On Optimization of Complete Social Networks

Araniyos Terefe Weldegebriel[1] Burak Yıldıran Stodolsky[2]

August 6, 2018

Abstract

A balanced social network is a social network where, for any member of the social network, the following two statements are true; a friend of my friend is my friend and an enemy of my enemy is my friend. In this paper we demonstrate a polynomial time greedy algorithm that balances any complete social network with n members by changing at most $\lceil \frac{n^2}{4} - \frac{n}{2} \rceil$ of the initial relationships between the members of the network. We also demonstrate that the problem of determining the minimum number of relationships that needs to change so that a complete social network, where each member has at least as many friends as enemies, becomes balanced is still NP-Complete.

1 Introduction

A *signed graph* H, written $H = (V, E, \sigma)$ consists of an underlying graph (V, E) together with a sign function σ which assigns to each edge the values $+1$ or -1. A given social network can be modeled using a signed graph, where we represent each person using a different vertex. If the relationship between two people is positive or negative, the edge between the two vertices representing them has weight $+1$ or -1, respectively. For undefined graph theoretical terminology appearing below, please consult any introductory text on graph theory.

Heider [6] introduced the notion of social networks. He defined a *balanced social network* as a social network where for each member of the network, a friend of a friend is a friend, an enemy of a friend is an enemy, a friend of an enemy is an enemy, and finally an enemy of an enemy is a friend. Social networks which are not balanced are called *unbalanced networks.* He pointed out that an unbalanced social network has a tendency to transform to a balanced network because of the stress felt by its members when they cannot consistently decide who their friends and enemies are. Using graph theory, Cartwright and Harary [3] summarized Heider's observations as follows: A signed graph is *balanced* if the product of the edge signs of any cycle is $+1$. See Fig. 1 for examples of possible social networks consisting of three members. In all the figures in this paper, solid edges represent positive edges, while dashed edges represent negative edges.

In [3], Cartwright and Harary also determined the structure of all balanced networks.

[1]Department of Mathematical Engineering, Istanbul Technical University, Maslak, Istanbul, Turkey. Email: weldegebriel@itu.edu.tr

[2]Department of Mathematical Engineering, Istanbul Technical University, Maslak, Istanbul, Turkey. Email: stodolsky@itu.edu.tr.

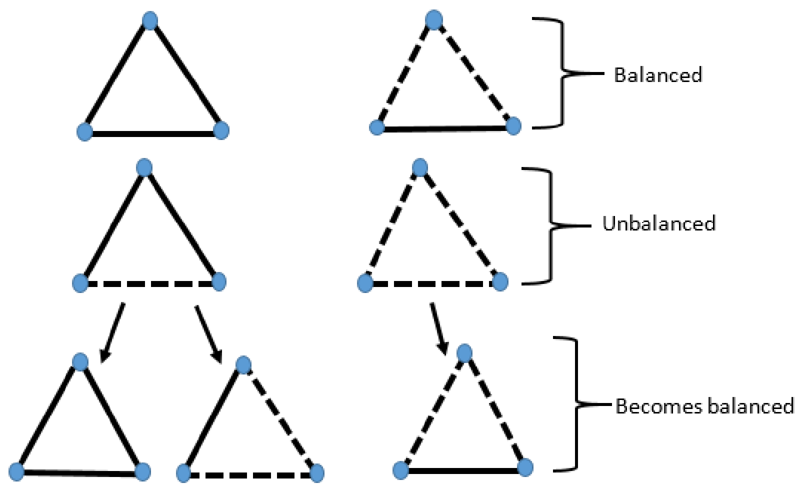

Figure 1: *Possible social networks consisting of three members.*

Theorem 1 (Structure Theorem)**.** *A signed graph is balanced if and only if its vertex set can be partitioned into two disjoint sets (one of which may possibly be empty) such that, edges inside each of the subsets (parts) are all positive and edges between vertices in different subsets are all negative.*

The *line index* of a signed graph G, $l(G)$, is the minimum number of edges of G whose signs have to be negated so that the resulting signed graph is balanced [5]. Therefore, $l(G)$ is a very reasonable parameter to consider if we wish to predict the actual evolution of social networks. Unfortunately, as shown in [2] and [7], calculating the line index of a signed graph is NP-Complete. As a consequence, new algorithms have been developed which only try to balance signed graphs without much consideration of their predictive power.

