Identifying Stable States of Large Signed Graphs

Muhieddine Shebaro
m.shebaro@txstate.edu
Texas State University
San Marcos, Texas, USA

Jelena Tešić
jtesic@txstate.edu
Texas State University
San Marcos, Texas, USA

ABSTRACT

Signed network graphs provide a way to model complex relationships and interdependencies between entities: negative edges allow for a deeper study of social dynamics. One approach to achieving balance in a network is to model the sources of conflict through structural balance. Current methods focus on computing the frustration index or finding the largest balanced clique, but these do not account for multiple ways to reach a consensus or scale well for large, sparse networks. In this paper, we propose an expansion of the frustration cloud computation and compare various tree-sampling algorithms that can discover a high number of diverse balanced states. Then, we compute and compare the frequencies of balanced states produced by each. Finally, we investigate these techniques' impact on the consensus feature space.

ACM Reference Format:

Muhieddine Shebaro and Jelena Tešić. 2023. Identifying Stable States of Large Signed Graphs. In *Companion Proceedings of the ACM Web Conference 2023 (WWW '23 Companion), April 30-May 4, 2023, Austin, TX, USA.* ACM, New York, NY, USA, 4 pages. https://doi.org/10.1145/3543873.3587544

1 INTRODUCTION AND RELATED WORK

State-of-the-art techniques for processing unsigned homogeneous graphs can handle trillions of edges and billions of nodes. In contrast, signed graph benchmarks are limited to a few thousand nodes and hundreds of thousands of edges. Signed network graphs provide a way to model complex relationships, and interdependencies between entities as negative edges allow for a deeper study of social dynamics and stability in domains such as friendship and enmity [Antal et al. 2006; Leskovec et al. 2010] or brain behavior [Saberi et al. 2021]. However, current benchmarks for signed graph analysis are too small and do not accurately reflect the complexity and diversity of real-world signed networks. Recent research has also shown that proposed algorithms for signed graph analysis lack a principled direction and make assumptions that do not apply to real-world data [Cucuringu et al. 2021; Tomasso et al. 2022] and are limited to narrow-band tasks such as finance [Aref et al. 2016], polypharmacy [Liu et al. 2021a], bioinformatics [Li et al. 2021] and sensor data analysis [Casas et al. 2020; Liu et al. 2021b]. Balance theory is a concept that describes how attitudes and relationships change over time. It suggests that people tend to become friends with friends of their friends and enemies with enemies of their enemies. The foundations of social balance theory were established by Heider [Heider 1946] and Harary [Cartwright and Harary 1956; Harary and Cartwright 1968] provided the mathematical foundations for signed graphs and introduced the concept of k-way balance. These concepts have been applied in various ways, such as predicting edge sentiment, recommending content and products, and identifying unusual trends [Derr et al. 2020; Interian et al. 2022]. In a balanced network, every fundamental cycle must contain an even number of negative edges. The frustration cloud analysis of a signed graph in [Rusnak and Tešić 2021] and the efficient data structure and algorithm to efficiently compute fundamental cycles in [Alabandi et al. 2021]. *Graph* $G = (V, E)$ is a set of vertices V connected by the set of edges E. The number of vertices in the graph G is $|V|$, and the number of edges is $|E|$. *Path* is a sequence of distinct edges that connect a sequence of distinct vertices. *Cycle* is a path that begins and ends at the same vertex. *Connected graph* is a graph in which a path joins two vertices. *Subgraph* is a graph with all edges and vertices in a larger graph, for example, Path and Cycle. *Signed graph* Σ is a tuple of a graph $G = (V, E)$ and an edge signing function $\sigma : E \rightarrow \{+1, -1\}$. The edge can be positive + or negative $-$, $e \in [e^+, e^-]$, E^+, and E^- are sets of positive and negative edges of G. *Sign* of a subgraph is *product* of the edges signs. *Balanced signed graph* is a signed graph where every cycle is positive. *Harary bipartition* separates the vertices of the balanced graph into two sets such that the vertices of both sets internally agree with each other but disagree with the vertices of the other set[Cartwright and Harary 1956]. *Near-balanced* graph Σ' is a balanced graph that requires a minimum number of edge sign switches to produce a balanced graph from the signed graph Σ. *Frustration cloud* $\mathcal{F}_\Sigma$ is a set of all signed near-balanced graphs of Σ: $\mathcal{F}_\Sigma = \{\Sigma'_i | i \in [1, N] \wedge Fr(\Sigma'_i) = 0\}$. Many previous tree-sampling techniques were proposed to generate randomized spanning trees. For instance, we have Wilson's algorithm, which creates a random walk with a random node. If it encounters a visited node, it erases the resulting loop before continuation [Wilson and Propp 1996]. Aldous-Broder algorithm produces a random uniform spanning

