Research report
GT-NDNetw: Game theoretical approaches for the critical node detection problem in social and economic networks

PN-III-P1-1.1-TE-2019-1633, 56/2020

August 30, 2022

Phase 1

Objective 1. Development of new algorithms for the CND problem based on game theory

- **A 1.1.** Documentation of existing algorithms in the literature; (1.1.2,1.2.3)
- **A 1.2.** Documentation of existing connectivity metrics σ; (1.1.1,1.2.1,1.3)
- **A 1.3.** Documentation of viral networks, financial networks; (1.1.3)
- **A 1.4.** Development of new algorithms based on non-cooperative game equilibria and on cooperative games;(1.4.1,1.5)
- **A 1.5.** Comparison with existing state-of-the-art methods on benchmark networks; (1.1.2,1.2.3,1.3)
- **A 1.6.** Application on viral and economic networks, and analysis of the obtained critical nodes; (1.1.3)

Phase 2

Objective 2. Extension of the CND problem to hypergraphs and multilayer networks (first part)

- **A 2.1.** Documentation of existing connectivity metrics for hypergraphs and multilayer networks; (2.1,3.1,2.6)
- **A 2.2.** Construction of hypergraph models and multilayer network models; (2.2)
- **A 2.3.** Development of suitable connectivity metrics; (2.2,3.2)

Phase 3

Objective 2. Extension of the CND problem to hypergraphs and multilayer networks (second part)

- **A 2.4.** Development of new algorithms; (2.3,2.6,3.3)
- **A 2.5.** Validation of algorithms on different networks; (2.4,3.3)
- **A 2.6.** Analysis of the critical nodes and trends with the developed algorithms in the new network models; (2.5,3.3.3)

Deliverables:

- research report;
- published articles:
  1. Gaskó, N., Képes, T., Suciu, M., & Lung, R. I. (2021, September). Critical Node Detection for Maximization of Connected Components: An Extremal Optimization Approach. In International Workshop on Soft Computing Models in Industrial and Environmental Applications (pp. 502-511). Springer, Cham. (ISI Proceeding)
  2. Suciu, M. A., Gaskó, N., Képes, T., & Lung, R. I. (2021, September). A Simple Genetic Algorithm for the Critical Node Detection Problem. In International Conference on Hybrid Artificial Intelligence Systems (pp. 124-133). Springer, Cham. (ISI Proceeding)
  3. Béczi, E., & Gaskó, N. (2021). Approaching the bi-objective critical node detection problem with a smart initialization-based evolutionary algorithm. PeerJ Computer Science, 7, e750 (ISI, Q1)
  4. Gaskó, N., Suciu, M., Lung, R. I., & Képes, T. (2022). An Evolutionary Approach for Critical Node Detection in Hypergraphs. A Case Study of an Inflation Economic Network. In International Conference on Intelligent Systems Design and Applications (pp. 1110-1117). Springer, Cham. (ISI Proceeding)

5. Képes, T., Gaskó, N., & Vekov, G. (2022). The Combined Critical Node and Edge Detection Problem. An Evolutionary Approach. In International Conference on Parallel Problem Solving from Nature (pp. 324-338). Springer, Cham. (ISI Proceeding)

- accepted articles:
  1. Gaskó, N., Képes, T., Suciu, M., &Lung, R. I. An Extremal Optimization Approach to the Pairwise Connectivity Critical Node Detection Problem, International Joint Conferences HAIS-SOCO-CISIS-ICEUTE & STARTUP OLÉ, accepted. (ISI Proceeding)
- submitted articles:
  1. Kis, N., Gaskó, N. Critical node detection in multilayer networks, submitted to Expert Systems with Applications
  2. Képes, T. The Critical Node Detection Problem in Hypergraphs Using Weighted Node Degree Centrality, submitted to PeerJ Computer Science
  3. Identification of influential nodes with Shapley Influence Maximization Extremal Optimization Algorithm, submitted to Applied Soft Computing
- web page: www.cs.ubbcluj.ro/˜gaskonomi/GT-NDNetw

Implementation

The main contributions of the project are presented in what follows.

1 Critical node detection problem

1.1 Critical node problem - kMaxComp

In the general formulation of the critical node detection problem we seek to find nodes which are more important than others with respect to a predefined measure. Several graph properties were studied as measures, e.g. the pairwise connectivity which needs to be minimized by deleting k nodes (which is one of the most popular), or minimizing the largest component size by deleting k nodes. In [4] three versions of the CNDP are studied: minimizing the largest components size, minimizing the pairwise connectivity and maximizing the number of connected components.

Firstly, we focus on the problem introduced in [74, 73], which is also described in [4]. The problem consists in removing k nodes such that the number of remaining components to be maximal (we denote this problem *kMaxComp*). Formally, if S denotes the set of deleted nodes, and $\mathcal{H}(G[V \setminus S])$ denotes the set of components of graph G without the set of nodes S, the *kMaxComp* problem consists in:

$$\max_{S \subset V} |\mathcal{H}(G[V \setminus S])|,$$
$$\text{such that } |S| \leq k,$$

where $|A|$ denotes the cardinality of set A.

As the CDNP was proved to be NP-hard for several connectivity measures [74] different solving methods has to be proposed. However, *kMaxComp* did not get very much attention. In [74] a Mixed Integer linear programming approach is proposed, and we find a a general integer programming framework in [82]. For a special class of graphs (trees and series-parallel graphs) a dynamic programming approach is designed [73]. In [4] a genetic algorithm is described to solve the problem and two greedy algorithms are presented.

1.1.1 Noisy Extremal Optimization

Extremal optimization (EO) [14, 15] is a simple and powerful combinatorial optimization algorithm which was successfully adapted for different practical problems, e.g. graph partitioning [13], load balancing problem [23]. A variant of EO called NoisyEO [59] was used successfully for the community detection problem.

In the standard variant of the EO two individuals are used during the search: s and s_{best}; s_{best} preserves the best solution found so far by s based on an overall fitness $f()$. EO individuals are represented as composed of several components that can be evaluated separately. The standard EO maximizes each component of a potential solution by randomly replacing the one with the worst fitness. The outline of the standard EO is presented in Algorithm 1.

In [59] NoisyEO is presented as a variant of EO that proposes the use of a network shifting mechanism to induce diversity in the search. We employ the same mechanism to escape local optima for the CDNP problem. The network shift consists in randomly deleting edges in the network with a probability p_{shift} whenever the search stagnates. The search takes place for a number of generations G on the shifted network, moving the solution away

Algorithm 1 Standard EO

Initialize s at random;
$s_{best} := s$; //s_{best} preserves the best solution found so far
while a termination condition is not met **do**
  find i_{min} component in s with the smallest fitness value
  randomly reassign $s_{i_{min}}$ in s;
  if $f(s) > f(s_{best})$ **then**
    $s_{best} := s$
  end if
end while

Algorithm 2 *CN-EO* algorithm

Parameters:

- Probability of shift - p_{shift};
- Number of generations between switching networks - G;
- Total number of generations - $NrGen$;

Randomly initialize (s, s_{best});
repeat
  if s_{best} does not change in G generations and there is no noise **then**
    Induce noise with probability p_{shift};* ;
    Reinitialize s_{best} with the current s value
  end if
  if There has been noise for G generations **then**
    Return to the original network
  end if
  Perform search using the current network (CN-EO(s, s_{best}) iteration);
until Maximum number of generations;
Return s_{best} with highest ***fitness*** achieved on a non-noisy network.

* Modify network by randomly deleting edges with probability p_{shift}

from a local optimum. We call this approach Critical Nodes - EO (CN-EO). The outline of CN-EO is presented in Alg. 2. A step of the CN-EO is outlined in Alg. 3.

Encoding Individual s is represented as a vector of integers of size k representing the critical nodes. Values are from the range 1 and the number of nodes in the graph.

Fitness function Within CN-EO there are two fitness functions used: one to evaluate individual s and a different one for evaluating each component of s. The overall fitness value of $s = (s_1, \ldots, s_k)$ is computed as the number of components of the graph, after removing the k nodes:

$$f(s) = |\mathcal{H}(G[V \setminus \{s_1, \ldots, s_k\}])| \tag{1}$$

The fitness value of a node i in s is computed as its marginal contribution to $f(s)$:

$$f_i(s) = f(s) - f(s \setminus i), \tag{2}$$

where $s \setminus i$ denotes the set of nodes in s without node i.

Network shift The network shift is used to induce diversity in the search by modifying the search space instead of the EO individuals. After a number of generations G the EO search stagnates (there in no change in s_{best}) the network is modified by removing edges with a probability p_{shift}. To preserve some information, the search of s continues, but s_{best} is randomly reinitialized in order to be easily replaced by the new values discovered by s. The search on the modified network takes place for G generations, after which it is resumed on the original network.

1.1.2 Numerical experiments

Parameter settings CN-EO has only three specific parameters: MaxGen is set to 5000, G is set to 10 and p_{shift} to 0.01. The effect of varying these parameters is expected to be similar to that found in [59] and is not a subject of this study.

Algorithm 3 CN-EO(s, s_{best}) iteration

For current configuration s evaluate $f_i(s)$, $i \in \{1, \ldots, k\}$.
Find the node with the worst fitness and replace it randmly with another one;
if $(f(s) > f(s_{best}))$ **then**
  set $s_{best} := s$.
end if

Benchmarks A set of synthetic benchmarks [1] for the CNDP problem were proposed in [80]. The benchmark set contains three different type of graphs: Barabási-Albert (BA), Erdős-Rényi (ER) and Forest-fire (FF) graphs. BA graphs are scale free networks, ER graphs are random networks, FF graphs simulate how fire spreads through a forest.

Table 1 describes basic network measures of the used benchmarks: number of nodes ($|V|$), number of edges ($|E|$), average degree ($\langle d \rangle$), density of the graph (ρ), and average path length (l_G). Table 2 describes the set of real networks used for numerical experiments, with the same measures as in the case of the synthetic networks. Real networks are from different research areas, e.g biological networks (Ecoli, HumanDis).

Table 1: Synthetic benchmark test graphs and basic properties.

Graph	$\|V\|$	$\|E\|$	$\langle d \rangle$	ρ	l_G
BA500	500	499	1.996	0.004	5.663
BA1000	1000	999	1.998	0.002	6.045
BA2500	2500	2499	1.999	0.001	6.901
BA5000	5000	4999	2.000	0.000	8.380
ER235	235	350	2.979	0.013	5.338
ER466	466	700	3.004	0.006	5.973
ER941	941	1400	2.976	0.003	6.558
ER2344	2344	3500	2.986	0.001	7.516
FF250	250	514	4.112	0.017	4.816
FF500	500	828	3.312	0.007	6.026
FF1000	1000	1817	3.634	0.004	6.173
FF2000	2000	3413	3.413	0.002	7.587

Table 2: Real graphs and basic properties.

Graph	$\|V\|$	$\|E\|$	$\langle d \rangle$	ρ	l_G	Ref.
Bovine	121	190	3.140	0.026	2.861	[69]
Circuit	252	399	3.167	0.012	5.806	[62]
EColi	328	456	2.780	0.008	4.834	[85]
USAir97	332	2126	12.807	0.038	2.738	[70]
HumanDis	516	1188	4.605	0.008	6.509	[38]
EUFlights	1191	31610	53.081	0.044	2.622	[66]

Comparison with other methods For comparisons we use three algorithms described in [4]: a greedy algorithm (G_1) based on node deletion from the candidate critical node set, another greedy algorithm (G_2), based on the node addition to the candidate critical node set and a genetic algorithm from an evolutionary algorithm framework using greedy rules (GA). The proposed genetic algorithm uses problem specific variation operators and it is combined with a local search mechanism.

