Discrete & Continuous Models
& Applied Computational Science
ISSN 2658-7149 (Online), 2658-4670 (Print)

2025, 33 (2) 130–143
http://journals.rudn.ru/miph

Research article

UDC 519.[176+178+16]:004.4
PACS 02.10.Ox 89.70.Eg 07.05.Tp
DOI: 10.22363/2658-4670-2025-33-2-130-143

EDN: BMIMTX

A technique of algorithms construction for solving a correlation clustering problem

Ellada I. Ibragimova, Daria V. Semenova, Aleksandr A. Soldatenko

Siberian Federal University, 79 Svobodny pr, Krasnoyarsk, 660041, Russian Federation

(received: January 24, 2025; revised: February 14, 2025; accepted: March 1, 2025)

Abstract. We propose a construction method for network structure based algorithms (NS-algorithms), aimed at solving the correlation clustering problem (CCP) specifically for signed networks. Our model assumes an undirected, unweighted simple signed graph. This problem is considered in optimization form with the error functional as a linear combination of inter-cluster and intra-cluster errors. It is known that this formulation of the problem is NP-hard. The technique of NS-algorithms constructing is grounded on the system approach presented in the form of a general scheme. The proposed scheme comprises six interconnected blocks, each corresponding to a stage in addressing the CCP solution. The main idea of the technique is to combine modules representing each block of the scheme. The proposed approach has been realized as a software package. The paper presents a model NS-algorithm constructed using the proposed technique. To evaluate its performance, computational experiments utilizing synthetic datasets are conducted, comparing the new algorithm against existing methods.

Key words and phrases: signed network, algorithm systematization, network structure based approach, correlation clustering problem

For citation: Ibragimova, E. I., Semenova, D. V., Soldatenko, A. A. A technique of algorithms construction for solving a correlation clustering problem. *Discrete and Continuous Models and Applied Computational Science* **33** (2), 130–143. doi: 10.22363/2658-4670-2025-33-2-130-143. edn: BMIMTX (2025).

1. Introduction

A network is a collection of real objects of an arbitrary nature, in which some pairs of these objects are connected. The network nodes can be objects from social, biological or telecommunications systems [1–3]. Signed networks are defined as an extension of networks that includes additional information about positive and negative links.

The following tasks are related to the analysis and modeling of signed networks: balance recognition; searching for a subnetwork with certain properties; clustering tasks; generating signed network with given properties; signed network mining [4–7].

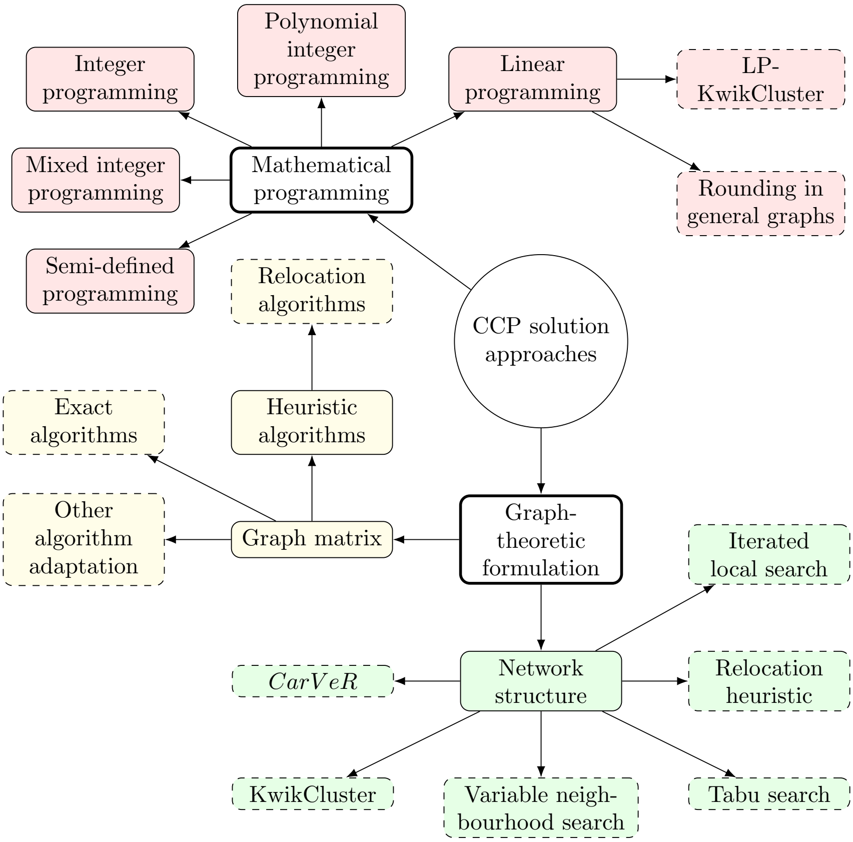

Figure 1. CCP solution approaches

Our research focuses on the Correlation Clustering problem (CCP). A historical overview of the problem is given in [8], an overview of existing solution methods is given in [9]. There are two main approaches to solving the correlation clustering problem [9]. Figure 1 shows the approaches to solving the correlation clustering problem. The first approach is based on the representing of the correlation clustering problem as a mathematical programming problem (e.g., linear, integer, semi-definite, etc.). [9, 10]. The second approach is based on a graph-theoretic representation of the signed network. There are two types of algorithm: those based on the graph's structure (NS-algorithms) [9, 11–13]; and those based on the graph's matrix representation [9, 14]. The following results refer to the former, i. e. NS-algorithms.