Algorithm 1: Tree-Based Signed Graph Balancing

Data: Signed graph Σ
Data: Spanning tree T of Σ
Result: Balanced graph Σ'_T

```
forall edges e, e ∈ Σ \ T do
    if fundamental cycle T ∪ e is negative then
        switch edge sign in Σ'_T: e⁻ → e⁺; e⁺ → e⁻ ;
end
```

Permission to make digital or hard copies of all or part of this work for personal or classroom use is granted without fee provided that copies are not made or distributed for profit or commercial advantage and that copies bear this notice and the full citation on the first page. Copyrights for components of this work owned by others than the author(s) must be honored. Abstracting with credit is permitted. To copy otherwise, or republish, to post on servers or to redistribute to lists, requires prior specific permission and/or a fee. Request permissions from permissions@acm.org.
WWW '23 Companion, April 30-May 4, 2023, Austin, TX, USA
© 2023 Copyright held by the owner/author(s). Publication rights licensed to ACM.
ACM ISBN 978-1-4503-9419-2/23/04…$15.00
https://doi.org/10.1145/3543873.3587544

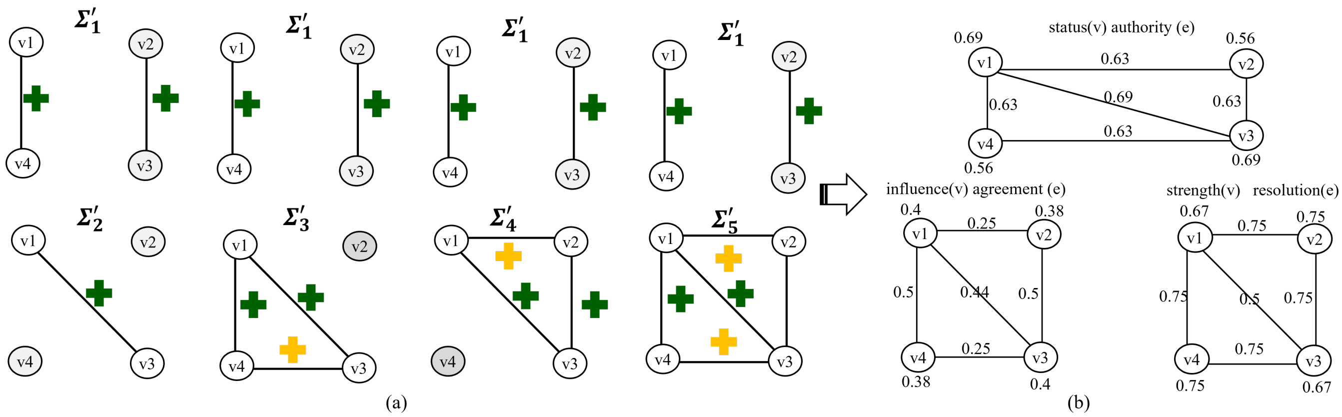


Figure 1: (a) Harary cuts of nearest balanced states (edges v1-v2 & v3-v4 are negative while edges v1-v4 & v2-v3 are positive) (b) consensus attributes derived for Σ: status(v)-authority(e); influence(v)-agreement(e); and resolution(v)-strength(e).

tree by performing a random walk on a finite graph with any initial vertex and stops after all vertices have been visited [Hu et al. 2021]. The partial rejection tree sampling algorithm randomly selects neighboring nodes in the graph. If the cycle is present in the selection, all edges of the loops are removed, and neighbors are selected again [Jerrum 2021]. Kruskal's algorithm with randomized weights produces a spanning tree that does not require a root node, is randomized in almost every run, and does not follow a static pattern [Hagberg et al. 2008]. Concerning consensus features, they are defined in [Rusnak and Tešić 2021].