Table 3 presents obtained results for synthetic and real world datasets. Mean value, standard deviation minimum and maximum value are reported over ten independent runs. We also indicate the best known results obtained in [4] and the corresponding method. It can be observed that CN-EO finds the maximum number of connected components when compared to the best known results from the literature. On average CN-EO is better than the best known results for seven datasets.

Figure 1 depicts the smallest test network, the bovine network, which represents protein interactions, and the obtained critical nodes by CN-EO. It can be seen that CN-EO identifies the nodes that, if removed, generated the maximum number of connected components in the network. We find that in this case they are also the nodes with the highest degree, but this is not always the case for all networks. We notice that the problem has multiple solutions that lead to the same maximal number of connected components for $k = 3$.

Table 3: CN-EO results. Best known results indicate the best value reported by other methods as well as the method reporting it.

		CN-EO				Best known result
Graph	k	Mean	Std.dev.	Min	Max	(from literature)
BA500	50	**314,8**	1,75	313	318	313 (GA, G_1, G_2)
BA1000	75	**594,5**	5,46	585	604	590 (GA, G_1, G_2)
BA2500	100	1091,3	37,74	1023	1146	**1129** (GA, G_1, G_2)
ER235	50	66,7	1,83	65	71	**68** (GA)
ER466	80	108,9	3,45	102	114	**110** (GA)
ER941	140	194,4	7,73	184	207	**206** (GA)
FF250	50	**93**	1,15	92	95	92 (GA, G_1, G_2)
FF500	110	201,3	36,02	99	217	**215** (GA)
FF1000	150	322,8	13,50	298	344	**340** (GA)
Bovine	3	**77,6**	0,70	77	79	77 (GA, G_1, G_2)
Circuit	25	**31,4**	2,95	28	36	31 (GA)
EUFlights	119	60,5	76,02	11	206	**211** (GA, G_2)
Ecoli	15	**171,8**	2,10	169	175	169 (GA, G_1)
USair97	33	40,6	44,84	3	106	**104** (GA, G_2)
HumanDis	53	**152,7**	3,89	148	158	148 (GA, G_2)

1.1.3 A stock market network application

There is a grown interest in constructing and and analyzing economic networks. Nodes can represent banks, directors, investigators, depending on the studied problem [27]. One of the first applications is the stock market analysis from a network perspective [21]. In [32] the Chinese stock market is analysed as a directed network with an influential model. We analyse the financial stock market network described in [53] from the critical nodes perspective. The network is obtained from the analysis of temporal correlations among the time-series of 62 stocks in the New York Exchange Market from the period 2012-2014. We use an unweighted version of the graph where we deleted all edges with weight smaller than 1.2 (edges mean the distance calculated based on the Pearson coefficient). The obtained a network with 62 nodes and 618 edges.

We obtained by NoisyEO the first 3, 4, 5, 6, 7 and 8 most critical nodes. The results are analysed also from a network measure perspective: degree of the nodes, betweenness and closeness centralities were obtained. We indicate the rank of the nodes ordered decreasing. The novelty of searching for critical nodes in a stock market example consists in finding important nodes, other ones as based on the above mentioned traditional measures. As seen in Table 4 based on the critical nodes we can set up another order of importance of the certain stock.

Table 4: Critical nodes and network centrality measures in the New York Exchange Market

Node no.	Stock Name	Critical node (k)	Degree (rank)	Closeness Centrality (rank)	Betweenness centrality (rank)
52	S	k=3	61 (1)	1 (1)	301.62 (1)
51	SO	k=3	48 (3-4)	0.82 (3-4)	164.151 (2)
25	ETR	k=3	48 (3-4)	0.82 (3-4)	140.60 (3)
6	AVP	k=4	51 (2)	0.85 (2)	127.76 (4)
54	UIS	k=5	44 (5-6)	0.78 (5-6)	68.25 (6)
59	WMT	k=6	44 (5-6)	0.78 (5-6)	82.50 (5)
5	ARC	k=7	41 (7)	0.75 (7)	52.61 (7)
1	AEP	k=8	38 (9)	0.72 (9)	52 (8)

1.2 Critical node problem - pairwise connectivity

The most studied version of the k-vertex-CNDP is the minimization of pairwise connectivity, called the Critical Node Problem. Given a graph $G = (V, E)$, where V denotes the set of nodes, E denotes the set of edges and a positive integer k, the problem consists of finding a subset of the vertices $S \subseteq V$ at most k nodes, such that the deletion of these nodes minimizes the pairwise connectivity in the remaining graph $G = (V \setminus S, E)$. Pairwise connectivity can be expressed as [52]:

$$f(S) = \sum_{C_i \in G[V \setminus S]} \frac{\delta_i(\delta_i - 1)}{2}, \tag{3}$$

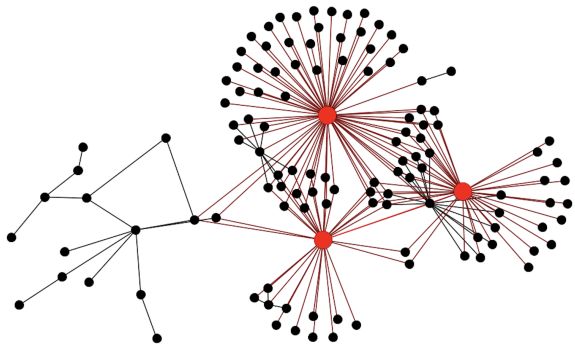


Figure 1: Bovine network (obtained critical nodes are marked with red, $k = 3$)

where C_i, $i = 1 \dots k$ are the connected components of the remaining graph $G = (V \setminus S, E)$, δ_i represents the size of the component C_i.

This variant is an NP-complete problem on general graphs [5]. As solving methods, an exact approach using Integer Linear Programming was proposed [5], in [82] a mixed integer programming formulation is described for several variants of the CNDP. As stochastic approaches, we mention a Greedy Randomized Adaptive Search Procedure [68], a Memetic Algorithm [88], and an evolutionary framework [4].

Another variant based on the pairwise connectivity is the bi-objective critical node detection problem, proposed in [81]. In [7] a cardinality-constrained variant is proposed. [83] proposed and formalizes the distance-based critical node detection problem.

1.2.1 Extremal Optimization

To escape from local optima another variant of the EO is used, the NoisyEO [59]. It uses a network shifting mechanism to induce diversity in the search. This mechanism was used successfully for the maximization of connected components, Critical Nodes -EO (CN-EO) in [34].

Improvement of EO In the standard version of EO the worst component (in our case the node) is randomly replaced. A natural way to improve this step is to guide the replacement to an "important" node. More central nodes (based on three network measures) will be chosen as new components, that is nodes which have their centrality above the average centrality value of the network will be chosen.

The following centrality measures are used:

- degree centrality - measures the degree of each node:

$$C_D(v) = \frac{deg(v)}{|V| - 1},$$

  where $deg(v)$ is the degree of node v and $|V|$ is the number of nodes.

- closeness centrality - captures the average shortest distances from a node to all other nodes:

$$C_C(v) = \frac{1}{\sum_{w \neq v} d(w, v)},$$

  where $d(w, v)$ denotes the shortest distance between node v and w.

- betweenness centrality - measures how a certain node lies on the shortest paths of other nodes:

$$C_B(v) = \sum_{s \neq v \neq t} \frac{g_{st}(v)}{g_{st}},$$

  where $s, t \in V$, g_{st} is the total number of shortest paths, $g_{st}(v)$ is the number of shortest paths, where v lies.

For NEO the same centrality measures are used, but the selection of the new node is based on a p_{rand} probability: with a probability of p_{rand} the nodes with the centrality value over the average value of the network are selected, in the rest a random value is chosen.

1.2.2 Numerical experiments

Parameter setting For each variant of the EO algorithm the maximum number of generations (iterations) is set to 5000. The algorithm does not have other parameters that need setting. In the case of the NEO MaxGen is set to 5000, G is set to 10, p_{shift} to 0.01, and p_{rand} to 0.8.

Benchmarks For the numerical experiments we use the benchmarks described in Section 1.1.2.

1.2.3 Comparisons with other methods

[4] presents three algorithms for the CNDP: two variants of a greedy algorithm (G_1 and G_2), one of them is based on node deletion and the other one is based on node addition and a third algorithm is a genetic algorithm from an evolutionary algorithm framework using greedy rules (GA). To the four variants of the EO algorithm, we refer them as: $R-EO$ the base algorithm, C_D-EO is the degree based approach, C_B-EO is the approach that uses betweenness and C_C-EO is the closeness based algorithm. Similarly, the four variants of the NEO algorithm are denoted by $R-NEO$, C_D-NEO, C_B-NEO, respectively C_C-NEO.

Table 5 presents the results obtained for the synthetic and real-world networks. Although we present best known results from the literature, the direct comparison with our methods is not fair as we report the averages and standard deviation over ten independent runs while the results reported by the literature are probably best obtained values (the authors of the study do not report average values, number of independent runs). As a general conclusion we can establish that using different centrality measures speeds up the convergence of the algorithm to the solutions, but it does not ensure better results in all cases. Another important conclusion is, that we cannot use only best nodes regarding a certain measure, critical nodes can appear between other nodes, as well. For example nodes with highest closeness centrality are not as good as randomly selected nodes.

Table 5: Results for different real and synthetic networks (mean and standard deviation for ten independent runs) for the EO and NEO variants. The column "Best known results" presents the best results found in the literature according to [4], the authors do not specify if these values represent means or minimum values over multiple runs

Network	k	$R-EO$ $R-NEO$	C_B-EO C_B-NEO	C_C-EO C_C-NEO	C_D-EO C_D-NEO	Best known result
BA500	50	195.00(±0.00) 201.00(±3.07)	199.00(±0.00) 195.80(±.98)	516.80(±26.84) 211.40(±11.82)	195.00(±0.00) 195.00(±.00)	195 (GA)
BA1000	75	563.30(±8.69) 617.20(±76.19)	558.60(±0.70) 558.50(±.50)	1395.50(±56.46) 708.20(±56.20)	558.70(±0.82) 559.20(±.87)	558 (GA)
ER250	50	324.70(±17.98) 322.00(±13.98)	374.30(±32.10) 301.80(±4.12)	412.40(±45.51) 308.20(±8.45)	340.60(±30.23) 302.90(±10.72)	295 (GA)
ER500	80	2068.60(±325.52) 2218.67(±160.60)	2032.10(±123.75) 1748.40(±75.01)	2535.70(±301.49) 1845.70(±100.09)	1877.30(±112.17) 1674.90(±44.42)	1560 (GA)
FF250	50	197.60(±4.03) 199.50(±0.71)	248.40(±9.61) 195.00(±2.05)	286.40(±15.26) 195.50(±2.16)	273.80(±21.67) 196.00(±2.10)	194 (GA)
FF500	110	262.10(±4.12) 277.50(±12.77)	416.10(±28.77) 282.40(±7.34)	455.40(±30.74) 277.40(±6.36)	276.90(±5.09) 262.90(±1.64)	257 (GA)
Bovine	3	268.00(±0.00) 268.00(±.00)	268.00(±0.00) 268.00(±.00)	268.00(±0.00) 268.00(±.00)	268.00(±0.00) 268.00(±.00)	268 (G_1,G_2,GA)
Circuit	25	2730.20(±428.16) 2317.00(±130.81)	3483.60(±664.97) 2284.00(±111.93)	6461.10(±982.88) 2388.40(±361.39)	4345.80(±2326.52) 2238.40(±97.01)	2099 (G_1,GA)
EColi	15	840.40(±18.13) 837.40(±16.74)	830.30(±21.41) 818.90(±19.71)	919.60(±29.95) 823.20(±21.07)	828.80(±20.45) 827.50(±21.50)	806(G_1,GA)
TrainsR	26	1004.60(±71.30) 937.20(±8.70)	1440.80(±150.75) 964.80(±19.14)	1286.90(±206.00) 948.70(±17.19)	1176.20(±202.76) 943.60(±7.70)	921 (GA)

1.3 Combined critical node and edge detection problem

The critical node and edge detection problem (CNEDP) consists of finding a set $W \subseteq V$ containing k nodes and a set $F \subseteq E$ having l edges in a given graph $G = (V, E)$, which deleted maximally degrades the graph according to a given measure σ. We denote this introduced problem by (k, l)-CNEDP.

