European Proceedings of
Social and Behavioural Sciences
EpSBS

www.europeanproceedings.com
e-ISSN: 2357-1330

DOI: 10.15405/epsbs.2021.09.02.190

ICEST 2021
II International Conference on Economic and Social Trends for Sustainability of Modern Society

COMPROMISE SOLUTION IN ECONOMIC COMPETITION

I. V. Zaitseva (a)*, O. A. Malafeyev (b), V. V. Bondar (c), O. I. Skvortsova (d)
*Corresponding author

(a) Russian State Hydrometeorological University, 79, Voronezhskaya Street, Saint Petersburg, Russia
irina.zaitseva.stv@yandex.ru
(b) Saint Petersburg State University, Saint Petersburg, Russia, malafeyevoa@mail.ru
(c) North Caucasus Federal University, Stavropol, Russia, viktori-bondar@yandex.ru
(d) Caucasus Federal University, Stavropol, Russia, olga-skvorcova2015@yandex.ru

Abstract

The idea arose to look at new approaches specifically to identify hidden societies, evaluate their effectiveness and explore the role of the results obtained in the economic field, that is, to address the problem of compromise using game theory. The idea behind the research is to apply a new set of methods to identify hidden communities, which are then presented to agents for their use. The agent, in turn, needs to choose an algorithm in such a way as to obtain maximum profit and minimum losses in competition in the economic resource market. In this paper, the main algorithm for identifying hidden communities is an algorithm based on modularity, in addition, the following algorithms are considered: The Bron-Kerbosch algorithm; The shortest open path algorithm; the Minimum covering tree search algorithm; the graph kernel search algorithm. For some algorithms, it is necessary to introduce some modification. This topic is important today, because with the help of a social network, hidden terrorist or fraudulent groups can be formed, which are subject to research and struggle in the future.

Keywords: Game theory, compromise solution, model, economic competition, social network

https://doi.org/10.15405/epsbs.2021.09.02.190
Corresponding Author: I. V. Zaitseva
Selection and peer-review under responsibility of the Organizing Committee of the conference
eISSN: 2357-1330

1. Introduction

First of all, it is necessary to build a mathematical model, based on the compromise problem will be formulated below. Let's turn to the theory of social graphs. A user is taken as $v_j, j = \overline{1, k}$; As the bracket for $e_{jl} = (v_j, v_l)$ there is a connection between the users j and l. By "Link" here is meant the presence of a comment from one user to another. For each e_{jl}, the weight w_r is determined as the total number of comments addressed to a given user. Let's denote it as V the set of nodes: $V = \{V_1, V_2, \dots, V_k\}$, as E - the set of edges: $E = \{E_1, E_2, \dots, E_l\}$, $W = \{W_{11}, W_{12}, \dots, W_{rp}\}$ −multiple weights. An edge is assigned to a pair of nodes, and a weight is assigned to each edge. The model is an ordered triple of sets $G = (V, E, W)$. Thus, we have a model. Thus, we have a model, applying to which certain algorithms for identifying hidden communities, we obtain information for further research (Agarkova et al., 2016; Bogoviz et al., 2018; Freeman et al., 1989).

2. Problem Statement

Let a set of agents be $I = \{I_1, I_2, \dots, I_n\}$, interested in identifying hidden communities. Any organization can be understood as an agent. So, We will assume that each agent has familiarized himself with the results of the operation of the algorithm A_t, $t = \overline{1, m}$ and made a decision about how useful this algorithm can be. Agents allocate a certain amount of resources for each of the algorithms, that is, each resource is assigned a weight. A resource can be understood here, for example, as a server or a storage area for storing data. The problem is that each agent needs to choose such an algorithm in order to get maximum profit and minimum loss when competing with other organizations (Al-Qurishi et al., 2017; Bondarenko et al., 2015; Ivanyukovich et al., 2020; Kostikova et al., 2016; Kuznetsov et al., 2016).

