BABEŞ–BOLYAI UNIVERSITY
FACULTY OF MATHEMATICS AND COMPUTER SCIENCE

Doctoral Thesis
Summary

Computational Intelligence Models for Influence Problems

SCIENTIFIC SUPERVISOR:
PROF. DR. HORIA F. POP

STUDENT:
TAMÁS-ZSOLT KÉPES

2024

Contents

Abstract

The thesis mainly focuses on graphs, and important components of graphs. Importance is defined in multiple ways, one approach of importance is influence, or a component's ability to spread information. The search for influential nodes in a graph has led us, to prepare multiple articles about the topic. These articles later formed the backbone for the first half of this thesis, focusing on the journey we took when investigating influential nodes, and influence in general. Our focus was multi faceted, from introductory works on approaches using Extremal Optimization, Cascade, Shapley value and a Monte Carlo approach, to the combination of these factors and the resulting algorithm.

The second definition of importance that we investigated was the problem of criticality. Critical nodes and edges along with critical hypergraph components were all investigated. The second part of the thesis focused on the problem of criticality in all sorts of networks, with a structure that focused on individual results more so than on the journey through time. At the end, several relevant applications and use-cases were presented.

Keywords

Influence, Influence Maximization, Criticality, Critical Node Detection, Extremal Optimization, Cascade, Genetic Algorithms, Shapley Value, Graphs, Hypergraphs

1. Chapter

Introduction

1.1 Introduction

The research assumed by this thesis is one grounded in years of previous works and publications. The idea of network metrics and network analysis is an interesting topic motivated by the potential usefulness in fields that relate to anything network-like, such as road networks [37], social networks [30] even communication networks [21] such as the Internet itself. The work will focus on two separate topics, that were both linked as network measures, but which were distinct enough since they differ in their objectives. The two major parts of this thesis focus on the two problems of Influence Maximization, presented in detail in Chapter 3, and Criticality with variants of the Critical Node Detection Problem, presented in Chapter 4.

A good example for the influence analysis can be an advertising firm looking for the most influential social media personalities or the analysis of virus spread. As for criticality, a good example can be the analysis of network failures or the analysis of prisoner interactions in order to predict critical inmates and prevent riots.

1.2 Objectives

The main objective assumed by this work was the complete assembly of all previous works done during the doctoral research process, grounded in years of publications, with several diverging, but still related research topics. The separate parts of the research, that focus on the two major problem families had distinct objectives, these objectives are presented below.

The Influence Maximization problem focuses on finding nodes with the highest importance in a network, importance in the context of influence can be defined as a node's information diffusion ability, in other words, the ability of a node to spread information. The main objective of Influence Maximization evolved throughout the research process. Initial objectives were the introduction of basic Game Theory concepts to the problem of Influence Maximization, and the identification of a good base optimization algorithm and propagation model to test our Game Theory concepts. With the introduction of these basic building blocks of our influence research, the follow-up for our initial processes was gradual, with experimentation of different diffusion models, different game theory concepts, and mathematical calculations. Finally, we arrived at the main objective of our Influence Maximization research, that being the introduction of a complete algorithm, that takes into account every bit of research, improvement, and general knowledge in the field of Influence, and applies it to one complete and complex algorithm. This journey and the culmination of the processes are presented in the Influence part of the thesis, Chapter 3.

The problem of Criticality started as the Critical Node Detection problem, a problem similar to that of influence maximization, with the main objective of Critical Node Detection being the ability to determine important nodes, this time importance being defined as criticality, where critical nodes are nodes that when removed, would maximally degrade the network. The initial explorations of this problem had their roots in the Critical Node Detection problem but they then evolved into a more general problem of Criticality, since we no longer talked about simple networks, or even just nodes in our research, but an increasingly generalized approach to the problem of Criticality. Two simple questions we asked: What can be critical in a network? and: What effect do different kinds of networks have on the concept of criticality? The objective of our research was to answer these questions with several variants of the Criticality problem being developed throughout the years. The main objective of the thesis is the collection and organization of these disunited researches, pointing out similarities and the evolution of our thoughts on Criticality, all presented in Chapter 4.

1.3 Original Contributions

The original contributions that this doctoral research process proposed to the world were numerous. Firstly, in general, the main contribution of our research to the literature was a plethora of varied and diverse optimization algorithms for the problems of Influence and Criticality, all being built on each other, and all providing increasingly good results, comparable or surpassing even state of the art algorithms' results.

Separating the two chapters, our main contributions to the field of Influence Maximization were the introduction and elaboration of Game Theory concepts together with the Extremal Optimization algorithm, to redefine and reformulate the Influence problem, as a cooperative game, with nodes as players. This novel concept exploration was very limited in the literature, which is why we tried to innovate in this direction. Further innovation came with the introduction of the Shapley value, a more specialized game theory value that can calculate individual contribution to a game, and the translation of the Shapley value to the language of Influence. Later, the introduction of a Shapley value approximation was the main innovation, since as far as we could tell, the process in which we approximated the Shapley value was very new and unique. Finally, the introduction of a few new improvements to our process led to the development of our final algorithm, which was the main contribution our research made to the field.

