 nodes are structurally identical, they will draw in any centrality measure. If one goes to the top of the rank, the others will go too. If one goes to the bottom, so do the others. Clearly, neither of these two alternatives is satisfying.

Borgatti (2006) recognized that simply filling the set of key players with the most central individuals does not suffice and articulated two different goals for the set of key players considered as a whole. The first, called KPP-Pos is related to optimally diffusing something through the network using the key players as seeds, while the second, called KPP-Neg, is related to maximizing the fragmentation the network with the removal of the key players. None of these goals is equivalent to finding a proportional representation of the network, but the first is somewhat closer and it is the only one we will consider here.

The algorithm proposed by Borgatti to solve KPP-Pos, searches for the set of key players, $\boldsymbol{L}$, that maximizes the sum of the inverses of the distances from $\boldsymbol{L}$ to each node outside it. The distance from $\boldsymbol{L}$ to any node outside it is considered to be the minimum distance from any member of $\boldsymbol{L}$ to the node.

For such an algorithm, picking a second central node from the big component in our example problem accomplishes nothing and, so, it chooses just one of them. The set of 3 key players is completed with one center node from the medium-sized component and one center node from any of the small ones.

Ortiz-Arroyo (2010) tackled KPP-Pos using a different strategy, which was to pick as key players the nodes whose individual removals would cause the greatest reduction in the connectivity entropy of the network as a whole. As shown in the paper, connectivity entropy does account for the existence of redundant nodes in the network. However, before picking a particular node as a key player, the proposed algorithm never considers if the redundant nodes had themselves been picked or not. As a consequence, each of the 3 central nodes of the biggest component in our example problem receives a low score for being redundant with the others and the component ends up without any key players at all.

Let's check what happens in our example when the political network electoral system is used to select the key players. The quota starts at 150/3=50 votes. Nobody has votes enough to be elected yet and the system proceeds eliminating the peripheral nodes of all components. After only the central nodes remain, no votes have been discarded, the central nodes of the biggest component have 20 votes each, the central node of the smallest components have 25 votes each and the central node of the medium-sized component has 40 votes. One of the central nodes of the big component is eliminated, what causes the other two to inherit its votes. With 30 votes each, they remain in the dispute while the central nodes of the smallest components are eliminated. With the vote discards that occur as a consequence of the elimination of the last nodes of the smallest components, the quota is reduced to 100/3 votes and the central node of the medium-sized component is elected. The surplus votes of the just elected candidate are discarded and the quota is reduced to 30 votes, causing the two remaining nodes to be elected.

In our example, Borgatti's solution, avoids redundancy in the set of key players, picking at most one central node from each component. If the biggest component in the example had 1000 nodes instead of 60, nothing would change.

The political network electoral system chooses 2 nodes from the big component, 1 from the medium-sized component and none from the smallest ones. If the big component had 1000 nodes, all of its 3 central nodes would have been picked. Under this criterion, some redundancy is accepted to represent better a big number of nodes or having more influence over them. The idea here is that one node cannot really play the role of three even if it has all required connections. That may or may not be true depending on the goal.

There are few methods for selecting sets of key players in social networks that go beyond centrality measures and the political network electoral system gives rise to a new one. We have not deeply investigated for which applications this method would be good, but we list two contexts where it could be used to, at least, provide some insight.

In the political network election process, a candidate transfers votes to others when he or she is eliminated or when he or she is already elected and has a surplus of votes. In disputes for relevant positions within an organization, potential contenders that don't believe to be strong enough to win, may withdraw and support someone else. At the same time, someone who has easily guaranteed his or her space may, very well, help a comrade to get a seat at the table too, showing that informal disputes somewhat resemble proportional representation elections. Since organizations can be modeled as social networks, the political network election system can be used to investigate the balance of power within them and find unofficial or future leaders. Of course, what is going to be used to define the strengths of the relations among people and what will count as the initial number of votes for each of them is domain specific and will be decisive for the relevance of any conclusions.

Now suppose that we are analyzing an economy. We could use the sizes of the companies as votes and commercial or similarity relations as edges to start a political network election. The election would distribute the whole economy evenly among key players (each of them would stand for a share that would be exactly equal to the value of final quota), what can provide an interesting view. Some of this key players would be there just for their own size. Others would be representing sets of related small companies, whose weights would have been transferred to their representatives. Some other players, would have been selected for having received the surplus weight of some very big companies.

We can obtain some extra enlightenment simplifying a network using key players. Such players could have been selected by a network election process, by any other automatic method or even having been picked manually. To make the simplification, we can look at the key players as the ultimate targets of all the flow in the network, what, in an election process, is the role of the remaining candidates. We can then eliminate all edges going to non key players using Procedure 9 and obtain a structure that is equivalent to the original one in respect to the flow toward the key players. Only edges going to key players would remain in the simplified network, but their strengths would reflect all removed ones. The indirect relations among key players would become explicit. The indirect relations between any individual non key player and any key player, would become explicit too. The modified structure would offer a simplified way to understand the role of any node in the network based on how it relates to the key players.

The time complexity of the election algorithm (see appendix, Theorem 6) is $O(M \cdot N^2 + N^3/K)$, where N is the number of candidates, K is the number of processors and M is the number of seats available (or the number of key players). This is cubic on the number of candidates (or number of nodes) and too slow for big social networks. Thus, an approximation may be necessary. In the N^3/K term, a quadratic factor is due to the number of possible neighbors of each candidate, that can be as big as the whole network. In the $M \cdot N^2$ term, a linear factor is due to the same thing. If we accept to drop the weakest links, we can limit the number of neighbors of any candidate to a constant U. In this case, the complexity of the algorithm becomes $O(M \cdot N \cdot U + N \cdot U^2/K)$, what is linear in the number of nodes and thus much more manageable.

8. Conclusion

We presented a new electoral system where party-lists are replaced by a fully flexible social network as a vote transfer mechanism. This system makes relationships among candidates explicit and thus allows their full liability for those who can benefit from their votes and provides to the electors intuitive and difficult to disguise information about them. We are not aware of any electoral system that induces a greater degree of transparency.

This system, neither increases the complexity of the act of voting like single transferrable voting systems, nor restrict voters options like closed party-list systems or single member district voting systems, allowing them to choose from possibly large ranges of candidates.

It is also relevant that the political network system can mirror the key properties of a most open party-list system through a particular set of neighbor choices, which can be freely made by the candidates or imposed by the parties, what reserves to the latter a more important role than in single transferrable voting systems. Moreover, there is a smooth path from this restricted network to a fully flexible one and it is even possible that semi flexible networks turn out to be the best structures for several electoral systems. Such semi flexible networks may reflect competing party policies that can evolve under the pressure of voters and that can benefit from the convenient guarantee of the minimum number of elected candidates for network solid coalitions.

The insensitivity of the number of current votes in any candidate to the order of prior eliminations and elections during the whole election process is also a pleasant mathematical property that prevents the introduction of randomness and unfairness in the election and eliminates, what would otherwise be, an important trigger for tactical voting. This reinforces our belief that political networks can be a useful instrument in different contexts.

As a collateral effect, the political network electoral system gives rise to a new method for selecting key players in social networks, whose goal is to find proportional representations of them. Besides that, the edge removal algorithm developed to handle vote transfer processes engenders a mechanism to transform the network into a simplified version where only relations involving key players exist, but where the strengths of such relations have been updated to reflect all relations among non key players in the original network. We listed two possible uses for these resources, but did not go deep into them. We leave this task as future research.

References

Bartels LM (2008) The irrational electorate. *The Wilson Quarterly* (Woodrow Wilson International Center for Scholars) 32, no. 4: 44-50.

Birch S (2005). Single-member district electoral systems and democratic transition. *Electoral Studies* 24, no. 2: 281-301.

Borgatti, Stephen P. "Identifying sets of key players in a social network." *Computational & Mathematical Organization Theory* 12, no. 1 (2006): 21-34.

Boldi P, Bonchi F, Castillo C and Vigna S (2009). Voting in Social Networks. *Proceedings of the 18th ACM conference on Information and knowledge management* (ACM) 18: 777-786.

Brasil (1965). " Lei 4.737." 15 de 7 de 1965.

Castro RC (2016). Identificação do Real Beneficiário e Busca Patrimonial por Grafos de Relacionamento no ContÁgil – Script "Análises Patrimoniais - SABRE". In: *Prêmio de Criatividade e Inovação da RFB*.

Downs A (1957). *An Economic Theory of Democracy.* 1st. New York: Harper and Row.

Dummett MAE (1984). *Voting Procedures.* 1st. Oxford: Clarendon Press.

Ford, B (2002). Multiple-Seat Delegative Elections. *Bryan Ford.* http://www.brynosaurus.com/log/2002/1021-DelegativeVoting/multi.html (accessed 2/19/2017).

Gallagher M (1991). Proportionality, disproportionality and electoral systems. *Electoral studies* 10, no. 1: 33-51.

Hardman H and Renwick A (2012). Latvia. *Electoral System Change in Europe since 1945*. http://www.electoralsystemchanges.eu/Files/media/MEDIA_151/FILE/Latvia_summary.pdf (accessed 3/5/2017).

Heidemann J, Klier M and Probst F (2010). Identifying key users in online social networks: A pagerank based approach.

Hill ID, Wichmann BA and Woodall DR. (1987). Algorithm 123, single transferable vote by Meek's method. *Computer Journal* 30, no. 3: 277-281.

Kemeny JG and Snell JL (1976). *Finite Markov Chains*. New York: Springer.

Leenknegt G and Schyff G (2007). Reforming the Electoral System of the Dutch Lower House of Parliament: An Unsuccessful Story. *German LJ* 8: 1133.

Ministry of Justice of Finland (2016). *Determination of the Election Results.* http://vaalit.fi/en/index/onelections/parliamentaryelections/resultinformation/determinationoftheelectionresults.html (accessed 3/5/2017).

Norris P (1997). Choosing electoral systems: proportional, majoritarian and mixed systems. *International political science review* (Sage Publications) 18, no. 3: 297-312.

Orr G (2002). Ballot order: Donkey voting in Australia. *Election Law Journal* 1, no. 4: 573-578.

Ortiz-Arroyo, Daniel. "Discovering sets of key players in social networks." In *Computational Social Network Analysis*, 27-47. Springer, 2010.

Page L, Brin S, Motwani R and Winograd T (1999). The PageRank citation ranking: Bringing order to the web. Stanford InfoLab.

Pedroche F (2010). "Ranking nodes in social network sites using biased PageRank." *2º Encuentro de Álgebra Lineal Análisis Matricial y Aplicaciones, ALAMA-2010.* Universidad Politécnica de Valencia.

Pildes RH and Donoghue KA (1995). Cumulative voting in the United States. *University of Chicago Legal Forum*: 241-313.

Popkin SL (1994). *The reasoning voter: Communication and persuasion in presidential campaigns*. University of Chicago Press.

Scott J (2012). *Social network analysis*. Sage.

Shenkman R (2009). *Just how Stupid are We?: Facing the Truth about the American Voter*. Basic Books.

Subelj L, Furlan S and Bajec M (2011). An expert system for detecting automobile insurance fraud using social network analysis. *Expert Systems with Applications* (Elsevier) 38, no. 1: 1039--1052.

Tideman N (1995). The single transferable vote. *The Journal of Economic Perspectives* 9, no. 1: 27-38.

Wikipedia (2016). "Open list." *Wikipedia, The Free Encyclopedia*. https://en.wikipedia.org/w/index.php?title=Open_list&oldid=756581891 (accessed 3/5/2017).

Wikipedia (2017). "Social network analysis software." *Wikipedia, The Free Encyclopedia*. https://en.wikipedia.org/w/index.php?title=Social_network_analysis_software&oldid=768741771 (accessed 3/5/2017).

Political Network Electoral System Appendix

In this appendix we model the political network electoral system mathematically and prove that:

1. the number of candidates that are declared elected by the political network system matches the number of available seats (Theorem 1);
2. any implementation that satisfies the definition of the political network system produces the same set of elected candidates, regardless of implementation details that are not specified in this definition (Theorem 2).
3. the system can be implemented by a finite computer procedure (Theorem 3), using acceptable amounts of processing time (Theorem 6) and space (Theorem 7);
4. to better understand the structure of the political network, we can generate an alternative network structure that, in terms of vote transferring, behaves exactly as the original network structure would behave if a certain arbitrary set of candidates had been elected or eliminated and where none of them play the role of middle men in any vote transferring path (Theorem 8);
5. the order in which candidates are declared elected or eliminated does not affect the final result of the election (Theorem 9);
6. solid network coalitions (among them party-list coalitions) are guaranteed to elect at least the floor of $\frac{\boldsymbol{V_p} \cdot \boldsymbol{M}}{\boldsymbol{V}}$, where $\boldsymbol{V_p}$ is the number of votes obtained by the party, $\boldsymbol{V}$ is the total number of valid votes and $\boldsymbol{M}$ is the number of seats available, just like in typical party-list systems and solid coalitions in single transferring voting systems (Theorem 4 and Corollary 10);
7. the elected candidates of each isolated party-list coalition are always those who received more votes individually, just like in most open party-list systems (Theorem 5);

We also show that the rounding errors introduced by the use of a finite precision computational implementation won't break the guarantee of the total number of elected candidates (Corollary 6) neither the guarantee of the minimum number of elected candidates in a solid network coalition (Corollary 9).

Basic Definitions

In this section, we present a definition for the political network electoral system that is totally equivalent to the one in the main text, but that is subdivided in useful concepts that will be referred from the theorems.

Definition 1: political network electoral system

The political network electoral system is an electoral system such that, starting from the conditions specified in Definition 2, the elected candidates are determined by Procedure 1 and where the operation that performs vote transfers, vote discards and adjusts in the current quota satisfies Definition 5.

Definition 2: conditions to start the processing of votes

The conditions to allow the start of vote processing in the political network electoral system are

- The candidates need to form a political network (Definition 3);
- Voters should have voted in one and only one candidate.

Definition 3: political network

A political network is a set of N candidates to M seats where each candidate, C, has a set of neighbor candidates and each neighbor is associated to a percentage of votes to be transferred to him or her if any transfer of votes takes place with origin in C. If the a set of neighbors is not empty, the sum of all percentages associated to its members is one.

Procedure 1: general procedure of the political network electoral system

1. Eliminate party virtual candidates
2. Transfer and discard votes
3. While there are remaining candidates (Definition 4)
    a. While there are remaining candidates with at least as many current votes (Definition 11) as the current quota (Definition 10)
        i. Declare candidates with at least as many current votes as the current quota to be elected
        ii. If there are no remaining candidates with more than zero votes and none of them can receive votes from any elected candidate or there are no remaining candidates with more than zero votes and all seats are filled
            1. Fill the remaining seats declaring some candidates elected using any arbitrary criterion, eliminate all remaining candidates, terminate the election and exit

    iii. Transfer and discard votes

  b. If there are remaining candidates

    i. Eliminate the candidate with least current votes

    ii. Transfer and discard votes

Definition 4: remaining candidate

A remaining candidate is a candidate that has not yet been declared to be elected or eliminated.

Definition 5: procedure to transfer votes and discard votes

The procedure to transfer and discard votes is any procedure that in a finite number of steps, results in the same number of votes in any candidate as a, possibly infinite, process that

- never transfer non transferrable votes (Definition 6);
- changes the number of votes in any candidate only through direct vote transfers (Definition 7) or votes discards (Definition 8);
- converges to a state where
  - o all transferable votes have reached remaining candidates except for the votes that cannot reach a remaining candidate through any number of direct vote transfers;

all transferrable votes that cannot reach a remaining candidate through any number of direct vote transfers have been discarded.

Definition 6: transferrable votes

The transferrable votes are:

- all current votes of an eliminated candidate;
- all current votes beyond the current quota (Definition 11) of an elected candidate.

Definition 7: direct vote transfer

A direct vote transfer is the subtraction of K votes from the current number of votes of a candidate C_i and the addition of $K \cdot P_{ij}$ votes to the current number of votes of each neighbor C_j of C_i, where P_{ij} is the percentage of votes that C_i chose to transfer to C_j.

Definition 8: vote discard

To discard an amount of votes is to subtract that amount from the total number of current valid votes and from the current number of votes of the candidate to whom they currently belonged.

Definition 9: current quota

The current quota is equal to the number of current valid votes (Definition 11) divided by the number of seats being disputed in the election.

Definition 10: current number valid votes

The current number of valid votes is the number of valid votes minus the number of votes that have already been discarded by the election procedure.

Definition 11: candidate current number of votes

The current number of votes of a candidate is the number of votes received by the candidate from the voters plus to the total number of votes transferred to the candidate from other candidates minus the total number of votes transferred from the candidate to other candidates minus the total number of votes of the candidate that were discarded during the execution of Procedure 1.

Mathematical representation

In the political network electoral system, each candidate $C_i \mid 1 \leq i \leq N$, where N is the number of candidates, chooses a set of neighbors, $\boldsymbol{Z}_i = \{Z_{ij} \mid 1 \leq j \leq K_i\}$, where Z_{ij} is a neighbor of C_i and K_i is the number of neighbors of C_i and indicates for every $1 \leq j \leq K_i$ the percentage of votes, P_{ij} that should be transferred to Z_{ij} if he or she is eliminated from the election or has more votes than necessary to be elected in such way that, $\forall \boldsymbol{Z}_i \neq \emptyset, \sum_{1 \leq j \leq K_i} P_{ij} = 1$.