Algorithm 2: Graph Balancing

Data: **Signed graph** Σ and spanning trees sampling method M
Result: frustration cloud $\mathcal{F}_\Sigma = \mathcal{B}$:($C$)
```
1  Generate set T_{M^k} of k spanning trees of Σ using M;
2  Empty F_Σ; ;
3  foreach spanning trees T, T ∈ T_k do
4      Find nearest balanced state Σ'_T using Alg. 1;
5      Transform Σ'_T balanced state to string B ;
6      if B ∉ ℬ then
7          add key B to ℬ;
8          C(B) = 1 ;
9      else
10         C(B)++ ;
11 end
```

2 METHODOLOGY

First, we extend the definition of frustration cloud $\mathcal{F}_\Sigma$ from a set to a *(key,value)* tuple collection $\mathcal{F}_\Sigma = \mathcal{B}$:($C$). The key is the unique balanced state $\mathcal{B}(i)$, and the value is the count of balanced states occurring in iteration $C(i)$. In each balancing iteration, we examine the resulting balance state (Alg. 2). Σ'_T in relation to $\mathcal{B}$. We represent the balanced state Σ'_T as a string B to make the process more efficient. The balanced state Σ'_T represents the 3 edge vectors (src,tgt,sign). If an edge i is defined by two vertices (u, v) and a sign s, the algorithm *graphB+* balances the graph by storing the edges as src(i)=u, tgt(i)=v, sign(i)=s [Alabandi et al. 2021].

We propose an efficient transform (O(|E|)) of the balanced state output Σ' to the string hash key B for comparison with other balanced states (Alg. 2 line 5). First, the triple edge vector is inserted into a set of tuple data structures to organize the edges and prepare for string conversion automatically. Then, it is transformed to a string format "src(i)->tgt(i): sign(i)," and then all edge strings are concatenated in order, separated by the delimiter "|" and stored as the B key in $\mathcal{B}$. If B is in $\mathcal{B}$, we increase the corresponding C(B) value count, where B is the existing balanced state Σ'_T. If Σ'_T is not in $\mathcal{B}$, we add (Σ'_T,(1)) pair to the collection. If the state was previously unseen, we add the new balanced state in an efficient matrix format to the hashmap as a string key as illustrated in Alg. 2. Then, we add 1 to the end of the count stack C. As the number of iterations increases, more elements will be added to $\mathcal{B}$, and the space complexity is linear. This can become an issue for graphs with millions of nodes and vertices as the frustration cloud is too big for the main memory.

We propose the randomization and hybridization of the standard tree sampling approaches to maximize the chances of discovering the optimal nearest balanced state in Alg. 2. **Depth first search (DFS)** algorithm [Cormen 2009] begins the traverse at the root node and proceeds through the nodes as far as possible until it reaches the node with all the nearby nodes visited. **Breadth first search (BFS)** algorithm [Cormen 2009] is a graph traversal approach in which the algorithm first passes through all nodes on the same level before moving on to the next level. The main drawback of efficient implementation of the efficient vertex search is the *static* order in accessing vertices in the adjacency lists. As a result, the efficient spanning tree algorithm can repeat the exact tree sampling and misses several unique trees. **Randomized Depth First Search (RDFS)** algorithm in Alg. 3 transforms DFS into a non-deterministic algorithm by eliminating the static ordering of the adjacency lists. The time complexity of the DFS is known to be O($|V| + |E|$), where $|V|$ is the number of vertices and $|E|$ is the number of edges in the signed network. The algorithm also runs in linear time O(n), where n

is the number of nodes adjacent to a specific node in the network, so the total time complexity is O($|V|+|E|$). Like RDFS, we also shuffled the adjacency lists for BFS in the **Randomized Breadth First Search (RBFS)** algorithm. The **Hybridized Sampler** randomly uses one of the two samplers (BFS & RDFS) in a specific iteration as shown in Alg 4.