In this article we study as a network connectivity measure the pairwise connectivity. The objective function, which needed to be minimized is the following:

$$f(A) = \sum_{C_i \in G[V \setminus A]} \frac{\delta_i(\delta_i - 1)}{2}, \tag{4}$$

where $A \subseteq V$, C_i is the set of connected components in the remaining graph, after the deletion of nodes and edges, and δ_i is the size of the connected component C_i.

1.3.1 Methods

We propose two algorithms to solve the (k, l)-CNEDP a Greedy algorithm and a Genetic Algorithm described in more details in [48].

1.3.2 Experimental set-up

Benchmarks The used synthetic networks are presented in Section 1.1.2. In Table 6 the set of real networks used for numerical experiments is presented, including the source of the network.

Table 6: Real graphs and basic properties.

Graph	$\|V\|$	$\|E\|$	$\langle d \rangle$	ρ	l_G	Ref.
Bovine	121	190	3.140	0.026	2.861	[69]
Circuit	252	399	3.167	0.012	5.806	[62]
Dolphins	62	159	5.1290	0.0841	3.3570	[51]
Ecoli	328	456	2.780	0.008	4.834	[60],[85]
Football	115	613	10.6609	0.0935	2.5082	[37],[51]
Hamsterster	2426	16631	13.7106	0.0057	2.4392	[70]
HumanDis	516	1188	4.605	0.008	6.509	[38]
Karate	34	78	4.5882	0.1390	2.4082	[51],[87]
Zebra	27	111	8.2222	0.3162	1.3590	[51],[76]

The number of critical nodes (k) is 5% of the total nodes, while the number of critical edges (l) is set to 3% of the total number of edges (proportions are set general, to emphasize critical nodes and edges on different type of networks). The maximum generation number for both GA variants was set to 5000.

1.3.3 Results and discussion

Table 7 presents the results obtained from the genetic algorithm (GA_1), genetic algorithm with time-varying mutation (GA_2) and from the greedy algorithm. GA_1 and GA_2 outperformed the greedy algorithm in most cases. Only in the case of the Football network did both algorithms perform in the same way (with standard deviation equal to 0). However, analyzing the results, in the case of the Football network the values for k and l were too small, because there was no change from the initial population in GA_1 and GA_2. The incorporation of time-varying mutation did not significantly improve the results.

Table 7: Results for synthetic and real graphs. Mean values and standard deviation over 10 independent runs are presented. A (*) indicates best result based on a Wilcoxon sign-rank test

Graph	GA_1	GA_2	Greedy
BA500	$650.70 \pm 82.62^*$	835.00 ± 259.57	5722.90 ± 33.20
BA1000	$1947.20 \pm 235.93^*$	$2021.10 \pm 255.62^*$	362555.90 ± 924.58
ER250	$15296.30 \pm 785.33^*$	$14976.70 \pm 623.52^*$	25066.70 ± 156.40
ER500	$61664.60 \pm 1532.08^*$	$62929.20 \pm 2215.60^*$	100357.40 ± 568.82
FF250	$7905.50 \pm 1212.81^*$	$7909.70 \pm 619.53^*$	25508.20 ± 1052.20
FF500	$12303.30 \pm 4858.39^*$	$12342.90 \pm 4493.16^*$	105350.50 ± 4396.26
Bovine	$16.50 \pm 20.66^*$	$29.20 \pm 25.50^*$	1337.10 ± 31.78
Circuit	$274.00 \pm 332.22^*$	$274.70 \pm 417.62^*$	24110.10 ± 1448.70
Dolphins	$1220.40 \pm 14.55^*$	$1230.20 \pm 15.75^*$	1711.00 ± 0.00
Ecoli	$447.60 \pm 465.91^*$	$1338.40 \pm 1467.07^*$	47649.30 ± 576.69
Football	$5995.00 \pm 0.00^*$	$5995.00 \pm 0.00^*$	$5995.00 \pm 0.00^*$
HumanDis	$531.70 \pm 859.40^*$	3927.30 ± 9258.98	113625.90 ± 1521.87
Karate	$355.90 \pm 41.98^*$	$315.00 \pm 10.54^*$	411.10 ± 80.67
Zebra	$165.20 \pm 7.32^*$	$166.40 \pm 8.98^*$	237.00 ± 0.00

Regarding the running time of both methods (GA_1 and greedy), in small networks, as expected, greedy runs faster (e.g. in the case of dolphins network 2.47 ± 0.02 seconds running time has the greedy algorithm, and 183.64 ± 0.48 seconds the GA_1), but in a larger network the GA_1 has better running time (e.g. for the FF500 network the greedy runs in average 1420.66 ± 63.48 seconds and the GA_1 1294.3 ± 25.15 seconds).

1.3.4 Application: new network robustness measure proposal

As an application of critical node and edge detection, we introduce a new network robustness measure. In the literature several robustness measure exist, trying to capture different properties of the networks. For example [26] describes different measures to characterize network robustness: k_v - vertex connectivity - the minimal number of vertices which need to be removed to disconnect the graph, k_e- edge connectivity - the same measure for edges, diameter of the graph (d), average distance ($d-$), average efficiency (E) - considering shortest paths, maximum edge betweenness (b_em), average vertex betweenness (b_v), average edge betweenness (b_e) - these measures considering

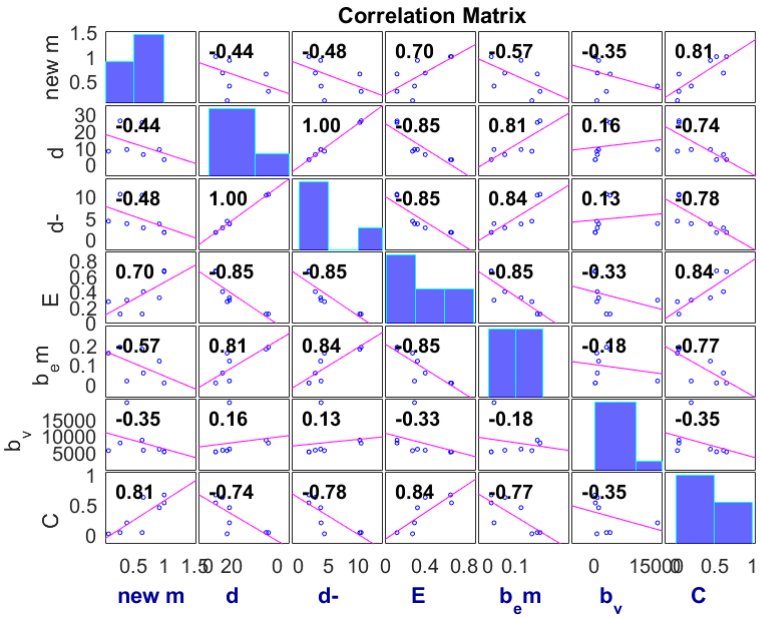


Figure 2: Pearson's correlation coefficients between network robustness mesures

shortest paths. The average clustering coefficient (C) is a proportion of triangles and connected triples. Algebraic connectivity (λ_2) is the second smallest eigenvalue of the Laplacian matrix of G, a number of spanning trees (ϵ) counts the possible different spanning trees of the graph, while effective graph resistance (Kirchhoff index) (R) investigates the graph as a network circuit.

We study several real-world networks from different application fields. The above mentioned network measures are calculated for the studied networks, as presented in Table 8. As we can see, the majority of the indices cannot be used for disconnected networks (in this example, the E-road network is disconnected), this is one of the motivations to introduce the new measure to analyze the network robustness, based on the (k,l)-CNEDP.

Table 8: Network robustness measures for studied networks

Measure	Networks								
	USAir97	E-road	Mouse_cortex	Cat_brain	494-bus	662-bus	Infect-dublin	Infect-hyper	Route_views
k_v	1	0	1	3	1	1	1	1	1
k_e	1	0	1	3	1	1	1	1	1
d	6	∞	8	3	26	25	9	3	9
$d-$	2.73	∞	4.27	1.69	10.47	10.24	3.63	1.65	3.70
E	0.40	0	0.27	0.66	0.11	0.11	0.32	0.67	0.29
$b_e m$	0.06	-	0.16	0.01	0.19	0.18	0.12	0.01	0.02
b_v	618.65	-	506.01	86.38	2827.38	3716.30	947	148.75	15227.74
b_e	70.76	-	369.78	4.84	1180.51	1429.46	110.10	4.77	5586.99
C	0.62	0	0.02	0.66	0.04	0.04	0.45	0.53	0.25
λ_2	0.12	0	0.03	2.88	-4.75	-5.14	0.19	0.99	-25.84
ϵ	3.37e+234	0	6.41e+10	1.45e+81	0	1.004	∞	1.7e+169	∞
R	45538.19	∞	47450.21	262.08	-4.56e+18	1.78e+18	30563.17	527.78	1587664.19
$NE_{k,l}$ [1]	0.66	0.33	0.09	1.00	0.28	0.64	0.92	1	0.39

The introduced measure ($NE_{k,l}$) has the following form:

$$NE_{k,l} = \frac{2 \cdot (k,l)\text{-CNEDP}}{(n-k-1)(n-k-2)}$$

$\frac{(n-k-1)(n-k-2)}{2}$ is the worst possible value of pairwise connectivity, after deleting k nodes, n is the number of nodes in the original network, $NE_{k,l} \in [0,1]$.

In the case of the USAir97 network, for example:

$$NE_{21,6} = \frac{2 \cdot 35264}{(332-21-1)(332-21-2)} = 0.66.$$

The $NE_{k,l}$ can be seen as a measure which based on the number of deletion of nodes and edges quantifies the network robustness. To analyse the results a correlation matrix was built (without the results of the E-road network), the new measure - $NE_{k,l}$ (new m) was compared with d,$d-$,E,$b_e m$,b_v and C. As presented in Figure 2 a weak correlation exists between the new measure and the clustering coefficient (C).

1.4 Critical nodes as influential nodes -cooperative game theoretic approach

The Influence Maximization Problem, connected to a diffusion model, can be described by using the influence function in the following manner.

Influence function Given a graph $G = (V, E)$, a spreading set $S \subseteq V$ and a propagation (diffusion) model D, the influence function $f_D(S)$ represents the expected number of nodes activated by the set S under the model D, where $f_D(S) : 2^V \to \mathbb{R}^+$.

IMP Given a graph $G = (V, E)$, a propagation model D, a number $k \in \mathbb{N}^*$, the influence maximization problem aims to maximize the influence function $f_D(S)$:

$$\max_{S \subseteq V, |S| \leq k} f_D(S),$$

where $|S|$ is the cardinality of the set S.

Propagation, or diffusion models, try to answer the central question of how information spreads in a network [39]. They can be divided in two main classes: deterministic and probabilistic. Examples of deterministic models are the majority threshold model [79], where a node will be active if half of the neighbors is already active, or the unanimous threshold model [19], where the activation depends on the degree of a certain node.