For the Criticality-related problems, our contributions did not follow a very linear approach, instead, we tried to innovate in all realms of Criticality detection, with several differing approaches, such as Genetic Algorithms, Greedy Approaches, and even variants of the Extremal Optimization algorithm being proposed. Our most original contribution to the field needs to be the exploration of Hypergraphs in relation to Criticality, and more specifically, the exploration of Hypergraph-related centrality metrics in relation to Criticality, which was yet again, an extremely novel idea.

We proposed a number of applications for both problems, such as a citation network analysis for the Influence Maximization problem, and the stock market, inflation, and political network analysis for the Criticality Detection problem. We also proposed a new network robustness metric, which uses one of our criticality algorithms in order to determine the robustness of a network, this new metric performs greatly in comparison to other, more well-established metrics, presenting itself as a valuable alternative to existing measurements.

2. Chapter

Basic Theoretical Notions

2.1 Simple graph related definitions

Definition 2.1.1 (Graphs). A Graph is represented as a $G = (V, E)$ pair, where $V = \{v_1, v_2, ..., v_n\}$ is the set of nodes or vertices, while $E = \{e_1, e_2, ..., e_n\}$ is the set of edges. In traditional graphs, the E set contains pairings of nodes. The values n and m denote the number of nodes and edges respectively.

A directed graph is a type of graph, where each edge has a direction, with a starting and an ending node. A directed edge is usually represented using an ordered pairing of two nodes.

Definition 2.1.2 (Neighbors). Given two nodes $v, w \in V$, v and w are considered neighbors if and only if $\exists e \in E$ edge so that $e = (v, w)$.

Definition 2.1.3 (Degree, In-Degree, Out-Degree). The degree of a node $v \in V$ can be given as the $|\{w \in V,$ where v and w are neighbors$\}|$, basically the cardinality of the set containing every node that is a neighbor of v.

In-degrees and out-degrees are only defined for directed graphs, the in-degree of a node v is the number of edges that point into the node whereas the out-degree of v is the number of edges that point out from node v.

Definition 2.1.4 (Paths). A path in a graph G can be defined as a sequence of vertices $v_1, v_2, v_3, ..., v_n$, where each adjacent pair of vertices (v_i, v_{i+1}) is connected by an edge in the graph. In other words, for each $i = 1..n - 1$, there exists an edge (v_i, v_{i+1}) in the graph.

Definition 2.1.5 (Circles). A circle in a graph G can be defined as a path $v_1, v_2, v_3, ..., v_n$, where $n \geq 3$, and the vertices v_1 and v_n are connected via an edge (v_n, v_1).

Definition 2.1.6 (Trees). A tree is an undirected, connected, and acyclic graph. A tree is the pair $T = (V, E)$, where V is the set of nodes and E is the set of edges.

Definition 2.1.7 (Connected components). Given an undirected graph $G = (V, E)$, where V is the set of vertices and E is the set of edges, a connected component C is a subset of vertices

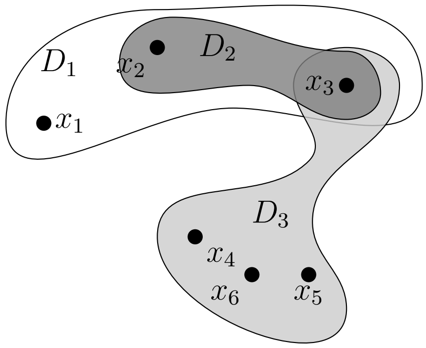

Figure 2.2.1: A simple example of a hypergraph with six nodes and three hyperedges.

$C \subseteq V$ such that for every pair of vertices $(u, v) \in C$, there exists a path in G that connects u and v.

2.2 Hypergraph definitions

Hypergraphs, introduced and formalized in [7], can be considered generalizations of simple graphs, with a definition similar to that of simple graphs, presented in Definition 2.2.1.

Definition 2.2.1 (Hypergraphs). A hypergraph can be defined as 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 the set of hyperedges, consisting of subsets of X, n and m refer to the number of nodes and hyperedges respectively.

Example 2.2.1. An example of a hypergraph can be seen in Figure 2.2.1. The hypergraph has six nodes, $X = \{x_1, x_2, x_3, x_4, x_5, x_6\}$, and three hyperedges $D = \{D_1, D_2, D_3\}$, $D_1 = \{x_1, x_2, x_3\}$, $D_2 = \{x_2, x_3\}$, $D_3 = \{x_3, x_4, x_5, x_6\}$.

Some classical graph definitions need to be redefined in a hypergraph environment.

Definition 2.2.2 (Neighbors). Given two nodes $u, v \in X$, u and v are considered neighbors if and only if $\exists D$ hyperedge so that $u \in D$ and $v \in D$.

Definition 2.2.3 (Degree). The degree of a node $u \in X$ can be given as the $|\{v \in V$, where u and v are neighbors$\}|$, basically the cardinality of the set containing every node that is a neighbor of u, if multiple nodes share the same hyperedge with u, every node in the same hyperedge counts as a separate degree.

Definition 2.2.4 (Paths). A path in a hypergraph $\mathcal{H}$ can be defined as a sequence of vertices $v_1, v_2, v_3, ..., v_n$, where each adjacent pair of vertices (v_i, v_{i+1}) share a hyperedge in the hypergraph. In other words, for each $i = 1..n-1$, there exists a hyperedge D so that $v_i \in D$ and $v_{i+1} \in D$.