Antal et al. [1] have proposed two algorithms that for complete social networks negate some edges on negative triangles to decrease the number of negative triangles. It was shown in [1, 9] that both of the algorithms may not balance some networks because the algorithms may get stuck in "jammed states", which are local but not global minima of the energy functions used to evaluate the amount of imbalance in a network. In [8], Marvel et al. developed an algorithm based on a continuous model, that balances most complete signed graphs. However, their algorithm yields only two kinds of balanced states depending on the density of positive edges in the signed network. If the initial signed graph has more positive edges than negative ones, the algorithm yields a signed graph in which all edges are positive; if not, then the algorithm yields a balanced signed graph in which the two parts have equal size. The approach in [8] has been the inspiration of many new research articles that try to balance social networks using continuous models.

In this note we exhibit a polynomial time greedy algorithm which balances any complete signed graph with n vertices by changing the signs of at most $\lceil \frac{n^2}{4} - \frac{n}{2} \rceil$ edges. Our algorithm is very similar to the greedy Max-Cut algorithm and has the advantage that it forces less number of sign changes of the edges of a signed graph than the algorithm in [8]. Therefore, the greedy algorithm might be more predictive of the transformations of real life social networks. We then demonstrate that the problem of determining the line indices of signed graphs where each vertex is incident to at least as many positive edges as negative edges, is still NP-Complete.

2 Methodology

Let $G = (V, E, \sigma)$ be a signed graph where $\sigma : E \to \{\pm 1\}$. For a bipartition $(S, \overline{S})$ of $V(G)$, let $P(v)$ be the set containing v. In other words, $P(v) = S$ if $v \in S$, and $P(v) = \overline{S}$ if $v \in \overline{S}$. We define the *stability degree of* v *for a partition* $(S, \overline{S})$ as

$$\partial_S(v) = \sum_{x \in P(v)} \sigma(vx) - \sum_{y \notin P(v)} \sigma(vy),$$

and the *stability degree of a partition* $(S, \overline{S})$ as

$$\mathcal{D}(S, \overline{S}) = \sum_{v \in V} \partial_S(v).$$

If $S = V(G)$, instead of $\partial_S(v)$ and $\mathcal{D}(S, \overline{S})$ we will simply write $\partial(v)$ and $\mathcal{D}(G)$ respectively. For the partition $(S, \overline{S})$, if the edge xv contributes positively/negatively to $\partial_S(v)$, x is called a *good/bad neighbor* of v and the edge xv is called a *good/bad edge*. Given a partition $(S, \overline{S})$ of $V(G)$, the set of good edges is composed of the positive edges that lie inside each part and the negative edges that lie between S and $\overline{S}$. Likewise, the set of bad edges is composed of the negative edges that lie inside each part and the positive edges that lie between S and $\overline{S}$. We make the following useful claim.

Claim 1. *H is a balanced signed graph if and only if $V(H)$ can be partitioned into sets S and $\overline{S}$ such that $\partial_S(v) = d(v)$ for all $v \in V(H)$ and $\mathcal{D}(S, \overline{S}) = 2|E(H)|$.*

Proof. By Theorem 1, a signed graph H is balanced if and only if $V(H)$ can be partitioned into sets S and $\overline{S}$ such that every edge is a good edge; in which case $\partial_S(v) = d(v)$ for all $v \in V(H)$. □

We now describe how to balance a signed graph $G = (V, E, \sigma)$ given a partition $(S, \overline{S})$ of V. The balanced signed graph G_S has V as its vertex set, E as its edge set, and the signs of the edges are obtained from G by negating the sign of each bad edge of the partition $(S, \overline{S})$ and leaving the remaining signs unchanged. In [10], the number of bad edges of $(S, \overline{S})$ is defined as $l(S, \overline{S})$. The following lemma describes the relationship between $\mathcal{D}(S, \overline{S})$ and $l(S, \overline{S})$.