The Independent Cascade Model (ICM) [47] is one of the most widely used diffusion models. ICM uses a probability p to activate nodes starting from a given node set A_0, of size k, of the graph $G = (V, E)$. A_0 is considered the initial current active node set and the aim is to construct set A of activated nodes influenced by A_0. In each step of the cascade, nodes in the current active set B activate their neighbors with probability p. Activated nodes are added to the cascade set A. The cascade stops if no node in the current active set B has activated any neighbor. The size of A, denoted by $\sigma(A_0)$ is returned as the result of the cascade and called the influence set of A_0.

Then the IMP problem based on ICM can be formulated as:

$$\max_{S \subseteq V, |S| \leq k} f_{ICM}(S),$$

where $f_{ICM}(S) = \sigma(S)$.

1.4.1 Proposed method - Shapley Influence Maximization Extremal Optimization (SIM-EO)

In this section we will describe our proposed algorithm named, the Shapely Influence Maximization Extremal Optimization approch, the SIM-EO.

Encoding: Each individual is composed from a set of nodes that represent a potential solution to the influence maximization problem of length k. Each node in this set can be considered a separate component to this solution, allowing us to potentially evaluate contribution on a per-node basis.

Fitness: There are two fitness functions used in the algorithm. Firstly we need to identify the value assigned to each solution, in order to compare our current solution to past results. We use the ICM as a good representation for the influence spread in a network. Secondly we need to identify the contribution for each node to a potential solution so that we can replace the components with extremal value (minimal contribution). A good possibility is the employment of the Shapley value of each node in relation to the total value of the whole solution.

Shapley Value Approximation: The Shapley value proposed in [72] is a popular solution concept used in cooperative game theory used to establish the contribution of each player to the total value of the game. A cooperative coalitional game can be described as a double $\Delta = (N, v)$, where:

- N is the set of players;
- $v : 2^N \to \mathbb{R}$ is the characteristic function, mapping subsets of players to real numbers.

The Shapley value ϕ_i of a player i measures the average contribution of player i to all possible players coalitions and can be computed as:

$$\phi_i(v) = \sum_{S \subseteq N \setminus \{i\}} \frac{|S|!(|N| - |S| - 1)!}{|N|!} (v(S \cup \{i\}) - v(S)) \tag{5}$$

where N is the set of players and $|\cdot|$ denotes the cardinality of the set.

While a typical Shapley value calculation would consider every possible coalition of nodes, it would be too costly, based on performance issues. We use an approximation of the Shapley Value, in which a few distinct orderings are selected of size k, for these nodes we calculate node contribution.

Monte Carlo Network Generation: Usually the tests are conducted on a single network using a standard variation of the Independent Cascade as the propagation model. This can be improved by introducing a Monte Carlo network generation. This works by starting from our initial , from which edges are removed, until a small percent of edges remain significantly improving our calculation speed. We use the probability factor from the Independent Cascade algorithm and introduce it at the point of network creation, creating multiple networks with

the aforementioned method and removing the probabilistic nature of the Cascade, since its randomness is now introduced in the network generation.

SIM-EO outline: The SIM-EO algorithm starts with the Monte Carlo Network Generation. We generate a starting individual with size k representing our potential solution. For each iteration, until a desired number of iteration have passed, we calculate the contribution values for every node and rank them based on the results. We randomly replace a number m of the worst values with new ones (the number of replaced nodes start high and settle at one per iteration). A high value of m (with respect to the number of components of the configuration) emphasizes the exploration capabilities of the algorithm, a small value puts in the foreground the exploitation.

The value of m is adapted using the maximum number of iterations (T_{max}) and the actual iteration number (T):

$$\min\left(\frac{m}{2}, \max\left(1, \frac{m}{10} + \frac{2 \cdot T}{T_{max}} \cdot \left(1 - \frac{m}{10}\right)\right)\right) \tag{6}$$

We evaluate our new result and compare the results with the globally best result. The final result will be the best result at the final iteration of the algorithm.

1.4.2 Numerical experiments

Synthetic networks The used synthetic networks are presented in Section 1.1.2.

Real networks For the numerical experiments we have chosen networks with different properties: directed (cit-DBLP, email-univ, Gnutella) and undirected (PGP, USAir97, hamsterster), different size (from 332 nodes up to 36682), and of different density (from sparse graphs to 0.0085 density) (Table 9).

Table 9: Real networks and basic properties.

Graph	$\|V\|$	$\|E\|$	$\langle d \rangle$	ρ	l_G
Hamsterster	2426	16631	13.71	0.006	3.588
Gnutella	36682	88328	4.815	0.00000	8.113
Email-univ	1133	5451	9	0.0085	3.852
Cit-DBLP	12600	49700	7.000	0.00062	5.427
PGP	10680	24316	4.55	0.00000	7.486
USAir97	332	2126	12	0.038	2.738

1.4.3 Parameter settings

For numerical experiments the following parameter setting were used: $nG = 30$, the number of iteration is set to 10000. We use for k six different values, $10, 20, 30, 40, 50, 60$. For the IC model the p takes two values: 0.01 and 0.03.

1.4.4 Comparison with other methods

For comparisons different type of algorithms were used: as a greedy approach Cost Effective Lazy Forward (CELF) [2] [54] was used, as the main idea exploiting the sub-modularity nature of the cascade function. Two heuristics that chooses nodes in the seed set in descending order of their degree were also used for comparisons: High degree (HighDeg) and Single discount (SDisc) [20].

As two other centrality based methods betweenness (Betw) [31] and PageRank[17] were selected. The first k nodes were selected based on these two measures. Betweenness centrality reflects the measure how a node lies on shortest paths, while PageRank measures the importance of nodes based on the incoming edges.

To explore the stochastic methods, two methods from this class of algorithms are used for comparisons, Influence detection via martingales (IMM)[3] [77], which has two main phases, a sampling phase and the node selection phase. As second method an evolutionary algorithm, a genetic algorithm (GA) [44] was used.

1.4.5 Results and discussion

In summary, in the case of the synthetic networks, for both $p = 1\%$ and $p = 3\%$ SIM-EO performed the best compared with the other seven methods. For $p = 1\%$ from 40 cases the SIM-EO turned out to be the best in all 40 cases, only in one case (BA1000, k=10) other three methods performed in the same way. Remaining by the synthetic networks, for $p = 3\%$ the proposed SIM-EO performed the best in all 40 cases, based on the Wilcoxon sign-rank test in 4 cases were no difference between SIM-EO and other methods.

[2] https://github.com/tsume82/Influence-Maximization, last accessed 7/3/2022
[3] https://sourceforge.net/projects/im-imm/, last accessed 10/03/2022

Regarding the real world networks for $p = 1\%$ from 36 different cases (six real world networks, six different values for k) in 36 SIM-EO performed the best, in 7 cases other methods achieved same performance, as well. In the case of $p = 3\%$ for three networks (Gnutella, email-univ, USAir97) for all 6 cases our method had the best results, totally for 18 cases, from which in 11 cases other methods performed in the same way. By the Hamsterster and pgp networks IMM had the best performance. In the case of the cit-DBLP network the GA algorithm outperformed significantly all methods. The explanation of this phenomenon can be the following: based on [50] the algorithm works well on larger denser graph, which properties are fulfilled in the case of the cit-DBLP network.

1.5 Critical node problem - non-cooperative game theoretic approach

In this case the optimization problem is to maximise the number of connected components of $G[V - S]$, where S represents the set of the deleted nodes.

A non-cooperative game theoretical approach Our method consists in a game theoretical approach.

We consider a game in normal form $\Gamma = (N, S, U)$, where:

- $N = \{1, ..., n\}$ is the set of players,
- s_i is the strategy of player i, $S = S_1 \times ... \times S_n$ is the set of strategy profiles of the game
- $u_i : S \to \mathbb{R}$ is the payoff of player i, and $U = (u_1, \ldots, u_n)$ are the players' utility functions

In our case, the set of players are nodes, the strategy consists in being or not a critical node and the payoff function is the number of connected components in which the graph will be fragmented without the certain node.

Generative relations characterize a certain equilibrium type by using the non-dominance concept. A binary relation τ is defined on S, $\tau = (S, S, G \subset S \times S)$. If we have $s\tau q$ (i.e. $(s, q) \in G$) then we say that s dominates q with respect to relation τ. If for s^*, $\nexists q$ such that $q\tau s^*$, we call s^* non-dominated with respect to relation τ. The set containing all $s^* \in S$ that are non-dominated with respect to relation τ is called the non-dominated set of S with respect to τ.

Relation τ is called *generative* for an equilibrium type of a game (e.g. Nash, strong Nash, etc.) if the set of non-dominated strategy profiles with respect to relation τ equals that set of equilibria of the game.

An easy way to construct a generative relation is to use a quality indicator $t : S \times S \to N$ such that: $s\tau q \Leftrightarrow t(s, q) < t(q, s)$

A generative relation for the Nash equilibrium was introduced in [58]:

Consider two strategy profiles s and q in S. Operator $\tau : S \times S \to \mathbb{N}$ defined as

$$t(s, q) = card\{i \in N | u_i(s) < u_i(q_i, s_{-i}), q_i \neq s_i\}$$

characterizes the NEs of the game.

The relation between the two strategy profiles s and q is:

- either s dominates q ($t(s, q) < t(q, s)$)
- either q dominates s ($t(q, s) < t(s, q)$)
- s and q are indifferent ($t(q, s) = t(s, q)$)

The used algorithm for the Nash equilibrium detection is the Nash Extremal Optimization.

Experimental results Experimental results illustrate the behaviour of EO and NEO by using the following set of networks: yeast[4], BA500, BA1000, ER250. For the numerical experiments, we used the EO and NEO, with random initialization and with a greedy initialization. Numerical results are presented in Table 10. Results indicate the potential of the proposed methods.

2 Critical node detection in hypergraphs

2.1 Introduction

The critical node detection problem (CNDP) [55] is a central topic in graph theory due to its large applicability in various fields, such as immunization [42], network vulnerability [46, 57], economics [6], social network analysis [16], etc. The problem arises from the fact that a node's importance varies in a graph and consists in finding sets

[4]http://archive.ics.uci.edu/ml/datasets/yeast, last accessed 10.10.2020

Table 10: Numerical results of the CNDP with four algorithms: EO, NEO, greedy initialization of EO based on degree distribution, greedy initialization of NEO

Graph	k	EO	NEO	Greedy+EO	Greedy+NEO
yeast	202	843	838	847	846
BA500	50	313	311	313	312
BA1000	75	590	588	590	589
ER250	50	68	65	68	65

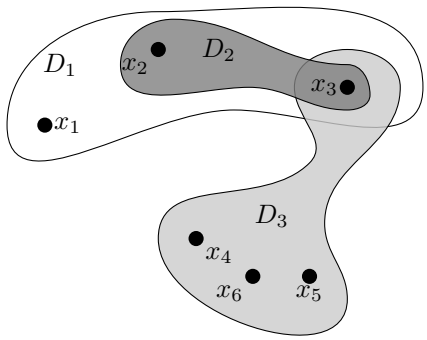


Figure 3: A simple example of hypergraph with six nodes and three hyperedges. If x_3 is deleted, the hypergraph will have three components.

of nodes that, when deleted from the graph, maximize a given network measure, which is usually related to the connectivity of the network.

In a general approach, given a graph $G = (V, E)$, the critical node detection problem consists in finding a set S of k nodes which deleted maximally degrades the graph according to a given measure $\sigma(G)$. Examples of such measures include the minimization of pairwise connectivity, minimizing the size of the largest component, bound the pairwise connectivity by a threshold, maximize the number of connected components, etc [52].