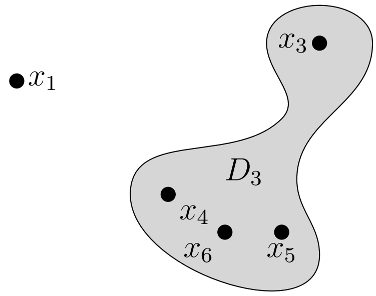

Figure 2.2.2: Strong detection of x_2 from the hypergraph presented in Fig. 2.2.1

Definition 2.2.5 (Connected components). Given a hypergraph $\mathcal{H} = (X, D)$, where X is the set of vertices and D is the set of hyperedges, a connected component C is a subset of vertices $C \subseteq X$ such that for every pair of vertices $(u, v) \in C$, there exists a path in the hypergraph $\mathcal{H}$ that connects u and v.

Definition 2.2.6 (Strong node deletion). Given a hypergraph $\mathcal{H} = (X, D)$, where X is the set of vertices and D is the set of hyperedges, strong node deletion consists in removing a node v from the set of nodes X, together with every hyperedge the node v is connected to. Other nodes from the same hyperedges are not considered in the deletion and remain in place.

Example 2.2.2. An example of strong node deletion on the hypergraph from Example 2.2.1 can be seen in figure 2.2.2.

Definition 2.2.7 (Weak node deletion). Given a hypergraph $\mathcal{H} = (X, D)$, where X is the set of vertices and D is the set of hyperedges, weak node deletion consists in removing a node v from the set of nodes X. The hyperedge containing the node v is only removed if v is the single node in that hyperedge.

Example 2.2.3. An example of weak node deletion on the hypergraph from 2.2.1 can be seen in figure2.2.3.

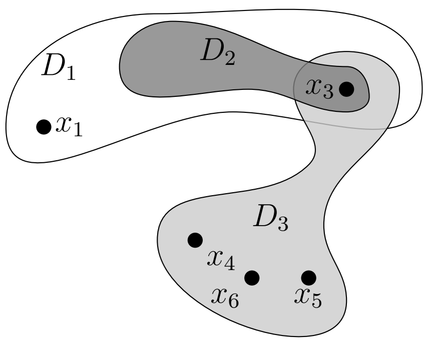

Figure 2.2.3: Weak deletion of x_2 from the hypergraph presented in Fig. 2.2.1

2.3 Optimization Problems

Optimization problems are a class of problems, where starting from a base state, a new, improved state must be reached, providing us with a solution for a given problem, this solution being one that gives an increased amount of satisfaction to some arbitrary metric.

Definition 2.3.1 (Optimization problems). Optimization problems can be defined as the following:

$$P = (X, f, \omega)$$

where, P is the optimization problem, X is the search space, f is the objective function and ω is the set of constraints.

Definition 2.3.2 (The objective function). The objective function f is the function that we would want to optimize and can be defined as: $f : X \to Y$, where X is the search space, while $Y \in \mathbb{R}$ is a real number.

Definition 2.3.3 (Global minimum). $x^* \in X$ is a global minimum for the objective function $f : X \to Y$ if $f(x^*) \leq f(x), \forall x \in X$.

Definition 2.3.4 (Single objective optimization problem). The single objective optimization problem, that minimizes can be described as:

$$\min_{x \in \mathbb{R}^n} f(x)$$

,

such as $h_j(x) = 0, j = 1, ..., J$ and $g_k(x) \leq 0, k = 1, ..., K$, where h_j and g_k are any number of contraint functions, while $x = (x_1, ..., x_n)$.

It is important to observe, that the above definitions only refer to minimization, but similar definitions can also be given for maximization problems.

3. Chapter

Influence Maximization

3.1 Introduction to the Influence Maximization Problem

The Influence Maximization Problem (IMP) is the problem that describes the spread of information in a network, nodes that spread the information at higher rates are called influential nodes. The main aim of IMP is to find a set of nodes from a network, that can spread the information at a maximal degree, thus maximizing the influence of the found set. The spread of information is simulated using diffusion models, which are also called propagation models. Using these models, we can define the Influence function, which is used to estimate the number of activated nodes, after the application of the specific diffusion model.

Definition 3.1.1 (Influence function). Given a graph $G = (V, E)$, the influence function $f_D(S) : 2^V \to \mathbb{R}^+$ can be defined as the average number of nodes activated by D using the nodes from set S as seeder nodes, where $S \subseteq V$ is a set of nodes, and D is a propagation (diffusion) model.

Definition 3.1.2 (IMP). Given a graph $G = (V, E)$, using D as the propagation model, the IMP can be defined as the problem of finding a set S of seeder nodes that will maximize the influence function:

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

where $|.|$ denotes the cardinality of a set, and $k \in \mathbb{N}^*$ is a parameter, setting the size of the seeder set to a fixed value.

The IMP is a complex and well-studied, NP-Hard optimization problem, with several proposed algorithms, such as in [28], [23], [11], [20], etc.

3.2 Propagation Models

The influence maximization problem uses propagation models to simulate the spread of information in an environment [18]. Two main classes of propagation algorithms exist, deterministic and probabilistic models.