It happens that a candidate is not forced to choose at least one neighbor. Only in this case, there is no way to transfer his or her votes and the equality above does not hold. Any transfer of votes with origin in a candidate without neighbors becomes a vote discard.

To handle this condition automatically, we defined C_0 as a virtual discard candidate that receives all votes that should be discarded. If a candidate C_i chose no neighbors, it is defined that $\boldsymbol{Z}_i = \{C_0\}$ and $P_{i0} = 1$, meaning that all transfers are done directly to the discard virtual candidate, sparing us from an explicit vote discard. Thus, we have that, $\sum_{0 \leq j \leq K_i} P_{ij} = 1$ for any candidate.

If, at any moment, there is a cycle in the network containing no remaining candidates, the transferable votes of any candidate in the cycle will have to be discarded. In this case, we do the discard subtracting such votes from their owners and adding them to C_0. The fact that C_0 is not their neighbor, for this operation, does not matter.

Relations in the political network are not symmetric, so $C_i \in \boldsymbol{Z}_j$ does not imply that $C_j \in \boldsymbol{Z}_i$.

We call, $\boldsymbol{F}$, the set of virtual candidates representing parties. They can be voted but cannot be elected and are eliminated in the very beginning of the election. Their votes are transferred to the members of the party.

So, $\forall C_f \in \boldsymbol{F},\ \boldsymbol{Z}_f = \{Z_{fj} \in \boldsymbol{F}_f\} \wedge P_{fj} = 1/|\boldsymbol{F}_f|$, where $\boldsymbol{F}_f$ is the set of candidates that are members of the party that is represented by the virtual candidate C_f.

The political network is thus a directed graph (Diestel 2000) where the nodes are the candidates (being one of them the virtual discard candidate and some of them the virtual candidates representing parties) and there is an edge going from C_i to C_j if and only if $C_j \in \boldsymbol{Z}_i$. Every edge is annotated with a real number between zero and one, which is the transfer percentage from C_i to C_j. The number annotated in the outgoing edges of any node sum one.

Let M, be the number of seats in dispute, V_i the number of individual votes received by candidate C_i and $V = \sum_{1 \le i \le N} V_i$, the total number of valid votes.

We call V_0 the number of votes in candidate C_0 that corresponds to the total number of discarded votes. Initially, $V_0 = 0$.

The number of current valid votes is $\hat{V} = V - V_0$ and is initially equal to V. The current quota is $\hat{Q} = \hat{V}/M$. At the same time, $\hat{V}_i$, is the current number of votes in C_i or the number of votes that currently belong to C_i.

We call the set of already elected candidates $\boldsymbol{L}$, the set of already eliminated candidates $\boldsymbol{E}$ and the set of remaining candidates $\boldsymbol{R}$. We will call $\boldsymbol{C}$ the set of all candidates including the virtual candidates representing parties, but not including the virtual discard candidate.

We call a vote package a structure $PV = (C, V, ID)$, where C is a candidate, V is a real number representing the amount of votes in the package and ID is an identifier that gives the package a non ambiguous identification.

Implementation of the political network electoral system

In this section we present an implementation for the political network electoral system which leaves some procedures open. We will show latter that any implementation that satisfies the definition of the political network system is equivalent to this implementation with some variation of the procedures left open.

As it can be directly verified, Procedure 2, Procedure 3 and Procedure 4 are equivalent to Procedure 1. They just operate over the mathematical structures defined in the previous section.

Procedure 2: central procedure of the political network election

4. $\boldsymbol{R} \leftarrow \boldsymbol{C}$ // Initially, all candidates are remaining candidates
5. $\boldsymbol{L} \leftarrow \boldsymbol{E} \leftarrow \emptyset$ // Nobody is initially elected neither eliminated
6. $\hat{V} \leftarrow V$ // The current number of valid votes is initially equal to the number of valid votes
7. $\forall \mathrm{i}, \hat{V}_{\boldsymbol{i}} \leftarrow V_i$ // Initially each candidate has the number of votes that he or she received individually
8. Call $\boldsymbol{EliminateVirtualPartyCandidatesAndTranferVotes}()$
9. While $\boldsymbol{R} \neq \emptyset$
    a. $shouldExit = \boldsymbol{IdentifyElectedAndTransferVotes}()$
    b. If $(shouldExit)$ $\boldsymbol{Exit}$
    c. If $\boldsymbol{R} \neq \emptyset$
        i. Call $\boldsymbol{EliminatedLastAndTransferVotes}(\ )$

Procedure 3: $\boldsymbol{IdentifyElectedAndTransferVotes}()$

10. $\dot{\boldsymbol{L}} \leftarrow \{C_i | C_i \in \boldsymbol{R} \wedge \hat{V}_{\boldsymbol{i}} \geq \hat{Q}\}$; // Creates a temporary set containing those candidates who have just reached the quota
11. While $\dot{\boldsymbol{L}} \neq \emptyset$
    a. $\boldsymbol{L} \leftarrow \boldsymbol{L} + \dot{\boldsymbol{L}}$ // Puts the recently elected in the set of elected candidates
    b. $\boldsymbol{R} \leftarrow \boldsymbol{R} - \dot{\boldsymbol{L}}$ // Removes elected from the set of remaining candidates
    c. If $\boldsymbol{DetectExitCondition}(\ )$
        i. Call $\boldsymbol{FillRemainingSeats}(\ )$
        ii. Call $\boldsymbol{EliminateRemainingCandidates}(\ )$
        iii. $\boldsymbol{return\ true}$
    d. Call $\boldsymbol{TransferVotesAndUpdateQuota}(\dot{\boldsymbol{L}})$;
    e. $\dot{\boldsymbol{L}} \leftarrow \{C_i | C_i \in \boldsymbol{R} \wedge \hat{V}_{\boldsymbol{i}} \geq \hat{Q}\}$; // others may be elected as a consequence of vote transfers and reduction of the current quota.
12. $\boldsymbol{return\ false}$

Procedure 4: $\boldsymbol{EliminateLastAndTransferVotes}()$

13. $\dot{\boldsymbol{E}} \leftarrow \{\min (C_i, \forall C_i \in \boldsymbol{R})\}$ // Creates a temporary set containing only the eliminated candidate
14. $\boldsymbol{E} \leftarrow \boldsymbol{E} + \dot{\boldsymbol{E}}$ **//** Adds that set to the set of eliminated candidates

15. $\boldsymbol{R} \leftarrow \boldsymbol{R} - \dot{\boldsymbol{E}}$ // Removes the set of eliminated candidates from the set of remaining candidates

16. Call $\boldsymbol{TransferVotesAndUpdateQuota}(\dot{\boldsymbol{E}})$

In Procedure 4, $\min(C_i, \forall C_i \in \boldsymbol{R})$ is a function that returns the remaining candidate with the least number of votes. It is assumed that if there is a tie some arbitrary criterion will choose a single candidate among the ones with least votes to be the result of $\min(C_i, \forall C_i \in \boldsymbol{R})$.

Procedure 5: $\boldsymbol{EliminateVirtualPartyCandidatesAndTranferVotes}()$

17. $\boldsymbol{E} \leftarrow \boldsymbol{E} + \boldsymbol{F}$ // Adds all virtual party candidates to the set of eliminated candidates

18. $\boldsymbol{R} \leftarrow \boldsymbol{R} - \boldsymbol{F}$ // Removes the virtual party candidates from the set of remaining candidates

19. Call $\boldsymbol{TransferVotesAndUpdateQuota}(\boldsymbol{F})$

We still need to write $\boldsymbol{TransferVotesAndUpdateQuota}(\dot{\boldsymbol{C}})$, where $\dot{\boldsymbol{C}}$ is the set of candidates that have been just elected or eliminated, in a way that incorporates all variations admitted by Definition 5 and that can help us to prove our target theorems latter. We do that through Procedure 6, Procedure 7 and Procedure 8.

Procedure 6: $\boldsymbol{TransferVotesAndUpdateQuota}(\dot{\boldsymbol{C}})$

20. Call $\boldsymbol{UpdateThePoliticalNetworkConveniently}(\dot{\boldsymbol{C}})$

21. $\dot{Q} \leftarrow (\hat{V} - \boldsymbol{CalculateTotalDiscards}())/M$ // Calculates the correct quota for the end of this procedure

22. Call $\boldsymbol{TransferVotesAccordingToQuota}(\dot{Q})$

23. $\hat{Q} \leftarrow \dot{Q}$

Procedure 6 starts calling procedure $\boldsymbol{UpdateThePoliticalNetworkConveniently}(\dot{\boldsymbol{C}})$. The definition of the political network electoral system does not authorize any changes in the network. Bellow, we define this procedure accordingly.

Definition 12: $\boldsymbol{UpdateThePoliticalNetworkConveniently}(\dot{\boldsymbol{C}})$

A suitable implementation of $\boldsymbol{UpdateThePoliticalNetworkConveniently}(\dot{\boldsymbol{C}})$ is any implementation that only changes the structure of the political network in ways that cannot make the result of the election different from the result that would be obtained if no changes were made.

Procedure 6 depends on the function $\boldsymbol{CalculateTotalDiscards}()$ that is defined bellow.

Definition 13: $\boldsymbol{CalculateTotalDiscards}()$

A suitable implementation of the $\boldsymbol{CalculateTotalDiscards}()$ is any implementation that returns a value that makes it possible for Procedure 6 to end in a way that is consistent with the definition of the electoral system.

We will show latter (Lemma 14) that the value that can be returned by $\boldsymbol{CalculateTotalDiscards}()$ is unique and can, indeed, be calculated with the information that is available in the beginning of Procedure 6.

Let's now write Procedure 7, which implements a vote transfer process for a given target quota, assuming that, in the end, it will be correct.

Procedure 7: $\boldsymbol{TransferVotesAccordingToQuota}(\dot{\boldsymbol{Q}})$

24. $\boldsymbol{H} \leftarrow \emptyset$ // creates a, initially empty, structure to contain all vote packages waiting to be transferred
25. $\forall C_i \in \boldsymbol{E} | \hat{V}_i > 0$ // For all eliminated candidates that still have any votes
    a. $\boldsymbol{AddPackage}\left(\boldsymbol{H}, PV_i = (C_i, \hat{V}_i, NID)\right)$ // puts, in $\boldsymbol{H}$, a vote package containing all votes in the eliminated candidate. NID represents a new identifier
    b. $\hat{V}_i \leftarrow 0$ // No votes remain in the eliminated candidate
26. $\forall C_i \in \boldsymbol{L} | \hat{V}_i > \dot{Q}$ // For all elected candidates with surplus votes
    a. $\boldsymbol{AddPackage}\left(\boldsymbol{H}, PV_i = (C_i, \hat{V}_i - \dot{Q}, NID)\right)$ // The packages associated to elected candidate contain only the surplus votes
    b. $\hat{V}_i \leftarrow \dot{Q}$ // A number of votes identical to the quota is kept in the candidate
27. Call $\boldsymbol{TransferVotes}(H)$

Procedure 8: $\boldsymbol{TransferVotes}(\boldsymbol{H})$

28. $NS \leftarrow 0$ // Variable NS is used to count the steps of the following loop till an arbitrary limit $MAXSTEPS$
29. While $\boldsymbol{H} \neq \emptyset \wedge NS < MAXSTEPS$ // while there are votes to be transferred and the arbitrary limit to the number of steps has not been reached
    a. $\left(VP_i = (C_i, \dot{V}_i, ID)\right) \leftarrow \boldsymbol{RemoveAPackage}(\boldsymbol{H})$ // VP_i is a package of votes built by a criterion that will be discussed latter, using votes in $\boldsymbol{H}$. $\dot{V}_i$ is the number of votes in the package and differs from $\hat{V}_i$ which is the number of votes in the candidate.
    b. $\forall C_{ij} | C_{ij} \in \boldsymbol{Z}_i$ // For all neighbors of C_i
        i. If $C_{ij} \in \boldsymbol{R} \cup \{C_0\}$

1. $\hat{V}_{ij} \leftarrow \hat{V}_{ij} + \dot{V}_i \cdot P_{ij}$ // Transfer the votes according to the specified percentage. Such transfer represents a discard when $C_{ij} = C_0$

ii. Else // votes will be transferred again

1. $\boldsymbol{AddPackage}\left(\boldsymbol{H}, PV_{ij} = \left(C_{ij}, \dot{V}_i \cdot P_{ij}, NID\right)\right)$

c. $NS \leftarrow NS + 1$

30. $V_0 \leftarrow V_0 + \sum_{\forall VP_i \in \boldsymbol{H}} \dot{V}_i$ // discards all votes that are still in $\boldsymbol{H}$

31. $\hat{V} \leftarrow V - V_0$ // adjusts the number of valid votes

In Procedure 7, we created a structure, $\boldsymbol{H}$**,** to contain all vote packages that are waiting to be transferred and passed such structure as a parameter to Procedure 8.

We say that a vote package was processed by Procedure 8 when it was removed from $\boldsymbol{H}$ and returned by $\boldsymbol{RemoveAPackage}(\boldsymbol{H})$ and consequently submitted to the commands within the loop defined in the procedure. We define $\boldsymbol{RemoveAPackage}(\boldsymbol{H})$ bellow.

Definition 14: $\boldsymbol{RemoveAPackage}(\boldsymbol{H})$

A suitable implementation of $\boldsymbol{RemoveAPackage}(\boldsymbol{H})$ is any implementation that returns a package of votes $PV = (C, V, ID)$ and removes it from $\boldsymbol{H}$ and that, if convenient, performs arbitrary join operations and arbitrary split operations to any packages in $\boldsymbol{H}$, as long as it allows no package to stays forever in $\boldsymbol{H}$**.**

Lemma 1: existence of a suitable implementation of $\boldsymbol{RemoveAPackage}(\boldsymbol{H})$

An implementation of $\boldsymbol{RemoveAPackage}(\boldsymbol{H})$ that performs no joins or splits and returns the vote packages in a first in first out manner satisfies Definition 14.

Proof:

When a vote package, VP, enters $\boldsymbol{H}$, the number of packages ahead of it is finite. If all packages that are generated latter are put after VP, it is guaranteed that for a sufficiently large value for $MAXSTEPS$, VP will be returned. ■

Definition 15: canonical variation of $\boldsymbol{RemoveAPackage}(\boldsymbol{H})$

The canonical variation of $\boldsymbol{RemoveAPackage}(\boldsymbol{H})$ is the variation that performs no joins or splits and returns the vote packages in a first in first out manner.

In the definition of the voting transfer process, transfers are direct. There is no intermediate structure. However, to prove our target theorems, we need to keep track of where each vote came from and not only of the total number of votes in each candidate. The structure $\boldsymbol{H}$ will help with that.

We will show in Lemma 3, that any procedure that satisfies Definition 5 is equivalent to a variation of Procedure 6. Before that, we need a preparatory lemma.

Lemma 2: amount of votes in each candidate after an operation of vote transferring

In the end of $\boldsymbol{TransferVotesAndUpdateQuota}(\dot{\boldsymbol{C}})$, any eliminated candidate C_j has zero votes and any elected candidate has $\hat{Q}$ votes, where $\hat{Q}$ is the current quota.

Proof:

By the definition of the political network electoral system, any votes in an eliminated candidate are transferable. The definition also says that all transferrable votes must be transferred till they reach a remaining candidate, or, if that is impossible, they must be discarded. In any case, no votes can stay in the eliminated candidate.

Any vote in an elected candidate beyond $\hat{Q}$ is also transferrable and thus cannot stay in the candidate. If such votes exist when $\boldsymbol{TransferVotesAndUpdateQuota}(\dot{\boldsymbol{C}})$ is called they must be transferred or discarded and they cannot exist in the end of the procedure.

At the same time, any candidate that is claimed to be elected, at that moment, has at least $\hat{Q}$ votes and since votes bellow $\hat{Q}$ are not transferrable there will never occur a vote transfer that will leave him or her with less than $\hat{Q}$ votes. The possibility that, latter, the current quota could be changed in a way that makes it larger than the number of votes in the candidate also does not exist, since, by definition, it can only go down. This proves the present lemma. ■

Lemma 3: covering of all possible vote transferring procedures by Procedure 6

For any arbitrary implementation of the procedure $\boldsymbol{TransferVotesAndUpdateQuota}(\dot{\boldsymbol{C}})$ that satisfies the definition of the political network electoral system there is a variant of Procedure 6 under which the number of votes left in every candidate C_j converges to the number of votes left by the arbitrary implementation, when $MAXSTEPS$ tends to infinite.

Proof:

Definition 5 says that any implementation of $\boldsymbol{TransferVotesAndUpdateQuota}(\dot{\boldsymbol{C}})$ must achieve the same results as a, possibly infinite, process that works under some restrictions that are enumerated in the definition itself. Let's call such canonical process $CanonicalProcess$.

In its end, $CanonicalProcess$ will have to achieve a value, $\dot{Q}$**,** for the current quota (Definition 9). The function $\boldsymbol{CalculateTotalDiscards}()$, which is still open, can return the value $M \cdot (\hat{Q} - \dot{Q})$, where $\hat{Q}$ is the current quota in the beginning of Procedure 6 and lead the very same quota.