While there are many studies that deal with the CNDP for weighted or unweighted, directed or undirected graphs, less attention has been given to the formulation and adaptation of the problem for hypergraphs. A hypergraph generalizes the concept of graph by considering edges that connect more than two nodes. In this setting there are many possible measures that can be considered in the formulation of the CNDP.

In this report we extend the CNDP for hypergraphs by considering the problem of removing k nodes in order to maximize the number of remaining components. Our goal is to use a genetic algorithm adapted to solve this problem. A macroeconomic inflation hypergraph is constructed and analysed in order to illustrate the applicability of the approach.

2.2 Critical node detection in hypergraphs

A hypergraph [12] is a generalization of a graph, where edges can join not only two, but any number of nodes. In recent years several application possibilities appeared, where hypergraphs can be used with success, like image classification [86], artificial intelligence [29], biology [30].

Formally, a hypergraph is a $\mathcal{H} = (X, \mathcal{D})$ double, where $X = \{x_1, x_2, ..., x_n\}$ is the set of nodes, $\mathcal{D} = \{D_1, D_2, ..., D_m\}$ is a set of the subsets of X, denoting the set of hyperedges. A simple example is depicted in Figure 3. The hypergraph has six nodes, $X = \{x_1, x_2, x_3, x_4, x_5, x_6\}$, $D = \{D_1, D_2, D_3\}$, $D_1 = \{x_1, x_2, x_3\}$, $D_2 = \{x_2, x_3\}$, $D_3 = \{x_3, x_4, x_5, x_6\}$

In the case of the hypergraphs two variants of node deletion exists: strong and weak deletion. Strong deletion means that not only the node x is removed, but also all edges containing that node. In the case of the weak deletion only the node x is removed. A simple example is illustrated in Figure 4.

The critical node detection problem for the hypergraphs can be formulated as follows: given a $\mathcal{H} = (X, \mathcal{D})$, the problem consists in removing weakly k nodes such that the number of remaining components to be maximal. Formally, if S denotes the set of deleted nodes, and $\mathcal{C}(\mathcal{H}[X \setminus S])$ denotes the set of components of graph $\mathcal{H}$ without the set of nodes S, the problem consists in:

$$\max_{S \subset X} |\mathcal{C}(\mathcal{H}[X \setminus S])|,$$

$$\text{such that } |S| \leq k,$$

where $|A|$ denotes the cardinality of set A.

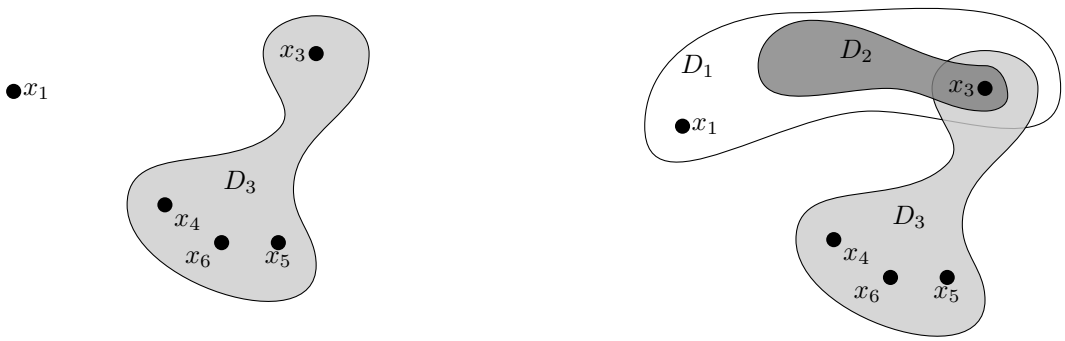


Figure 4: Strong detection of x_2 (left) and weak deletion of x_2 (right) from the hypergraph presented in Fig. 3

Table 11: Hyp-GA Genetic operators

Mutation:	flip-bit mutation
Crossover:	two point crossover
Selection:	tournament selection (tournament size 3)

Example Considering the hypergraph presented in Figure 3, if $k = 1$ x_3 need to be deleted to maximize the number of remaining components.

2.3 Hyp-GA

Evolutionary algorithms are powerful optimization tools to solve computationally hard problems. In the next we will use a simple genetic algorithm to solve the critical node detection problem. For the hypergraph we use a clique representation, which means that a hyperedge is transformed to 'traditional' edge (e.g. if a hyperedge has 4 nodes, there will be an edge between each node). Table 11 presents the elements of the genetic algorithm, while specific settings are described in what follows.

Encoding An individual x is encoded as a bit string of size equal to the size of the network. Each node is represented in x either as 0 or 1, the value 1 indicates that the corresponding node is included in the set of critical nodes encoded by x. We denote this set by S_x.

Fitness function To evaluate the fitness of an individual we remove all nodes in the critical set it encodes, i.e. nodes with value 1 and count the number of connected components in the remaining graph. We have:

$$f(x) = |\mathcal{H}(G[V \setminus S_x])|. \tag{7}$$

Constraint handling To maintain the size of S_x less that the threshold k we remove excess nodes by using the marginal contribution of a node to the fitness of the individual. The marginal contribution of a node i from the critical set S_x of x, denoted by $u_i(x)$ is:

$$u_i(x) = f(x) - |\mathcal{H}(G[V \setminus \{S_x \setminus \{i\}\}])|,$$

where $f(x)$ is the fitness defined in eq. (7). $u_i(x)$ measures the difference between the fitness of x and the fitness of x when removing node i from S_x. Nodes with lowest marginal contributions are removed from S_x, such that only k nodes remain in the set before each individual evaluation.

2.4 Numerical experiments

For numerical experiments we tested the proposed method on synthetic data and on a new macroeconomic dataset.

Benchmarks In order to construct hypergraphs with known properties we started from benchmarks constructed for the overlapping community detection problem. LFR benchamrks[5] are baseline test graphs for community detection algorithms. The communities in the graphs show similarity to a hyperedge in a hypergraph, as communities

[5]https://www.santofortunato.net/resources, last accessed 1/9/2021

Algorithm 4 Hyp-GA outline

Initialize population P of size p_{size} at random.
for i=1 to MaxGen **do**
  $P^{(i)}$ = Select p_{size} individuals for variation;
  Offsping= variation operators on $P^{(i)}$;
  Correct and Evaluate Offspring;
  $P^{(i)}$ = offspring;
end for

Table 12: Critical countries found by the proposed approach.

2010-2012	2013-2015	2015-2017	2017-2019
Afghanistan	Bahamas	Antigua and Barbuda	Burundi
Bangladesh	Guinea	Aruba	Ethiopia
Bhutan	India	Brazil	Haiti
Ghana	Jamaica	Dominica	Iran
India	Japan	France	Jamaica
Iran	Jordan	Greece	Lebanon
Jamaica	Latvia	Iraq	Luxembourg
Moldova	Luxembourg	Israel	Morocco
Morocco	St. Kitts and Nevis	Japan	Sri Lanka
Spain	St. Lucia	Luxembourg	United Kingdom

can contain several nodes that are more connected. We generate test graphs of 150 nodes with different characteristics: the proportion of edges that a node has with nodes from the same community $\{0.1, \ldots, 0.6\}$ - low values indicate dense communities (nodes are more connected with nodes from the same community), the higher the number the node has more edges in other communities than in the one he belongs; the number of overlapping communities a node belongs to, 2 for our experiments; how many nodes belong in multiple communities, $\{13, 50, 70\}$ nodes from the 150 nodes of the graph. The generated communities are the input hypergraphs of our algorithm.

Parameters For our experiments we use the following parameters for the genetic algorithm are: population size $\{25, 40, 100\}$, maximum number of generations 500, crossover $\{0, 0.5, 0.8\}$ and mutation probability 0.5, probability to mutate a bit $\{0, 0.01, 0.02, 0.03\}$, and tournament size of 3. The values tested for k are 5 and 10.

Results We find that they are consistently converging to the same number of components across different parameter values, except when using a mutation rate of 0, indicating the known importance of this operator. When more nodes overlap, we find less components for this value of k, as there are more links across communities in the benchmarks. However, results show that the approach is robust, converging in various settings and may be further extended to practical applications.

2.5 Inflation hypergraph

As a new application we constructed a hypergraph from the world inflation rate (consumer prices). Data about 123 countries is publicly available[6] and contains information about inflation rate from 1960 until 2019. We analyse only the last ten years, $2010 - 2019$, and we eliminate the countries with missing information, after this preprocessing 98 countries remain. The hypergraph is built in the following way: the countries are the nodes of the hypergrah, a hyperedge exists between nodes if they have inflation values in the same year within an interval (one hyperedge contains nodes/countries that have negative inflation for the studied year, the other hyperedges contain the nodes with inflation in the intervals 0-5, 5-10, 10-15 and greater then 15. Because we do not want to obtain one component we divided the 10 years in four intervals (2010-2012,2013-2015,2015-2017,2017-2019) thus obtaining four hypergraphs. The main advantage of this construction is that in this hypergraph the whole dynamics of the inflation of countries appear, because in a single hypergraph several values for each year can be presented. Using the proposed approach we search for the ten most critical nodes in each hypergraph, we present the result in table 12.

A hyperedge in the graph means that countries have similar inflation rate in a certain year. Critical nodes can appear if in a country or in several countries from the same hyperedge the inflation rate changes, thereby it can give a glimpse in a collective set of unstable countries, the countries are investigated as a community.

A new problem, the critical node detection problem for hypergraphs is proposed and a genetic algorithm to solve it. New benchmarks are constructed and as an application a macroeconomic dataset is transformed to a

[6]https://dice.ifo.de/en/node/358439, last accessed 20/09/2021

hypergraph, and critical nodes are obtained, which can reveal new information about the dataset.

Experiments demonstrate the potential of the proposed method. As further work other variants of hypergraph critical node detection problem will be studied.

2.6 Weighted node centrality based critical node detection

The Weighted Node Degree Centrality was proposed is an extension of traditional centrality measures to hypergraphs.

The simple degree centrality for nodes in a hypergraph can be defined as the number of nodes adjacent to the primary node. This definition disregards the strength of the ties between nodes, so a weighted degree centrality is proposed to mitigate this problem. The final formal definition of the weighted node degree centrality incorporates weights to hyperedges and can be described in the following way:

$$C_d^h(i) = \sum_{j=1}^{N} \sum_{k=1}^{L} w_k, if\{v_i, v_j\} \subset e_k$$

where, N is the number of nodes, L is the number of hyperedges, e_k is a specific hyperedge, w_k is the weight of the specific hyperedge and v_i is a specific node. Multiple hyperedge weight can be described, all of them make use of the frequency of the hyperedge's appearance or it's multiplicity (m_j) and the cardinality of the hyperedge (c_j). The following weights are used in the next:

- **Constant:** $w_j = 1$
- **Frequency based:** $w_j = m_j$
- **Newman's definition of ties strength in collaboration networks:** $w_j = \frac{m_j}{|c_j|-1}$
- **Network theory:** $w_j = 1 - (1 - \frac{1}{c_j - 1})^{m_j}$

2.6.1 Results

Two real-world networks are tested, a house-committees and a senate-committees hypergraph were studied. Results are obtained with a simple genetic algorithm. In Tables 13 and 14 a * represents the best run in each category; the best run is chosen instead of an average in order to get a concrete set of nodes that can be tested with the other weight types.

Table 13: Result comparisosns for the house committees network

Original weight type	Constant	Frequency	Newman	Network
Constant	*292.91	293.24	8.32	8.31
Frequency based	290.28	*290.56	8.31	8.30
Newman approach	301.10	301.26	*8.27	8.25
Network approach	299.16	299.39	8.25	*8.24

Table 14: Result comparisosns for the senate committees network

Original weight type	Contant	Frequency	Newman	Network
Constant	*246.14	252.94	18.43	18.18
Frequency based	245.20	*251.54	18.27	18.07
Newman approach	247.77	254.00	*18.15	17.97
Network approach	256.85	262.23	18.07	*17.94

3 Critical node detection in multilayer networks

3.1 Related work

In this section, we give the formal definitions of networks and multilayer networks, and the critical node detection problem is presented in more detail.