The definitions and the formulation of the problem of correlation clustering of signed networks necessary for further exposition are given in section 2. Section 3.1 describes the general scheme of algorithm construction and analysis, as well as possible variants of blocks. Section 3.2 describes the program complex for solving the CCP problem and investigates a new algorithm *CarVeR*. Section 3.3 presents the results of computational experiments for the algorithm built according to the scheme are given.

2. Theoretical Basis

This paper investigates signed networks $\Sigma = (G, \sigma)$, where $G = (V, E)$ is undirected, unweighted graph, with a vertex set V, $|V| = n \geqslant 2$, and a set of edges E, $|E| = m \geqslant 1$. We will consider simple graphs, i.e., graphs without multiple edges and loops. In the graph G each edge is uniquely represented by an unordered pair $e = (u, v)$, where $e \in E$, $u, v \in V$. In this case, edge e is said to be incident to vertices u and v, which are adjacent.

On edges $(u, v) \in E$ of the graph G the sign function $\sigma : E \to \{+, -\}$ is given, which generates a partition of the set of edges of the graph $E = E^+ \cup E^-$, where E^+ is a set of positive edges, E^- is a set of negative edges.

A signed network is k-balanced, if its vertex set can be divided into k pairwise non-intersecting non-empty clusters such that all positive edges lie within clusters and all negative edges lie between them [15].

We denote the system of sets that form a partition of the vertex set V into k subsets as follows:

$$C = \left\{ C_i \subseteq V : \bigcup_{i=1}^{k} C_i = V, C_i \cap C_j = \varnothing, i \neq j;\ i = \overline{1, k} \right\}.$$

It is known that the k-balance property may not be fulfilled for an arbitrary sign network. In this case, it is interesting to find a partition of the vertex set of the network such that changing the sign of the minimum number of edges results in a k-balanced network. This problem is considered as a graph clustering problem with a special kind of error functional. The elements of the partition $C_i \in C$ will be called clusters.

We define the positive error $P(C)$ of a partition (2) as the number of positive edges between subsets $C_1, \ldots, C_k$. Note that $P(C)$ is the inter-cluster error calculated by the formula

$$P(C) = \sum_{i=1}^{k} \sum_{u \in C_i} \sum_{v \in V \setminus C_i} [(u, v) \in E^+],$$

where $[\cdot]$ is Iverson bracket [16].

We define the negative error $N(C)$ as the number of negative edges in the subsets of partition (2). Negative error is the intra-cluster error calculated by the formula

$$N(C) = \sum_{i=1}^{k} \sum_{\{u,v\} \subseteq C_i} [(u, v) \in E^-].$$

In [17] the authors propose representing the total error as a convex combination of positive and negative errors depending on the parameter $\alpha \in [0, 1]$:

$$Q_\alpha(C) = \alpha N(C) + (1 - \alpha) P(C).$$

Note that the error functional (2) always satisfies the following inequality

$$0 \leq Q_\alpha \leq \alpha |E^-| + (1 - \alpha)|E^+|.$$

In this paper, the signed network clustering problem is considered in the following formulation [17].

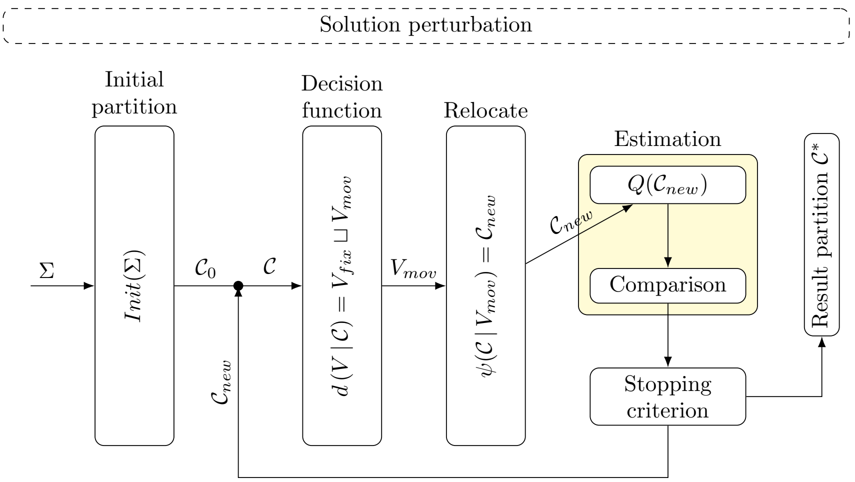

Figure 2. General NS-algorithms scheme

2.1. Correlation Clustering problem (CCP)

Condition: the signed network $\Sigma = (G, \sigma)$ is given, where $G = (V, E)$ is an undirected graph, $n = |V| \geqslant 2$, $m = |E| \geqslant 1$.