Lemma 1. *Let G be a signed graph and $(S, \overline{S})$ be a partition of $V(G)$. Then,*

$$l(S, \overline{S}) = \frac{2|E(G)| - \mathcal{D}(S, \overline{S})}{4}.$$

Proof. For each vertex $v \in V(G)$, $\partial_S(v)$ is the difference between the number of good edges and bad edges incident to v. Consequently, $d(v) - \partial_S(v)$ counts each bad edge incident to v twice. Summing over all $v \in V(G)$ we get $2|E(G)| - \mathcal{D}(S, \overline{S}) = 4 \cdot l(S, \overline{S})$ because each bad edge gets counted twice more, once for each endpoint. □

Lemma 1 shows that as parameters $\mathcal{D}(S, \overline{S})$ and $l(S, \overline{S})$ are equivalent. Also as stated in [10], it is clear that $l(G) = \min_{(S, \overline{S})} l(S, \overline{S})$, which immediately yields that $l(G)$ is attained only by a partition which maximizes $\mathcal{D}(S, \overline{S})$. For a given signed graph G, a partition $(S, \overline{S})$ is said to be an *optimal partition* if $l(S, \overline{S}) = l(G)$.

The advantage of using $\partial_S(v)$ is that it is a local parameter which measures the happiness of each individual with a given partition of the social network. If we see social networks as environments where individuals choose their alliances so as to maximize their own happiness, then the natural conclusion is that each person v makes choices only based on their $\partial_S(v)$ because he/she is primarily concerned with his/her relationships with others and not with the relationships between other individuals. This idea forms the basis of our algorithm. A signed graph G can be *split* if there exists a proper subset S of the vertex set such that $\mathcal{D}(G) < \mathcal{D}(S, \overline{S})$. In the next section we discuss this algorithm and show that even if each individual tries to maximize his/her happiness, then the network may not reach an optimal balanced state.

3 Result and Discussion

The algorithm is inspired from the greedy Max-Cut algorithm. We move a vertex v from one partite set to the other, if $\partial_S(v) < 0$; in other words, the number of good edges incident to v is less than the number of bad edges incident to v. Once no such vertex is left, we output a balanced signed graph after minor modifications.

Algorithm.

Input: A signed graph G.

Output: A balanced signed graph G_{S_k}, obtained from a partition $(S_k, \overline{S_k})$ of $V(G)$ by negating the signs of every bad edge of $(S_k, \overline{S_k})$.

Initialization: Set $S_0 = V(G)$. (The algorithm could also take any bipartition of the vertex set as the initial condition.)

Iteration: At step i identify vertex v such that $\partial_{S_{i-1}}(v) = \min\limits_{w \in V(G)} \partial_{S_{i-1}}(w)$ and $\partial_{S_{i-1}}(v) \leq 0$. If no such vertex exists, construct $G_{S_{i-1}}$ and terminate. If such a vertex v exists and $\partial_{S_{i-1}}(v) < 0$, then construct S_i by removing v from the partite containing it and adding v to the other partite, and iterate. If $\min\limits_{w \in V(G)} \partial_{S_{i-1}}(w) = 0$, then let $A = \{x : 0 \leq \partial_{S_{i-1}}(x) \leq 1\}$. If there are vertices u and v such that $\partial_{S_{i-1}}(u) = 0$ and $v \in A$ and uv is a good edge, remove u from the partite containing it and add it to the other partite to construct S_i and $\overline{S_i}$, and iterate. If there are no such vertices, construct $G_{S_{i-1}}$ and terminate. See Fig 2 for an example.