To solve this problem, all the possible options for the distribution of algorithms among organizations are considered (the case without intersections). Let us denote the set of options as $X = \{x_1, x_2, \dots, x_p\}$, where $p -$ is the number of all possible options for the distribution of algorithms for identifying hidden communities, and call it the set of feasible solutions (Jebabli et al., 2015). The decision here is the choice of a specific distribution of algorithms $x^* = x_j$ between all agents. Let's introduce the utility function $H_j(x_l)$ such that $H_j(X)\colon X \to R^1, j = \overline{1, m}$. The utility matrix is compiled from the values of this function:

$$\begin{pmatrix} H_1(x_1) & \cdots & H_1(x_p) \\ \vdots & \ddots & \vdots \\ H_m(x_1) & \cdots & H_m(x_p) \end{pmatrix}$$

Let us introduce the concept of a compromise solution to the problem posed. As $M_i, i = \overline{1, n}$, we take the maximum income that agent I_k receives as a result of applying the algorithm A_j from the set A:

$$M_i = \max_{x \in X} H_i(x), \qquad i = \overline{1, n}$$

Next, an ideal vector$M = (M_{1,} M_2, \dots\ M_n)$, is constructed each element of which determines the maximum income for each agent. Let us introduce a residual vector whose element is the value $M_k - H_k(x)$for$\forall x \in X, k = \overline{1, p}$. Vector elements are ordered in ascending order. The first element of the current vector is the value, which shows that there is no difference between the maximum income and the value of the utility function. The very last element of the vector indicates the so-called "offended" agent, that is, an agent is determined whose difference between the maximum income and the value of the utility function is maximum:

$$\max_{i\in I}\big(M_i - H_i(x)\big), \qquad i = \overline{1,n}$$

A compromise solution is understood as the distributionC_X^H:

$$\mathrm{C}_X^H = \underset{x\in X}{\operatorname{argmin}} \max_{i\in I}(M_i - H_i(x))\,, i = \overline{1,n}$$

Now, let’s formulate an algorithm for finding a compromise solution (Malafeyev et al., 2019a):

1. Form a utility table, where each agent i- agent associates assigns the weight wr to the j-th algorithm.
2. We compose the set of all possible distributions of algorithms for organizations. This set is a table where $x_k$is considered as a row - a variant of the distribution of algorithms between agents, as a column - the current agent, as a value - the algorithm that is used.
3. We form a table in which the current variant of distribution acts as rows, agents as columns, and the value of the utility function is determined as values.
4. Using the constructed table for each agent, we determine the maximum payoff $M_J$and form an ideal vector $M = (M_1, \dots, M_n)$
5. Form a non-viscosity table, where $M_k - H_k(x)$ is taken as values for $\forall x \in X, k = \overline{1,p}$
6. From this table for each distribution (by rows) we determine the maximum value:

$$\max_{i\in I}\big(M_i - H_i(x)\big), \qquad i = \overline{1,n}$$

7. From the obtained values, we select the minimum deviation (by lines):

$$\mathrm{C}_X^H = \underset{x\in X}{\operatorname{argmin}} \max_{i\in I}(M_i - H_i(x))\,, i = \overline{1,n}$$

Thus, we have determined the optimal variant of the distribution of algorithms for all agents, that is, we have found a compromise solution to the problem posed.

3. Research Questions

In The work, the main algorithm for identifying hidden communities was an algorithm based on modularity.

4. Purpose of the Study

We have a mathematical model and many algorithms that can be used to identify hidden communities on a social network. The question arises: how can this be used in practice.

5. Research Methods

5.1. Bron-Kerbosch algorithm

This algorithm is intended for searching in the graph of clicks (Zubov, 2007). Click - is a subset of graph nodes in which any two vertices are connected by an edge. The point of highlighting clicks is that we identify powerful enough connections - communities in which everyone contacts everyone. Thus, the task of finding hidden communities is reduced to the task of finding maximum clicks by inclusion. The task of searching for clicks is reduced to the task of finding an independent set of vertices by constructing the complement of the graph. By an independent set of vertices we mean such a subset, any two elements of which are not adjacent.