In deterministic models, we can compute the set of nodes that have been activated by the propagation process. The results from these computations have been used various applications in the literature, a main use for these types of models is social networks, and the analysis done on social networks, such as in [19].

Probabilistic propagation models aim to estimate the information spread and generally give a more realistic result. They achieve this using a probabilistic variable or other stochastic methods in order to include variability in the results. Some of the more popular propagation models currently investigated in the literature are the cascade models such as the Independent Cascade Model (ICM) [23], the Weighted Cascade Model (WCM) [23] or the linear threshold model [32]. For most of our influence-related research, the cascade algorithm variants were used, as the main form of propagation simulation.

3.2.1 The Cascade

The cascade algorithm is a propagation model, used in the simulation of information diffusion in a social network. It is widely used in the literature for many different applications [23]. The main algorithm is extremely similar to a breadth-first search on a network, with three main differentiating factors, those being an increased number of starting nodes and an increase to the number of investigated nodes in any given iteration, together with a different, probabilistic, traversal process. This differing traversal process allows the otherwise simplistic traversal algorithm, to simulate information diffusion. The activation step is present in the neighbor selection process, the idea being, that every neighbor of the currently investigated nodes should be considered, but not every neighbor should be visited. The selection of which neighbor to visit (or activate) is done using a probabilistic variable p, meaning that for every neighbor of the currently investigated node set, there is a probability of p that the neighbor will be activated.

Connecting the above functionalities, the cascade algorithm, regardless of the chosen variant, will always provide a number, which is the cardinality of the set of successfully activated nodes. A more mathematical description:

$$\sigma(A_0) = |A|$$

where A_0 is the starting set of nodes, while A is the set of activated nodes.

An important observation should be made: the set of starting nodes is always guaranteed to be activated, meaning that the cardinality of the set of activated nodes will always be at least equal to the cardinality of the starting set of nodes.

$$\sigma(A_0) \geq |A_0|$$

Another important observation should be, that given the probabilistic nature of the cascade algorithm, one run does not provide statistically consistent results, the average of multiple runs of the cascade algorithm is necessary in order to average out any anomalies and to provide conclusive results. This means that, despite the cascade algorithm's result being a cardinality, that is a whole number, in any given context, the result of the average, and therefore the result of the information spread investigation will be a real number.

Independent Cascade Model

The independent cascade model (ICM) [23] is a widely used cascade algorithm variant, and it is the propagation model of choice throughout most of the research process.

The Independent Cascade uses a global, static probability of p (values most commonly range from 1% to 5%).

Weighted Cascade Model

The Weighted Cascade Model (WCM) as proposed in [23] is a diffusion model contained in the wider cascade algorithm family. Its functionality is similar to that of the Independent Cascade Model with the main difference being observed in the probability of propagation p used in the cascade algorithm. The Weighted Cascade uses a per-node probability $p_{w'}$ which gives the probability of activating neighbor w'. This probability is calculated as

$$p_{w'} = \frac{1}{in\_degree(w')}$$

3.3 Extremal Optimization

Nature and many physical phenomena have strong self-optimizing characteristics [5], and many co-dependent natural environments are optimized by the selection of the undesirable or "bad" individuals and their random replacement in the system.

The main research conducted for this thesis was focused on the IMP's interpretation as a co-dependent environment, and the aforementioned selection of individuals led to the use of our first proposed algorithm family that was considered, the family of Extremal Optimization (EO) [9] [8] algorithm variants. EO is defined in Definition 3.3.1.

Definition 3.3.1 (Extremal optimization). Extremal Optimization is an optimization algorithm, with the premise of dividing the solution of a proposed problem into multiple smaller components. Each component has to have a calculable contribution to the goodness of any proposed solution. Furthermore, in any given instance of the EO algorithm, there are multiple solution candidates that are analyzed at any given time.

The main challenge of the conversion from graph-based problems, such as the influence maximization problem, to the world and terminology of optimization algorithms, more specifically EO, was the interpretation of data and the translation of terminology between the two worlds. Another problem was the linking of the propagation model terminology with both graph and EO terminology.

A solution s and the best solution s_{best} are both sets of nodes, that are subsets of V. These sets are the individuals from the EO terminology and they have a size of k, which is the number that limits the size of s from IMP terminology. Each individual s is composed of nodes, which are the components of the EO terminology.

Both individuals and components of an individual need a fitness function, these fitness functions evolved throughout our research history. The fitness function of each individual should be related to the influence of the set of nodes that each individual is composed of. This influence is calculated using the propagation model of choice. If we consider our main propagation method of the Independent Cascade Method our influence function and therefore the fitness function of each individual can be represented as

$$f_{ICM}(s) = \bar{\sigma}(s),$$

where $\bar{\sigma}(s)$ is the average of multiple runs of the propagation model.