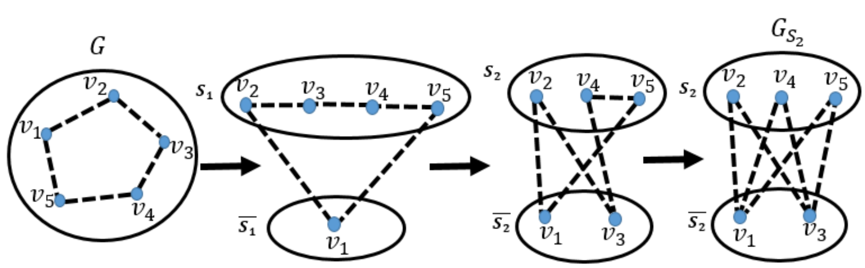

Figure 2: *G is a complete signed graph on 5 vertices. Missing edges have positive sign. Initially, $\partial(v) = 0$ for all $v \in V$ and the edge v_1v_3 is a good edge, hence, the algorithm moves v_1. In the partition $(S_1, \overline{S_1})$, $\partial_{S_1}(v_3) = -2$ and therefore we move v_3. In the partition $(S_2, \overline{S_2})$, $\partial_{S_2}(v_4) = \partial_{S_2}(v_5) = 0 = \min\limits_{w \in V(G)} \partial_{S_2}(w)$, but the edge v_4v_5 is a bad edge. As a result, the algorithm terminates by switching the signs of v_1v_4, v_3v_5 and v_4v_5, which constructs G_{S_2}.*

Next we discuss some properties of the algorithm.

Claim 2. *Let v be a vertex that moves at the i^{th} iteration. Then,* $\partial_{S_i}(v) = -\partial_{S_{i-1}}(v)$ *and* $\mathcal{D}(S_i, \overline{S_i}) = \mathcal{D}(S_{i-1}, \overline{S_{i-1}}) - 4\partial_{S_{i-1}}(v)$.

Proof. Since we move v from one partite to the other, each of its good edges become bad edges, and vice versa. In addition, for each neighbor x of v, the contribution made by vx to $\partial_{S_{i-1}}(x)$, is negated when we calculate $\partial_{S_i}(x)$. In summary, for each good edge incident to v, there is a contribution of -2, and for each bad edge there is a contribution of +2 to $\mathcal{D}(S_i, \overline{S_i})$. Since the difference between the number of good and bad neighbors of v is $\partial_{S_{i-1}}(v)$, the contribution of every vertex other than v to $\mathcal{D}(S_i, \overline{S_i})$ is $-2\partial_{S_{i-1}}(v)$. The contribution of v to $\mathcal{D}(S_i, \overline{S_i}))$ is also $-2\partial_{S_{i-1}}(v)$. □

Note that, $\mathcal{D}(S_i, \overline{S_i})$ increases at each iteration except when $\min_{w \in V(G)} \partial_{S_{i-1}}(w) = 0$. In that case, either the algorithm terminates, or $\mathcal{D}(S_i, \overline{S_i})$ increases again in the next iteration. By Lemma 1, $\mathcal{D}(S_i, \overline{S_i}) \leq 2|E(G)|$, and therefore, the algorithm will terminate after a finite number of steps. Once the algorithm terminates, $\partial_{S_k}(v) \geq 0$ for all $v \in V(G)$. As a result, to construct the balanced graph G_{S_k}, the algorithm would negate at most half the edges of $E(G)$. We will slightly improve this bound below.

We next examine the case of complete signed graphs. We will show that the maximum value of $l(G)$ among all complete signed graphs with n vertices is attained by the graph K_n^-, whose underlying graph is K_n, and whose edges all have negative signs. We will then prove that our algorithm will never negate more edges than this value while balancing complete signed graphs.