Algorithm 3: RDFS: Random Edge Switching for Depth-First Search Spanning Tree Sampling

Data: Signed graph Σ and root node n
Result: Spanning tree T of Σ
Set visited[n] $\rightarrow True$;
get uniformly distributed random number, z;
shuffle adjacency list of n using seed z, adj[n];
forall *nodes* N, $x \in adj[n]$ **do**
  if *visited[x] is false* **then**
    Recursively call Alg.3 on signed graph Σ and root node x
end

Algorithm 4: Hybridized RDFS-BFS Sampling

Data: Signed graph Σ and a root Node n
Result: Spanning tree T of Σ
Get uniformly distributed random number 0 or 1, z ;
if z *is* 0 **then**
  Run BFS algorithm [Burtscher 2021]
else if z *is* 1 **then**
  Run Alg. 3

3 EXPERIMENTS AND RESULTS

3.1 Variation of the Frustration Cloud Size Using Different Tree-Sampling Techniques

signed graph	BFS	RBFS	DFS	RDFS	Hybrid
highland	125	125	8	**522**	290
sampson18	164	164	13	**957**	496
rainFall	306	306	175	**1000**	762
wikiElec	1000	1000	969	**1000**	1000
slashdot	1000	1000	943	**1000**	984
epinions	1000	1000	965	**1000**	1000
S&P1500	1000	1000	631	**1000**	1000
wikiRfa	1000	1000	915	**1000**	1000

Table 2: Number of unique balanced states for various signed network signed graphs with different samplers after 1000 iterations.

Executing algorithm 2 using different sampling techniques defined in section 2 in many signed networks and for 1000 iterations yielded different sizes for the frustration cloud. We can observe from table 2 that RDFS has produced the highest number of unique balanced states in all signed networks, whereas DFS has the lowest. In addition, this indicates that for smaller datasets (highland and sampson18), 1000 iterations are sufficient to induce the repetition in the resulting balanced states when all the nodes have already been selected as roots for BFS and DFS.

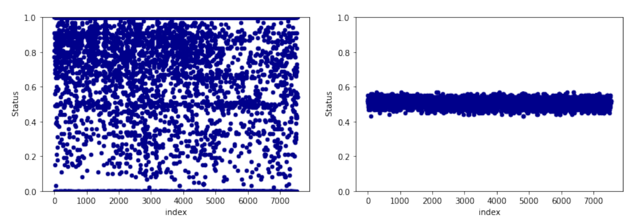
3.2 Frequencies of Nearest Balanced States in Frustration Cloud

In this subsection, we compare the balanced states along with their frequencies for the same number of iterations. We ran Alg. 2 over 1000 iterations for 5 different tree-spanning sampling algorithms on a signed network in Figure 1. For BFS, only three of the five closest balanced states were recovered, e.g. Σ_1, Σ_3, and Σ_4 with respective frequencies of 500, 250, and 250. RDFS was able to recover the five balanced states in the frustration cloud with frequencies 612, 111, 83, 86, and 108. We summarize the results incorporating other samplers in Table 3. BFS and RBFS seem to perform the same, and applying randomization to the adjacency lists did not affect the result.

Sampler	Σ_1	Σ_2	Σ_3	Σ_4	Σ_5
BFS	500	0	250	250	0
RBFS	500	0	250	250	0
DFS	500	250	250	0	0
RDFS	**612**	**111**	**83**	**86**	**108**
Hybrid	565	68	151	173	43

Table 3: Comparison of the frequencies of balanced states computed by different samplers for Σ in 1000 iterations

3.3 Impact of Tree-Sampling Techniques on Consensus Features



(a) BFS **(b) RDFS**

Figure 2: 1D Status Space for Wiki-Elec after 1000 iterations.

Because RDFS produced the most balanced states in the frustration cloud, more ways to cut the graph will exist. Hence, The same node will have less probability of winding up in the larger Harary bipartition. The result would be that the nodes will be less spread than those in the status space. It is also true that the more iterations we apply to a signed network, the more unique and nearest balanced states produced, the higher chance a node that was in, the smaller bipartition in the previous iteration to be in the larger bipartition, the narrower and tighter the nodes will be in the status space. Figure 2 (a) shows that with BFS, with 1000 iterations, nodes' status ranges from 0.0 to 1.0. This signifies that some nodes will never make it to the larger Harary bipartition because of the restricted number of balanced states and Harary cuts. These same nodes will also be falsely labeled as outliers in some applications. However, in Figure 2 (b), with RDFS and with the same number of iterations, it is observed that the nodes' status fluctuates approximately between 0.4 and 0.6.