For the fitness of each component, we need to calculate the contribution of each node to the fitness of the complete individual. The main idea of our research was, to consider the problem of influence as a coalitional cooperative game from game theory, and then to use this game theory understanding to calculate the contributions of each node as players of the game. Our first algorithm uses a more simplistic function. The fitness value of an active node i in s is computed as its marginal contribution to $\bar{\sigma}$:

$$f_i(s) = \bar{\sigma}(s) - \bar{\sigma}(s \setminus i) - 1,$$

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

Time-Varying Extremal Optimization

The main drawback of the Extremal Optimization algorithm is its struggles with local optima. One of the proposed solutions or remedies to this problem was the utilization of a Time-Varying version of EO, which aims at changing a set of minimal components from any individual instead of one component at a time. The number of components changed in any given iteration is labeled as q. Three versions of the parameter modification process were considered:

Basic Linear version:

$$q = max(1, \lfloor \frac{k * (T_{Max} - T)}{T_{Max}} \rfloor),$$

where q is the number of nodes designed to be replaced in the current generation, k is the size of set s, the number of nodes in any given individual from the EO algorithm, T is the current generation count, T_{Max} is the total number of generations, while $\lfloor \cdot \rfloor$ is the floor of a real number

Alternative Linear version:

$$q = max(1, \lfloor \frac{k * (T_{Max} - 2 * T)}{T_{Max}} \rfloor),$$

Exponential version:

$$q = \lfloor max(1, \frac{1}{2} * k * (k - 1)^{\frac{-T}{T_{Max}}} \rfloor$$

The alternative linear was chosen as it gave the best results for our purposes, with a small modification, of ensuring that the number of nodes replaced was not above half of the total nodes.

3.4 The Shapley Value

The Shapley value was proposed in [33], and it is used to calculate the contribution of each player to a cooperative game in game theory, making it a popular solution concept for the game. It has many descriptions, but for our purposes, the Shapley value can be described as the method of fair division of the earnings of a cooperative game, where the payoff can be divided. This fair division should relate to the individual contribution of each player.

Definition 3.4.1 (Cooperative coalitional game). A cooperative coalitional game $\Delta = (N, v)$ contains two elements:

- The players of the game, contained in set N;
- A characteristic function $v : 2^N \to \mathbb{R}$ that assigns real values to subsets of players.

Definition 3.4.2 (Shapley Value). The Shapley Value ϕ_i can be defined in the context of a cooperative coalitional game, where it measures the average contribution of player i in all coalitions of the game and it is calculated as the following:

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

where $|\cdot|$ represents the cardinality of a set.

Since the Shapley value can be used to calculate the extent of each player's contribution to the final value of the game, the concept should be translatable to the extremal optimization language, and we should be able to calculate each component's contribution i.e. their fitness, to the final value of the individual's fitness.

3.4.1 The Shapley Value Approximation

While the Shapley value gives great results, mainly because it closely approximates each node's contribution to the final result, thus making it trivial to choose the nodes with minimal value, there is a major problem with the Shapley value calculation. In each instance of the Shapley value calculation, every coalition of nodes is considered. This makes the calculation extremely costly in any practical application. To overcome this, we used an approximation method, which is shown to sufficiently approximate the results of a complete Shapley value calculation. This approximation uses a number of distinct orderings of size k, containing the same nodes as s but in different permutations, we then calculate the nodes' contribution based on only these different orderings and the coalitions from these orderings. Since our calculation depends on the order of nodes in the coalition, ensuring that there were a sufficient number of orderings considered was one of our main challenges.

3.5 The Monte Carlo Approach

The usual approach to the algorithm uses the entire original network given as an input, to calculate the results of the Cascade run, resulting in the need for multiple Cascade runs. Since the propagation model has a probabilistic nature if we wish to provide a level of certainty to the results, we would need the averaging of multiple propagation runs. This is both a slow and unpredictable process, with runtime increasing with the increase in desired predictability. We proposed an improvement, using a Monte Carlo approach to introduce a network generation phaes.

The scheme works by creating a number n_G of new networks from the original network, this n_G is similar in scale to the number of Cascade runs that would be necessary for statistically accurate results in the original scenario. Each newly created network $G_i = (V, E_i)$ contains every node of the original network V and a small percentage of edges from the original network $E_i \subset E$. The edges that are kept for a new network are selected at random from E, and the probability of selecting an edge to be kept is p, the same probability used in the original independent cascade.

Using these newly generated networks greatly decreases complexity and runtime, since we modified the search space. The cascade algorithm used for these new networks is reduced to a non-probabilistic version, with the probabilistic nature of the search being incorporated in the newly generated network space, on which the algorithm is being run. This change in perspective allows us to prepare a large number of generated networks in the setup phase of the algorithm, and we can then later use the averages of the runs on each network, to get a final result for a given seeder set. We can still use the same seeder set since the nodes in the generated networks did not change.

The Shapley Value approximation is also affected, with the modification of the cascade algorithm. The revision of introducing the new Monte Carlo Cascade to the Shapley value approximation algorithm did not modify the validity or the effectiveness of the approximation.

3.6 The SIM-EO Algorithm

Throughout our research on the influence maximization problem, the intentions were clear to summarize every and all possible improvements proposed throughout the iterative steps done on the algorithms, and a final algorithm was developed using these improvements. The algorithm proposed was named Shapley Influence Maximization - Extremal Optimization (SIM-EO), acknowledging both the algorithm type which is an Extremal Optimization variant and also the specialty fitness used, that being the use of the Shapley value as component fitness.