Lemma 2. $l(K_n^-) = \lceil \frac{n^2-2n}{4} \rceil$. *Furthermore, if $(S, \overline{S})$ is an optimal partition of K_n^-, then $\mathcal{D}(S, \overline{S}) = n$ if n is even, and $\mathcal{D}(S, \overline{S}) = n - 1$ if n is odd.*

Proof. A partition $(S, \overline{S})$ of K_n^- will be optimal if it contains the maximum number of edges between the two parts. As a result $|S| = \lceil n/2 \rceil$ and $|\overline{S}| = \lfloor n/2 \rfloor$, or vice versa. Since only the edges inside each part have to be negated we have, $l(K_n^-) = 2\binom{n/2}{2}$ if n is even and $l(K_n^-) = \binom{\lceil n/2 \rceil}{2} + \binom{\lfloor n/2 \rfloor}{2}$ if n is odd, as desired. The second statement can easily be proved by using Lemma 1, which we leave to the reader. □

Theorem 2. *For any signed graph G without isolated vertices $l(G) < |E|/2$. Additionally, if G is a complete signed graph on n vertices and $(S, \overline{S})$ is the partition of $V(G)$ identified by our algorithm, then $l(S, \overline{S}) \leq l(K_n^-)$.*

Proof. We can assume G is connected because the line index of a graph is the sum of the line indices of its components. Let $(S, \overline{S})$ be the partition of $V(G)$ identified by our algorithm. If G does not have a vertex v with $\partial_S(v) = 0$, then $\mathcal{D}(S, \overline{S}) \geq n$ and by Lemma 1, $l(G) \leq \frac{|E|}{2} - \frac{n}{4}$. So assume there is a vertex v such that $\partial_S(v) = 0$. Note that $\partial_S(v) = 0$ if and only if v has an equal number of good and bad neighbors. Since G is connected and not an isolated vertex, $d(v) \geq 2$ and each of its $d(v)/2$ good neighbors must have stability degree at least 2, otherwise the algorithm would not have terminated. Consequently, $\mathcal{D}(S, \overline{S}) \geq d(v)$ and $l(G) \leq \frac{|E|}{2} - \frac{d(v)}{4}$.

If G is a complete signed graph and n is even, then $d(v)$ is odd for all vertices in $V(G)$. Therefore, $\partial_S(v) \geq 1$ for all $v \in V(G)$ and $\mathcal{D}(S, \overline{S}) \geq n$, which yields $l(S, \overline{S}) \leq l(K_n^-)$. On the other hand if n is odd and there exist a vertex v with $\partial_S(v) = 0$, then each of its $\frac{n-1}{2}$ good neighbors must have stability degree

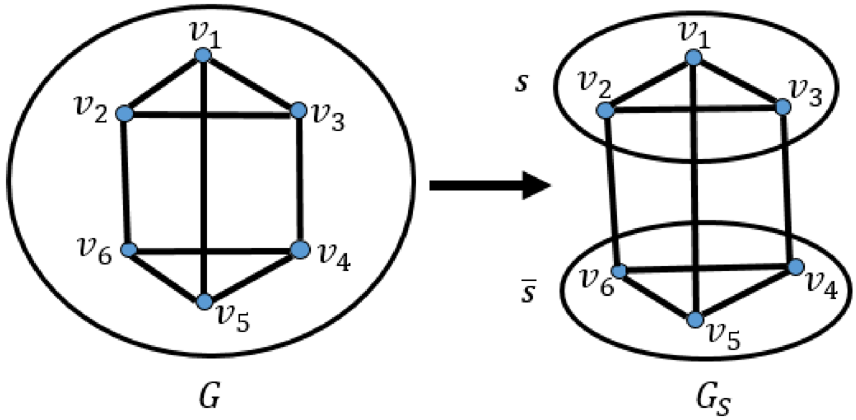

Figure 3: *G is a complete signed graph on* 6 *vertices. The missing edges have negative sign. Originally every vertex has positive stability degree but the partition $(V, \emptyset)$ is not optimal. The optimal partition is attained by $(S, \overline{S})$, where $S = \{v_1, v_2, v_3\}$.*

at least 2 so, $\mathcal{D}(S, \overline{S}) \geq n - 1$ and $l(S, \overline{S}) \leq l(K_n^-)$. □