Question: for given $\alpha \in [0, 1]$ is required to find a partition $\mathcal{C}$ of vertex set V of signed network Σ with minimum total error $Q_\alpha(\mathcal{C})$.

As shown in [18], the problem of correlation clustering of signed network with an error functional in the form of (2) at $\alpha = 0.5$ is NP-hard.

The solution of the problem is the set of clusters $\mathcal{C}^*$ that provides the minimum of the error functional (2):

$$\mathcal{C}^* = \arg\min_{\mathcal{C}\in\Phi}\big[\alpha N(\mathcal{C}) + (1-\alpha)P(\mathcal{C})\big],$$

where $\Phi = \bigcup\limits_{k=1}^{n} \Phi_k$ is a set of all possible partitions V, Φ_k is a set of partitions into k subsets. The power of the solution space Φ is equal to Bell's number B_n. It should be noted that the solution (2.1) may not be single.

3. Results

3.1. A technique for constructing algorithms for solving CCP

To solve the CCP, a systematic approach involving the construction and analysis of algorithms based on network structure (NS-algorithms) was proposed in [19]. This approach was realized in the form of a general scheme, which is presented in Fig. 2.

The scheme consists of six blocks that are executed sequentially until the stopping condition is satisfied. In [19], five known and two our NS-algorithms for solving the CCP were analyzed using this scheme.

This paper proposes a detailed breakdown of the blocks in the general scheme. The modules that mimic the behavior of known NS-algorithms are highlighted for each block. A brief description of each block and its corresponding modules is then presented. For the modules that can be evaluated, an upper bound on the execution time for a naive implementation is given.

It should be noted that our systematic approach enables us to combine different modules to create new algorithms, and the topology of the overall scheme simplifies the analysis and proof of the computational complexity of the constructed algorithms.

The "Initial partition" block contains the function $Init(\Sigma)$, which performs the initial partitioning of the vertex set V of the signed network Σ according to a certain rule. This block is usually executed once at the start of the algorithm and forms the initial partitioning $C_0 \in \Phi$ in the form (2).

The following modules are highlighted for this block.

- *Trivial partition.* The original graph is randomly partitioned into k non-overlapping subsets. The subsets number k can be either given initially or generated, and $k < |V|$. The module is executed in time $\mathcal{O}(|V|)$.
- *Prepared partition.* The result of another algorithm or an input file is used as the initial partitioning.
- *Connectivity components.* The clusters are the connectivity components of the graph $\Sigma^+ = (V, E^+)$. The module is executed in time $\mathcal{O}(|V| + |E^+|)$.
- *Ranking and cut.* The score $I(v)$ is evaluated for each vertex v, and the cluster to which it should belong is determined. Then, for each selected cluster, all the vertices are ordered according to their score. A random vertex set among the first $\alpha < |V|$ vertices, where α is an integer is selected from the ordered vertex set. This process is repeated until all the vertices have been assigned to a cluster. The module can be completed in time $\mathcal{O}(|V|^2 \cdot \log|V| \cdot |E|)$.
- *KwikCluster.* Partitioning according to the *KwikCluster* algorithm [11]. At each step, a random vertex is selected to form a cluster with all positively connected vertices. This process is repeated until all the vertices have been assigned to a cluster. The module is executed in time $\mathcal{O}(|V| \cdot |E|)$.

The "Decision function" block is preparatory for the "Relocation" block. The decision function divides the set of vertices V taking into account the current partitioning C into two non-intersecting subsets:

$$d(V \,|\, C) = V_{fix} \sqcup V_{mov},$$

where V_{fix} is the set of vertices blocked for moving between clusters, and V_{mov} is the set of vertices allowed to move between clusters.

The following modules are allocated to this block.

- *Trivial function.* The solving function returns $V_{mov} = V$. The execution time of the module is $\mathcal{O}(1)$.
- *Tabu.* While the algorithm is running, the vertices that were affected are memorized, and are added to the tabu list V_{tabu} for a given number of iterations or until the end of the algorithm. The solving function returns $V_{mov} = V \setminus V_{tabu}$. The execution time of the module is $\mathcal{O}(|V|)$.
- *Negative error.* All vertices that contribute a negative error in (2) are returning. The execution time of the module is $\mathcal{O}(|V| \cdot |E|)$.
- *Potential function.* This block returns the vertices for which the value of some potential function satisfies the given condition.

The "Relocation" block contains the function $\psi(\mathcal{C} \,|\, V_{mov})$, which is responsible for moving vertices from the set V_{mov} between clusters of the current partition $\mathcal{C}$ and thus forming a new partition $\mathcal{C}_{new} \in \Phi$. It is worth noting that vertices can move not only between existing clusters, but also form new clusters.

The following modules are highlighted for this block. All the described strategies allow for the formation of new clusters.