SIM-EO outline The algorithm draws on every previous iteration. Firstly, its base is the simplified version of our earliest EO-based algorithm, discarding the more complex s_{best} selection present there, but maintaining the similar framework of the algorithm.

SIM-EO starts with the Monte Carlo Network Generation algorithm, utilizing the proposed Monte Carlo improvements, creating multiple networks using the Cascade probability as the base for network generation probability and a parameter n_G to give the required network count.

Each individual's fitness is calculated using the Independent Cascade algorithm's variant, which is modified by replacing the probabilistic nature of the Cascade and the need for multiple runs on the same network, with the many networks approach, each network differing, each being generated using the Monte Carlo method.

The algorithm then uses the Shapley value algorithm, more specifically the Shapley value approximation algorithm using the Monte Carlo networks, as the fitness value for each component.

The number of changed components is represented by the number m and it is calculated using the improved equation from the time-varying extremal optimization, which is based on the alternate linear version from the equation 3.3.

As for the parameters of the algorithm, we differentiate between inputs, which refer to the influence maximization problem, and parameters which are specific to the algorithm in question. Parameters include the n_G, $MaxIterations$, which is the number of maximum generations the algorithm will run for, and m. The inputs include the initial network G, and the values k and p.

3.7 Numerical experiments and a proposed application

While the numerical experiments are too large in scope to fit this smaller summary, a short description of the results can be given. The SIM-EO algorithm and its earlier variants all gave great result that were comparable with, or even better then, state of the art algorithm that were investigated. This improvement was in terms of results, as for time complexity, our algorithms were all slower and computationally much more intensive. This provides a trade-off between current best heuristic approaches, that give fast but inaccurate results or our EO variants, that give better results, although slower. Both real-world inspired and synthetic network testing was done, and almost every network type showed the same relations between the algorithms, that were described earlier.

Highly cited journals network analysis

A singular real-world based test-case was proposed for the InfEO variant of the algorithm, as the rest of the influence based research was mainly theoretical. For the real-world testing, a network was constructed using data from the Web of Science (WoS)[1] article database, specifically, a citation network was created using articles from the domain of computer science. The articles were selected using a specific search query, incorporating category, year and the the *highly cited* label.[2].

1. (https://apps.webofknowledge.com/), last accessed 29.06.2023
2. https://images.webofknowledge.com/images/help/WOS/hs_citation_applications.html, last accessed 29.06.2023

In the network that resulted from this data, nodes were journals while links were citations between articles of different journals, a directed link existed between two nodes if the starting node had an article that referenced the ending node. This way the network was constructed from 606 articles, resulting in a number of nodes being 7482, out of which 131 had positive out-degrees. The network also contained and 14479 links. The 131 journals that had a positive out-degrees, were the prime candidates to be identified as influential nodes by our algorithm.

The analysis that was done on the highly cited journals network provided an interesting insight into a possible application of the influence maximization problem and shoved the usefulness of the InfEO algorithm in a pseudo-real-world scenario.

4. Chapter

Criticality in Networks

4.1 The Problem of Criticality and Critical Node Detection

One of the main research fields that concern network analysis is the field that investigates the problem of criticality in networks. The Critical Node Detection Problem (CNDP) is described in [27], or in the survey by [25], and it is the most common criticality problem that is investigated in the literature. CNDP can be simply described as a problem of identification. We need to identify important nodes in a network. Importance is defined according to a specific metric. It is crucial to distance this definition from the definition of influence, where the nodes needed to have maximal information diffusion ability. Here, the nodes need to be important in a network integrity sense.

The above description is vague on purpose since almost every part of the critical node detection problem can be modified, we could be looking for one or more nodes or even other components of the network such as in the survey [38], where the authors describe other critical element types and among them critical nodes. We can define an important node according to several metrics, we can even use different kinds of network types.

This variability is one of the causes of the CNDP being used in a large variety of studies in the literature. It was used for social network analysis in [10], [16], network vulnerability studies in [14] and network risk management in [3].

One of the main components of the CNDP is the measure used for detecting critical nodes. In the literature a number of measures were proposed, and even more are possible, with the main question asked by the researchers being: Why is a node critical? What makes a node critical?

In [2] there are three versions discussed as the answer to our questions, these are the most popular variations of the CNDP in the literature and they are the following: The *kMaxComp* problem, pairwise connectivity and the *MinMaxC* problem. These will be described in more detail shortly.

As a matter of complexity, several connectivity measures were looked at, and the CNDP was proved to be NP-hard for all of the investigated measures in [34], originating from this dilemma, different solving methods have been proposed, however *kMaxComp* was not thoroughly investigated.

Definition 4.1.1 (Critical Node Detection Problem (CNDP)). The Critical Node Detection Problem can be defined as the problem of finding a set of nodes, the set having a fixed size of k, in any given graph, such as, after the removal of the selected set, the graph would maximally degrade, according to an arbitrary measure σ.

Together with the *kMaxComp* problem, three distinct forms of σ were studied for traditional networks throughout the research process presented in this thesis. These problems defined by these σ measures are the following:

Definition 4.1.2 (kMaxComp problem). The kMaxComp problem consists of removing a the set of nodes, that would lead to the maximal number of remaining connected components in the damaged graph. Formally, if S denotes the set of deleted nodes, having a size of k, and $\mathcal{H}(G[V \setminus S])$ denotes the set of connected components of graph G after the removal of the selected set of nodes, basically the damaged graph, the *kMaxComp* can be described using the following equation:

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

where $|\cdot|$ denotes the cardinality of a set.