Definition A graph is a tuple $G = (V, E)$, where V is the set of nodes and E is the set of edges $E \subseteq V \times V$.

Based on the survey [52] critical node detection problem is classified into two main groups: k-vertex-CNDP, where for a given $k \in \mathbb{N}^*$, and for a given σ connectivity metric the main purpose is to delete a set of k nodes such as the function $f(\sigma)$ to be optimized (minimized or maximized). The other main class is the β-connectivity-CNDP, where for a given connectivity metric σ and for a given integer β the goal is to bound the function $f(\sigma)$ to β, such as the number of deleted nodes to be minimized. In the next section we will discuss in more detail the k-vertex-CNDP, while an extension of this problem is the subject of this article and we give a formal definition:

Definition For a given graph $G = (V, E)$ the k-vertex-CNDP problem consists in finding k nodes, such that $f(\sigma)$ to be minimized (or maximized), where σ is a centrality measure, and S is the set of deleted nodes:

$$\min_{S \subset V} f(\sigma)$$

The most studied version of this problem is the Critical Node Problem (CNP), where the aim is to minimize the pairwise connectivity, and it was proven to be NP-complete on general graphs [5].

Different solutions were proposed, in [82] a mixed integer programming formulation is described for several variants of the CNDP. As stochastic approaches, we mention a Greedy Randomized Adaptive Search Procedure [68], a Memetic Algorithm [88], and an evolutionary framework [4].

Several variants of this problem were studied, [81] proposes the bi-objective critical node detection problem, for which a memetic approach was proposed in [11]. Another bi-objective variant, described in [56], studies the problem of minimizing the pairwise connectivity and the cost of removing the nodes.

[83] introduces the distance-based critical node detection problem, an Integer Linear Programming method is proposed to solve it in [3], and in [43] a decomposition technique is proposed. In [36] an imperfect information variant of this problem is proposed and studied.

Another variant, introduced in [74, 73], and described in more detail in [4] (denoted by kMaxC) studies the removal of k nodes so that the number of remaining components is maximal. In [4] a genetic algorithm is described to solve the problem and two greedy algorithms are presented. In [33] an extremal optimization approach is proposed to solve this problem. [35] extends the problem to hypergraphs, and it solves it with a simple genetic algorithm. Another connectivity measure is the minimization of the largest connected component size (denoted by kMinMaxC). This problem was also proved to be NP-complete on general graphs [74]. In [65] a greedy framework is proposed to solve this problem. [73] proposed a polynomial time algorithm to solve the problem on trees and series-parallel graphs. In [78] an Isolated Centrality measure is introduced, and the kMaxC problem is studied. In [28] this problem is unified with the kMaxC problem and with minimization of the attack cost, obtaining a three-objective problem, a heuristic approach is proposed and tested on an infrastructure network.

The measure σ can also be a centrality measure. In [2] high centrality nodes from networks were removed. In [64] basic centrality measures were modified to use them for power grid networks. Most articles use the centrality measures to obtain topological network information and to incorporate them into the kMaxC or KMinMaxC problem, for example, [46], [65].

For multilayer networks, we first give a formal definition.

Definition A multilayer network can be defined as a quadruplet $MG = (V_M, E_M, V, L)$ [49], where V is the set of nodes, $L = \{L_a\}_{a=1}^{d}$ is the set of layers defined by d aspects, $V_M \subseteq V \times L_1 \times \cdots \times L_d$ is the set of node-layer combinations, $E_M \subseteq V_M \times V_M$.

Remark If $d = 0$, M reduces to a single layer network, if $d = 1$, M reduces to a multiplex network.

Remark A graph $G = (V, E)$ can be considered as a single layer network.

The given definition is the most general formulation of the multilayer network, multiplex networks [75] is a special case of multilayer networks. Other types of multilayer networks are for example: interconnected networks [25], multi-relational networks [18], and multivariate networks [67].

The literature is not rich in critical node detection in the case of multilayer networks, most of which consider only centrality measures extended to deal with multilayer networks as well. [8] presents a method to identify essential proteins in a protein-protein multiplex network based on multiplex eigenvector centrality.

In [71] the authors present an Integer Linear Programming approach to detect the k most vulnerable node in an interdependent network. As a case study, they studied Maricopa Country, where the first layer is the power network, and the second layer is the communication network. In [22] a mathematical framework is described to calculate different centrality measures and ranks nodes based on these measures. In this study, interconnected multilayer networks were analyzed.

[84] proposes a tensor decomposition-based technique to identify key nodes in a multilayer network, a yeast landscape, an inflammatory and a lung cancer multilayer network is analyzed. In [10] a new measure is proposed to detect the most influential nodes, and some real-world multiplex networks and semi-synthetic datasets are used for numerical experiments.

3.2 Problem formulation

The problem to find critical nodes in a multilayer network can be traced back to the 'traditional' problem if the multilayer network is transformed into a simple graph. Naturally in this way, some information will be lost from the network. In the CNDP for a multilayer network, we need to emphasize the importance of layers, therefore one objective is to minimize the number of connections between layers, and the other is to optimize another objective (e.g. maximize the number of components).

Definition Formally, the bi-objective formulation can be described in the following way, for a given multilayer network MG and a given k and a connectivity metric σ: f_1 : minimize the number of connections between layers in graph MG; f_2 : optimizing (minimizing/maximizing) the function $f(\sigma)$;

For the σ connectivity metric we will study two connectivity metrics:

- the number of components - maximizing the number of connected components: deleting k nodes such as the number of connected components in the multilayer graph $MG[V_M \setminus S]$ to be maximized, where S is the set of the deleted k nodes, and $\mathcal{H}(MG[V_M \setminus S])$ denotes the set of components of the graph MG without the set of nodes S, the $kMaxC$ problem consists in:

$$\max_{S \subset V_M} |\mathcal{H}(MG[V_M \setminus S])|,$$

  such that $|S| \leq k$, where $|A|$ denotes the cardinality of set A.

- the size of the largest component - minimizing the size of the largest component: deleting k nodes such as the largest component size h in the multilayer graph $MG[V_M \setminus S]$ to be minimized, where S is the set of the deleted k nodes:

$$\min_{S \subset V_M} \{\max\{|h|, h \in \mathcal{H}(MG[V_M \setminus S])|\}\}$$

The first variant (where the number of connected components needs to be maximized) will be denoted by $P1$ in the rest of the paper, while the second variant, where the size of the largest component needs to be minimized is denoted by $P2$.

Because the problem is a bi-objective optimization problem a straightforward solution is to obtain the Pareto front.

Generally, in a problem with M objectives and p decision variables the Pareto dominance and the solution of a multi-objective optimization problem can be described as follows.

Definition A solution $x \in \mathbb{R}^{'}$ Pareto dominates (when the problem is to minimize) $y \in \mathbb{R}^{'}$ if and only if

$$f_i(x) \leq f_i(y),$$

for $i \in \{1, \ldots M\}$, and $\exists j \in \{1, \ldots M\}$:

$$f_j(x) < f_j(y)$$

Definition A solution $x \in \mathbb{R}^{'}$ is said to be Pareto optimal, if and only if there is no other solution that Pareto dominates it.

Example Let us consider a simple example from Figure 5. If $k = 1$ a solution would be to obtain node 1 (on the first layer). In this case, the number of connected components is 4, and 3 edges between different layers are deleted. Deleting node 4 (on the first layer) the number of connected components would be 4, but no edge between layers would be deleted.

3.3 Numerical experiments

Benchmarks For the numerical experiments, two kinds of datasets are used. Firstly, we have generated multilayer networks, where the critical nodes are known, test the algorithm. Secondly, real-world networks were studied to obtain the critical nodes.

Synthetic benchmarks To test the algorithms scale-free networks [9] were adopted to multilayer networks. Such networks contain hubs, nodes with several connections, which can be considered critical nodes (based on certain measures). Based on the Barabasi-Albert graph generator[7] using the preferential attachment algorithm we can generate multilayer scale-free networks (the source code is made publicly available on a gitlab repository[8]). To

[7] `https://networkx.org/documentation/stable/\_modules/networkx/generators/random\_graphs.html\#barabasi\_albert\_graph`, last accessed 09/08/2022

[8] `https://gitlab.com/kisnandor2/multilayercndp/-/blob/main/MultilayerScaleFreeNetworkGenerator.ipynb`

Figure 5: Two multilayer (3 layer) networks. The right one shows how the network would look like if we removed the critical node (Layer 1, node 1) from the left network

Figure 6: Two small real-world multilayer networks

be sure that we know in advance the critical nodes, we modified the base algorithm in a way that some nodes are preferred over others when doing the preferential selection. The network generator algorithm takes the following parameters as input: number of layers, minimum number of nodes on each layer, maximum number of nodes on each layer, number of critical nodes in total, maximum number of critical nodes on each layer, and a switch that decides if the critical nodes should be spread as much as possible between the layers or if they should appear condensed as much as possible on a single layer.

Real-world benchmarks For numerical experiments, two small networks from biology (see Figure 6) and two larger networks which represent movie scripts under the form of temporal multilayer networks extracted from aligning scripts and subtitles [63]were used. Table 15 presents basic properties of the networks. For the two larger networks, Table 16 presents detailed information about the number of nodes and edges on each layer.

Table 15: Basic properties and source of real-world multilayer networks

Name	Nodes	Edges	Layers	Ref
Metabolite	9	10	3	[41]
Disease	18	29	2	[41],[40]
Avengers	339	4110	3	[63]
Star Wars	220	1954	3	[63]

Table 16: Number of nodes and edges for The Avengers 2012 and the Star Wars IV real-world multilayer networks

Description	Avengers	Star Wars
Character layer nodes	39	65
Keyword layer nodes	232	118
Location layer nodes	68	37
Character Character edges	728	1000
Keyword Keyword edges	1000	126
Location Location edges	192	70
Character Location edges	272	189
Character Keyword edges	1000	211
Keyword Location edges	918	358

Table 17: Top 1% of the obtained nodes (name of the node and the occurrence proportion) over the total 160 runs (10 runs for each parameter configuration) for the Star Wars network

StarWars			
P1		P2	
Node name	Prop.	Node name	Prop.
Blocationkade Runner	0.71	Blocationkade Runner	0.74
Cockpit	0.56	Luke	0.54
Luke	0.41	Cockpit	0.42

3.3.1 Method

To solve the proposed bi-objective problem a popular and effective Pareto-based multi-objective optimization algorithm is used, the NSGA-II [24]. NSGA-II uses a crowding distance-based diversity-preserving mechanism and an elitist selection approach. Based on different studies comparing several multi-objective algorithms [45] the NSGA-II was chosen due to its good performance results, but every Pareto-based multi-objective approach can be used.

3.3.2 Performance evaluation

For performance evaluation two measures are used, the first one is the hypervolume indicator (HI) [89], [90], which measures the volume of the region of the dominated points in the objective space and a reference point.

The second measure is the overall Pareto spread (OS) [61] which measures how widely the observed Pareto solution set spreads over the objective space when the design objective functions are considered.