	vertices	edges		vertex degrees			attributes	
signed graph	$\|V\|$	$\|E\|$	% positive	average	median	max	density	bal_3
Σ Fig. 1	4	5	60	2.5	2.5	3	0.833	0.0
test10 [Alabandi et al. 2021]	10	13	53.85	2.6	2.5	4	0.288	0.5
highland [Read 1954]	16	58	50	7.25	7.5	10	0.483	0.868
sampson18 [Sampson 1968]	18	112	54.4	12.44	12.50	16	0.732	0.6
rainFall [Cucuringu et al. 2021]	306	93,636	68.78	305.00	305	305	1.0	0.717
S&P1500 [Cucuringu et al. 2021]	1,193	711,028	75.13	1,192	1,192	1,192	0.833	0.718
wikiElec [Leskovec and Krevl 2014]	7,539	112,058	73.33	28.16	15	1,079	0.004	0.798
wikiRfa [Leskovec and Krevl 2014]	7,634	175,787	77.91	43.99	13	1,233	0.005	0.717
epinions [Leskovec and Krevl 2014]	119,130	704,267	83.23	11.82	2	3,558	< 0.001	0.890
slashdot [Leskovec and Krevl 2014]	82,140	500,481	77.03	12.19	2	2,548	< 0.001	0.856

Table 1: Signed graph attributes: $|V|$ is several vertices; $|E|$ is a number of edges in a graph; *% positive* is the number of positive edges divided by e; Vertex degree statistics are calculated in terms of the average, mean and maximum node degree; graph density d is calculated by dividing $2 * |E|$ by $|V| * |V-1|$ and bal_3 is the percentage of triangles in the graph that are balanced.

4 CONCLUSION

In this work, we show that the selection of the spanning tree sampling methods can influence the outcome of the analysis based on the balancing algorithms. The implication lies in the alteration of the values of the consensus features and the frequency in which every possible nearest balanced condition occurs. Because one sampler might capture a balanced state that the other cannot, one should harness multiple sampling techniques when balancing a signed network regardless of network size. We also show that the randomized depth-first search generates the most unique, balanced states for a given signed graph and a balancing method.

REFERENCES

Ghadeer Alabandi, Jelena Tešić, Lucas Rusnak, and Martin Burtscher. 2021. Discovering and Balancing Fundamental Cycles in Large Signed Graphs. In *Proceedings of the International Conference for High Performance Computing, Networking, Storage and Analysis* (St. Louis, Missouri) *(SC '21)*. Association for Computing Machinery, New York, NY, USA, Article 68, 17 pages. https://doi.org/10.1145/3458817.3476153

T. Antal, P. L. Krapivsky, and S. Redner. 2006. Social Balance on Networks: The Dynamics of Friendship and Enmity. *Physica D: Nonlinear Phenomena* 224, 1 (2006), 130–136. https://doi.org/10.1016/j.physd.2006.09.028 arXiv:physics/0605183

Samin Aref, Andrew J. Mason, and Mark C. Wilson. 2016. An exact method for computing the frustration index in signed networks using binary programming. *CoRR* abs/1611.09030 (2016). http://arxiv.org/abs/1611.09030

Martin Burtscher. 2021. graphB+: Balancing Algorithm for Large Graphs. https://userweb.cs.txstate.edu/~burtscher/research/graphB/.

Dorwin Cartwright and Frank Harary. 1956. Structural balance: a generalization of Heider's theory. *Psychological Rev.* 63 (1956), 277–293.

Sergio Casas, Cole Gulino, Renjie Liao, and Raquel Urtasun. 2020. Spagnn: Spatially-aware graph neural networks for relational behavior forecasting from sensor data. In *2020 IEEE International Conference on Robotics and Automation (ICRA)*. IEEE, 9491–9497.

Thomas H. Cormen. 2009. *Introduction to algorithms.* MIT Press. https://libproxy.txstate.edu/login?url=https://search.ebscohost.com/login.aspx?direct=true&db=cat00022a&AN=txi.b5115051&site=eds-live&scope=site

Mihai Cucuringu, Apoorv Vikram Singh, Déborah Sulem, and Hemant Tyagi. 2021. Regularized spectral methods for clustering signed networks. *Journal of Machine Learning Research* 22, 264 (2021), 1–79.

Tyler Derr, Zhiwei Wang, Jamell Dacon, and Jiliang Tang. 2020. Link and interaction polarity predictions in signed networks. *Social Network Analysis and Mining* 10, 1 (2020), 1–14.

Aric Hagberg, Pieter Swart, and Daniel S Chult. 2008. Exploring network structure, dynamics, and function using networkx. https://networkx.github.io/documentation/stable/reference/algorithms/index.html.