A bipartition $(S, \overline{S})$ of a signed graph G is called *semi-optimal* if $\partial_S(v) \geq 0$ for each vertex v, and no vertex x with $\partial_S(x) = 0$ has a good neighbor y with $\partial_S(y) \leq 1$. Our algorithm finds a semi-optimal partition of any signed graph. Note that as seen in Fig 3, the algorithm does not necessarily find the optimal partition of a signed graph even if it is complete. However, by using the algorithm we can show that the problem of determining the line indeces of signed graphs, in which each vertex is incident to at least as many positive edges as negative edges, is still NP-Complete. We make use of the notion of switching for signed graphs developed by Zaslavsky [11]. Let G be a signed graph and $S \subseteq V(G)$. If G' is the signed graph obtained from G by negating the signs of the edges between S and $\overline{S}$ and keeping the signs of the remaining edges unchanged, we say G' is obtained from G by *switching* S. If a signed graph H can be obtained from a signed graph G by switching some $S \subseteq V(G)$, G and H are said to be *switching equivalent*, written $G \sim H$, and H is called the *unified representation* of the partition $(S, \overline{S})$ of G. Note that if $G \sim H$, then each good/bad edge of $(S, \overline{S})$ in G has positive/negative sign in H, respectively. Consequently, $\partial(v)$ in H is equal to $\partial_S(v)$ in G and $l(G) = l(H)$ as was shown in [10].

Theorem 3. *Determining the line index of signed graphs, in which each vertex is incident to at least as many positive edges as negative edges, is NP-Complete.*

Proof. Let G be a signed graph and $(S, \overline{S})$ be the partition of $V(G)$ identified by our algorithm. Since $\partial_S(v) \geq 0$ for all $v \in V(G)$, each vertex is incident to at least as many good edges as bad edges. Therefore, in the unified semi-optimal representation H of the partition $(S, \overline{S})$ of G each vertex is incident to at least as many positive edges as negative edges. If $l(H)$ could be computed in polynomial time, so could $l(G)$ since $l(G) = l(H)$. However, as mentioned before, computing $l(G)$ is an NP-Complete problem. □

Note that Theorem 3 reduces the problem of determining the line indeces of signed graphs to determining the line indeces of graphs in which each vertex is incident to at least as many positive edges as negative edges. We can also investigate the optimality of a bipartition of a signed graph G by considering its unified representation. Given a signed graph G and a subset S of $V(G)$, the *stability degree of* S, $\partial(S)$, is the sum of the weights of the edges which only have one endpoint in S. Recall that given a graph G and a subset S of its vertex set, the induced graph on S is $G[S]$. This leads us to our next results:

Lemma 3. *Let $G = (V, E, \sigma)$ be a signed graph and $S \subseteq V$. Then, $\mathcal{D}(G) = \mathcal{D}(G[S]) + 2\partial(S) + \mathcal{D}(G[\overline{S}])$.*

Proof. $\mathcal{D}(G) = \sum_{v \in V} \partial(v) = \sum_{v \in S} \partial(v) + \sum_{w \in \overline{S}} \partial(w) = \mathcal{D}(G[S]) + \partial(S) + \mathcal{D}(G[\overline{S}]) + \partial(\overline{S})$. □

Theorem 4. *Let $G = (V, E, \sigma)$ be a signed graph and $(S, \overline{S})$ be a partition of its vertex set and H be the unified representation of the partition $(S, \overline{S})$. The following conditions are equivalent:*

1. *$(S, \overline{S})$ is an optimal partition of G.*
2. *H cannot be split.*
3. *For any $A \subseteq V$, $\partial(A) \geq 0$ in H.*
4. *For any $A \subseteq V$, $\mathcal{D}(H[A]) \leq \sum_{v \in A} \partial(v)$.*

Proof. $(1 \Leftrightarrow 2)$ H splits if and only if $(V, \emptyset)$ is not an optimal partition of $V(H)$. As a result, $l(G) = l(H) < l_H(V, \emptyset) = l_G(S, \overline{S})$ if and only if $(S, \overline{S})$ is not an optimal partition of G.