Definition 5 determines that all vote transfers should be done through direct vote transfers (Definition 7). Other than that, $CanonicalProcess$ can only discard votes that could not be transferred to a remaining candidate by any number of direct transfers.

Thus, $CanonicalProcess$ can only change the number of votes in any candidate through a sequence of operations, $\boldsymbol{SO} = (O_1, O_2, \dots, O_i, \dots)$, where $\forall i, O_i$ is a direct vote transfer or a vote discard. Given the source candidate C_{ti} and the amount of votes, $\dot{V}_{ti}$, that are going to be transferred from C_{ti}, a direct transfer is fully determined, that is, we already know how many votes will go to each other candidate. A vote discard can also be described by the source candidate, C_{ti}, and amount of votes, $\dot{V}_{ti}$ to be discarded. So we can specify O_i by $O_i = (IsDiscard_i, C_{ti}, \dot{V}_{ti})$, where $IsDiscard_i$ is a Boolean separating transfers from discards.

At any point between any two operations, the state of the full transfer process can be described by the number of votes, V_j, in each candidate C_j. Note that the number of votes in C_0 determines the number o current valid votes and thus the current quota.

The sequence of operations is restricted by the fact that only transferrable votes can be transferred or discarded. Thus,

$\forall O_i, C_{ti} \in \boldsymbol{L} \cup \boldsymbol{E}$,

$\forall O_i | C_{ti} \in \boldsymbol{L}, \dot{V}_{ti} \leq V_i - \tilde{Q} \leq V_i - \dot{Q}$ and

$\forall O_i | C_i \in \boldsymbol{E}, \dot{V}_{ti} \leq V_i$,

where $\tilde{Q}$ is the current quota immediately before the application of O_i.

Let's show that we can simulate $\boldsymbol{SO}$ with a variation of Procedure 6. We will do that, starting with the canonical implementation of $\boldsymbol{RemoveAPackage}(\boldsymbol{H})$ and performing convenient package joins and splits. Let's call the process that is driven by the variation of Procedure 6, $CoverProcess$.

Let's call $\boldsymbol{D}$ the set of candidates from whom there is no path to a remaining candidate in the political network and $\overline{\boldsymbol{D}}$ the set of candidates from whom there is such a path**.**

Clearly, votes in candidates that belong to $\boldsymbol{D}$ can never reach a remaining candidate. They can be discarded by the $CanonicalProcess$ at any moment and must be discarded by it at some point. Votes in a candidate C that belongs to $\overline{\boldsymbol{D}}$ cannot be discarded. With enough direct vote transfers, at least a certain percentage $P > 0$ of them would eventually reach a remaining candidate. If the $CanonicalProcess$ chose to discard $v > 0$ votes of the V_C votes in C, even if v was very small, after the discard, it would still only count on direct vote transfers to transfer votes. Direct vote transfers are inflexible, thus the $CanonicalProcess$ could not force more than the same percentage P of the votes left in C to reach remaining candidates. That would be equal to $P \cdot (V_C - v) < P \cdot V_C$, violating the definition of the political network system, which requires that all votes that can be transferred to remaining candidates are transferred to them.

Let's call $\acute{V}_j$ the number of votes in candidate C_j in the $CoverProcess$, while V_j is the number of votes in candidate C_j in the $CanonicalProcess$. Let's call $\ddot{V}_j$ the total number of votes in packages associated to C_j in $\boldsymbol{H}$ in the $CoverProcess$.

We say that the state of the $CoverProcess$ is equivalent to the state of the $CanonicalProcess$ if and only if

$$\forall j | C_j \in \boldsymbol{R} \cup \{C_0\}, V_j = \acute{V}_j \wedge \ddot{V}_j = 0,$$

$$\forall j | C_j \in \boldsymbol{E} \cap \overline{\boldsymbol{D}}, V_j = \ddot{V}_j \wedge \acute{V}_j = 0,$$

$$\forall j | C_j \in \boldsymbol{L} \cap \overline{\boldsymbol{D}}, V_j = \ddot{V}_j + \dot{Q} \wedge \acute{V}_j = \dot{Q},$$

$$V_0 + \sum_{\forall j | C_j \in \boldsymbol{E} \cap \boldsymbol{D}} V_j + \sum_{\forall j | C_j \in L \cap \boldsymbol{D}} V_j - \dot{Q} = \acute{V}_0 + \sum_{\forall j | C_j \in \boldsymbol{E} \cap \boldsymbol{D}} \ddot{V}_j + \sum_{\forall j | C_j \in L \cap \boldsymbol{D}} \ddot{V}_j$$

$$\forall j | C_j \in \boldsymbol{E} \cap \boldsymbol{D}, \acute{V}_j = 0$$

$$\forall j | C_j \in \boldsymbol{L} \cap \boldsymbol{D}, \acute{V}_j = \dot{Q},$$

By Lemma 2, in the end of $\boldsymbol{TransferVotesAndUpdateQuota}(\dot{\boldsymbol{C}})$ the number of votes in any eliminated candidate is zero and the number of votes in any elected candidate is equal to the current quota. This means that under the $CanonicalProcess$ their number of votes must converge to such values.

Thus, in the $CanonicalProcess$, after a large enough number of steps, $LARGENUMSTEPS$, we will have that $\forall j | C_j \in \boldsymbol{E}, 0 \leq V_j < \epsilon$ and $\forall j | C_j \in \boldsymbol{L}, 0 \leq V_j - \dot{Q} < \epsilon$, for any small ϵ. At the same time, when , $MAXSTEPS$ is reached $CoverProcess$ ends and all votes in $\boldsymbol{H}$ are discarded, making $\forall j, \ddot{V}_j = 0$.

If the state of $CanonicalProcess$ after $LARGENUMSTEPS$ is equivalent to the state of $CoverProcess$ at its end, it is algebraically implied that $\forall j \neq 0, |V_j - \acute{V}_j| < \epsilon$, and $|V_0 - \acute{V}_0| < N \cdot \epsilon$, where N is the number of candidates. Thus the two process must be converging to the same state when $MAXSTEPS$ tends to infinite.

We must show that the equivalence between the two processes can always be kept.

Before the first direct transfer occurs, Procedure 7 puts in $\boldsymbol{H}$, vote packages that establish the equivalence directly.

From that point on, it is easy to check that for every O_i, if $O_i$is a vote discard, nothing needs to be done, if it is a vote transfer, $\boldsymbol{RemoveAPackage}(\boldsymbol{H})$, can always return the vote package $VP_i = (C_{ti}, \dot{V}_{ti}, ID)$ what leads Procedure 8, to keep the state of the $CoverProcess$ equivalent to the state in the canonical process immediately after the application of O_i.

Note that, if the states were equivalent before the transfer operation, the necessary number of votes, is always available in $\boldsymbol{H}$**.** Since $\boldsymbol{RemoveAPackage}(\boldsymbol{H})$ can join and split packages, it can always form a single package with the correct number of votes and return it.

The single package to be returned can be formed taking k packages, associated to C_{ti} in their order of entrance in $\boldsymbol{H}$, in such way that $\sum_{l=0}^{k} V_l \leq \dot{V}_{ti}$, and join them, summing all their votes into the new package. If $\sum_{l=0}^{k} V_l = \dot{V}_{ti}$, the new package is done. If $\sum_{l=0}^{k} V_l < \dot{V}_{ti}$, we can take the $k + 1^{th}$package associated to C_{ti} in H and split it in two, leaving the first package with $\dot{V}_{ti} - \sum_{l=0}^{k} V_l$ votes. Adding this package to the package to be returned, it will be done.

It is important to notice that, forming packages in the way described above guarantees that no package can be left in $\boldsymbol{H}$ forever, what is a requirement of Definition 14.

So the equivalence can be kept proving the present lemma. ■

Useful lemmas

To easy the discussions to come, in this section, we show some useful lemmas.

Lemma 4: election constant

During the execution of Procedure 2, the sum of the total votes in real candidates, with the total votes in virtual party candidates, with the total number of discarded votes and with the total number of votes in vote packages in $\boldsymbol{H}$ is a constant equal to the original number of valid votes.

Proof:

Initially all votes are in real candidates or virtual party candidates and they correspond to the number of valid voters in the election, thus, at this moment the present lemma holds. It is easy to verify that additions to any of the values that are relevant to the present lemma are accompanied by equivalent subtractions some other of this values, keeping the lemma valid. ■

Lemma 5: determination of additions and discards by the total amount of votes in processed packages

In an execution of Procedure 8, the total number of votes, TV_r, added to any candidate $C_r \in \boldsymbol{R} \cup \{C_0\}$ are given by $TV_r = \sum_{j=1}^{N} TV_j \cdot P_{jr}$, where TV_j is the total number of votes associated to candidate C_j in packages that were processed by Procedure 8.

Proof:

It is direct from Procedure 7 and Procedure 8, that any addition of votes to any $C_r \in \boldsymbol{R} \cup \{C_0\}$ is a consequence of the processing of some package VP_j and is given by $V_j \cdot P_{jr}$, where V_j is the number of votes in VP_j and P_{jr} is the percentage of transfer from C_j to C_r specified in the structure of the political network. where C_j is the candidate associated to VP_j. The veracity of the present lemma is immediate. ■

We will show in Lemma 8 and Corollary 4 , that we can join and split vote packages without changing the results of Procedure 8 and Procedure 7. This conclusion can be used both to prove that the joins and splits performed within $\boldsymbol{RemoveAPackage}(\boldsymbol{H})$ in the proof of Lemma 3 do not change the results of the election and to prove that adjusts in the structure of the network performed within $\boldsymbol{UpdateThePoliticalNetworkConveniently}(\dot{\boldsymbol{C}})$ that happen to have an effect that is equivalent to joins of vote packages also do not change such results.

Before proving Lemma 8, we will need a few more definitions and preparatory lemmas. Every package VP_j processed by Procedure 8 can generate new packages and add them to $\boldsymbol{H}$. The processing of these packages, in turn, generate even more packages. Since every package is generated by the processing of a single parent package, the complete set of packages indirectly generated by VP_j form a tree whose root is VP_j.

Definition 16: vote package tree

A vote package tree AV_j , with root VP_j, is the tree composed by VP_j and any package generated by Procedure 8 when it process any package belonging to AV_j and where the parent of any package is the package from which it was generated.

Corollary 1: identification of a vote package in a vote package tree

Writing $VP_e \equiv SC_{je}$, we can uniquely indentify any package VP_e in the vote package tree with root VP_j by the sequence of candidates $SC_{je} = \{C_{jet} | 0 \leq t \leq h\}$, indicated in the packages that are in the path of length h, from VP_j till VP_e in the tree, where $C_{je0} = C_j$ e $C_{jeh} = C_e$.

Proof:

Since the processing of each candidate by Procedure 8 cannot generate more than one package associated to the same candidate, we can identify the package by its parent and the candidate associated to it. Since the structure is a tree, the veracity of the present lemma is direct. ■

Corollary 2: identification of a vote package in a forest of vote packages

Writing $VP_e \equiv (VP_j, SC_{je})$, we can uniquely identify a vote package VP_e in a forest of vote packages by the pair (VP_j, SC_{je}), where VP_j is the root of the tree to which VP_e belongs and SC_{je} is the sequence of candidates the identifies VP_e in this tree.

Proof:

Procedure 8 process a package VP_j only once, so the tree associated to VP_j is unique. From this and Corollary 1 follows the present corollary. ■

Lemma 6: amount of votes in a packages belonging to a tree

The number of votes in package $VP_e \equiv (VP_j, SC_{je})$, V_e is given by $V_e = V_j \prod_{t=1}^{h} P_{(t-1)t}$, where, h is the length of SC_{je} and $P_{(t-1)t}$ is the percentage of transfer to the t[th] candidate in SC_{je} from the t[th-1] candidate in SC_{je}, as defined in the structure of the network.

Proof:

If the length of SC_{je} is 1, $VP_e = VP_j$ and the result is trivial.

Procedure 8**,** defines the number of votes of each package VP_j generated by the processing of a package VP_i, by $VP_j = VP_j \cdot P_{ij}$, where P_{ij} is the percentage of transfer from VP_i to VP_j. If the present lemma holds for paths of length $h-1$, then the number of votes in the parent of VP_e, VP_{e-1} is given by $V_{e-1} = V_j \prod_{t=1}^{h-1} P_{(t-1)t}$.

Using $V_e = V_{e-1} \cdot P_{(e-1)e}$, the result is immediate. ■

Corollary 3: alignment of vote packages trees

If we take two packages VP_j and VP_i such $C_j = C_i$, for any package VP_{je} belonging to a tree of root VP_j, identified by the sequence of candidates SC, there is one and only one package VP_{ie} belonging to a tree of root VP_i identified by the same sequence.

Proof:

The candidates indicated in the packages that are generated in the processing of any package VP_j, by Procedure 8 only depend on C_j, the candidate associated to VP_j and not on V_j, the amount of votes in VP_j so the structures of the generated trees are identical proving the present corollary. ■

Definition 17: package equivalent to a finite set of packages

We say that a package $VP_L = (C, V_L, ID_L)$ is equivalent to a finite set of packages $SP = \{VP_k | VP_k = (C, V_k, ID_k), 1 \leq k \leq s\}$ if and only if $\sum_{k=1}^{s} V_k = V_L$.

Definition 18: package equivalent to an infinite set of packages

We say that a package $VP_L = (C, V_L, ID_L)$ is equivalent to an infinite set of packages $SP = \{VP_k | VP_k = (C, V_k, ID_k), 1 \leq k\}$ if and only if $\lim_{s\to\infty} \sum_{k=1}^{s} V_k = V_L$.

Definition 19: equivalence of a tree to a forest

We say that a tree of votes AV_L, of root VP_L is equivalent to a finite or infinite forest FV, whose k[th] tree we call AV_k of root VP_k, if and only if $C_L = C_k, \forall k$ and if every package $VP_{Le} \equiv (VP_L, SC)$ is equivalent to the set $SP_e = \{VP_{ke} \equiv (VP_k, SC), \forall VP_k \in SP\}$.

Lemma 7: automatic equivalence of a tree to a forest

If a package of votes VP_L is equivalent to a set of finite or infinite packages, SP, the tree AV_L, with root VP_L is equivalent to the forest composed by the trees whose roots are the elements, VP_k, belonging to SP.

Proof:

By Lemma 6, the number of votes in any package $VP_{Le} \equiv (VP_L, SC)$, belonging to AV_L is given by $V_{Le} = V_L \prod_{t=1}^{h} P_{(t-1)t}$ and the number of votes in a package $VP_{ke} \equiv (VP_k, SC)$, where VP_k is any element of SC, is given by $V_{ke} = V_k \prod_{t=1}^{h} P_{(t-1)t}$.

If SP is finite, $\sum_{k=1}^{s} V_k = V_L$, if it is infinite $\lim_{s\to\infty} \sum_{k=1}^{s} V_k = V_L$. In any case, with algebraic manipulations and using the definitions of package equivalence (Definition 17 and Definition 18) we conclude that VP_{Le} is equivalent to the set $SP_e = \{VP_{ke} \equiv (VP_k, SC), \forall VP_k \in SP\}$.

By the definition of equivalence of trees to forests (Definition 19), the present lemma is immediate. ■

Definition 20: package join

We say that we joined a set a packages, SP, if we remove all packages belonging to SP from $\boldsymbol{H}$, prevent any other members of SP from being put in $\boldsymbol{H}$ and put in $\boldsymbol{H}$ a package VP_L that is equivalent to SP

Lemma 8: insensitivity to package joins

If we join a set of packages, SP, into a package VP_L when $MAXSTEPS$ tends to infinite, for every $C_r \in \boldsymbol{R} \cup \{C_0\}$, the value to which V_r tends in the end of Procedure 8 and in the end of Procedure 7 does not change.

Proof:

By Lemma 5, it suffices to show that the total votes, TV_j, in packages associated to any candidate C_j, processed by Procedure 8, when $MAXSTEPS$ tends to infinite, tends to a value that does not depend on whether we have performed or not the join of SP.

By Lemma 7, each $VP_{Le} \equiv (VP_L, SC)$ generated as a direct or indirect consequence of the addition of VP_L to $\boldsymbol{H}$ is equivalent to the set of packages $SP_e = \{VP_{ke} \equiv (VP_k, SC), \forall VP_k \in SP\}$ whose elements will be either removed from $\boldsymbol{H}$ or prevented from being generated by Procedure 8 and put in $\boldsymbol{H}$ as a consequence of the joining operation of SP into VP_L.

Let TO be line of processing where the joining operation did not take place and TA be a line o processing where it did.

Let TO_e be the value to which tends TV_e the total amount of votes in packages associated to a candidate C_e processed in Procedure 8 when $MAXSTEPS$ tends to infinite in line TO.

Let TA_e, be the value to which tends TV_e the total amount of votes in packages associated to a candidate C_e processed in Procedure 8, when $MAXSTEPS$ tends to infinite in line TA.

No package can stay forever in $\boldsymbol{H}$ in line TO, since that is a requirement of Definition 14. Since the join only adds one package to H and does not change the order of the other packages, package can stay forever in $\boldsymbol{H}$ in line TA also.

Let's suppose that $TO_e > TA_e$ and reach a contradiction.

If in line TO, TV_e, tends to TO_e, so for some value, ot, of $MAXSTEPS$, TV_e becomes greater than TA_e.