The kMaxComp problem was the main form of the possible CNDP variants that we used in basic criticality detection researches, such as the CN-EO algorithm, MAXC-GA algorithm, while also providing the base for our Hypergraph focused Hyp-GA algorithm. It is probably the most widely used metric out of the three main metrics found in the literature, and can be used for all sort of criticality detection, no matter the network in question.

Definition 4.1.3 (CNP - Pairwise Connectivity). In this case, we need to minimize the following objective function:

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

, where C_i is the set of nodes that are in a concrete connected component after the deterioration of the graph, while $|\cdot|$ denotes the cardinality of a set, the size of the component

For this equation we consider the connected components after the degradation of the graph, by the removal of the nodes selected by the CNDP problem.

The pairwise connectivity was used as a metric in our interpretation of the combined critical node and edge detection problem, since it can be greatly used to detect network deterioration. Our focus for that research was the use of the combined problem as a mean to detect critical network deterioration points, providing a possible network security metric. The use of the pairwise connectivity in terms of criticality is explained in detail paragraph 4.2.1, where an example is given for the calculations needed in this case.

Definition 4.1.4 (MinMaxC). This problem consists of minimizing the size of the largest component after the removal of the nodes selected by the algorithm as being potentially critical. Formally:

$$\min |(\max_{C \in \mathcal{H}} C)|,$$

where $\mathcal{H}$ is the set containing the connected components of the graph, while $|\cdot|$ denotes the cardinality of a set.

While testing was done using the MinMaxC problem, no concrete research was completely based on this variant of the CNDP, since we found, that for our purposes, the MinMaxC did not provide desired results, with the combination that this problem was computationally intensive, at least the initial implementation of it, a decision was taken, not to dive deeper into this variant. Nevertheless, it is still one of the more used approaches to the CNDP problem, meaning that a definition was needed.

Besides these three main use-cases, a fourth approach was also proposed by us in the form of using the hypergraph specific Weighted Node Degree Centrality as our metric, in the research that resulted the WNDC CNDP problem, presented in section 4.2.2. This was a novel approach, and as such, a detailed explanation was given in the appropriate section.

The CN-EO algorithm

As the name implies, CN-EO is an algorithm mainly based around the Extremal Optimization algorithm, that formed the basis of our research in the influence maximization side of things. Nonetheless, as we tried to innovate on our earlier approaches, we introduced a new Extremal Optimization variant as the basis for CN-EO, that being the NoisyEO algorithm from [29]. In that paper, the NoisyEO was successfully used for the community detection problem, and it proved to by adaptable to the problem of criticality, with a few notable changes. NoisyEO works by introducing a shifting procedure into the algorithm. Once the results seemingly stagnate for an extended number of iterations, the shifting procedure is initialized.

The shift procedure works by randomly modifying the network $G = (V, E)$ creating $G' = (V', E')$, where $V' \subseteq V$ and $E' \subseteq E$ are the set of new nodes and edges respectively. These new sets are acquired by the removal of random nodes and edges from the original V and E, the removal is done with a probability of p_{shift} which is a parameter of our algorithm. The new network G' is now used as the network in the EO algorithm for a set number of iterations nrG. During these iterations, the best solution so far can be modified, importantly we do not retain the result, just the set of potentially critical nodes. After these iterations, the original network is returned, but the proposed solution should be changed, and this way the search can escape a local optima. The algorithm was tested on some of the previously introduced benchmark networks and provided good result compared to the state of the art at the time.

The MAXC-GA algorithm

Another early approach for solving the CNDP was the MAXC-GA algorithm, which can be described as a simple genetic algorithm. The specific goal of this research was to create a solution for the CNDP while using the least amount of problem-specific information during the search phase. As the proposed algorithm is a genetic algorithm, there are some general pieces of information about the algorithm structure and use that need to be elaborated. We used a binary encoding, a two point crossover scheme, a uniform mutation scheme and tournament-based selection. The fitness value of an individual was computed using the number of remaining components after the graph deterioration caused by the individual's marked nodes.

While benchmark testing was successfully done on MAXC-GA, proving that it is a good contender for CNDP algorithm, for the paper that created the MAXC-GA algorithm, there was no real-world use case proposed in a similar fashion to the results proposed for the EO approach. Instead, the paper was proposed as a proof-of-concept paper, that advocated for the use of minimal problem-specific information for evolutionary algorithm design.

4.2 Generalizations of the Critical Node Detection Problem

4.2.1 The Combined Critical Node and Edge Detection Problem

Criticality in a network can relate to any form of network components, not just nodes. One of the components that is also investigated in the literature, and should provide interesting results are the edges of a graph. Besides the critical node detection problem, we can define the critical edge detection problem.

Definition 4.2.1 (Critical edge detection problem (CEDP)). Given a graph $G = (V, E)$, the objective of the problem is to find a set of l edges, that can be considered critical according to any given metric, that measures the degradation of graphs after the edges from l are removed.