Before describing this algorithm, we introduce the following notation:

$V -$ set of vertices, v - considered vertex;

$M -$ is the current independent set of vertices;

$\Gamma(M)$ – is the set of vertices adjacent to vertices from M;

$K -$ is the set of considered candidates (those who can be added to M);

P –is the set of viewed vertices that are not added to the set M.

Algorithm. The original set of vertices V is considered as candidates: K = V.

Until K or M is not empty.

If K is not empty:

1. Remove the first element from the set K.
2. Put it in M.
3. Remove from K, P all vertices adjacent to the current one.

Otherwise.

If P is not empty: output M:

1. Remove the last element from the set M.
2. Return to the line where M consists of the set of vertices that we have left at the current step.
3. Remove the current vertex from the set K.
4. Add the current vertex to the set P.

An important point of this algorithm is that it is designed for an undirected graph. However, it can be applied to the model under consideration by converting a directed graph to an undirected one. This is done by introducing the fact that an edge is constructed between two vertices only if these nodes are closely connected, (there is an incoming and outgoing arc).

5.2. Open Shortest Path Algorithm

This algorithm is designed to divide the original graph into K clusters (the number of K is set in advance) (Zaitseva, 2019; Zaitseva, et al., 2019a, 2019b, 2019c). The essence of the algorithm is that first, we find the shortest open path, and then we remove the (K-1) edge with the maximum weight. A path is called the shortest and non-closed path if the total value of the edge weights is minimal and there are no cycles.

Algorithm.

1. Determine a pair of vertices with the minimum weight and connect them with an edg.
2. While there are isolated points in the sample:
2.1 Find an isolated point with the smallest distance to a non-isolated.
2.2 Connecting them with an edge.
3. Remove (K-1) from the longest edges.

However, in our case, it is necessary to look not at the minimum weights of the edges, but at the maximum. In this regard, the algorithm can be considered from the other side:

1. Determine a pair of vertices with the maximum weight and connect them with an edge.
2. While there are isolated points in the sample:
2.1. Find the isolated point with the greatest distance to the non-isolated.
2.2. We connect them with an edge.
3. remove the K-1 from the shortest edges.

Thus, we get K clusters that can be considered as communities.

5.3. Minimum Spanning Tree Search Algorithm

A minimal covering tree is a subgraph of a graph consisting of the same set of vertices and such a set of edges that there is a route between any two vertices and there are no cycles (Malafeyev et al., 2020). There are Kruskal and Prim algorithms for finding such a tree. They are designed to work with an undirected graph. For the oriented case, there is an algorithm called the "two Chinese algorithm". However, it was Kruskal's algorithm that was chosen with the reduction of a directed graph to an undirected one. This is done due to the fact that an edge is built between two vertices, the weight of which is the total value of the incoming and outgoing edges (if one of them is absent, we take 0). A feature of Kruskal's algorithm is the fact that there is no restriction on the fact that the edge weights must be strictly positive. In this regard, the algorithm can be reduced to finding the maximum spanning tree.

Algorithm:

1. Change the weight of each edge to a negative value.
2. Forming a list of edges sorted in ascending order.

3. Iterate over the elements of the list.

3.1. Check: if we add the current edge to the tree, whether a cycle is formed. If not- add, otherwise- skip.

4. In the constructed tree, remove the (K-1) edge with the minimum weight.

As a result of this algorithm, K clusters are obtained.

5.4. Graph kernel search algorithm

Interesting in the case of identifying hidden communities is the concept of the core of the graph (Malafeyev et al., 2019b, 2019c). First, it is necessary to introduce such concepts as internally independent and externally independent set of graph vertices. An internally stable set of vertices is a subset of a set of vertices, each element of which is not adjacent to any other element. An outwardly stable set of vertices is a set in which either any vertex is included in this set, or a vertex is not included in this set, but there is an arc from this vertex to this set.

The core of a graph is a subset of the set of vertices of the original graph, the nodes of which are both externally stable and internally stable. The Magu algorithm is used to search for these sets.

Magu's algorithm for finding an externally stable set of vertices:

1. Composing the adjacency matrix.
2. On the main diagonal we put down the units.
3. Disjunctions are written out for each line.
4. We reduce the obtained factors (reduce to DNF).
5. All vertices included in the elementary conjunction form a set of external stability.