For some value, at, of $MAXSTEPS$, any package VP_j not involved in the join operation which is processed till step ot in line TO, will also have been processed in line TA.

At the same time, in line TA, for a large enough at, any package VP_{Le} equivalent to some set SP_e which has had at least one element processed in line TO till step ot will have been processed.

Since the amount of votes in VP_{Le} is equal to the total amount of votes in all packages belonging to SP_e, The sum of all processed packages associated to candidate C_e, TV_e , would have to be greater than TA_e in line TA, what is a contradiction.

Now let's suppose that $TO_e < TA_e$ and also reach a contradiction.

If in line TA, TV_e tends to TA_e, then for some value, at, of $MAXSTEPS$, TV_e becomes larger than TO_e. Let's call the value of TV_e at step at, TV_{eat}.

Since no package can stay forever in $\boldsymbol{H}$, for some value ot of $MAXSTEPS$, any package VP_j not involved the join operation which is processed till step at in line TA, will also have been processed in line TO.

If SP is finite, in line TO, for a large enough ot any package $VP_{ke} \in SP_e$, where SP_e is equivalent to some package VP_{Le} which has been processed in line TA till step at will have been processed. That would make TV_e larger than TO_e in line TO, what is a contradiction.

If SP is infinite, for a large enough ot, for any VP_{Le} processed in line TA till step at, any package $VP_{ke} \in SP_e$, k < l, where SP_e is equivalent to VP_{Le} and l is as large as desired will have been processed in line TO. Let's call the value of TV_e at step ot, TV_{eot}. Under this conditions, we can choose l in order to guarantee that

$$V_{Le} - \left(\sum_{VP_{ke} \in SP_e, \mathrm{k}<l} V_{ke} \right) < \delta, \forall VP_{Le}$$

for any chosen δ.

Consequently, calling SP_{at} the set of all VP_{Le} belonging to the tree of root VP_L which have been processed in line TA till step at

$$\sum_{\forall VP_{Le} \in SP_{at}} \left(V_{Le} - \sum_{VP_{ke} \in SP_e, \mathrm{k}<l} V_{ke} \right) < \delta \cdot |SP_{at}|$$

and

$$\sum_{\forall VP_{Le} \in SP_{at}} V_{Le} = TV_{eat}$$

If we choose $\delta = \frac{TV_{eat} - TO_e}{|SP_{at}|}$, we have that

$$\sum_{\forall VP_{Le} \in SP_{at}} \left(V_{Le} - \sum_{VP_{ke} \in SP_e, \mathrm{k}<l} V_{ke} \right) < TV_{eat} - TO_e$$

$$\sum_{\forall VP_{Le} \in SP_{at}} V_{Le} - \sum_{\forall VP_{Le} \in SP_{at}} \sum_{VP_{ke} \in SP_e, \mathrm{k}<l} V_{ke} < TV_{eat} - TO_e$$

$$TV_{eot} \geq \sum_{\forall VP_{Le} \in SP_{at}} \sum_{VP_{ke} \in SP_e, \mathrm{k}<l} V_{ke} > TO_e$$

So, TV_e becomes greater than TO_e in line TO, what is a contradiction.

The only remaining possibility is that $TO_e = TA_e$, what proves the present lemma for Procedure 8.

Since, except for Procedure 8, there is nothing in the scope of Procedure 7, that can be affected by the join operation, the lemma also holds for it. ■

Definition 21: package finite split

We say that we split a package, VP_L, into a set of packages, SP, if we remove VP_L from $\boldsymbol{H}$ and put in $\boldsymbol{H}$ all packages belonging to a finite set, SP, which is equivalent to VP_L.

Corollary 4: insensitivity to package finite splits

If we split a package VP_L in a finite set of packages, SP, when $MAXSTEPS$ tends to infinite, for every $C_r \in \boldsymbol{R} \cup \{C_0\}$, the value to which V_r tends in the end of Procedure 8 and in the end of Procedure 7 does not change.

Proof:

By Lemma 8, we can join SP into VP_L and the results won't change.

This corollary is just the reflex of that and its veracity is valid as long as we can guarantee that after the split no package can stay forever in $\boldsymbol{H}$. Since we are adding a finite number of packages to $\boldsymbol{H}$ and are not changing the order of processing of any other packages the present corollary follows. ■

Uniqueness of the Result of the Election and Implementation in a Finite Number of Steps

In this section, we show that the result of the an election where the political network system is employed is unique and that such result can be found in a finite number of steps. These are the essential requirements to claim that the proposed system is sound and computationally possible.

To build a finite variation of Procedure 7, we will show that

1. it is possible to build a modified version of the political network that guarantees that Procedure 8 ends in a finite number of steps without changing its result and
2. we can correctly write the procedure $\boldsymbol{CalculateTotalDiscards}()$ used by Procedure 7.

Procedure 9 removes an edge from the political network going from a candidate C_i to a candidate C_k (removes C_k from $\boldsymbol{Z}_i$). We will prove, in Lemma 9, that its use cannot change the results of Procedure 8 or Procedure 7.

Procedure 9: $\boldsymbol{RemoveFromNeighborSet}(\boldsymbol{C_k}, \boldsymbol{C_i})$

3. If $\widehat{\boldsymbol{Z}}_i = \{C_k\}$ and $\widehat{\boldsymbol{Z}}_k = \{C_i\}$
    a. $\widehat{\boldsymbol{Z}}_i \leftarrow \{C_0\}$
    b. $P_{i0} \leftarrow 1$

4. Else
    a. $\widehat{\boldsymbol{Z}}_i \leftarrow \widehat{\boldsymbol{Z}}_i - \{C_k\} + \widehat{\boldsymbol{Z}}_k - \{C_i\}$ // eliminates C_k from the neighbor set of C_i, $\widehat{\boldsymbol{Z}}_i$, but adds all neighbors of C_k to $\widehat{\boldsymbol{Z}}_i$, excepts for C_i itself.
    b. $\forall j | C_j \in \widehat{\boldsymbol{Z}}_i$
        i. $\hat{P}_{ij} \leftarrow (\hat{P}_{ij} + \hat{P}_{ik} \cdot \hat{P}_{kj})/(1 - P_{ki} \cdot P_{ik})$ // corrects the percentages of transfer to each neighbor

In Procedure 9, if C_j wasn't a neighbor of C_i, the old value of $\hat{P}_{ij}$ is considered zero. If C_j wasn't a neighbor of C_k them the old value of $\hat{P}_{kj}$ is considered zero. At the same time, if C_i wasn't a neighbor of C_k , P_{ki} is treated as being zero.

Lemma 9: edge removal

The application of Procedure 9 by $\boldsymbol{UpdateThePoliticalNetworkConveniently}(\dot{\boldsymbol{C}})$, for $C_k \in \boldsymbol{E} \cup \boldsymbol{L}$, removes C_k from the neighbor set of C_i (removes an edge from C_i to C_k) and the value to which the total amount of votes added to any $C_r \in \boldsymbol{R} \cup \{C_0\}$ tends during the execution of Procedure 7, when MAXSTEPS tends to infinite, does not change.

Proof:

The edge removal does not affect the processing of any vote package that is not associated to C_i (not even when they are associated to C_k), thus it suffices to show that when a package VP_i associated to C_i, is processed the use of the modified network structure is equivalent to the use of the original structure accompanied by some operations of package joining that do not affect the results of interest. It is worth noting that a package associated to C_i can only be processed if $C_i \in \boldsymbol{E} \cup \boldsymbol{L}$. Since $C_k \in \boldsymbol{E} \cup \boldsymbol{L}$ by hypothesis, for the present lemma we can assume both C_i and C_k have been elected or eliminated.

Case 1

If C_k and C_i were the only neighbors of each other, any vote package VP_i associated to C_i, when being processed to Procedure 8, would generate a single package associated to C_k, containing all its votes and that when processed would generate a single package associated C_i, closing an infinite cycle and would never transfer votes to any other candidates. Thus, regardless of how big MAXSTEPS is, the number of votes in VP_i would eventually be discarded. In this context, Procedure 9, immediately discards VP_i, what proves the present lemma for this particular case.

Case 2

In the case where $C_i \notin \widehat{\boldsymbol{Z}}_i$, Procedure 9 adds to the percentage of votes that would already be transferred from C_i to each $C_j \in \widehat{\boldsymbol{Z}}_k$, the value $\hat{P}_{ik} \cdot \hat{P}_{kj}$, that is the percentage that would normally be transferred from C_i to C_k (through a package put in $\boldsymbol{H}$) and, latter, from C_k to C_j (when the package were processed).

$\forall C_j, C_j \in (\boldsymbol{L} \cup \boldsymbol{E})$, the processing of the package $VP_i = (C_i, \dot{V}_i, ID)$ by Procedure 8 acting over the original structure would generate a package $VP_j^1 = (C_j, \dot{V}_i \cdot \hat{P}_{ij}, ID)$ and would generate a package $VP_k = (C_k, \dot{V}_i \cdot \hat{P}_{ik}, ID)$. When VP_k were processed it would generate the package $VP_j^2 = (C_j, \dot{V}_i \cdot \hat{P}_{ik} \cdot \hat{P}_{kj}, ID)$. The join of VP_j^1 and VP_j^2 results in $VP_j = (C_j, \dot{V}_i(\hat{P}_{ij} + \hat{P}_{ik} \cdot \hat{P}_{kj}), ID)$ which is identical to the package associated to C_j that is generated by Procedure 8 when it acts over the structure that was modified by Procedure 9. Since by Lemma 8, the join of packages does not affect the results of interest, the modifications made by Procedure 9 also do not.

When $\forall C_j, C_j \in (\boldsymbol{R} \cup \{C_0\})$, the processing of the package $VP_i = (C_i, \dot{V}_i, ID)$ by Procedure 8 acting over the original structure would immediately add $\dot{V}_i \cdot \hat{P}_{ij}$ votes to C_j and would generate a package $VP_k = (C_k, \dot{V}_i \cdot \hat{P}_{ik}, ID)$. When VP_k were processed it would add $\dot{V}_i \cdot \hat{P}_{ik} \cdot \hat{P}_{kj}$ votes to C_j. When Procedure 8 acts over the structure that was modified by Procedure 9, $\dot{V}_i(\hat{P}_{ij} + \hat{P}_{ik} \cdot \hat{P}_{kj})$ votes are added to C_j. The total number of votes added to C_j is the same, thus, the present lemma is proved for the cases in which C_i was not a neighbor of C_k.

Case 3

If C_i was a neighbor of C_k, using the original structure, any package of votes associated to C_i when processed by Procedure 8 would generate a single package associated to C_k and a package associated to each of its neighbors belonging to $\boldsymbol{L} \cup \boldsymbol{E}$. Besides that, the neighbors of C_i belonging to $\boldsymbol{R} \cup \{C_0\}$ would receive direct increments to their votes.

The package associated to C_k, when processed, would, in turn, generate a package associated to C_i and a package associated to each of the other neighbors of C_k belonging to a $\boldsymbol{L} \cup \boldsymbol{E}$. Besides that, the neighbors of C_k belonging to $\boldsymbol{R} \cup \{C_0\}$ would receive direct increments to their votes.

The alternate processing of packages associated to C_i and C_k, in the cycle that would be formed, would generate, for each neighbor of C_i belonging to $\boldsymbol{L} \cup \boldsymbol{E}$ that is not C_k, an infinite set of packages. Besides that, the neighbors of C_i belonging to $\boldsymbol{R} \cup \{C_0\}$ that are not C_k would receive infinite direct increments to their votes. For each neighbor of C_k

belonging to $\boldsymbol{L} \cup \boldsymbol{E}$ that is not C_i another infinite set of packages would be generated. The neighbors of C_k belonging to $\boldsymbol{R} \cup \{C_0\}$ would also receive an infinite number of direct increments to their votes.

$\forall C_j, C_j \in (\boldsymbol{L} \cup \boldsymbol{E}), C_j \in \widehat{\boldsymbol{Z}}_k \cup \widehat{\boldsymbol{Z}}_i$ the processing of any package $VP_i = (C_i, \dot{V}_i, ID)$ by Procedure 8 acting over the structure that was modified by Procedure 9 generates a single package $VP_j = (C_j, VT_j, ID)$, associated to C_j.

For $\forall C_j, C_j \in (\boldsymbol{R} \cup \{C_0\}), C_j \in \widehat{\boldsymbol{Z}}_k \cup \widehat{\boldsymbol{Z}}_i$ the processing of any package $VP_i = (C_i, \dot{V}_i, ID)$ by Procedure 8 acting over the structure that was modified by Procedure 9 increments the amount of votes in C_j, V_j, by the value of VT_j.

It is direct from Procedure 8 that the amount of transferred votes, VT_j, does not depend on whether $C_j \in (\boldsymbol{R} \cup \{C_0\})$ or $C_j \in (\boldsymbol{L} \cup \boldsymbol{E})$. In all cases, $VT_j = \frac{\hat{P}_{ij} + \hat{P}_{ik} \cdot \hat{P}_{kj}}{1 - P_{ki} \cdot P_{ik}}$.

Thus, it suffices to show that VT_j is equal to the sum of the votes transferred to C_j (either directly or by the generation of packages) in the alternate processing of packages associated to C_i and C_k when Procedure 8 acts over the original structure and MAXSTEPS tends to infinite. In the case in which $C_j \in (\boldsymbol{L} \cup \boldsymbol{E})$ that means that the use of the modified structure has the effect o joining all packages associated to C_j, generated in the alternating process, into an equivalent package, what by Lemma 8 does not change the results. In the case where $C_j \in \boldsymbol{R} \cup \{C_0\}$ that means that the sum of the increments to V_j is the same whether or not the structure is modified.

Let $C_{\hat{k}}$ be a neighbor of C_k that is not C_i. Let $C_{\hat{\imath}}$ be a neighbor of C_i that is not C_k. Let's observe the alternating process between packages associated to C_i and C_k and sum the votes transferred to $C_{\hat{k}}$ and $C_{\hat{\imath}}$.

In the first step, $VP_i = (C_i, \dot{V}_i, ID)$ is processed. In this step, a number of votes equal to $\dot{V}_i \cdot P_{i\hat{\imath}}$ are transferred to $C_{\hat{\imath}}$ and it is generated a package associated to C_k containing $\dot{V}_i \cdot P_{ik}$ votes.

In the second step, a package associated to C_k containing $\dot{V}_i \cdot P_{ik}$ votes is processed. So, $C_{\hat{k}}$ receives $\dot{V}_i \cdot P_{ik} \cdot P_{k\hat{k}}$ votes. It is also generated a package associated to C_i containing $\dot{V}_i \cdot P_{ik} \cdot P_{ki}$ votes.

In the third step, a number of votes equal to $\dot{V}_i \cdot P_{ik} \cdot P_{ki} \cdot P_{i\hat{\imath}}$ are transferred to $C_{\hat{\imath}}$ and it is generated a package associated to C_k containing $\dot{V}_i \cdot P_{ik} \cdot P_{ki} \cdot P_{ik}$.

In the forth step, a number of votes equal to $\dot{V}_i \cdot P_{ik} \cdot P_{ki} \cdot P_{ik} \cdot P_{k\hat{k}}$ are transferred to $C_{\hat{k}}$. It is also generated a package associated to C_i containing $\dot{V}_i \cdot P_{ik} \cdot P_{ki} \cdot P_{ik} \cdot P_{ki}$ votes. We can observe that the amount of votes transferred to $C_{\hat{k}}$ in each even step is equal to the amount transferred in the prior even step multiplied by $P_{ki} \cdot P_{ik}$ what is the result of a movement from C_k to C_i and another in the inverse direction. This way, we have that the sum of votes transferred from C_k to a $C_{\hat{k}}$, $TV_{k\hat{k}}$, in the infinite process is given by

$$TV_{k\hat{k}} = \lim_{l\to\infty} \sum_{t=0}^{l} \dot{V}_i \cdot P_{ik} \cdot \ P_{k\hat{k}} \cdot (P_{ki} \cdot P_{ik})^t = \dot{V}_i \cdot \frac{P_{k\hat{k}} \cdot P_{ik}}{1 - P_{ki} \cdot P_{ik}}$$

At the same time, we can observe that the amount of votes transferred to $C_{\hat{\imath}}$ in an odd step is equal to the amount of votes transferred in the prior odd step multiplied by $P_{ik} \cdot P_{ki}$. Thus, we have that the sum of votes transferred from C_i to $C_{\hat{\imath}}$ in the infinite process, $TV_{i\hat{\imath}}$ is given by

$$TV_{i\hat{\imath}} = \lim_{l\to\infty} \sum_{t=0}^{l} \dot{V}_i \cdot \ P_{i\hat{\imath}} \cdot (P_{ki} \cdot P_{ik})^t = V_i \cdot \frac{P_{i\hat{\imath}}}{1 - P_{ki} \cdot P_{ik}}$$

Therefore, $\forall C_j \in (\widehat{\boldsymbol{Z}}_k \cup \widehat{\boldsymbol{Z}}_i)$, the sum of votes transferred to C_j in the infinite process, TV_j is given by

$$TV_j = TV_{kj} + TV_{ij} = V_i \cdot \frac{\hat{P}_{ij} + \hat{P}_{ik} \cdot \hat{P}_{kj}}{1 - P_{ki} \cdot P_{ik}}.$$

Since that is that is exactly the amount of votes transferred to C_j when Procedure 8 acts over the structure that was modified by Procedure 9, the present lemma is proved. ■

With Procedure 9, we can remove all candidates that have already been elected or eliminated from the neighbor sets of all candidates, that is, remove all edges going to non remaining candidates. To do that we use Procedure 10 and Procedure 11.