$(2 \Leftrightarrow 3)$ By Lemma 3, we know that for any $A \subseteq V$, $\mathcal{D}(H) = \mathcal{D}(H[A]) + 2\partial(A) + \mathcal{D}(H[\overline{A}])$. For the partition $(A, \overline{A})$ of $V(H)$, each good edge between the two sets becomes a bad edge of the partition, and vice versa. Therefore, $\mathcal{D}(A, \overline{A}) = \mathcal{D}(H[A]) - 2\partial(A) + \mathcal{D}(H[\overline{A}])$. Also, H splits if and only if there exists $A \subseteq V$ such that $\mathcal{D}(H) < \mathcal{D}(A, \overline{A})$. But the latter is true if and only if $0 < -4\partial(A)$, which in turn is true if and only if $\partial(A) < 0$.

$(3 \Leftrightarrow 4)$ By Lemma 3, for any $A \subseteq V(H)$, $\sum_{v \in A} \partial(v) = \mathcal{D}(H[A]) + \partial(A)$. Consequently, $\mathcal{D}(H[A]) \leq \sum_{v \in A} \partial(v)$ if and only if $\partial(A) \geq 0$. □

In terms of actual social networks, Theorem 4 predicts that; if members of a subset of a social network are more loyal to their group than to the society as a whole, then polarization of the society may be unavoidable.

4 Conclusion

The simple greedy algorithm discussed in this paper has several advantages compared to other recent algorithms that balance signed graphs. Although much like Antal et al. [1] and other graph theoretic algorithms, the algorithm cannot always find an optimal partition of the signed network, it nevertheless does balance any signed graph. Compared to the algorithm of Marvel et al. [8], our algorithm also forces fewer relationship changes on the social network because a signed graph with mostly positive edges might still contain vertices that are incident to many negative edges. Therefore, the negation of all negative edges for such graphs might not be reasonable as suggested in [8].

To illustrate this point we consider the social network formed by the major state participants of the Syrian conflict. Serdar G. et al. in [4] describes in detail the evolution of the social network which can be summarized in Fig. 4 below.

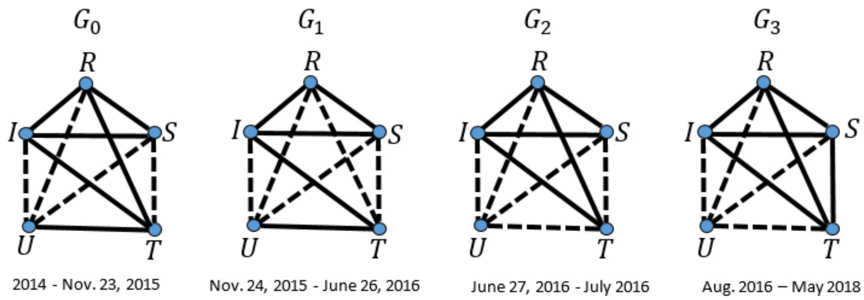

Figure 4: *These five graphs are complete signed graphs showing the evolution of the relationships between the five countries Turkey, USA, Russia, Iran and Syria, represented by T, U, R, I and S respectively, from 2014 to May 2018.*

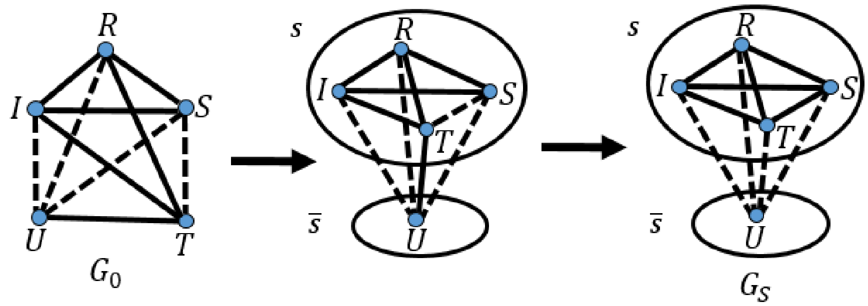

Figure 5: *The original graph G_0 of Fig. 4, and the partition $(S, \overline{S})$ identified by the algorithm, and the balanced graph G_S predicted by the algorithm.*