- *By contribution*. The vertex from the set V_{mov} that makes the maximum contribution to the negative error is selected. A new cluster is chosen so that the target function (2) is minimized when a vertex joins it. The execution time of the module is $\mathcal{O}(|V|^2 \cdot |E|)$.
- *By objective*. A vertex from the set V_{mov} is selected and moved to another existing cluster so that the target function (2) is minimized. For this purpose, all possible vertex moves from the set of allowable vertices are evaluated. The execution time of the module is $\mathcal{O}\left(|V|^2 \cdot |E|\right)$.
- *Relocation of r vertices*. We select $1 \leq r \leq r_{max}$ vertices from the set V_{mov}, which are simultaneously moved to other existing clusters so that the target function (2) is minimized. In this case, r_{max} is set initially. The module is executed in the time $\mathcal{O}\left(|V|^{2r} \cdot |E|\right)$.

The "Evaluation" block consists of two stages. At the first stage, the error functional $Q(\mathcal{C}_{new})$ is calculated.

The second stage compares the current partition with previously found partitions by the value of the error functional. This block will also be present in the algorithm that finds several partitions sequentially or in parallel.

The "Stopping criterion" block allows to exclude the search over the whole Φ set. The stopping criterion can be: time, number of iterations, value of the error functional, residual of the error functional, etc.

The following modules are highlighted for this block:

- *Stop by time*. When the algorithm is started, a running time limit is set. Once the algorithm exceeds the specified time, the process stops and the current partitioning is the result.
- *Stop by iteration*. The number of iterations of the main loop of the algorithm is set when it starts, and the result is the partitioning obtained at the final iteration.
- *Stop by vertices*. The algorithm terminates when the tabu list contains all vertices. The result of the algorithm is the partition obtained at the last iteration.

The "Perturbation" block aims at getting out of the error functional local minimum by shuffling the vertices of the current partition, and the method of shuffling may determine the main idea of the algorithm. This block may follow after any other block of the overall scheme and may be repeated many times.

For this block, only the "Vertex shuffling" module is currently provided. A given number of vertices are moved between clusters randomly. The execution time of the module is $\mathcal{O}(|V|)$.

The table 1 represents known algorithms according to the modules described above. Comparisons of the first seven algorithms on synthetic and real data were presented in the papers [19–21].

The proposed technique allows us to construct both deterministic algorithms and algorithms with an element of random solution search. Analysis of existing NS-algorithms (see Table 1) revealed that elements of randomness appear in the "Initial partition" and "perturbation" blocks. In particular, the algorithms *CarVeR*, *SGClust*$_\alpha$, *Tabu Search* are deterministic, i.e., the same input leads to the same output. In the algorithms *Relocation heuristic*, *Variable neighborhood search* and *Iterated local search*, randomness is present at the initial partitioning stage and in the process of perturbing the solution to escape from local minima. The theoretical derivations and possible limitations of these algorithms are presented in the corresponding articles of the authors presented in the table 1.

The proposed system approach provides the possibility of combining different modules to develop new algorithms. In the last row of the 1 table, a representation of the new deterministic algorithm *Custom* is given. The algorithm *Custom* consists of the following modules according to the scheme in Fig. 2:

Table 1

Representation of algorithms for solving the CCP by scheme phases 2

Algorithm	Initial partition	Decision function	Relocation	Stopping Criterion	Perturbation
Relocation heuristic (*RH*) [17]	*Trivial partition*	*Trivial function*	*By objective*	*Stop by time*	–
Tabu search [12]	*Prepared partition*	*Tabu*	*By objective*	*Stop by time*	–
Variable neighborhood search [12]	*Prepared partition*	*Trivial function*	*By objective*	*Stop by time*	+
KwikCluster [11]	*KwikCluster*	–	–	*Stop by vertices*	–
Iterated local search (*ILS*) [13]	*Ranking and cut*	*Trivial function*	*Relocation of r vertices*	*Stop by iteration, time*	+
SGClust$_\alpha$ [21]	*Connectivity components*	*Negative error*	*By contribution*	*Stop by vertices*	–
CarVeR [19]	*Connectivity components*	*Potential function*	*By contribution*	*Stop by vertices*	–
Custom	*Connectivity components*	*Potential function*	*Relocation of r vertices*	*Stop by iteration, time*	–

- “Connectivity components” of the “Initial partition” block;
- “Relocation of *r* vertices” of the block “Relocation”;
- “Stop by iteration” of the “Stopping criterion” block.

Computational experiments on testing the new algorithm and comparing it with the known ones will be presented in section 3.3.

3.2. CCP solving software package

The technique of building and testing NS-algorithms for the CCP is implemented as software package in Python 3.10.2. The complex consists of three structural elements (Figure 3): NS-algorithms for the CCP solving, methods for signed graphs obtaining, comparison and analysis of the NS-algorithms results.