Procedure 10: $\boldsymbol{RemoveFromNeighborSets}(\boldsymbol{C_k})$

1. $\forall C_i | C_k \in \widehat{\boldsymbol{Z}}_i$
   a. $\boldsymbol{RemoveFromNeighborSet}(C_k, C_i)$

Lemma 10: removal of all edges going to a non remaining candidate

After the application of Procedure 10, the candidate C_k, is not in the neighbor set of any other candidate and the use of the modified structure does not affect the result of Procedure 8 when $MAXSTEPS$ tends to infinite.

Proof:

Procedure 9 was called to eliminate C_k from the neighbor set of each existing candidate and, by Lemma 9, did that in a way that does not affect the results of Procedure 8.

It is direct from Procedure 9, that it never creates edges going to C_k in this process what proves the present lemma. ■

Procedure 11: $\boldsymbol{RemoveFromNeighborSets}(\dot{\boldsymbol{C}})$

1. $\forall C_k | C_k \in \dot{\boldsymbol{C}}$
   a. $\boldsymbol{RemoveFromNeighborSets}(C_k)$

Lemma 11: removal of all edges going to a subset of non remaining candidates

Using Procedure 11 with any $\dot{\boldsymbol{C}} \subset (\boldsymbol{E} \cup \boldsymbol{L})$ as a parameter we obtain a network structure where no candidate $C_k \in \dot{\boldsymbol{C}}$ appears in the neighbor set of any candidate and that never changes the results of Procedure 8 when $MAXSTEPS$ tends to infinite.

Proof:

By Lemma 10, Procedure 10 removes a non remaining candidate, C_k, received as a parameter from the neighbor sets of all candidates in a way that does not affect the result of Procedure 8. If C_k wasn't in any neighbor set, Procedure 10 does nothing.

It is direct from Procedure 11, Procedure 10 and Procedure 9 that it is never created an edge going to a candidate that wasn't already the destiny of some edge. Thus, a candidate that has been removed from the neighbor sets of all candidates is never put back in any of them.

Since Procedure 11 calls Procedure 10 for every candidate in $\dot{C}$ a sequence, the structure produced by Procedure 11, cannot contain, any edges to a candidate in $\dot{C}$ what proves the present lemma. ■

Lemma 12: removal of all edges going to non remaining candidates in an election process

Using Procedure 11 as $\boldsymbol{UpdateThePoliticalNetworkConveniently}(\dot{\boldsymbol{C}})$ we obtain a network structure where no non remaining candidate appears in the neighbor set of any candidate and the result of Procedure 8, when $MAXSTEPS$ tends to infinite, does not change.

Proof:

By Lemma 10, Procedure 10 removes a non remaining candidate, C_k, received as a parameter from the neighbor sets of all candidates in a way that does not affect the result of Procedure 8. If C_k wasn't in any neighbor set, Procedure 10 does nothing.

It is direct from Procedure 11, Procedure 10 and Procedure 9 that it is never created an edge going to a candidate that wasn't already the destiny of some edge. Thus, a candidate that has been removed from the neighbor sets of all candidates is never put back in any of them.

It is direct from the implementations of Procedure 3, Procedure 4, Procedure 5 and Procedure 6 that all candidates that have been eliminated or elected must have been once include in $\dot{\boldsymbol{C}}$ in a call to $\boldsymbol{UpdateThePoliticalNetworkConveniently}(\dot{\boldsymbol{C}})$

This way, the structure produced by Procedure 11, cannot contain, any edges to a non remaining candidate what proves the present lemma. ■

In Lemma 13, we show that a complete execution of Procedure 7 can be done in a finite number of steps as long as we can, in advance, calculate the total amount of votes that will be discarded. In Lemma 14 we show how to do that calculation.

Lemma 13: transfer of votes in a finite number of steps

If we implement procedure $\boldsymbol{UpdateThePoliticalNetworkConveniently}(\dot{\boldsymbol{C}})$ as Procedure 11 and implement $\boldsymbol{RemoveAPackage}(\boldsymbol{H})$ without package splits, Procedure 7 will always end in a finite number of steps and its results will be the same that we would get if $\boldsymbol{UpdateThePoliticalNetworkConveniently}(\dot{\boldsymbol{C}})$ was doing nothing with $MAXSTEPS$ tending to infinite.

Proof:

By Lemma 12 the structure modified by $\boldsymbol{UpdateThePoliticalNetworkConveniently}(\dot{\boldsymbol{C}})$ will not contain edges going to non remaining candidates and the results of Procedure 8, when $MAXSTEPS$ tends to infinite won't change because of the modifications.

It is direct from Procedure 8 that in the absence of such edges, no vote package is put in $\boldsymbol{H}$ during its execution.

Before calling Procedure 8, Procedure 7, adds to $\boldsymbol{H}$ a number of packages that is equal to the number of elected candidates possibly added by 1 if a candidate has just been eliminated.

So, as long as $\boldsymbol{RemoveAPackage}(\boldsymbol{H})$ does not do any divisions of packages (what it has no reason to do) Procedure 8 ends in a number of steps that is proportional to the number of elected candidates what proves the present lemma. ■

Lemma 14: number of discarded votes and current quota

Let $\tilde{V}$ be the value returned by $\boldsymbol{CalculateTotalDiscards}()$ and $\dot{Q}$ the value calculated in Procedure 6 for the quota in the end of Procedure 7. The values of $\dot{Q}$ and $\tilde{V}$ must be given by

$$\dot{Q} = \frac{\hat{V} - \sum_{\forall C_i \in \boldsymbol{L} \cup \boldsymbol{E}} \hat{V}_i \cdot P_{i0}}{\left(M - \sum_{\forall C_i \in \boldsymbol{L}} P_{i0}\right)}$$

and

$$\tilde{V} = \sum_{\forall C_i \in \boldsymbol{L}} (\hat{V}_i - \dot{Q}) \cdot P_{i0} + \sum_{\forall C_i \in \boldsymbol{E}} \hat{V}_i \cdot P_{i0}$$

where M is the number of available seats, $\boldsymbol{L}$ is the set of already elected candidates, $\boldsymbol{E}$ is the set of already eliminated candidates, $\hat{V}$ is the current number of valid votes in the beginning of Procedure 6 and $\hat{V}_i$ is the current number of votes of candidate C_i in the beginning of Procedure 6.

Proof:

In the end of Procedure 7, we must have that $\hat{Q} = \dot{Q}$, where $\hat{Q}$ is the current quota, that is, we should have

$$\dot{Q} = (\hat{V} - \tilde{V})/M$$

where $\tilde{V}$ it the total number of votes discarded by Procedure 7 and $\hat{V}$ in the current number of valid votes in the beginning of Procedure 6.

It is direct from Procedure 6, Procedure 7 and Procedure 8 that if $\dot{Q}$ is smaller than that value, non transferrable votes will be transferred from elected candidates. If it is larger, some transferrable votes will fail to be transferred.

In the end of Procedure 7, each elected candidate will have exactly $\dot{\boldsymbol{Q}}$ votes**.** Thus, $\hat{V}_i - \dot{Q}$ is the number of votes in the vote package that is put in $\boldsymbol{H}$ by Procedure 7 for any candidate $C_i \in \boldsymbol{L}$**.**

In the case of eliminated candidates, Procedure 7 creates packages with all their votes (in practice there can be at most one candidate with more than zero votes at this point, but that is not relevant).

By Lemma 13, without changing the results of Procedure 7, we can replace the original political network by a network where no edge goes to a non remaining candidate, except for the virtual discard candidate. In the modified network, the percentage of transfer from any candidate C_i to the discard virtual candidate is P_{i0} and, since remaining candidates keep all their votes, there are no indirect vote discards.

This way, the total number of votes discarded by Procedure 7 is given by

$$\tilde{V} = \sum_{\forall C_i \in \boldsymbol{L}} (\hat{V}_i - \dot{Q}) \cdot P_{i0} + \sum_{\forall C_i \in \boldsymbol{E}} \hat{V}_i \cdot P_{i0}$$

Solving the equation

$$M \cdot \dot{Q} = \hat{V} - \sum_{\forall C_i \in \boldsymbol{L}} (\hat{V}_i - \dot{Q}) \cdot P_{i0} - \sum_{\forall C_i \in \boldsymbol{E}} \hat{V}_i \cdot P_{i0}$$

we get a single solution

$$M \cdot \dot{Q} = \hat{V} - \sum_{\forall C_i \in \boldsymbol{L}} \hat{V}_i \cdot P_{i0} + \dot{Q} \sum_{\forall C_i \in \boldsymbol{L}} P_{i0} - \sum_{\forall C_i \in \boldsymbol{E}} \hat{V}_i \cdot P_{i0}$$

$$\dot{Q}\left(M - \sum_{\forall C_i \in \boldsymbol{L}} P_{i0}\right) = \hat{V} - \sum_{\forall C_i \in \boldsymbol{L}} \hat{V}_i \cdot P_{i0} - \sum_{\forall C_i \in \boldsymbol{E}} \hat{V}_i \cdot P_{i0}$$

$$\dot{Q} = \frac{\hat{V} - \sum_{\forall C_i \in \boldsymbol{L} \cup \boldsymbol{E}} \hat{V}_i \cdot P_{i0}}{\left(M - \sum_{\forall C_i \in \boldsymbol{L}} P_{i0}\right)}$$

what proves the present lemma. ■

In Theorem 1, we show that the final number of elected candidates matches the number of available seats. To prepare for that, in Corollary 5, we state some restrictions related the value of the current quota.

Corollary 5: some restrictions to the value of the current quota

The current quota, $\hat{Q}$:

- is zero from the beginning of the election, if there are no valid votes at all;
- becomes zero if no remaining candidates have any votes at all, no remaining candidates can be reached by vote transfers originated from elected candidates and there are less than M elected candidates,
- stays positive if any remaining candidate have any votes at all or can be reached by vote transfers originated from elected candidates and there are less than M elected candidates.

Proof:

From Lemma 14, the formula to calculate $\hat{Q}$ is

$$\dot{Q} = \frac{\hat{V} - \sum_{\forall C_i \in \boldsymbol{L} \cup \boldsymbol{E}} \hat{V}_i \cdot P_{i0}}{\left(M - \sum_{\forall C_i \in \boldsymbol{L}} P_{i0}\right)}$$

The initial quota is calculated before there are any elected or eliminated candidates. It is given by $Q = V/M$ and is only zero if the total amount of valid votes is zero.

If there are less than M elected candidates, since each P_{i0} is at most 1 and $\hat{V} = \sum_{\forall C_i \in \boldsymbol{L} \cup \boldsymbol{E} \cup \boldsymbol{R}} \hat{V}_i$, $\dot{Q}$ can only become zero if, $\forall C_i \in \boldsymbol{L}, P_{i0} = 1 \vee \hat{V}_i = 0$, $\sum_{\forall C_i \in \boldsymbol{R}} \hat{V}_i = 0$ and $\sum_{\forall C_i \in E} \hat{V}_i = 0$. We can ignore the last equality because, if remaining candidates have no votes, eliminated candidates, even before vote transfers, must have no votes too.

While there are less than M candidates elected, the denominator of the formula, stays positive.

This proves the present corollary. ■

Theorem 1: total number of elected candidates

If there are at least M candidates and at least one candidate receives a vote, at the end of the execution of Procedure 2, exactly M candidates are elected.

Proof:

Candidates are declared elected in a loop, the elections loop, that alternates between detecting those that have reached the current quota and operations of vote transfer, that cause the current quota to be updated. To declare that a candidate is elected, the current quota calculated in the prior step is used.

While this loop is running the number of eliminated candidates, $|\boldsymbol{E}|$, is fixed so is $|\boldsymbol{L}| + |\boldsymbol{R}|$, the sum of the number of elected and remaining candidates.

Let's call the total number of votes in remaining candidates V_R.

Let's consider an iteration of the elections loop such that before it starts $|\boldsymbol{L}| < M$ and $|\boldsymbol{L}| + |\boldsymbol{R}| > M$ and $V_R \neq 0$**.** Let's call this conditions **typical conditions**.

Since $V_R \neq 0$, by Corollary 5, $\hat{Q} \neq 0$.

As it can be directly verified, before the loop begins, a votes transfer, must have just occurred. Thus, by Lemma 2, each already elected candidate has exactly $\hat{Q}$ votes. If the set of remaining candidates with at least $\hat{Q}$ votes, $\boldsymbol{R_Q}$**,** were such that $|\boldsymbol{L}| + |\boldsymbol{R_Q}| > M$, the number of votes in $\boldsymbol{L} \cup \boldsymbol{R_Q}$ would be at least $(M+1)\hat{Q}$ votes. Since $M\hat{Q} = \hat{V}$, that would be more votes than the current number of valid votes, what is impossible.

If the set of remaining candidates with at least $\hat{Q}$ votes, $\boldsymbol{R_Q}$**,** is such that $|\boldsymbol{L}| + |\boldsymbol{R_Q}| = M$, the number of votes in $\boldsymbol{L} \cup \boldsymbol{R_Q}$ would be at least $M\hat{Q}$ votes. Since $M\hat{Q} = \hat{V}$, that would be at least as many votes as the current number of valid votes, that means that all candidates in $|\boldsymbol{R_Q}|$ have exactly $\hat{Q}$ votes and no other remaining candidates have any votes at all. After the candidates in $\boldsymbol{R_Q}$ are declared elected, we have that $|\boldsymbol{L}| = M$, $|\boldsymbol{L}| + |\boldsymbol{R}| > M$ and $V_R = 0$. This causes the end of the election with exactly M elected candidates and without further vote transfers. Actually there are no votes to be transferred, but there are candidates to be eliminated, what is promptly done . The current quota can become undefined, but it is never used again.

If the set of remaining candidates with at least $\hat{Q}$ votes, $\boldsymbol{R_Q}$, is such that $|\boldsymbol{L}| + |\boldsymbol{R_Q}| < M$, after the candidates in $\boldsymbol{R_Q}$ are declared elected we have to consider two possibilities: there are other remaining candidates with more than zero votes or that can receive transfers from elected candidates or there are not.

If there are no other remaining candidates with more than zero votes or that can receive transfers from elected candidates, the remaining seats are filled promptly, using an arbitrary criterion, and the election is terminated with exactly M candidates elected .

If there are other remaining candidates with more than zero votes or that can receive transfers from elected candidates, by Corollary 5, the vote transfers will end with $\hat{Q} \neq 0$. Thus, the elected candidates will have at most $(M-1)\hat{Q}$ votes. Since $M\hat{Q} = \hat{V}$, that would be less votes than the current number of valid votes, what means that $V_R \neq 0$.

So we have that $|\boldsymbol{L}| < M$ and $|\boldsymbol{L}| + |\boldsymbol{R}| > M$ and $V_R \neq 0$, which are exactly the conditions of the beginning o the iteration. All the analysis presented is valid for the next iteration of the loop.

The loop will end in one of the special conditions that terminate the election or when there no more remaining candidates with at least as many votes as the current quota. So, if the election continues, in the end of the loop, it will still be valid that $|\boldsymbol{L}| < M$ and $|\boldsymbol{L}| + |\boldsymbol{R}| > M$ and $V_R \neq 0$. This implies that $|\boldsymbol{R}| \geq 2$.

After that, one elimination take place and the loop is run again.

An elimination can never reduce V_R to zero, because, if there are any candidates with zero votes, one of them must be the eliminated. If no remaining candidates has zero votes,$V_R$could only become zero if $\boldsymbol{R}$ became empty, but in the end of the elections loop $|\boldsymbol{R}| \geq 2$, so a single elimination cannot do that.

So, if after the elimination, $|\boldsymbol{L}| + |\boldsymbol{R}| > M$, the loop will run again under the same conditions and by the prior analysis will either terminate the election correctly or end again with the same conditions.

So, eliminations will go on one by one and eventually we will have that $|\boldsymbol{L}| + |\boldsymbol{R}| = M$.

The loop will start with $|\boldsymbol{L}| < M$ and $|\boldsymbol{L}| + |\boldsymbol{R}| = M$, $V_R \neq 0$. Let's call this conditions, **ending conditions**.

Under this conditions, by Corollary 5, $\hat{Q} \neq 0$.

We needed the condition that $V_R \neq 0$ to deduce that $\hat{Q} \neq 0$. After that we can ignore that condition and assume that the loop started simply with $|\boldsymbol{L}| < M$ and $|\boldsymbol{L}| + |\boldsymbol{R}| = M$ and $\hat{Q} \neq 0$.

The average number of votes in $\boldsymbol{L} \cup \boldsymbol{R}$ is $\frac{\hat{V}}{M} = \hat{Q}$. Thus there must be at least one remaining candidate with at least $\hat{Q}$ votes and that candidate will be declared elected. Since $|\boldsymbol{L}| + |\boldsymbol{R}| = M$, there is no possibility that more than M candidates are elected.