3.3.3 Results and discussion

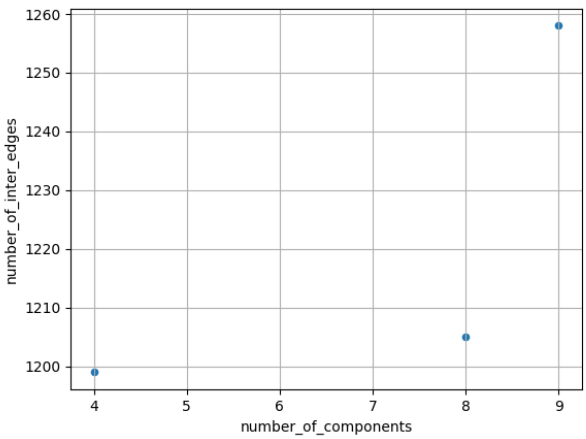
In the case of the two small real-world multilayer networks, we searched for one single critical node, since the networks are very small. In the case of these networks the overall spread of the results, in any parameter combination, was 0, meaning that all the results converged into a single point. The hypervolume value was also the same for any combination. In the case of the first network from Figure 6 our algorithm always resulted in finding node 1 from Layer 1 as the critical node. In the case of the second network from Figure 6 for P_1 the obtained critical node is either node 4 or node 9 from Layer 2, while in the case of P_2 the algorithm always obtained node 5 from Layer 1. In the analysis of the movie datasets two test cases were proposed: searching for $k = 1\%$ of critical nodes out of total

Table 18: Top 1% of the obtained nodes (name of the node and the occurrence proportion) over the total 160 runs (10 runs for each parameter configuration) for Avengers network

Avengers			
P1		P2	
Node name	Prop.	Node name	Prop.
Tony	0.76	Tony	0.73
Nick Fury	0.63	Nick Fury	0.60
Banner's Lab	0.61	Banner's Lab	0.58
Captain America	0.5	Captain America	0.44

Table 19: Overall Pareto spread (OS) and hypervolume index (HI) for the two bi-objective problems (P_1 and P_2), and for different k values ($k = 1\%$ and $k = 5\%$) for the Star Wars network

	$P1$				$P2$			
	$k = 1\%$		$k = 5\%$		$k = 1\%$		$k = 5\%$	
#	OS	HI	OS	HI	OS	HI	OS	HI
1	0.0284	1796	0.0298	4123	0.0024	2097	0.0028	4574
2	0.0236	1781	0.0245	4134	0.0026	2101	0.0046	4530
3	0.0142	1792	0.0146	4231	0.0026	2097	0.0026	4615
4	0.0177	1764	0.0187	4225	0.0017	2097	0.0024	4727
5	0.0165	1305	0.1058	2440	0.0049	1609	0.0088	2376
6	0.0426	1563	0.0459	1900	0.0021	1315	0.0104	2879
7	0.0217	1528	0.0193	4016	0.0021	2013	0.0022	3457
8	0.0210	1660	0.0331	3642	0.0034	2073	0.0079	4307
9	0.0177	1764	0.0146	4222	0.0014	2101	0.0044	4484
10	0.0479	1764	0.0164	4225	0.0014	2097	0.0038	4591
11	0.0213	1764	0.0142	4177	0.0011	2097	0.0024	4660
12	0.0142	1764	0.0104	4228	0.0011	2101	0.0024	4660
13	0.0433	1470	0.0621	3824	0.0020	1883	0.0163	2964
14	0.0296	1351	0.0503	2990	0.0023	1669	0.0173	4215
15	0.0420	1764	0.0128	4100	0.0022	1818	0.0042	4485
16	0.0296	1792	0.0245	4200	0.0024	2060	0.0034	4532



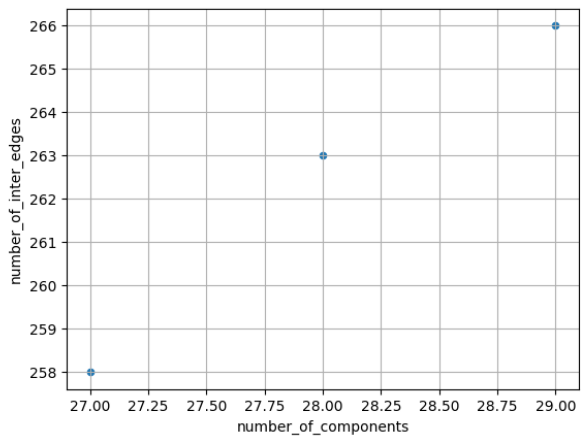


Figure 7: Pareto front example for Avengers network, $k = 1\%$ (left), Star Wars network, $k = 5\%$ (right) for the P_1 problem

nodes and for $k = 5\%$. This means that in the Star Wars IV dataset 3 and 12 nodes were searched for as critical nodes, while for the Avengers dataset 4 and 17 critical nodes. Table 19 presents the results for the 16 parameter configuration for the two bi-objective problems (P_1 and P_2), and for different k values ($k = 1\%$ and $k = 5\%$) for the Avengers and the Star Wars networks respectively. The overall Pareto spread and the hypervolume indicators were reported over 10 independent runs. For the hypervolume indicator, the reference point was set to the outlier coordinates of the Pareto front. As expected in the parameter setting section of the synthetic networks, the results slightly differ for each configuration. Figure 7 presents the obtained Pareto fronts in a single run for the Avengers and Star Wars network, where the population size is 20 and the generation number is set to 100. Table 17 presents the obtained 1% of nodes in a total of 160 experiments based on the Pareto fronts for both $P1$ and $P2$ problems. In [63] the top 5 nodes, sorted by relevance, are shown for different centrality metrics. Compared to their results it is worth mentioning that characters like 'Luke' from Star Wars and 'Nick Fury', 'Captain America' and 'Tony' from the Avengers are top nodes in our results. We also have to mention that in both datasets the chance of selecting a node from the Keyword layer as critical is also very low.

References

[1] www.individual.utoronto.ca/mventresca/cnd.html.

[2] R. Albert, H. Jeong, and A.-L. Barabási. Error and attack tolerance of complex networks. *nature*, 406(6794):378–382, 2000.

[3] G. U. Alozie, A. Arulselvan, K. Akartunalı, and E. L. Pasiliao Jr. Efficient methods for the distance-based critical node detection problem in complex networks. *Computers & Operations Research*, 131:105254, 2021.

[4] R. Aringhieri, A. Grosso, P. Hosteins, and R. Scatamacchia. A general evolutionary framework for different classes of critical node problems. *Engineering Applications of Artificial Intelligence*, 55:128–145, 2016.

[5] A. Arulselvan, C. W. Commander, L. Elefteriadou, and P. M. Pardalos. Detecting critical nodes in sparse graphs. *Computers & Operations Research*, 36(7):2193–2200, 2009.

[6] A. Arulselvan, C. W. Commander, P. M. Pardalos, and O. Shylo. Managing network risk via critical node identification. *Risk management in telecommunication networks, Springer*, 2007.

[7] A. Arulselvan, C. W. Commander, O. Shylo, and P. M. Pardalos. Cardinality-constrained critical node detection problem. In *Performance models and risk management in communications systems*, pages 79–91. Springer, 2011.

[8] K. Athira and G. Gopakumar. An integrated method for identifying essential proteins from multiplex network model of protein–protein interactions. *Journal of Bioinformatics and Computational Biology*, 18(04):2050020, 2020.

[9] A.-L. Barabási and R. Albert. Emergence of scaling in random networks. *Science*, 286(5439):509–512, 1999.

[10] P. Basaras, G. Iosifidis, D. Katsaros, and L. Tassiulas. Identifying influential spreaders in complex multilayer networks: A centrality perspective. *IEEE Transactions on Network Science and Engineering*, 6(1):31–45, 2017.

[11] E. Béczi and N. Gaskó. Approaching the bi-objective critical node detection problem with a smart initialization-based evolutionary algorithm. *PeerJ Computer Science*, 7:e750, 2021.

[12] C. Berge. *Hypergraphs: combinatorics of finite sets*, volume 45. Elsevier, 1984.

[13] S. Boettcher and A. G. Percus. Extremal optimization for graph partitioning. *Physical Review E*, 64(2):026114, 2001.

[14] S. Boettcher and A. G. Percus. Optimization with Extremal Dynamics. *Physical Review Letters*, 86:5211–5214, 2001.

[15] S. Boettcher and A. G. Percus. Extremal optimization: an evolutionary local-search algorithm. In *Computational Modeling and Problem Solving in the Networked World*, pages 61–77. Springer, 2003.

[16] S. P. Borgatti. Identifying sets of key players in a social network. *Computational & Mathematical Organization Theory*, 12(1):21–34, 2006.

[17] S. Brin and L. Page. Reprint of: The anatomy of a large-scale hypertextual web search engine. *Computer Networks*, 56(18):3825–3833, 2012. The WEB we live in.

[18] D. Cai, Z. Shao, X. He, X. Yan, and J. Han. Community mining from multi-relational networks. In *European conference on principles of data mining and knowledge discovery*, pages 445–452. Springer, 2005.

[19] N. Chen. On the approximability of influence in social networks. *SIAM Journal on Discrete Mathematics*, 23(3):1400–1415, 2009.

[20] W. Chen, Y. Wang, and S. Yang. Efficient influence maximization in social networks. In *Proceedings of the 15th ACM SIGKDD international conference on Knowledge discovery and data mining*, pages 199–208, 2009.

[21] K. T. Chi, J. Liu, and F. C. Lau. A network perspective of the stock market. *Journal of Empirical Finance*, 17(4):659–667, 2010.

[22] M. De Domenico, A. Solé-Ribalta, E. Omodei, S. Gómez, and A. Arenas. Ranking in interconnected multilayer networks reveals versatile nodes. *Nature communications*, 6(1):1–6, 2015.

[23] I. De Falco, E. Laskowski, R. Olejnik, U. Scafuri, E. Tarantino, and M. Tudruj. Extremal optimization applied to load balancing in execution of distributed programs. *Applied Soft Computing*, 30:501–513, 2015.

[24] K. Deb, A. Pratap, S. Agarwal, and T. Meyarivan. A fast and elitist multiobjective genetic algorithm: Nsga-ii. *IEEE transactions on evolutionary computation*, 6(2):182–197, 2002.

[25] M. Dickison, S. Havlin, and H. E. Stanley. Epidemics on interconnected networks. *Physical Review E*, 85(6):066109, 2012.

[26] W. Ellens and R. E. Kooij. Graph measures and network robustness. *arXiv preprint arXiv:1311.5064*, 2013.

[27] F. Emmert-Streib, S. Tripathi, O. Yli-Harja, and M. Dehmer. Understanding the world economy in terms of networks: A survey of data-based network science approaches on economic networks. *Frontiers in Applied Mathematics and Statistics*, 4:37, 2018.

[28] L. Faramondi, R. Setola, S. Panzieri, F. Pascucci, and G. Oliva. Finding critical nodes in infrastructure networks. *International Journal of Critical Infrastructure Protection*, 20:3–15, 2018.

[29] Y. Feng, H. You, Z. Zhang, R. Ji, and Y. Gao. Hypergraph neural networks. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 33, pages 3558–3565, 2019.

[30] N. Franzese, A. Groce, T. Murali, and A. Ritz. Hypergraph-based connectivity measures for signaling pathway topologies. *PLoS computational biology*, 15(10):e1007384, 2019.

[31] L. C. Freeman. A set of measures of centrality based on betweenness. *Sociometry*, pages 35–41, 1977.

[32] Y.-C. Gao, Y. Zeng, and S.-M. Cai. Influence network in the chinese stock market. *Journal of Statistical Mechanics: Theory and Experiment*, 2015(3):P03017, 2015.

[33] N. Gaskó, T. Képes, M. Suciu, and R. I. Lung. Critical node detection for maximization of connected components: An extremal optimization approach. In *International Workshop on Soft Computing Models in Industrial and Environmental Applications*, pages 502–511. Springer, 2021.