The first structural element implements the block modules shown in Figure 2. Modules are realized independently and can be combined according to the flow of the scheme taking into account their

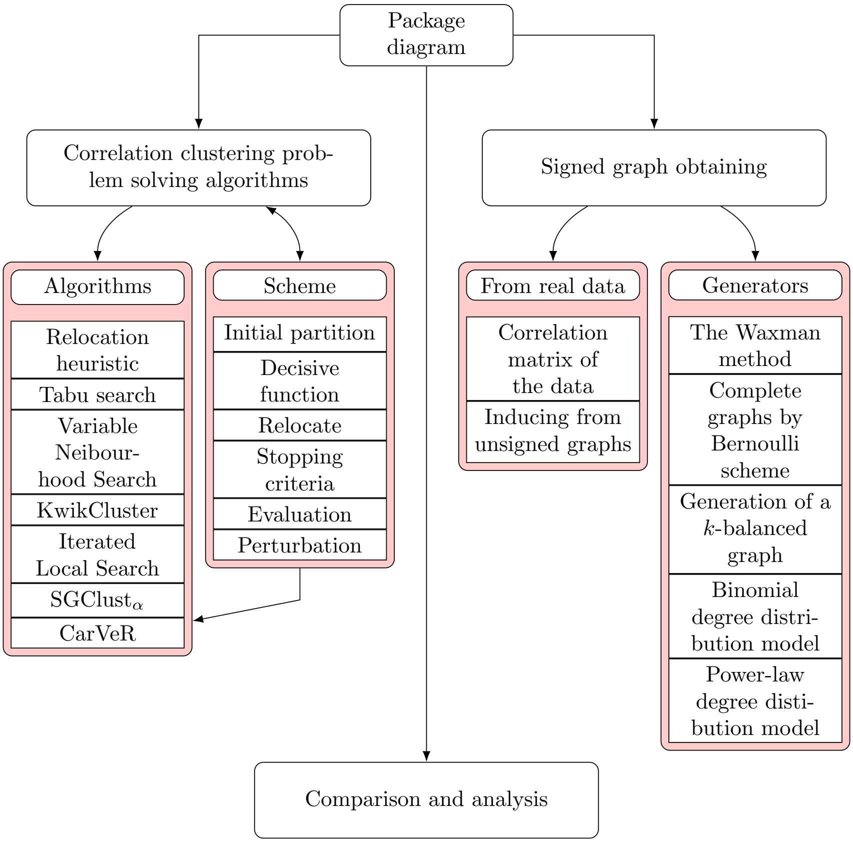

Figure 3. Software package structure

peculiarities. By combining the corresponding modules the known NS-algorithms according to the table 1 and the model algorithm *Custom* are realized in the program complex.

The second structural element implements both generation methods and methods of obtaining signed graphs from real data. The program provides the well-known generation methods, as well as methods for obtaining signed graphs from unsigned graphs described below.

The following methods of signed graph generation are implemented in the program complex.

- *Waxman's method [22].* To generate a graph consisting of n vertices, we generate n points on the two-dimensional plane. For each pair of vertices $\{u, v\}$ the probability of their connection by an edge is calculated by the following formula:

$$\mathbf{P}(\{u, v\}) = \beta \exp \frac{-d(u, v)}{L \cdot \alpha},$$

  where $d(u, v)$ is the distance between vertices u and v, L is the maximum distance between two vertices. The parameter β adjusts the density of edges, and the parameter α adjusts the density of long edges relative to short edges, $\alpha, \ \beta \in (0, 1]$. The sign of an edge is generated by Bernoulli scheme with a given probability p for a positive sign.
- *Complete sign graphs by Bernoulli scheme [21].* For each pair of vertices of a complete graph, the sign of an edge is generated by Bernoulli scheme with a given probability of positive sign p.

- *k-balanced.* The input to the algorithm is k is the number of clusters, n_k is the number of vertices in a cluster, p is the probability of an edge occurring. The edges inside the clusters are assigned a positive sign, while the edges between them are assigned a negative sign.
- *2-balanced with binomial distribution of vertices [23].* An unsigned graph with binomial distribution of vertices is generated. Then, the set of vertices is randomly divided into two groups. Edges within groups are assigned a positive sign and edges between groups are assigned a negative sign.
- *2-balanced with power law distribution of vertices [23].* An unsigned graph is generated with a degree distribution of vertices. Then, the set of vertices is randomly divided into two groups. The edges within each group are assigned a positive sign, while the edges between the two groups are assigned a negative sign.
- *Binomial distribution of vertices.* An unsigned graph with binomial distribution of vertices is generated. The sign of an existing edge is determined by Bernoulli scheme with parameter p — the probability of positive sign.
- *Power law distribution of vertices.* An unsigned graph with a power law distribution of vertices is generated. The sign of an existing edge is determined by Bernoulli scheme with parameter p — the probability of positive sign.

Some methods of obtaining sign graphs from real data are implemented in the software package.

- *Direct assignment.* The data is represented as a sign graph without modifications.
- *Correlation matrix.* The initial object-property table is considered. The nodes of the graph are the properties of the objects. For each pair of properties, the correlation is calculated and the edge is labeled based on it.
- *Inducing from unsigned graphs.* The papers [24, 25], propose methods for obtaining signed graphs from unsigned graphs. In the proposed methods, existing edges in a graph are assigned a sign based on some vertex-dependent functional. For example, its degree or eccentricity.

The third structural element enables comparative analysis of algorithms. For example, a given number of graphs are generated according to the specified method for the selected NS-algorithms. On the obtained graphs the solution of the CCP is searched by each of the selected algorithms. A summary table containing the averaged results for each algorithm and the characteristics of the graphs is output to a file. For better perception of the experimental results, their visualization is provided.