After that, if there are no other remaining candidates with more than zero votes or that can receive transfers from elected candidates, the remaining seats are filled promptly and the election is terminated with exactly M candidates elected. Otherwise, the we have that $|\boldsymbol{L}| < M$ and $|\boldsymbol{L}| + |\boldsymbol{R}| = M$ and $\hat{Q} \neq 0$. This are

conditions of the beginning that were used in the prior analysis. So the loop goes on, always electing at least one candidate, till the termination condition is met and the election is ended with M candidates elected.

When the election loop is run for the first time, since $\boldsymbol{E} = \boldsymbol{L} = \emptyset$, $|\boldsymbol{R}| \geq M$ and there is at least one candidate the received more than zero votes we must have that either the **typical conditions,** $|\boldsymbol{L}| < M$ and $|\boldsymbol{L}| + |\boldsymbol{R}| = M$, $V_R \neq 0$ or

or the **ending conditions,** $|\boldsymbol{L}| < M$ and $|\boldsymbol{L}| + |\boldsymbol{R}| = M$, $V_R \neq 0$**,** must be met. Both lead the election to an end where there are exactly M elected candidates.

This proves the present theorem. ■

Corollary 6, is useful to implement the electoral system in a computer that is performing finite precision calculations. Typically, the last candidate, may have a number of votes that is extremely close to the quota, but, because of rounding errors not exactly the quota. With Corollary 6, such candidate may be claimed to be elected without worrying about that.

Corollary 6: election of the last remaining candidates

If the number of remaining candidates is equal to $M - |L|$ then all remaining candidates will eventually be elected.

Proof:

This corollary is immediate from Theorem 1. ■

We will show in Theorem 2, that the result of an election using the political network system is unique.

Theorem 2: uniqueness of the result of a political network election

The result of an election that follows the definition of the political network electoral system is unique.

Proof:

The definition of the political network electoral system leads directly to Procedure 2, Procedure 3, Procedure 4 and Procedure 5, which are completely deterministic except for the use of $\boldsymbol{TransferVotesAndUpdateQuota}(\dot{\boldsymbol{C}})$, which has some freedom. However, Lemma 3 guarantees that any implementation of $\boldsymbol{TransferVotesAndUpdateQuota}(\dot{\boldsymbol{C}})$ must be equivalent to a variation of Procedure 6. Procedure 6, has some freedom through the functions $\boldsymbol{CalculateTotalDiscards}()$ and $\boldsymbol{RemoveAPackage}(\boldsymbol{H})$ and through the procedure $\boldsymbol{UpdateThePoliticalNetworkConveniently}(\dot{\boldsymbol{C}})$. The latest, by Definition 12, cannot affect the result of the election. The function $\boldsymbol{RemoveAPackage}(\boldsymbol{H})$, by Lemma 8 and Corollary 4, cannot change the result of Procedure 7, that is used to perform vote transfers assuming that a pre-calculated value for the current quota will be correct at the end of the transferring process. Such value is defined by the return of $\boldsymbol{CalculateTotalDiscards}()$, which by Lemma 14 is unique.

Therefore, regardless of implementation variations, the value of Procedure 6 cannot change, what proves the present theorem. ■

Theorem 3: election in a finite number of steps

The result of an election that uses the political network electoral system can be found in a finite number of steps.

Proof:

The definition of the political network electoral system leads directly to Procedure 2, Procedure 4 and Procedure 5, which clearly involve finite numbers of steps, except for their use of the procedure $\boldsymbol{TransferVotesAndUpdateQuota}(\dot{\boldsymbol{C}})$.

By Lemma 3 this procedure can be implemented as variation of Procedure 6, which clearly involve only finite steps directly, but uses Procedure 7. Procedure 7, in turn, by Lemma 13, can be implemented in a way that ends in a finite number of steps.

Procedure 3, uses the function $\boldsymbol{DetectExitCondition}(\ )$, whose implementation we didn't show. It needs to check if any remaining candidates have any votes and if all percentages of transfer from elected candidates to the virtual discard candidate are equal to 1. This can clearly be done in finite steps.

This proves the present theorem. ■

Lemma 15 is also useful to implement the electoral system with finite precision calculations. It shows that eliminated candidates can never be too close to the current quota.

Lemma 15: minimum vote gap

If the number of non eliminated candidates is at least $M+1$, then, in Procedure 2, immediately after the execution of Procedure 3, there is at least one candidate with at most $\frac{M}{M+1}\cdot\hat{Q}$ votes, where $\hat{Q}$ is the current quota and M is the number of seats available.

Proof:

Since, by Lemma 2, eliminated candidates have zero votes, the average number of votes in non eliminated candidates is

$$\frac{\hat{V}}{N-|\boldsymbol{E}|}\leq\frac{\hat{V}}{M+1}=\frac{M}{M+1}\cdot\hat{Q}.$$

Since some candidate must have a number of votes that is at most equal to the average, the present theorem is proved. ■

Relation to most open party-list systems

In this section, we will establish a floor to the number of elected candidates of a network solid coalition (Theorem 4) that is immediately applicable to party party-list coalitions (Corollary 10).

We will also show that the elected members of a the party-list coalition are always those who received more individual votes (Theorem 5).

Definition 22: party-list coalition

We say that a set of candidates $\boldsymbol{N}_p$ form a party-list coalition if and only if $\forall i|C_i\in\boldsymbol{N}_p,\boldsymbol{Z}_i=\boldsymbol{N}_p-\{C_i\}$ and $\forall i,j|C_i\in\boldsymbol{N}_p,C_j\in\boldsymbol{N}_p,P_{ij}=1/|\boldsymbol{N}_p-1|$.

Definition 23: network solid coalition

We say that a set of candidates $\boldsymbol{N}_p$ form a network solid coalition if and only if there is a path from any candidate in the coalition to any other candidate in the coalition that only passes through candidates within the coalition and $\forall C_i\in\boldsymbol{N}_p,\boldsymbol{Z}_i\subset\boldsymbol{N}_p$.

The definition above is equivalent to saying that the induced subgraph (Diestel 2000) on the political network by $\boldsymbol{N}_p$ is strongly connected (Tarjan 1972) and that the coalition has no outgoing edges.

Corollary 7: solidity of party-list coalitions

Party-list coalitions are network solid coalitions

Proof:

In a party-list coalition there is a direct path from any candidate to any other and there are no edges going to a candidate outside the coalition.

This proves the present corollary. ■

Lemma 16: resilience of network solid coalitions

If a set of candidates $\boldsymbol{N}_p$ form a network solid coalition in the original political network then, in any network structure modified by Procedure 11 where $\boldsymbol{N}_p \cap \boldsymbol{R} \neq \emptyset$**,** there is a path from any candidate in $\boldsymbol{N}_p$ to any candidate in $\boldsymbol{N}_p \cap \boldsymbol{R}$ that only passes through candidates within $\boldsymbol{N}_p$ and there are no edges going from any candidate inside the original coalition to any candidate outside it, except for one edge going from the last remaining candidate in the coalition to the virtual discard candidate, C_0.

Proof:

Since the candidates in $\boldsymbol{N}_p$ form a network solid coalition then there is a path from any candidate in $\boldsymbol{N}_p$ to any other candidate in $\boldsymbol{N}_p$ that only passes through candidates within $\boldsymbol{N}_p$.

We are interested in the fact that there is a path from any candidate in $\boldsymbol{N}_p$ to any candidate in $\boldsymbol{N}_p \cap \boldsymbol{R}$ that only passes through candidates within $\boldsymbol{N}_p$, what is a particular case of the assertion above. Thus, initially, it is true.

It is also initially true that

When a candidate C_j stops being a remaining candidate, Procedure 10 is called by Procedure 11 and modifies the structure of the network usually removing all incoming edges of C_j and adding edges going from C_k to C_l, $\forall C_k, C_l | k \neq l, C_j \in \boldsymbol{Z}_k \wedge C_l \in \boldsymbol{Z}_j$, what clearly adds no edges going away from the coalition, if there were no edges going away from the coalition already.

In the only special case in which $\exists C_k | \boldsymbol{Z}_j = \{C_k\} \wedge \boldsymbol{Z}_k = \{C_j\}$ Procedure 10 adds an edge from C_k to C_0, where C_0 is the virtual discard candidate. However, since there is a path from C_k to any candidate in $\boldsymbol{N}_p \cap \boldsymbol{R}$ and since there can only be an edge going from C_j to C_k if C_k is, itself, a remaining candidate, the special case can only happen if $\boldsymbol{N}_p \cap \boldsymbol{R} = \{C_j, C_k\}$. In this case $\boldsymbol{N}_p \cap \boldsymbol{R}$ would become equal to C_k and the present lemma's statement admits an edge going from the last remaining candidate in $\boldsymbol{N}_p$ to C_0.

Latter, when C_k ceases to be a remaining candidate, Procedure 11 will add edges from other candidates in $\boldsymbol{N}_p$ to C_0, but that is also admitted.

We still have to prove that, after the execution of Procedure 10, there is still a path from any candidate in $\boldsymbol{N}_p$ to any candidate in $\boldsymbol{N}_p \cap \boldsymbol{R}$ that only passes through candidates within $\boldsymbol{N}_p$.

We can see that that is true noting that if the old path from a candidate C_s to a candidate C_t with $C_s \in \boldsymbol{N}_p \wedge C_t \in \boldsymbol{N}_p \cap \boldsymbol{R}$ did not pass through C_j it remains unaffected by Procedure 10. If it passed through C_j then there was a path from C_s till some $C_k |\ C_j \in \boldsymbol{Z}_k$ only passing through candidates in $\boldsymbol{N}_p$and there was a path from some $C_l | C_l \in \boldsymbol{Z}_j$ to C_t only passing through candidates in $\boldsymbol{N}_p$. Procedure 10 adds an edge from C_k to C_l, $\forall C_k, C_l | k \neq l$ guaranteeing that a path from C_s to C_t still exists.

A path from C_s to C_j itself ceases to exist, but that does not matter since C_j stops belonging to $\boldsymbol{R}$.

This proves the present lemma. ■

Corollary 8: resilience of votes in network solid coalitions

If a set of candidates $\boldsymbol{N}_p$ form a network solid coalition, then all votes received by any candidates belonging to $\boldsymbol{N}_p$ remain in candidates belonging to $\boldsymbol{N}_p$ while there is still $C_j | C_j \in \boldsymbol{N}_p \cap \boldsymbol{R}$ (there is still a remaining candidate in the coalition).

Proof:

By Lemma 16, while $\boldsymbol{N}_p \cap \boldsymbol{R} \neq \emptyset$, $\boldsymbol{N}_p$ has no outgoing edges except for one edge going from the last remaining candidate in the coalition to the virtual discard candidate, C_0. Since remaining candidates never transfer any votes, no votes originally received by candidates belonging to $\boldsymbol{N}_p$ can be transferred to candidates outside $\boldsymbol{N}_p$ (including C_0). So, these votes can only be transferred to other candidates within $\boldsymbol{N}_p$.

This proves the present corollary. ■

Lemma 17: minimum number of votes in network solid coalition candidates

If the number of non eliminated candidates in a network solid coalition is $\left\lfloor \frac{V_p \cdot M}{V} \right\rfloor$, then, in Procedure 2, immediately after the execution of Procedure 3, where candidates are claimed to be elected, each candidate in this coalition have exactly $\hat{Q}$ votes, where $\hat{Q}$ is the current quota, M is the number of seats available, V_p is the total number of votes of the party and V is the total number of valid votes.

Proof:

By Corollary 8, all votes originally received by candidates in $\boldsymbol{N}_p$ are always in candidates in $\boldsymbol{N}_p$. So the average number of votes of the candidates in $\boldsymbol{N}_p$ is

$$\frac{V_p}{\left\lfloor\frac{V_p \cdot M}{V}\right\rfloor} \geq \frac{V_p}{\frac{V_p \cdot M}{V}} = \frac{V}{M} \geq \frac{\hat{V}}{M} = \hat{Q}.$$

All candidates with $\hat{Q}$ or more votes are elected in the loop that precedes the call to Procedure 4 in Procedure 2. By Lemma 2, all elected candidates have exactly $\hat{Q}$ votes after the vote transfer that happens within that loop. Thus, no candidate in can $\boldsymbol{N}_p$ can have more them $\hat{Q}$. To achieve the calculated average, they all must have exactly $\hat{Q}$. ■

Theorem 4: minimum number of elected candidates in a network solid coalition

If a set of candidates $\boldsymbol{N}_p, |\boldsymbol{N}_p| \geq \left\lfloor\frac{V_p \cdot M}{V}\right\rfloor$, form a network solid coalition, a number $M_p \geq \left\lfloor\frac{V_p \cdot M}{V}\right\rfloor$ of its members are elected, where M is the number of seats available, V_p is the total number of votes of the party and V is the total number of valid votes.

Proof:

By Lemma 17, immediately after the call to Procedure 3, if the number of non eliminated candidates in a network solid coalition is $\left\lfloor\frac{V_p \cdot M}{V}\right\rfloor$, all candidates in $\boldsymbol{N}_p$ have exactly $\hat{Q}$, what implies directly that they all must have been elected. Since eliminations happen one by one and $|\boldsymbol{N}_p| \geq \left\lfloor\frac{V_p \cdot M}{V}\right\rfloor$, at some point, before the number of non eliminated candidates of $\boldsymbol{N}_p$ becomes smaller them $\left\lfloor\frac{V_p \cdot M}{V}\right\rfloor$, it must become equal, causing the elections of them all. This proves the present theorem. ■

In Corollary 9, we show that there is a relatively big gap between the number of votes in any candidate in a network solid coalition whose number of non eliminated candidates has reached the theoretical minimum and the remaining candidate with the least number of votes. This means that rounding errors cannot break the guarantee of the minimum number of elected candidates for network solid coalitions. If the candidates that should be elected immediately for reaching the current quota happen not to be so because of rounding errors, they cannot be eliminated either. They stay as remaining candidates till they receive more votes, the current quota is reduced, or Corollary 6 is applied. In any case they are eventually elected.

Corollary 9: vote gap for network solid coalitions

If the number of non eliminated candidates in a network solid coalition is $\left\lfloor\frac{V_p \cdot M}{V}\right\rfloor$, then, in Procedure 2, immediately after the execution of Procedure 3, where candidates are claimed to be elected, each candidate in this coalition have at least $(1-\frac{M}{M+1})\hat{Q}$ votes beyond the number of votes of the remaining candidate with the least number of votes , where $\hat{Q}$ is the current quota, M is the number of seats available.

Proof:

By Lemma 15,, there is at least one remaining candidate with at most $\frac{M}{M+1}\cdot\hat{Q}$. By Lemma 17, the number of votes of any candidate in the network solid coalition is $\hat{Q}$. ■

Corollary 10: minimum number of elected candidates in a party-list coalition

If a set of candidates, $\boldsymbol{N_p}$, form a most open party-list coalition, a number $M_p \geq \left\lfloor\frac{V_p \cdot M}{V}\right\rfloor$ of its members are elected, where M is the number of seats available, V_p is the total number of votes of the party and V is the total number of valid votes.

Proof:

By Corollary 7, a party-list coalition is a network solid coalition. From Theorem 4, the present corollary is immediate. ■

Theorem 5: set of elected candidates in a party-list coalition without incoming edges

If a set of candidates, $\boldsymbol{N_p}$, form a party-list coalition, and none of its members receive votes transfers from candidates that are not in $\boldsymbol{N_p}$ ($\boldsymbol{N_p}$ has no incoming edges), the subset $\boldsymbol{M_p}$ of size M_p containing all elected candidates of $\boldsymbol{N_p}$, contains exactly the M_p candidates that received more individual votes in $\boldsymbol{N_p}$.

Proof:

All candidates that are not elected in the political network system are eliminated when they become the candidate with least votes of all remaining candidates.

When any transfer of votes with origin in a candidate of $\boldsymbol{N_p}$ takes place, the number of votes transferred to each remaining candidate of $\boldsymbol{N_p}$ is the same, since in a party-list coalition, all percentages of transfer are equal.

Since no candidates in $\boldsymbol{N_p}$ receives transfers from outside $\boldsymbol{N_p}$, at any moment in the election, the number of votes transferred to any candidate in $\boldsymbol{N_p}$ is equal to a constant, K.

Thus, if C_1 and C_2, are two candidates belonging to $\boldsymbol{N_p}$, that received respectively V_1 and V_2 individual votes such that $V_1 > V_2$, the values $\hat{V}_1$ and $\hat{V}_2$ representing their current number of votes are such that $\hat{V}_1 = V_1 + K > V_2 + K = \hat{V}_2$.

So C_1 always has more votes then C_2 and C_1 can never be eliminated before C_2.

This proves the present theorem. ■

Definition 24: most open party-list network structure

We say that a political network has a most open party-list structure if and only of all candidates are in party-list coalitions.

Corollary 11: elected candidates of a party-list coalition in a most open party-list network structure

If the political network has a most open party-list structure the elected candidates of any party-list coalition are exactly those who received more individual votes in the coalition.

Proof:

Every candidate is in a party-list coalition and party-list coalitions have no outgoing edges. This implies that no coalitions can have incoming edges either. The application of Theorem 5 to each coalition proves the present corollary. ■

Computational complexity of a political network election and speed test

In Theorem 6, we show that the computational time complexity (Papadimitriou 2003) of a political network election is $O(M \cdot N^2 + N^3/K)$, where N is the number of candidates, M is the number of seats available and K is the number of processors. In Theorem 7, we show that its computational space complexity is $O(N^2)$. Such complexities are not low, but not prohibitive for a modern computer, what proves that the proposed systems is feasible in practice.