As seen in Fig. 5, starting from the original network G_0, the algorithm of this paper would yield the partitions $(S, \overline{S})$. Hence, the prediction of the algorithm would be that the relationship between US and Turkey would change from friendly to hostile, while the relationship between Turkey and Syria would change from hostile to friendly. These predictions correspond exactly to the final state G_3 of the network in Fig. 4. One could make the observation that the algorithm fails to predict the evolution of the network from state G_0 to G_1. However, since $\partial_S(T) = 0$ in Fig. 5, the easiest way for any party wishing to externally influence the evolution of the network would be to try to change the relationships between Turkey and the other countries. Indeed, Turkey officially blames members of the Fethullah Gulen movement for the shooting down of the Russian warplane. This incident caused the network to change to state G_1.

In the initial state G_0 of the network, most of the relationships are friendly. Consequently, the algorithm in [8] would predict a final network state where all parties are friendly to each other. It is clear that this prediction is not reflective of the evolution of the actual social network. One could easily provide an infinite family of signed graphs in which the density of positive edges is higher than the density of negative edges, but whose optimal partitions are clearly not $(V, \emptyset)$ as would be predicted by [8]. The same statement is also true of graphs where the density of negative edges is higher than the density of positive edges. Such a graph might have an optimal partition where the two partite sets do not have the same size, as would be predicted by [8]. See Fig. 6 for examples.

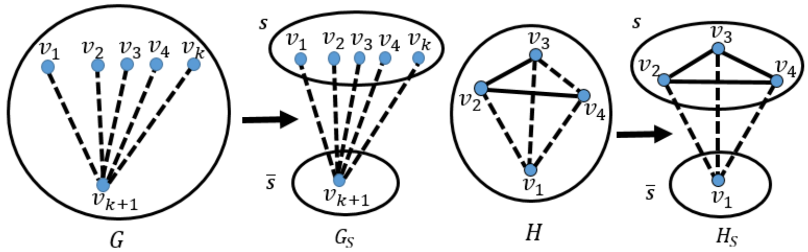

Figure 6: *G is a complete signed graph on $k+1$ vertices. The missing edges have positive sign. Note that G has $\binom{k}{2} = \frac{k^2-k}{2}$ positive edges and k negative edges, but the graph can be split by moving v_{k+1}. The signed graph H has optimal partition as shown; where the two partite sets do not have the same size.*

References

[1] T. Antal, P. L. Krapivsky, and S. Redner. Dynamics of social balance on networks. *Phys. Rev. E (3)*, 72(3):036121, 10, 2005.

[2] Francisco Barahona. On the computational complexity of ising spin glass models. *Journal of Physics A: Mathematical and General*, 15(10):3241, 1982.

[3] Dorwin Cartwright and Frank Harary. Structural balance: a generalization of heider's theory. *Psychological review*, 63(5):277, 1956.

[4] Serdar Ş Güner and Dilan E Koç. The syrian conflict: Driving forces of balances and imbalances. *MERIA Journal*, 33(3), 2017.

[5] Frank Harary. On the measurement of structural balance. *Behavioral Science*, 4(4):316–323, 1959.

[6] Fritz Heider. Attitudes and cognitive organization. *The Journal of Psychology*, 21(1):107–112, 1946. PMID: 21010780.

[7] Emmanuel Loukakis. A dynamic programming algorithm to test a signed graph for balance. *International journal of computer mathematics*, 80(4):499–507, 2003.

[8] Seth A Marvel, Jon Kleinberg, Robert D Kleinberg, and Steven H Strogatz. Continuous-time model of structural balance. *Proceedings of the National Academy of Sciences*, 108(5):1771–1776, 2011.

[9] Seth A Marvel, Steven H Strogatz, and Jon M Kleinberg. Energy landscape of social balance. *Physical review letters*, 103(19):198701, 2009.

[10] Shaoji Xu. The line index and minimum cut of weighted graphs. *European Journal of Operational Research*, 109(3):672–685, 1998.

[11] Thomas Zaslavsky. Signed graphs. *Discrete Applied Mathematics*, 4(1):47–74, 1982.