3.3. Computation experiments

This section presents the results of computational experiments for the new model NS-algorithm *Custom*, which was proposed in section 3.1.

All experiments were carried out on a computer with 8GB RAM, an Intel(R) Core(TM) i7-10510U CPU @ 1.80GHz processor running the Windows 10 operating system using single-threaded mode.

First series. The first series of experiments was conducted for the algorithms $SGClust_{\alpha}$, *CarVeR* and *Custom* on 200 graphs of 25 vertices each generated using the Waxman method. All three algorithms were run for each graph and the time taken, as well as the main clustering parameters (number of clusters, positive and negative errors, and total error), were measured. The stopping criterion was applied to the relocation block of the *Custom* algorithm. The set time was equal to 5 s.

Figure 4 compares the algorithms $SGClust_{\alpha}$, *CarVeR* and *Custom* by inter-cluster and intra-cluster errors. From the plots, we can conclude that the algorithms produce comparable error rates.

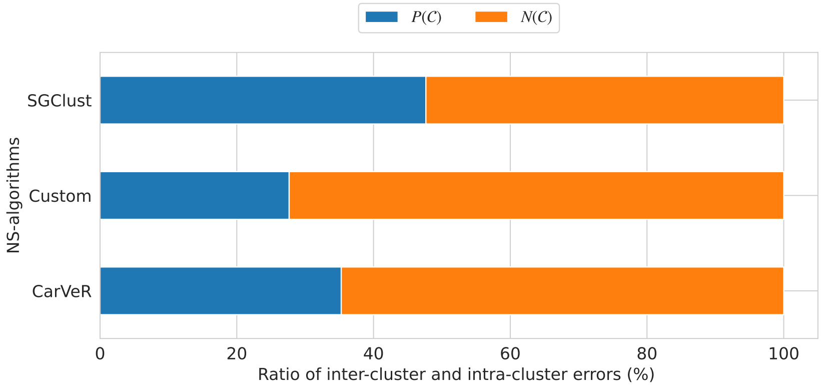

Figure 4. Comparison of NS-algorithms by inter-cluster $P(\mathcal{C})$ and intra-cluster $N(\mathcal{C})$ errors

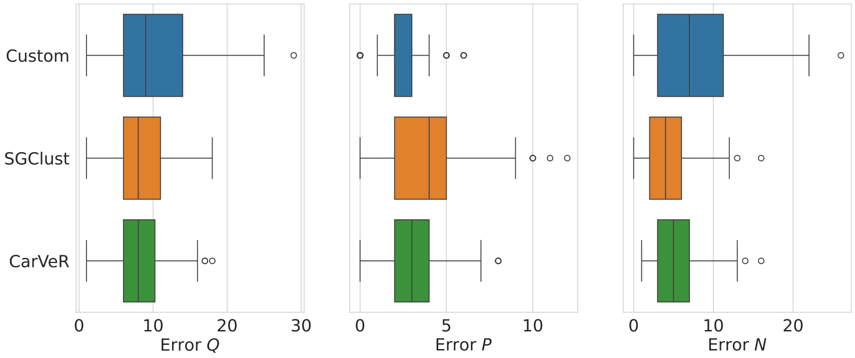

Figure 5. Comparison of algorithms by error functional

are comparable in terms of error.

Figure 5 compares the algorithms $SGClust_\alpha$, $CarVeR$ and $Custom$ in terms of total, positive and negative error values.

Second series. The second series of experiments was performed on the full signed graph, with parameters: $|V| = 20$, $|E| = 190$, $|E^+| = 94$, $|E^-| = 96$. The *Custom* algorithm was run with different specified times ranging from the running time of the *CarVeR* algorithm to 15 seconds with a step of 3 seconds. Some of the results are presented in the table 2. As can be seen from the table, the error decreases as the running time of the *Custom* algorithm increases. Therefore, when using time-dependent algorithms, it is necessary to select the running time.

4. Discussion

This paper builds upon our ongoing research into the development and analysis of heuristic NS-algorithms for solving the correlation clustering problem for signed networks. We have developed a technique for constructing NS-algorithms for solving CCP based on a systematic approach, which is

Table 2

Comparison of algorithms at different runtimes

Algorithm	Cluster count	Error	Negative error	Positive error	Time, s
Custom	2	84	72	12	2,828
Custom	2	72	52	20	3,464
Custom	3	59	33	26	6,045
SGClust	4	59	21	38	$< 0,001$
CarVeR	3	59	33	26	$< 0,001$

presented in the form of a generalized scheme comprising six interrelated components. The key idea of the proposed technique is a modular architecture, which enables the combination of different algorithmic solutions of existing and new NS-algorithms at each stage. This provides a flexible platform for the development and analysis of NS-algorithms.

The proposed technique has been implemented as a Python program. This complex provides the possibility of constructing new NS-algorithms, applying existing ones and comparing the results of their execution. The modular structure of the complex provides flexibility of approach: different modules can be more effective for certain types of networks. Due to this, searching for a solution by different NS algorithms can help to find a solution with better performance (lower error, in less time, etc.). Note that some of the modules used can be resource-intensive, which limits their application to high dimensional networks. Besides, the complex supports generation of sign networks with specified parameters, the induction of signed networks from unsigned graphs, construction of signed networks from real data. This enables the formation of a library of signed networks of varying complexity, which is necessary for testing and the comparative analysis of NS algorithms.