[34] N. Gaskó, T. Képes, M. Suciu, and R. I. Lung. Critical node detection for maximization of connected components: An extremal optimization approach. In *16th International Conference on Soft Computing Models in Industrial and Environmental Applications (SOCO 2021)*, pages 502–511, Cham, 2022. Springer International Publishing.

[35] N. Gaskó, M. Suciu, R. I. Lung, and T. Képes. An evolutionary approach for critical node detection in hypergraphs. a case study of an inflation economic network. In *International Conference on Intelligent Systems Design and Applications*, pages 1110–1117. Springer, 2022.

[36] C. P. Gillen, A. Veremyev, O. A. Prokopyev, and E. L. Pasiliao. Robustness of solutions to critical node detection problems with imperfect data: a computational study. *Optimization Methods and Software*, 32(2):250–273, 2017.

[37] M. Girvan and M. E. Newman. Community structure in social and biological networks. *Proceedings of the national academy of sciences*, 99(12):7821–7826, 2002.

[38] K.-I. Goh, M. E. Cusick, D. Valle, B. Childs, M. Vidal, and A.-L. Barabási. The human disease network. *Proceedings of the National Academy of Sciences*, 104(21):8685–8690, 2007.

[39] A. Guille, H. Hacid, C. Favre, and D. A. Zighed. Information diffusion in online social networks: A survey. *ACM Sigmod Record*, 42(2):17–28, 2013.

[40] A. Halu, M. De Domenico, A. Arenas, and A. Sharma. The multiplex network of human diseases. *NPJ systems biology and applications*, 5(1):1–12, 2019.

[41] Z. Hammoud and F. Kramer. Multilayer networks: aspects, implementations, and application in biomedicine. *Big Data Analytics*, 5(1):1–18, 2020.

[42] J. He, H. Liang, and H. Yuan. Controlling infection by blocking nodes and links simultaneously. In *International workshop on internet and network economics*, pages 206–217. Springer, 2011.

[43] F. Hooshmand, F. Mirarabrazi, and S. MirHassani. Efficient benders decomposition for distance-based critical node detection problem. *Omega*, 93:102037, 2020.

[44] G. Iacca, K. Konotopska, D. Bucur, and A. Tonda. An evolutionary framework for maximizing influence propagation in social networks. *Software Impacts*, 9:100107, 2021.

[45] H. Ishibuchi, Y. Nojima, and T. Doi. Comparison between single-objective and multi-objective genetic algorithms: Performance comparison and performance measures. In *2006 IEEE International Conference on Evolutionary Computation*, pages 1143–1150. IEEE, 2006.

[46] S. Iyer, T. Killingback, B. Sundaram, and Z. Wang. Attack robustness and centrality of complex networks. *PloS one*, 8(4):e59613, 2013.

[47] D. Kempe, J. Kleinberg, and É. Tardos. Maximizing the spread of influence through a social network. In *Proceedings of the ninth ACM SIGKDD international conference on Knowledge discovery and data mining*, pages 137–146. ACM, 2003.

[48] T. Képes, N. Gaskó, and G. Vekov. The combined critical node and edge detection problem. an evolutionary approach. In *International Conference on Parallel Problem Solving from Nature*, pages 324–338. Springer, 2022.

[49] M. Kivelä, A. Arenas, M. Barthelemy, J. P. Gleeson, Y. Moreno, and M. A. Porter. Multilayer networks. *Journal of complex networks*, 2(3):203–271, 2014.

[50] K. Konotopska and G. Iacca. Graph-aware evolutionary algorithms for influence maximization. In *Proceedings of the Genetic and Evolutionary Computation Conference Companion*, pages 1467–1475, 2021.

[51] J. Kunegis. Konect: The koblenz network collection. In *Proceedings of the 22nd International Conference on World Wide Web*, WWW '13 Companion, page 1343–1350, New York, NY, USA, 2013. Association for Computing Machinery.

[52] M. Lalou, M. A. Tahraoui, and H. Kheddouci. The critical node detection problem in networks: A survey. *Computer Science Review*, 28:92–117, 2018.

[53] V. Latora, V. Nicosia, and G. Russo. *Complex networks: principles, methods and applications.* Cambridge University Press, 2017.

[54] J. Leskovec, A. Krause, C. Guestrin, C. Faloutsos, J. VanBriesen, and N. Glance. Cost-effective outbreak detection in networks. In *Proceedings of the 13th ACM SIGKDD international conference on Knowledge discovery and data mining*, pages 420–429, 2007.

[55] J. M. Lewis and M. Yannakakis. The node-deletion problem for hereditary properties is np-complete. *Journal of Computer and System Sciences*, 20(2):219 – 230, 1980.

[56] J. Li, P. M. Pardalos, B. Xin, and J. Chen. The bi-objective critical node detection problem with minimum pairwise connectivity and cost: Theory and algorithms. *Soft Computing*, 23(23):12729–12744, 2019.

[57] M. Lozano, C. García-Martínez, F. J. Rodriguez, and H. M. Trujillo. Optimizing network attacks by artificial bee colony. *Information Sciences*, 377:30–50, 2017.

[58] R. I. Lung and D. Dumitrescu. Computing nash equilibria by means of evolutionary computation. *Int. J. of Computers, Communications & Control*, 3(suppl. issue):364–368, 2008.

[59] R. I. Lung, M. Suciu, and N. Gaskó. Noisy extremal optimization. *Soft Computing*, 21(5):1253–1270, 2017.

[60] D. Lusseau, K. Schneider, O. J. Boisseau, P. Haase, E. Slooten, and S. M. Dawson. The bottlenose dolphin community of doubtful sound features a large proportion of long-lasting associations. *Behavioral Ecology and Sociobiology*, 54(4):396–405, 2003.

[61] H.-y. Meng, X.-h. Zhang, and S.-y. Liu. New quality measures for multiobjective programming. In *International Conference on Natural Computation*, pages 1044–1048. Springer, 2005.

[62] R. Milo, S. Itzkovitz, N. Kashtan, R. Levitt, S. Shen-Orr, I. Ayzenshtat, M. Sheffer, and U. Alon. Superfamilies of evolved and designed networks. *Science*, 303(5663):1538–1542, 2004.

[63] Y. Mourchid, B. Renoust, H. Cherifi, and M. El Hassouni. Multilayer network model of movie script. In *International Conference on Complex Networks and their Applications*, pages 782–796. Springer, 2018.

[64] A. Nasiruzzaman, H. Pota, and M. Mahmud. Application of centrality measures of complex network framework in power grid. In *IECON 2011-37th Annual Conference of the IEEE Industrial Electronics Society*, pages 4660–4665. IEEE, 2011.

[65] D. T. Nguyen, Y. Shen, and M. T. Thai. Detecting critical nodes in interdependent power networks for vulnerability assessment. *IEEE Transactions on Smart Grid*, 4(1):151–159, 2013.

[66] T. Opsahl. Why anchorage is not (that) important: Binary ties and sample selection.

[67] P. Pattison and S. Wasserman. Logit models and logistic regressions for social networks: Ii. multivariate relations. *British journal of mathematical and statistical psychology*, 52(2):169–193, 1999.

[68] D. Purevsuren, G. Cui, N. N. H. Win, and X. Wang. Heuristic algorithm for identifying critical nodes in graphs. *Advances in Computer Science: an International Journal*, 5(3):1–4, 2016.

[69] J. Reimand, L. Tooming, H. Peterson, P. Adler, and J. Vilo. Graphweb: mining heterogeneous biological networks for gene modules with functional significance. *Nucleic Acids Research*, 36:452–459, 05 2008.

[70] R. A. Rossi and N. K. Ahmed. The network data repository with interactive graph analytics and visualization. In *AAAI*, 2015.

[71] A. Sen, A. Mazumder, J. Banerjee, A. Das, and R. Compton. Identification of k most vulnerable nodes in multi-layered network using a new model of interdependency. In *2014 IEEE Conference on Computer Communications Workshops (INFOCOM WKSHPS)*, pages 831–836. IEEE, 2014.

[72] L. S. Shapley. A value for n-person games. *Contributions to the Theory of Games*, 2(28):307–317, 1953.

[73] S. Shen and J. C. Smith. Polynomial-time algorithms for solving a class of critical node problems on trees and series-parallel graphs. *Networks*, 60(2):103–119, 2012.

[74] S. Shen, J. C. Smith, and R. Goli. Exact interdiction models and algorithms for disconnecting networks via node deletions. *Discrete Optimization*, 9(3):172–188, 2012.

[75] L. Solá, M. Romance, R. Criado, J. Flores, A. García del Amo, and S. Boccaletti. Eigenvector centrality of nodes in multiplex networks. *Chaos: An Interdisciplinary Journal of Nonlinear Science*, 23(3):033131, 2013.

[76] S. R. Sundaresan, I. R. Fischhoff, J. Dushoff, and D. I. Rubenstein. Network metrics reveal differences in social organization between two fission–fusion species, grevy's zebra and onager. *Oecologia*, 151(1):140–149, 2007.

[77] Y. Tang, Y. Shi, and X. Xiao. Influence maximization in near-linear time: A martingale approach. In *Proceedings of the 2015 ACM SIGMOD international conference on management of data*, pages 1539–1554, 2015.

[78] O. Ugurlu. Comparative analysis of centrality measures for identifying critical nodes in complex networks. *Journal of Computational Science*, 62:101738, 2022.

[79] T. W. Valente. Social network thresholds in the diffusion of innovations. *Social networks*, 18(1):69–89, 1996.

[80] M. Ventresca. Global search algorithms using a combinatorial unranking-based problem representation for the critical node detection problem. *Computers & Operations Research*, 39(11):2763–2775, 2012.

[81] M. Ventresca, K. R. Harrison, and B. M. Ombuki-Berman. The bi-objective critical node detection problem. *European Journal of Operational Research*, 265(3):895–908, 2018.

[82] A. Veremyev, O. A. Prokopyev, and E. L. Pasiliao. An integer programming framework for critical elements detection in graphs. *Journal of Combinatorial Optimization*, 28(1):233–273, 2014.

[83] A. Veremyev, O. A. Prokopyev, and E. L. Pasiliao. Critical nodes for distance-based connectivity and related problems in graphs. *Networks*, 66(3):170–195, 2015.

[84] D. Wang, H. Wang, and X. Zou. Identifying key nodes in multilayer networks based on tensor decomposition. *Chaos: An Interdisciplinary Journal of Nonlinear Science*, 27(6):063108, 2017.

[85] R. Yang, L. Huang, and Y.-C. Lai. Selectivity-based spreading dynamics on complex networks. *Physical review e*, 78(2):026111, 2008.

[86] J. Yu, D. Tao, and M. Wang. Adaptive hypergraph learning and its application in image classification. *IEEE Transactions on Image Processing*, 21(7):3262–3272, 2012.

[87] W. W. Zachary. An information flow model for conflict and fission in small groups. *Journal of anthropological research*, 33(4):452–473, 1977.

[88] Y. Zhou, J.-K. Hao, and F. Glover. Memetic search for identifying critical nodes in sparse graphs. *IEEE transactions on cybernetics*, 49(10):3699–3712, 2018.

[89] E. Zitzler and L. Thiele. Multiobjective optimization using evolutionary algorithms—a comparative case study. In *International conference on parallel problem solving from nature*, pages 292–301. Springer, 1998.

[90] E. Zitzler and L. Thiele. Multiobjective evolutionary algorithms: a comparative case study and the strength pareto approach. *IEEE transactions on Evolutionary Computation*, 3(4):257–271, 1999.

Project director,
Conf. dr. Gaskó Noémi