Theorem 6: computational time complexity

The computational complexity of the a political network election is $O(M \cdot N^2 + N^3/K)$, where N is the number of candidates and K is the number of processors.

Proof:

Let's consider an implementation of Procedure 8 where $\boldsymbol{RemoveAPackage}(\boldsymbol{H})$ does no splits and where $\boldsymbol{UpdateThePoliticalNetworkConveniently}(\dot{\boldsymbol{C}})$ is implemented as Procedure 11.

Procedure 8 contains a main loop that removes one element of the structure $\boldsymbol{H}$ in each step and since, by Lemma 12, the structure modified by $\boldsymbol{UpdateThePoliticalNetworkConveniently}(\dot{\boldsymbol{C}})$ does not contain any edges going to non remaining candidates, puts no packages back into $\boldsymbol{H}$.

Since, Procedure 7, put in $\boldsymbol{H}$, at most one element for every elected candidate and one element for the most recently eliminated candidate, this loop is executed at most M

times, where M is the number of available seats (the elected candidates are at most $M - 1$, or the election would be already over).

Inside the main loop there is a sub loop that is executed once for every neighbor of the node removed from $\boldsymbol{H}$**.** Every node may have at most N neighbors.

So the computational time complexity of Procedure 7 is $O(M \cdot N)$, where N is the number of candidates and M is the number of available seats.

Other than calling Procedure 7 once, Procedure 6 only executes constant time commands, calls to $\boldsymbol{UpdateThePoliticalNetworkConveniently}(\dot{\boldsymbol{C}})$ and calls $\boldsymbol{CalculateTotalDiscards}()$. The latest, by Lemma 14,

only needs to apply a linear time formula. Thus, except for $\boldsymbol{UpdateThePoliticalNetworkConveniently}(\dot{\boldsymbol{C}})$, whose analysis we will do separately , Procedure 6 is also $O(M \cdot N)$.

At the same time, Procedure 6, is only called when a candidate is eliminated or when at least one candidate is elected. In both cases, at least one candidate is removed from the set of remaining candidates. Since there is no return to this set, Procedure 6 can be called at most N times in the whole election process.

Thus, the computational time complexity of all vote transfers in the full election process, is $O(M \cdot N^2)$.

Procedure $\boldsymbol{UpdateThePoliticalNetworkConveniently}(\dot{\boldsymbol{C}})$, which is implemented as Procedure 11, is called once for every call to Procedure 6 and thus, at most, N times.

Procedure 11 calls Procedure 10 for every candidate, C_k, in the set $\dot{\boldsymbol{C}}$. Candidates in this set are those who have just been elected or eliminated, thus, each candidate, must be is this set once and only once in the whole election. This way, Procedure 10 is called N times in the election.

Every time Procedure 10 is called, it calls Procedure 9 for every node that has C_k as a neighbor. They are at most N, thus Procedure 9 is called at most N^2 times. Procedure 9 includes loops where neighbor sets are scanned and each of them can be of length at most N.

During the execution of Procedure 9, the neighbor sets of the candidates that had C_k as a neighbor are updated based on their prior values and on the neighbor set of C_k, which is fixed. This means that these updates can be run in parallel, without any synchronization delays.

Thus, the time complexity of the $\boldsymbol{UpdateThePoliticalNetworkConveniently}(\dot{\boldsymbol{C}})$ for the full election process is $O(N^3/K)$.

It can be checked directly that other than vote transfers and updates in the network the election involves only less complex operations. The only point that may be unclear is the use of the function $\boldsymbol{DetectExitCondition}(\ )$, whose implementation we didn't show, by Procedure 3. It needs to check if any remaining candidates have any votes and if all percentages of transfer from elected candidates to the virtual discard candidate are all equal to 1. This can clearly be done, at worst, in linear time. The calls to $\boldsymbol{DetectExitCondition}(\ )$ happen only when at least one candidate is elected. So its full complexity is $O(M \cdot N)$.

Therefore, the time complexity of the full election algorithm is

$O(M \cdot N^2 + N^3/K)$. ■

Theorem 7: computational space complexity

The computational space complexity of the a political network election is $O(N^2)$.

Proof:

It is easy to see that the political network election requires the storage of some data whose amount is fixed, like the current quota, some data whose amount is proportional to the number of candidates, like the set of remaining candidates, the set of eliminated candidates and the number of current votes for every candidate and some data whose amount is proportional to the number of relations among candidates like the sets of neighbors and the percentages of transfer for each neighbor.

Thus, the space complexity of the election is $O(C) + O(N) + O(N^2) = O(N^2)$. ■

Analysis of the network structure

To investigate the political network it is convenient to be able to remove all edges going to arbitrary sets candidates as if they had been eliminated or elected and observe the relations among the candidates that remain. Procedure 11 removes all edges going to eliminated or elected candidates from the network without changing the result of the election. We can apply this procedure to a set of arbitrary candidates, $\boldsymbol{C_e}$ and just observe the results. This makes sense because in the modified structure no votes could go from a candidate C_i to a candidate C_j passing through any candidate in $\boldsymbol{C_e}$**.** This way, relations among candidates that may be disguised by the interposition of elements of $\boldsymbol{C_e}$ will have to show up**.**

We will show in Theorem 8, that the resulting network is always the same regardless of whether any candidate $C_k | C_k \in \boldsymbol{C_e}$ is treated as eliminated or elected.

Theorem 8: removal of all incoming edges of arbitrary candidates

If Procedure 11 is called with any set of candidates, $\boldsymbol{C_e}$, arbitrarily chosen as a parameter, the result will be a network structure that is exactly the one we would get if the candidates in $\boldsymbol{C_e}$ had been elected or eliminated during the execution of Procedure 2, regardless of the order of the eliminations and elections and regardless of which candidates were elected and which were eliminated.

Proof:

It is easy to see that Procedure 11 is not even aware of which candidates were eliminated and which were elected, so its behavior cannot be affected by that.

By Lemma 11, the use of Procedure 11 when any subset of $\boldsymbol{E} \cup \boldsymbol{L}$ is passed as a parameter never changes the results of Procedure 8.

Let suppose that Procedure 11 could result in two different structures, $\boldsymbol{S}^A$ and $\boldsymbol{S}^B$. In this case, the use of both structures would have to always lead Procedure 8 to the same results. Procedure 11 applies several modifications the network without ever changing the results of Procedure 8, so, indeed, there can be many different structures that can lead Procedure 8 always to the same results. We will prove that two different structures that could be the final result of Procedure 11 cannot.

If $\boldsymbol{S}^A \neq \boldsymbol{S}^B$ There would be two candidates C_i and C_j such that $P_{ij}^A \neq P_{ij}^B$, where P_{ij}^A is the percentage of transfer from C_i to C_j in $\boldsymbol{S}^A$ and P_{ij}^B is the percentage of transfer from C_i to C_j in $\boldsymbol{S}^B$.

Since in the end of the execution of Procedure 11 there are only edges going to remaining candidates, then C_j would have to be a remaining candidate.

Suppose that latter, a transfer of votes with $\boldsymbol{C_i}$ as the origin takes place and that no other transfers take place in the same round, what can happen, for example when $\boldsymbol{C_i}$ is eliminated. The amount of votes transferred to $\boldsymbol{C_j}$ in the end of Procedure 8 would be different in $\boldsymbol{S^A}$ and $\boldsymbol{S^B}$, contradicting Lemma 12.

There is still the possibility that the structures $\boldsymbol{S}^A$ and $\boldsymbol{S}^B$ differ only in respect to outgoing edges of candidates that have already been eliminated, since that would not affect the results of Procedure 8. However, it is direct from the implementations of Procedure 9, Procedure 10 and Procedure 11 that they treat outgoing edges of any candidates in the same way. So, if $\boldsymbol{S}^A$ and $\boldsymbol{S}^B$ cannot differ in respect to the outgoing edges of remaining candidates, they cannot differ in respect to outgoing edges of any candidates.

This proves the present theorem. ■

Insensitivity to the order of eliminations and elections

In Theorem 9 we show that the number of votes in any remaining candidate and the number of discarded votes do not depend on the order of elections and eliminations that have already taken place in an election process. Before that, we need some lemmas and a corollary.

Lema 18: insensitivity to the postponing of the elimination of a single candidate

If we sequentially execute an operation of vote transferring through the network, claim one arbitrary candidate C_j to be eliminated and execute another operation of vote transferring, than $\forall C_r \in \boldsymbol{R} \cup \{C_0\}$ the final number of votes in candidate C_r, $\dot{V}_r$, is identical to $\acute{V}_r$, where $\acute{V}_r$ is the final number of votes in candidate C_r when first claim C_j to be eliminated and them execute a single operation of vote transferring.

Proof:

By Lemma 13, we can replace the original political network by a network where no edges go to a non remaining candidate, except for the virtual discard candidate. We assume that throughout the present proof.

The first proposed sequence of operations involve transferring votes, eliminating a candidate and transferring votes again. We will need to calculate some values immediately after the first vote transferring, so that we can use them latter to calculate the final quota, $\dot{Q}$ and $\dot{V}_r$, $\forall C_r \in \boldsymbol{R} \cup \{C_0\}$. We will refer to such values by $\ddot{Q}$, $\ddot{V}$, $\ddot{V}_r$ with meanings that can easily be deduced by the contexts.

Using Lemma 14 after the first vote transferring operation we have

$$\ddot{Q} = \frac{\hat{V} - \sum_{\forall C_i \in \boldsymbol{L} \cup \boldsymbol{E}} \hat{V}_i \cdot P_{i0}}{\left(M - \sum_{\forall C_i \in \boldsymbol{L}} P_{i0}\right)}$$

Also by Lemma 14, the current number of valid votes at this point is given by

$$\ddot{V} = \hat{V} - \sum_{\forall C_i \in \boldsymbol{L}} (\hat{V}_i - \ddot{Q}) \cdot P_{i0} - \sum_{\forall C_i \in \boldsymbol{E}} \hat{V}_i \cdot P_{i0}$$

At the same time, by Lemma 2, after the first vote transfer process, all elected candidates have exactly $\ddot{\boldsymbol{Q}}$ votes and all eliminated candidates have exactly 0 votes.

Thus, $\hat{V}_i - \ddot{Q}$ is the number of votes in the vote package put in $\boldsymbol{H}$ by Procedure 7 for any candidate $C_i \in \boldsymbol{L}$**.** In the case of eliminated candidates Procedure 7 creates a package with all their votes. Thus, the number of votes in C_j after the first vote transferring is given by

$$\ddot{V}_j = \hat{V}_j + \sum_{\forall C_i \in \boldsymbol{L}} (\hat{V}_i - \ddot{Q}) \cdot P_{ij} + \sum_{\forall C_i \in \boldsymbol{E}} \hat{V}_i \cdot P_{ij}$$

Since no edges go to elected or eliminate candidates, for any candidate $C_k | C_k \in \boldsymbol{L} \cup \boldsymbol{E}$, $P_{jk} = 0$.

After we eliminate C_j, the network is adjusted by Procedure 11. For any candidate C_k we have that

$$\ddot{P}_{k0} = \frac{P_{k0} + P_{kj} \cdot P_{j0}}{1 - P_{kj} \cdot P_{jk}}$$

For the candidates C_k that are in $\boldsymbol{L} \cup \boldsymbol{E}$, since $P_{jk} = 0$,

$$\ddot{P}_{k0} = P_{k0} + P_{kj} \cdot P_{j0}$$

Lets now calculate $\dot{Q}$.

To obtain the current quota after the second transfer, we apply the formula for the current quota again but start from the amounts of votes in each candidate after the first transfers and remember that C_j has been eliminated. We obtain

$$\dot{Q} = \frac{\ddot{V} - \ddot{V}_j \cdot P_{j0} - \sum_{\forall C_i \in \boldsymbol{L}} \ddot{Q} \cdot \ddot{P}_{i0}}{\left(M - \sum_{\forall C_i \in \boldsymbol{L}} P_{i0}\right)}$$

$$\dot{Q} = \frac{\left(\hat{V} - \sum_{\forall C_i \in \boldsymbol{L}} (\hat{V}_i - \ddot{Q}) \cdot P_{i0} - \sum_{\forall C_i \in \boldsymbol{E}} \hat{V}_i \cdot P_{i0}\right) - \ddot{V}_j \cdot P_{j0} - \sum_{\forall C_i \in \boldsymbol{L}} \ddot{Q} \cdot \ddot{P}_{i0}}{\left(M - \sum_{\forall C_i \in \boldsymbol{L}} P_{i0}\right)}$$

$$\dot{Q} \cdot \left(M - \sum_{\forall C_i \in \boldsymbol{L}} P_{i0}\right) =$$

$$\left(\hat{V} - \sum_{\forall C_i \in \boldsymbol{L}} (\hat{V}_i - \ddot{Q}) \cdot P_{i0} - \sum_{\forall C_i \in \boldsymbol{E}} \hat{V}_i \cdot P_{i0}\right)$$

$$-\left(\hat{V}_j + \sum_{\forall C_i \in \boldsymbol{L}} (\hat{V}_i - \ddot{Q}) \cdot P_{ij} + \sum_{\forall C_i \in \boldsymbol{E}} \hat{V}_i \cdot P_{ij}\right) \cdot P_{j0}$$

$$- \sum_{\forall C_i \in \boldsymbol{L}} \ddot{Q} \cdot (P_{i0} + P_{ij} \cdot P_{j0})$$

$$\dot{Q} = \frac{\hat{V} - \hat{V}_j \cdot P_{j0} - \sum_{\forall C_i \in \boldsymbol{L} \cup \boldsymbol{E}} \hat{V}_i \cdot (P_{i0} + P_{ij} \cdot P_{j0})}{\left(M - \sum_{\forall C_i \in \boldsymbol{L}} P_{i0}\right)}$$

We have calculated the current quota after the second vote transfer for the case where we transfer votes, eliminate a candidate and transfer votes again.

Now, let's calculate $\acute{Q}$ the current quota after the vote transfer for the case where we first eliminated the candidate and them perform a single vote transfer.

We start by adjusting the network to reflect the elimination of $\boldsymbol{C_j}$, what, as we have seen, for candidates $\boldsymbol{C_k}$ that are in $\boldsymbol{L} \cup \boldsymbol{E}$, gives us $\ddot{\boldsymbol{P}}_{\boldsymbol{k0}} = \boldsymbol{P_{k0}} + \boldsymbol{P_{kj}} \cdot \boldsymbol{P_{j0}}$. Then we apply the formula of what yields to

$$\acute{Q} = \frac{\hat{V} - \sum_{\forall C_i \in \boldsymbol{L} \cup \boldsymbol{E} \cup \{\boldsymbol{C_j}\}} \hat{V}_i \cdot \left(P_{i0} + P_{ij} \cdot P_{j0}\right)}{\left(M - \sum_{\forall C_i \in \boldsymbol{L}} P_{i0}\right)}$$

Since $P_{jj} = 0$,

$$\acute{Q} = \frac{\hat{V} - \hat{V}_j \cdot P_{j0} - \sum_{\forall C_i \in \boldsymbol{L} \cup \boldsymbol{E}} \hat{V}_i \cdot (P_{i0} + P_{ij} \cdot P_{j0})}{\left(M - \sum_{\forall C_i \in \boldsymbol{L}} P_{i0}\right)}$$

Thus $\acute{Q} = \dot{Q}$.

We are ready to calculate $\dot{V}_r$ and $\acute{V}_r$, $\forall C_r \in \boldsymbol{R} \cup \{C_0\}$. Let's start by $\dot{V}_r$.