Based on the proposed approach, a model NS-algorithm was constructed, and its efficiency was tested in two series of computational experiments using synthetic data. In the first series of experiments, the model algorithm was compared with *CarVeR* and $SGClust_\alpha$ algorithms in terms of error functional. The obtained results indicate that the value of the error functional for the model algorithm is comparable to the results of known algorithms, which confirms its competitiveness. In the second series of experiments, the effect of the stopping criterion on the quality of clustering was studied. In particular, it was found that the use of the time stopping criterion requires careful tuning, since the limitation of the algorithm running time can significantly affect the quality of the obtained solution. It should be noted that the constructed *Custom* algorithm is not intended to compete with existing NS-algorithms. Our objective was to showcase the potential of the proposed approach for developing such algorithms. Therefore, we can conclude that the *Custom* algorithm can be used as a basis for NS-algorithms creating.

5. Conclusion

A promising area for further research is the development of a formal method for obtaining an upper estimate of the computational complexity of algorithms constructed on the basis of the proposed scheme. It is also advisable to extend the software system's functionality by including other approaches to solving CCP, developing new methods for generating signed networks, adding new metrics for evaluating graph properties and clustering, and adapting the proposed technique to solve other problems on signed networks. Specific examples of extensions to the software package include: adding alternative error functionals to the “Evaluation” block, including different potential functions in the “Decision Function” block, adding new generation algorithms to the “Signed graph obtaining” block, and introducing different metrics for analyzing the sign network properties and clustering quality in the “Comparison and Analysis” block.

Author Contributions: Conceptualization, D.V. and A.A.; methodology, D.V. and A.A.; Software, E.I.; validation A.A. and E.I.; formal analysis D.V. and A.A.; investigation E.I.; data curation, E.I.; writing - original draft, E.I.; writing - review & editing, D.V. and A.A.; visualization, A.A. and E.I.; supervision, D.V.; project administration, D.V.; funding acquisition, D.V. All authors have read and agreed to the published version of the manuscript.

Funding: This work is supported by the Krasnoyarsk Mathematical Center and financed by the Ministry of Science and Higher Education of the Russian Federation (Agreement No. 075-02-2024-1429).

Data Availability Statement: Data are openly available in a repository. Data supporting this study are openly available from Zenodo at 10.5281/zenodo.15206744 and 10.5281/zenodo.15206779.

Conflicts of Interest: The authors declare no conflict of interest.

References

1. Maatouk, A., Hajri, S. E., Assaad, M. & Sari, H. On optimal scheduling for joint spatial division and multiplexing approach in fdd massive mimo. *IEEE Trans Signal Process* **67,** 1006–10213 (2018).
2. Arinik, N., Figueiredo, R. & andLabatut, V. Multiple partitioning of multiplex signed networks: Application to European parliament votes. *Social Networks* **60,** 83–102 (2020).
3. Abdelnasser, A., Hossain, E. & Kim, D. I. Clustering and resource allocation for dense femtocells in a two-tier cellular OFDMA network. *IEEE Trans Wireless Communications* **13,** 1628–1641 (2014).
4. Hüffner, F., Betzler, N. & Niedermeier, R. *Optimal Edge Deletions for Signed Graph Balancing* in *Experimental Algorithms* (ed Demetrescu, C.) (Springer Berlin Heidelberg, Berlin, Heidelberg, 2007), 297–310.
5. Figueiredo, R. & Frota, Y. *An improved Branch-and-cut code for the maximum balanced subgraph of a signed graph* 2013. ArXiv: abs/1312.4345.
6. Figueiredo, R. & Frota, Y. The maximum balanced subgraph of a signed graph: Applications and solution approaches. *European Journal of Operational Research* **236** (2014).
7. Tang, J., Chang, Y., Aggarwal, C. C. & Liu, H. A Survey of Signed Network Mining in Social Media. *ACM Computing Surveys CSUR* **49,** 1–37 (2015).
8. Il’ev, V., Il’eva, S. & Kononov, A. *Short Survey on Graph Correlation Clustering with Minimization Criteria* in *Lecture Notes in Computer Science* (Germany, Heidelberg, Springer-Verlag, 2016), 25–36.
9. Wahid, D. F. & Hassini, E. A Literature Review on Correlation Clustering: Cross-disciplinary Taxonomy with Bibliometric Analysis. *Oper. Res. Forum* **47** (2022).
10. Queiroga, E., Subramanian, A., Figueiredo, R. & Frota, Y. T. Integer programming formulations and efficient local search for relaxed correlation clustering. *Journal of Global Optimization* **81,** 919–966 (2021).
11. Ailon, N., Charikar, M. & Newman, A. *Aggregating inconsistent information: ranking and clustering* in *Symposium on the Theory of Computing* (2005).
12. Brusco, M. J. & Doreian, P. Partitioning signed networks using relocation heuristics, tabu search, and variable neighborhood search. *Social Networks* **56,** 70–80 (2019).
13. Levorato, M., Figueiredo, R., Frota, Y. & Drummond, L. Evaluating balancing on social networks through the efficient solution of correlation clustering problems. *EURO Journal on Computational Optimization* **5,** 467–498 (2017).
14. Doreian, P. & Mrvar, A. Partitioning signed social networks. *Soc. Networks* **31,** 1–11 (2009).