Frank Harary and Dorwin Cartwright. 1968. On the Coloring of Signed Graphs. *Elemente der Mathematik* 23 (1968), 85–89. http://eudml.org/doc/140892

Fritz Heider. 1946. Attitudes and cognitive organization. *J. Psychology* 21 (1946), 107–112.

Yiping Hu, Russell Lyons, and Pengfei Tang. 2021. A reverse Aldous-Broder algorithm. *ANNALES DE L INSTITUT HENRI POINCARE-PROBABILITES ET STATISTIQUES* 57, 2 (2021), 890 – 900. https://libproxy.txstate.edu/login?url=https://search.ebscohost.com/login.aspx?direct=true&db=edswsc&AN=000677592900014&site=eds-live&scope=site

Ruben Interian, Ruslan G Marzo, Isela Mendoza, and Celso C Ribeiro. 2022. Network polarization, filter bubbles, and echo chambers: An annotated review of measures, models, and case studies. *arXiv preprint arXiv:2207.13799* (2022).

Mark Jerrum. 2021. Fundamentals of Partial Rejection Sampling. https://doi.org/10.48550/ARXIV.2106.07744

Jure Leskovec, Daniel Huttenlocher, and Jon Kleinberg. 2010. Predicting Positive and Negative Links in Online Social Networks. In *Proceedings of the 19th International Conference on World Wide Web* (Raleigh, North Carolina, USA) *(WWW '10)*. ACM, 641–650.

Jure Leskovec and Andrej Krevl. 2014. SNAP Datasets: Stanford Large Network Dataset Collection. http://snap.stanford.edu/data.

Rui Li, Xin Yuan, Mohsen Radfar, Peter Marendy, Wei Ni, Terence J. O'Brien, and Pablo M. Casillas-Espinosa. 2021. Graph Signal Processing, Graph Neural Network and Graph Learning on Biological Data: A Systematic Review. *IEEE Reviews in Biomedical Engineering* (2021), 1–1. https://doi.org/10.1109/RBME.2021.3122522

Taoran Liu, Jiancong Cui, Hui Zhuang, and Hong Wang. 2021a. Modeling polypharmacy effects with heterogeneous signed graph convolutional networks. *Applied Intelligence* 51 (2021), 8316–8333.

Tianya Liu, Andong Sheng, Guoqing Qi, and Yinya Li. 2021b. Admissible Bipartite Consensus in Networks of Singular Agents Over Signed Graphs. *IEEE Transactions on Circuits and Systems II: Express Briefs* 68, 8 (2021), 2880–2884. https://doi.org/10.1109/TCSII.2021.3058803

Kenneth Read. 1954. Cultures of the Central Highlands, New Guinea. *Southwestern Journal of Anthropology* 10, 1 (1954), 1–43.

Lucas Rusnak and Jelena Tešić. 2021. Characterizing Attitudinal Network Graphs through Frustration Cloud. *Data Mining and Knowledge Discovery* 6 (November 2021). https://doi.org/10.1007/s10618-021-00795-z

Majid Saberi, Reza Khosrowabadi, Ali Khatibi, Bratislav Mišić, and Gholamreza Jafari. 2021. Topological impact of negative links on the stability of resting-state brain network. *Scientific Reports* 11, 1 (2021), 2176. https://doi.org/10.1038/s41598-021-81767-7

S. Sampson. 1968. A novitiate in a period of change: An experimental and case study of relationships. Ph.D. thesis, Cornell University.

Maria Tomasso, Lucas Rusnak, and Jelena Tešić. 2022. Advances in Scaling Community Discovery Methods for Signed Graph Networks. *Journal of Complex Networks* 10, 3 (06 2022). https://doi.org/10.1093/comnet/cnac013 arXiv:https://academic.oup.com/comnet/article-pdf/10/3/cnac013/44082209/cnac013.pdf cnac013.

David Bruce Wilson and James Gary Propp. 1996. How to get an exact sample from a generic Markov chain and sample a random spanning tree from a directed graph, both within the cover time. *Proceedings of the seventh annual ACM-SIAM symposium on Discrete algorithms* (1996), 448 – 457. https://libproxy.txstate.edu/login?url=https://search.ebscohost.com/login.aspx?direct=true&db=edscma&AN=edscma.314102&site=eds-live&scope=site