Magu's algorithm for finding an internally stable set of vertices:

1. Composing the adjacency matrix.
2. From the matrix we write out all the paired disjunctions.
3. We reduce the obtained factors (reduce to DNF).
4. Elements that are lacking in each conjunction are included in the set of internal stability.

Graph kernel search algorithm:

1. Find an internally stable set of vertices.
2. Find an externally stable set of vertices.
3. Find the intersection of the resulting sets.

Thus, using this algorithm, it turns out to select such a set of vertices that can be designated as a community. The case is possible when the kernel of the graph does not exist, or there are several of them.

Let's consider an example. Let's say we have 3 agents (n = 3), 3 algorithms (m = 3), and the following utility Table 1, 2 and 3.

Table 1. Organization weights for algorithms

	Algorithm 1	Algorithm 2	Algorithm 3
Agent 1	2	6	7
Agent 2	1	7	2
Agent 3	3	2	9

Consider the possible options for the distribution of algorithms.

Table 2. Variants distribution of algorithms by agents

Xi	Agent 1	Agent 2	Agent 3
1	A1	A2	A3
2	A1	A3	A2
3	A2	A3	A1
4	A3	A2	A1
5	A3	A1	A2
6	A2	A1	A3

Let's form a matrix of winnings.

Table 3. Values of winnings for distribution participants

Xi	Agent 1	Agent 2	Agent 3
1	H1(x1) = 2	H2(x1) = 7	H3(x1) = 9
2	H1(x2) = 2	H2(x2) = 2	H3(x2) = 2
3	H1(x3) = 6	H2(x3) = 2	H3(x3) = 7
4	H1(x4) = 7	H2(x4) = 7	H3(x4) = 7
5	H1(x5) = 7	H2(x5) = 1	H3(x5) = 2
6	H1(x6) = 6	H2(x6) = 1	H3(x6) = 9

Let us find for each column an element of the ideal vector M (maximum value): $M_1 = 7$, $M_2 = 7$, $M_3 = 9$ M = (7, 7, 9). Let's calculate the deviation values $M_k - H_k(x)$.

Table 4. Values of the deviation of the gain from the maximum

Xi	**Agent 1**	**Agent 2**	**Agent 3**
1	7-2 = 5	0	0
2	7-2 = 5	5	7
3	7 – 6 = 1	5	2
4	7 – 7 = 0	0	2
5	7 – 7 = 0	6	7
6	7 – 6 = 1	6	0

According to this table, it determines the maximum deviation and from the obtained values we select the minimum. It will be 2, that is, the optimal distribution is x* = x4. Compromise set: {A3, A2, A1} - 1st organization chooses 3rd algorithm, second - 2nd algorithm and 3rd - 1st algorithm, thus we can say that they have made a compromise. The case is possible when there are several optimal solutions, namely

$$x^* = \{x_1^*, \dots, x_q^*\}$$

In this case, the same algorithm is applied, but to the current options by redefining the utility.

6. Findings

Thus, using the game theory, we can not only apply the listed algorithms to identify hidden communities, but also distribute them among organizations so that they receive the maximum profit and minimum losses in relation to other agents.

7. Conclusion

At the moment, new algorithms for identifying hidden communities have been considered, an algorithm for finding the graph kernel has been implemented and an optimization problem has been formulated, which takes into account these algorithms and their results for further use. Further, it is planned to implement the remaining algorithms and provide the results obtained for research by experts.

References

Agarkova, L., Gurnovich, T., Filonich, V., Shmatko, S., & Podkolzina, I. (2016). Priority directions of development of innovation education cluster in the regional agro-industrial complex. *International Journal of Economics and Financial Issues*, *6*(2), 718-727.

Al-Qurishi, M., Alrakhami, M., Alamri, A., & Sybil, M. A. (2017). Defense techniques in online social networks: A Survey. *IEEE Access,* 99, 12-24.

Bogoviz, A. V., Taranov, P. M., & Shuvaev, A. V. (2018). Innovational tools for provision of food security through state support for the agro-industrial complex in the conditions of digital economy. *Advances in Intelligent Systems and Computing*, *622*, 659-665.