15. Harary, F. Structural Balance: A Generalization of Heider's Theory. *Psychological Review* **63,** 227–293 (1956).
16. Graham, R. L., Knuth, D. E. & Patashnik, O. *Concrete Mathematics: A Foundation for Computer Science 2nd edn.* 657 pp. (Addison-Wesley, Massachusetts, USA, 1994).
17. Doreian, P. & Mrvar, A. A partitioning approach to structural balance. *Social Networks* **18,** 149–168 (1996).
18. Bansal, N., Blum, A. & Chawla, S. Correlation Clustering. *Machine Learning* **56,** 89–113 (2002).
19. Soldatenko, A. A., Semenova, D. V. & Ibragimova, E. I. An approach to analyzing and constructing algorithms for solving a one clustering problem on signed graphs. *Applied discrete mathematics,* 112–127 (2024).
20. Ibragimova, E. I., D. V. Soldatenko, A. A. & Semenova. *Comparison of Two Heuristic Algorithms for Correlation Clustering Problem Solving* in *2023 5th International Conference on Problems of Cybernetics and Informatics (PCI)* (IEEE, 2023), 1–4.
21. Soldatenko, A., Semenova, D. & Ibragimova, E. *On heuristic algorithm with greedy strategy for the Correlation Clustering problem solution* in *Lecture Notes in Computer Science* (Germany, Heidelberg, Springer-Verlag, 2023), 462–477.
22. Waxman, B. M. Routing of multipoint connections. *IEEE Journal on Selected Areas in Communications* **6,** 1617–1622 (1988).
23. Ciotti, V., Bianconi, G., Capocci, A., Colaiori, F. & Panzarasa, P. Degree correlations in signed social networks. *Physica A: Statistical Mechanics and its Applications,* 25–39 (2015).
24. Aniyan, A. & Naduvath, S. Induced signed graphs of some classes of graphs. *Proceedings of the Jangjeon Mathematical Society* **23,** 283–291 (2020).
25. Sudheer, N., George, A. C., Aniyan, A. & Naduvath, S. Some new results on colour-induced signed graphs. *Acta Univ. Sapientiae Informatica* **14,** 338–353 (2022).

Information about the authors

Ellada I. Ibragimova—Assistant of Department of Higher and Applied Mathematics, Siberian Federal University (e-mail: EIbragimova@sfu-kras.ru, ORCID: 0000-0002-6852-2281, Scopus Author ID: 58766400800)

Daria V. Semenova—Candidate in Physics and Mathematics, Associate Professor of the Department of Higher and Applied Mathematics, Siberian Federal University (e-mail: DVSemenova@sfu-kras.ru, ORCID: 0000-0002-8670-2921, ResearcherID: O-3107-2019, Scopus Author ID: 49261191500)

Aleksandr A. Soldatenko—Candidate in Physics and Mathematics, Associate Professor of the Department of Higher and Applied Mathematics, Siberian Federal University (e-mail: ASoldatenko@sfu-kras.ru, ORCID: 0000-0001-6708-4753, ResearcherID: O-8550-2019, Scopus Author ID: 57194285182)

УДК 519.[176+178+16]:004.4

PACS 02.10.Ox 89.70.Eg 07.05.Tp

DOI: 10.22363/2658-4670-2025-33-2-130-143 EDN: BMIMTX

О методе построения алгоритмов для решения задачи корреляционной кластеризации

Э. И. Ибрагимова, Д. В. Семенова, А. А. Солдатенко

Сибирский федеральный университет, д. 79, пр. Свободный, г. Красноярск, 660041, Российская Федерация

Аннотация. Развивается техника построения алгоритмов, основанных на структуре сети (NS-алгоритмы), для решения задачи корреляционной кластеризации (CCP) для знаковых сетей. Моделью знаковой сети выступает неориентированный и невзвешенный простой знаковых графов. Эта задача рассматривается в оптимизационной форме с функционалом ошибки в виде линейной комбинации межкластерной и внутрикластерной ошибок. Известно, что в данной постановке задача является NP-трудной. Техника построения NS-алгоритмов основана на системном подходе, представленном в виде общей схемы. Схема состоит из шести взаимосвязанных блоков, каждый из которых отражает основные этапы решения CCP. Основная идея техники заключается в комбинировании модулей, представляющих каждый блок схемы. Предложенный подход реализован в виде программного комплекса. В работе представлен модельный NS-алгоритм, построенный с помощью предлагаемой техники. Проведены вычислительные эксперименты на синтетических данных по сравнению модельного алгоритма с уже известными.

Ключевые слова: знаковая сеть, задача корреляционной кластеризации, NS-алгоритм, техника построения