After the first transferring of votes we have that

$$\ddot{V}_r = \hat{V}_r + \sum_{\forall C_i \in \boldsymbol{L}} (\hat{V}_i - \ddot{Q}) \cdot P_{ir} + \sum_{\forall C_i \in \boldsymbol{E}} \hat{V}_i \cdot P_{ir}$$

After the elimination of C_j, we have that

$$\dot{V}_r = \ddot{V}_r + \ddot{V}_j \cdot P_{jr} + \sum_{\forall C_i \in \boldsymbol{L}} (\ddot{Q} - \dot{Q}) \cdot \ddot{P}_{ir}$$

$$\dot{V}_r = \hat{V}_r + \sum_{\forall C_i \in \boldsymbol{L}} (\hat{V}_i - \ddot{Q}) \cdot P_{ir} + \sum_{\forall C_i \in \boldsymbol{E}} \hat{V}_i \cdot P_{ir} +$$

$$\left(\hat{V}_j + \sum_{\forall C_i \in \boldsymbol{L}} (\hat{V}_i - \ddot{Q}) \cdot P_{ij} + \sum_{\forall C_i \in \boldsymbol{E}} \hat{V}_i \cdot P_{ij}\right) \cdot P_{jr} +$$

$$\sum_{\forall C_i \in \boldsymbol{L}} (\ddot{Q} - \dot{Q}) \cdot (P_{ir} + P_{ij} \cdot P_{jr})$$

$$\dot{V}_r = \hat{V}_r + \hat{V}_j \cdot P_{jr} + \sum_{\forall C_i \in \boldsymbol{L}} (\hat{V}_i - \dot{Q}) \cdot (P_{ir} + P_{ij} \cdot P_{jr}) + \sum_{\forall C_i \in \boldsymbol{E}} \hat{V}_i \cdot (P_{ir} + P_{ij} \cdot P_{jr})$$

Now let's go to $\acute{V}_r$. It is given by

$$\acute{V}_r = \hat{V}_r + \hat{V}_j \cdot P_{jr} + \sum_{\forall C_i \in \boldsymbol{L}} (\hat{V}_i - \dot{Q}) \cdot \ddot{P}_{ir} + \sum_{\forall C_i \in \boldsymbol{E}} \hat{V}_i \cdot \ddot{P}_{ir}$$

$$\acute{V}_r = \hat{V}_r + \hat{V}_j \cdot P_{jr} + \sum_{\forall C_i \in \boldsymbol{L}} (\hat{V}_i - \dot{Q}) \cdot (P_{ir} + P_{ij} \cdot P_{jr}) + \sum_{\forall C_i \in \boldsymbol{E}} \hat{V}_i \cdot (P_{ir} + P_{ij} \cdot P_{jr})$$

Thus $\dot{V}_r = \acute{V}_r$, what proves the present lemma. ■

Lema 19: insensitivity to postponing of the election of a single candidate

If we sequentially execute an operation of vote transferring through the network, claim one arbitrary candidate C_j to be elected and execute another operation of vote transferring, than $\forall C_r \in \boldsymbol{R} \cup \{C_0\}$ the final number of votes in candidate C_r, $\dot{V}_r$, is identical to $\acute{V}_r$, where $\acute{V}_r$ is the final number of votes in candidate C_r when we first claim C_j to be elected and them execute a single operation of vote transferring.

Proof:

By Lemma 13, we can replace the original political network by a network where no edges go to a non remaining candidate, except for the virtual discard candidate. We assume that throughout the present proof.

The first proposed sequence of operations involve transferring votes, claiming a candidate to be elected and transferring votes again. We will need to calculate some values immediately after the first vote transferring, so that we can use them latter to calculate $\dot{Q}$ and $\dot{V}_r$, $\forall C_r \in \boldsymbol{R} \cup \{C_0\}$. We will refer to such values by $\ddot{Q}$, $\ddot{V}$, $\ddot{V}_r$ with meanings that can easily be deduced by the contexts.

Using Lemma 14, after the first vote transferring operation we have

$$\ddot{Q} = \frac{\hat{V} - \sum_{\forall C_i \in \boldsymbol{L} \cup \boldsymbol{E}} \hat{V}_i \cdot P_{i0}}{\left(M - \sum_{\forall C_i \in \boldsymbol{L}} P_{i0}\right)}$$

Also by Lemma 14, the current number of valid votes at this point is given by

$$\ddot{V} = \hat{V} - \sum_{\forall C_i \in \boldsymbol{L}} (\hat{V}_i - \ddot{Q}) \cdot P_{i0} - \sum_{\forall C_i \in \boldsymbol{E}} \hat{V}_i \cdot P_{i0}$$

At the same time, by Lemma 2, after the first vote transfer process, all elected candidates have exactly $\ddot{\boldsymbol{Q}}$ votes and all eliminated candidates have exactly 0 votes.

Thus, $\hat{V}_i - \ddot{Q}$ is the number of votes in the vote package put in $\boldsymbol{H}$ by Procedure 7 for any candidate $C_i \in \boldsymbol{L}$. In the case of eliminated candidates Procedure 7 creates a package with all their votes. Thus, the number of votes in C_j after the first vote transferring is given by

$$\ddot{V}_j = \hat{V}_j + \sum_{\forall C_i \in \boldsymbol{L}} (\hat{V}_i - \ddot{Q}) \cdot P_{ij} + \sum_{\forall C_i \in \boldsymbol{E}} \hat{V}_i \cdot P_{ij}$$

Since no edges go to elected or eliminate candidates, for any candidate $C_k | C_k \in \boldsymbol{L} \cup \boldsymbol{E}$, $P_{jk} = 0$.

After we claim that C_j has been elected, the network is adjusted by Procedure 11. For any candidate C_k we have that

$$\ddot{P}_{k0} = \frac{P_{k0} + P_{kj} \cdot P_{j0}}{1 - P_{kj} \cdot P_{jk}}$$

For the candidates C_k that are in $\boldsymbol{L} \cup \boldsymbol{E}$, since $P_{jk} = 0$,

$$\ddot{P}_{k0} = P_{k0} + P_{kj} \cdot P_{j0}$$

Lets now calculate $\dot{Q}$.

To obtain the current quota after the second transfer, we apply the formula for the current quota again but start from the amounts of votes in each candidate after the first transfers and remember that C_j has been elected. We obtain

$$\dot{Q} = \frac{\ddot{V} - \ddot{V}_j \cdot P_{j0} - \sum_{\forall C_i \in \boldsymbol{L}} \ddot{Q} \cdot \ddot{P}_{i0}}{\left(M - P_{j0} - \sum_{\forall C_i \in \boldsymbol{L}} P_{i0}\right)}$$

$$\dot{Q} = \frac{\left(\hat{V} - \sum_{\forall C_i \in \boldsymbol{L}} (\hat{V}_i - \ddot{Q}) \cdot P_{i0} - \sum_{\forall C_i \in \boldsymbol{E}} \hat{V}_i \cdot P_{i0}\right) - \ddot{V}_j \cdot P_{j0} - \sum_{\forall C_i \in \boldsymbol{L}} \ddot{Q} \cdot \ddot{P}_{i0}}{\left(M - P_{j0} - \sum_{\forall C_i \in \boldsymbol{L}} P_{i0}\right)}$$

$$\dot{Q} \cdot \left(M - P_{j0} - \sum_{\forall C_i \in \boldsymbol{L}} P_{i0}\right) =$$

$$\left(\hat{V} - \sum_{\forall C_i \in \boldsymbol{L}} (\hat{V}_i - \ddot{Q}) \cdot P_{i0} - \sum_{\forall C_i \in \boldsymbol{E}} \hat{V}_i \cdot P_{i0}\right)$$

$$-\left(\hat{V}_j + \sum_{\forall C_i \in \boldsymbol{L}} (\hat{V}_i - \ddot{Q}) \cdot P_{ij} + \sum_{\forall C_i \in \boldsymbol{E}} \hat{V}_i \cdot P_{ij}\right) \cdot P_{j0}$$

$$- \sum_{\forall C_i \in \boldsymbol{L}} \ddot{Q} \cdot (P_{i0} + P_{ij} \cdot P_{j0})$$

$$\dot{Q} = \frac{\hat{V} - \hat{V}_j \cdot P_{j0} - \sum_{\forall C_i \in \boldsymbol{L} \cup \boldsymbol{E}} \hat{V}_i \cdot (P_{i0} + P_{ij} \cdot P_{j0})}{\left(M - P_{j0} - \sum_{\forall C_i \in \boldsymbol{L}} P_{i0}\right)}$$

We have calculated the current quota after the second vote transfer for the case where we transfer votes, claim that a candidate is elected and transfer votes again.

Now, let's calculate $\acute{Q}$ the current quota after the vote transfer for the case where we first claim the candidate to be elected and them perform a single vote transfer.

We start by adjusting the network to reflect the election of $\boldsymbol{C_j}$, what, as we have seen, for candidates $\boldsymbol{C_k}$ that are in $\boldsymbol{L} \cup \boldsymbol{E}$, gives us $\ddot{\boldsymbol{P}}_{\boldsymbol{k0}} = \boldsymbol{P_{k0}} + \boldsymbol{P_{kj}} \cdot \boldsymbol{P_{j0}}$. Then we apply the formula of , what yields to

$$\acute{Q} = \frac{\hat{V} - \sum_{\forall C_i \in \boldsymbol{L} \cup \boldsymbol{E} \cup \{\boldsymbol{C_j}\}} \hat{V}_i \cdot \left(P_{i0} + P_{ij} \cdot P_{j0}\right)}{\left(M - P_{j0} - \sum_{\forall C_i \in \boldsymbol{L}} P_{i0}\right)}$$

Since $P_{jj} = 0$,

$$\acute{Q} = \frac{\hat{V} - \hat{V}_j \cdot P_{j0} - \sum_{\forall C_i \in \boldsymbol{L} \cup \boldsymbol{E}} \hat{V}_i \cdot (P_{i0} + P_{ij} \cdot P_{j0})}{\left(M - P_{j0} - \sum_{\forall C_i \in \boldsymbol{L}} P_{i0}\right)}$$

Thus $\acute{Q} = \dot{Q}$.

We are ready to calculate $\dot{V}_r$ and $\acute{V}_r$ $\forall C_r \in \boldsymbol{R} \cup \{C_0\}$**.** Let's start by $\dot{V}_r$.

After the first transferring of votes we have that

$$\ddot{V}_r = \hat{V}_r + \sum_{\forall C_i \in \boldsymbol{L}} (\hat{V}_i - \ddot{Q}) \cdot P_{ir} + \sum_{\forall C_i \in \boldsymbol{E}} \hat{V}_i \cdot P_{ir}$$

After the election of C_j, we have that

$$\dot{V}_r = \ddot{V}_r + (\ddot{V}_j - \dot{Q}) \cdot P_{jr} + \sum_{\forall C_i \in \boldsymbol{L}} (\ddot{Q} - \dot{Q}) \cdot \ddot{P}_{ir}$$

$$\dot{V}_r = \hat{V}_r + \sum_{\forall C_i \in \boldsymbol{L}} (\hat{V}_i - \ddot{Q}) \cdot P_{ir} + \sum_{\forall C_i \in \boldsymbol{E}} \hat{V}_i \cdot P_{ir} +$$

$$\left(\hat{V}_j - \dot{Q} + \sum_{\forall C_i \in \boldsymbol{L}} (\hat{V}_i - \ddot{Q}) \cdot P_{ij} + \sum_{\forall C_i \in \boldsymbol{E}} \hat{V}_i \cdot P_{ij}\right) \cdot P_{jr} +$$

$$\sum_{\forall C_i \in \boldsymbol{L}} (\ddot{Q} - \dot{Q}) \cdot (P_{ir} + P_{ij} \cdot P_{jr})$$

$$\dot{V}_r = \hat{V}_r + (\hat{V}_j - \dot{Q}) \cdot P_{jr} + \sum_{\forall C_i \in \boldsymbol{L}} (\hat{V}_i - \dot{Q}) \cdot (P_{ir} + P_{ij} \cdot P_{jr}) + \sum_{\forall C_i \in \boldsymbol{E}} \hat{V}_i \cdot (P_{ir} + P_{ij} \cdot P_{jr})$$

Now let's go to $\acute{V}_r$. It is given by

$$\acute{V}_r = \hat{V}_r + (\hat{V}_j - \dot{Q}) \cdot P_{jr} + \sum_{\forall C_i \in \boldsymbol{L}} (\hat{V}_i - \dot{Q}) \cdot \ddot{P}_{ir} + \sum_{\forall C_i \in \boldsymbol{E}} \hat{V}_i \cdot \ddot{P}_{ir}$$

$$\acute{V}_r = \hat{V}_r + (\hat{V}_j - \dot{Q}) \cdot P_{jr} + \sum_{\forall C_i \in \boldsymbol{L}} (\hat{V}_i - \dot{Q}) \cdot (P_{ir} + P_{ij} \cdot P_{jr}) + \sum_{\forall C_i \in \boldsymbol{E}} \hat{V}_i \cdot (P_{ir} + P_{ij} \cdot P_{jr})$$

Thus $\dot{V}_r = \acute{V}_r$, what proves the present lemma. ■

Corollary 12: insensitivity to postponing the elimination or the election of an element of a set of candidates to be elected or eliminated

If we simultaneously claim all candidates in a set $\dot{\boldsymbol{E}}$ to be eliminated and all candidates in a set $\dot{\boldsymbol{L}}$ to be elected and perform a vote transfer process, $\forall C_r \in \boldsymbol{R} \cup \{C_0\}$ the final number of votes in candidate C_r, $\dot{V}_r$, is identical to $\acute{V}_r$, where $\acute{V}_r$ is the final number of votes in candidate C_r when we simultaneously claim all candidates in a set $\dot{\boldsymbol{E}} - \{C_j\}, C_j \in \dot{\boldsymbol{E}}$ to be eliminated and all candidates in a set $\dot{\boldsymbol{L}}$ to be elected, perform a vote transfers process and latter we claim C_j to be eliminated and perform another vote transfer and is also identical to $\grave{V}_r$, where $\grave{V}_r$ is the final number of votes in candidate C_r when we simultaneously claim all candidates in a set $\dot{\boldsymbol{E}}$ to be eliminated and all candidates in a set $\dot{\boldsymbol{L}} - \{C_j\}, C_j \in \dot{\boldsymbol{L}}$ to be elected and perform a vote transfer process and latter we claim C_j to be elected and perform another vote transfer process .

Proof:

Let's consider the modified structure specified in Lemma 13.

If we postpone the treatment of C_j, the structure of the network will, first, be changed considering only the other eliminations and elections. If we treat C_j together with all others, we can still let the modifications related to it to be the last, since Procedure 11 does not require any particular order. Thus we can consider the structure that was modified to reflect the other eliminations and elections as a starting point to apply the modifications related to C_j for both cases.

In one case, we should transfer the votes of the other candidates while $\boldsymbol{C_j}$ is treated as a remaining candidate, modify the structure to reflect the elimination or election of $\boldsymbol{C_j}$ and apply another vote transfer process to the network. In the other case, we should modify the structure to reflect the elimination or election of $\boldsymbol{C_j}$ and apply a single vote transfer process to the network. By Lema 18 and Lema 18 the two ways of proceeding yield the same results, proving the present corollary. ■

Theorem 9: general insensitivity to the order election and elimination of candidates

If $(\dot{\boldsymbol{E}}, \dot{\boldsymbol{L}})$ is a pair where $\dot{\boldsymbol{E}}$ is a set of candidates to be eliminated and $\dot{\boldsymbol{L}}$ a set of candidates to be elected, we can execute the elections and eliminations in any order including arbitrary simultaneous operations of elimination and election, execute vote transfers process between any two consecutive operations and $\forall C_r \in \boldsymbol{R} \cup \{C_0\}$ the final number of votes in candidate C_r, $\dot{V}_r$, will always be the same.

Proof:

Let's start form an arbitrary sequence of pairs of length k, $\left((\boldsymbol{E_0}, \boldsymbol{L_0}), \dots, (\boldsymbol{E_i}, \boldsymbol{L_i}), \dots, (\boldsymbol{E_{k-1}}, \boldsymbol{L_{k-1}})\right)$, where $\forall i, 0 \le i < k, (\boldsymbol{E_i}, \boldsymbol{L_i})$ is such that all candidates in $\boldsymbol{E_i}$ are to be eliminated, all candidates in $\boldsymbol{L_i}$ are to be elected and all elections and eliminations in $(\boldsymbol{E_i}, \boldsymbol{L_i})$ are to be performed simultaneously and where the elections and eliminations specified in $(\boldsymbol{E_i}, \boldsymbol{L_i})$ are to be performed before the elections and eliminations specified in $(\boldsymbol{E_{i+1}}, \boldsymbol{L_{i+1}})$ and vote transfer process are to be executed between the handling of any two consecutive pairs. We will show that any other sequence of pairs would lead to the same results.

First, we can apply Corollary 12, to separate the candidates that were part of a pair $(\boldsymbol{E_i}, \boldsymbol{L_i})$. We can do that choosing any individual candidate, C_j, of $\boldsymbol{E_i}$ or $\boldsymbol{L_i}$ and postponing his or her election or elimination to immediately after $(\boldsymbol{E_i}, \boldsymbol{L_i})$ without changing any results. This way the sequence would become

$$\left((\boldsymbol{E_0}, \boldsymbol{L_0}), \dots, (\boldsymbol{E_i} - \{C_j\}, \boldsymbol{L_i}), (\{C_j\}, \emptyset) \dots, (\boldsymbol{E_{k-1}}, \boldsymbol{L_{k-1}})\right)$$

or

$$\left((\boldsymbol{E_0}, \boldsymbol{L_0}), \dots, (\boldsymbol{E_i}, \boldsymbol{L_i} - \{C_j\}), (\emptyset, \{C_j\}) \dots, (\boldsymbol{E_{k-1}}, \boldsymbol{L_{k-1}})\right).$$

We can keep doing that till we have a sequence of pairs where one set is always empty and the other unitary. That is equivalent to a sequence of individual elections and eliminations.

Then, because the second pair in the list contains only one candidate, we can use Corollary 12 to join it with the first pair, making their operations simultaneous without changing the results.

Latter we can add the next pair, which still contains only one candidate, to the first pair. By Corollary 12, that won't change the results either. We can keep doing this till we have only one pair, specifying that all eliminations and elections are simultaneous.

This set is identical for any initial sequence, what proves the present theorem. ■

References

Diestel, Reinhard. *Graph theory.* Vol. 173. Springer, 2000.

Papadimitriou, Christos H. *Computational complexity.* John Wiley and Sons Ltd., 2003.

Tarjan, Robert. "Depth-first search and linear graph algorithms." *SIAM journal on computing* 1, no. 2 (1972): 146-160.