Bondarenko, L. A., Zubov, A. V., Zubova, A. F., Zubov, S. V., & Orlov, V. B. (2015). Stability of quasilinear dynamic systems with after effect. *Biosciences Biotechnology Research Asia*, *12*(1), 779-788.

Freeman, L. C., White, D. R., & Romney, A. K. (1989). *Research methods in social network analysis*. VA George Mason University.

Ivanyukovich, G. A., Malafeyev, O. A., Zaitseva, I. V., Zakharov, V. V., & Zakharova, N. I. (2020). To the evaluation of the parameters of the regression equation between the radiometric and geological testing. *Journal of Physics: Conference Series*, *1515*(3), 032079.

Jebabli, M., Cherifi, H., Cherifi, C., & Hamouda, A. (2015). User and group networks on YouTube: A comparative analysis. ACS. *12th International Conference of Computer Systems and Applications (AICCSA).*

Kostikova, A. V., Tereliansky, P. V., Shuvaev, A. V., Parakhina, V. N., & Timoshenko, P. N. (2016). Expert fuzzy modeling of dynamic properties of complex systems. *ARPN Journal of Engineering and Applied Sciences, 11*(17), 10222-10230.

Kuznetsov, S. Y., Tereliansky, P. V., Shuvaev, A. V., Natsubidze, A. S., & Vasilyev, I. A. (2016). Analysis of innovative solutions based on combinatorial approaches. *ARPN Journal of Engineering and Applied Sciences, 11*(17), 10601-10608.

Malafeyev, O. A., Ivanyukovich, G. A., Redinskikh, N. D., Zaitseva, I. V., Pichugin, Y. A., & Shulga, A. A. (2020). Estimation of the Regression Equation Parameters X-Ray Radiometric and Geological Testing on Deposits of Rare and Precious Metals. *IOP Conference Series: Earth and Environmental Science, 459*(3), 032073.

Malafeyev, O., Redinskikh, N., Zaitseva, I., Smirnova, T., & Kolesov, D. (2019a). Model of information multi-stage process in structured system. *AIP Conference Proceedings, 2186*, 170013.

Malafeyev, O., Zaitseva, I., Onishenko, V., Petrova, V., & Kirjanen, A. (2019b). Optimal location problem in the transportation network as an investment project: A numerical method. *AIP Conference Proceedings, 2116*, 450058.

Malafeyev, O., Zaitseva, I., Parfenov, A., Strekopytova, M., & Strekopytov, S. (2019c). Game-theoretical model of cooperation between producers. *AIP Conference Proceedings, 2116*, 450059.

Zaitseva, I. (2019). Numerical method of distribution of labor resources by game-theoretic model. *AIP Conference Proceedings, 2116*, 450057.

Zaitseva, I., Malafeyev, O., Dolgopolova, A., Zhukova, V., & Vorokhobina, Y. (2019a). Numerical method for computing equilibria in economic system models with labor force. *AIP Conference Proceedings, 2116*, 450060.

Zaitseva, I., Malafeyev, O., Poddubnaya, N., Shlaev, D., & Bogdanova, S. (2019b). Deterministic and stochastic models of labor forces dynamics in the industry with pure competition. *AIP Conference Proceedings, 2186*, 170012.

Zaitseva, I., Malafeyev, O., Zubov, A., Bondarenko, L., & Orlov, V. (2019c). Statistical game-theoretic model of the optimal labor resources distribution. *Journal of Physics: Conference Series, 1399*(3), 033054.

Zaitseva, I. V., Malafeyev, O. A., Kostyukov, K. I., Orlov, V. B., & Yupatova, K. V. (2020). Dynamic programming method in the tasks of optimal labor capital distribution programs. *IOP Conference Series: Earth and Environmental Science, 421*(3), 032024.

Zubov, A. V. (2007). Stabilization of program motion and kinematic trajectories in dynamic systems in case of systems of direct and indirect control. *Automation and Remote Control, 68*(3), 386-398.