The CNDP and the CEDP do coexist in the literature, we proposed an innovative approach to the problem of criticality, by combining the two problems into one, obtaining the combined critical node and edge detection problem (CNEDP), which is a much less studied approach to the criticality problem family. CNEDP can be used to simulate real-world scenarios, meaning it can be useful in various applications (e.g. road networks, computer networks, etc.) It can work by deleting not only nodes or edges but a combination of the two.

The problem of (k, l)-CNEDP

Definition 4.2.2 (The critical node and edge detection problem (CNEDP)). Given graph $G = (V, E)$, the CNEDP consists of finding a set W having a size of k, containing nodes from the original network, and a set F having a size of l, containing edges from the original network, which when simultaneously deleted will degrade the graph in a maximal manner, according to a given measure σ. The introduced problem is denoted as (k, l)-CNEDP.

We can identify an interesting problem that comes up from deleting two connected but separate graph components: nodes and edges. It should be obvious that removing a node will remove all edges that are connected to it, since there can be no edge between inexistent nodes, we needed to consider if removing an edge would remove the nodes connected to it. This possibility was quickly disregarded, since removing nodes would remove edges, and we would quickly end up with an empty graph.

The network connectivity measure that was used for this research was pairwise connectivity. In our case, we therefore needed to minimize the following objective function:

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

where $A \subseteq V$, C_i is a set that contains all connected components in the deteriorated graph, with δ_i being the size of the connected component C_i.

We can observe that finding an ideal set for critical components would lead to the complete disintegration of the graph, with the component sizes all being reduced to 1, which would result in an objective function value of 0. We can also identify, that setting the value of k, respectively l to 0 would reduce our problem to CNDP or CEDP respectively. Finally we can also observe that the CNEDP should be NP-complete since in [4], a variant of the CNP was proven to be NP-complete, with the CNP being a sub-task of our current problem.

Greedy approach to the (k, l)-CNEDP

In the research proposed in [2] there were three non-evolutionary greedy solutions that were presented for the CNDP. An adaptation of the second approach is used by us during this research, which made the second approach greedy work on the (k, l)-CNEDP. We used the pairwise connectivity in the greedy process, which works based on the following function:

$$GR2(S_X) = argmax\{(f(S_X) - f(S_X \cup \{t\}) : t \in X \setminus S_X\}$$

where S_X can be one of the proposed solution sets for nodes or edges, meaning that X represents either the original set of nodes respectively the original set of edges of the network. This function is described in [2] and it was adopted by us.

Genetic algorithm for the (k, l)-CNEDP

As an additional approach, we created a simple genetic algorithm in order to approach the CNEDP problem. The main operators are the following: list-based encoding, the fitness is calculated as the pairwise connectivity after deterioration, tournament-based selection, two mutation variants, both replacement based, with no need for a repair operator. A $(\mu + \lambda)$ selection scheme was also used.

Comparisons between the two methods Early comparisons were done using a set number of fitness calculations as a cut-off, but to provide fair comparisons, we needed to allow both algorithms to run their full course, however, the greedy algorithm was unable to realistically process larger networks thanks to the scaling of the algorithm being too exponential. More concrete comparison results followed suit, were the greedy calculations were not limited and the results were derived from full runs in cased were the greedy algorithm could finish. The results showed that in most cases the GA was better suited, but for some specific cases the greedy approach was better.

4.2.2 Critical Node Deletion in Hypergraphs

Attempts were made in order to combine the concept of Hypergraphs and the CNDP. Two main approaches were given for the Hypergraph based CNDP.

Definition 4.2.3 (The critical node detection problem in hypergraphs). Given a hypergraph $\mathcal{H} = (X, \mathcal{D})$, the CNDP for hypergraphs consists of weakly removing a set of k nodes in order to maximize the number of remaining connected components in the deteriorated graph.

Hyp-GA

Our earlier research on the topic of hypergraphs in critical nodes resulted in the Hyp-GA algorithm. As the name implies, it is yet another genetic algorithm implementation, this time having the focus on hypergraphs instead of new algorithm types. This algorithm worked by using a basic representation of a hypergraph, as a graph containing complete sub-graphs, or formally a *clique representation*, and then working on this representation as a simple graph. It uses all the same components and parameters as our previous GA based approach. While giving good results, the inefficient representation together with the blunt algorithm did not provide outstanding results.

Hypergraph critical nodes with weighted node degree centrality

Yet another hypergraph and criticality focused GA approach, with much more interesting results. A few main evolutionary tactics were introduced that were missing in the earlier approach: better representation using dedicated hypergraph libraries; a better algorithm scheme in general with an improved GA; and the introduction of the Weighted node degree centrality as a metric along which we could calculate criticality.

Weighted Node Degree Centrality The main scope of this research and the main differentiator compared to previous work is the introduction of a hypergraph-specific centrality measure as a criticality measure for our problem. This new metric is named the Weighted Node Degree Centrality (WNDC), which was presented in [22] as an attempt to extend the traditional centrality measures, common in graph research, to the realm of hypergraphs.

Generally, the w weights of a hyperedge take into account two metrics that describe the given edge. Multiplicity (m_j), which describes the frequency of a given hyperedge's appearance in the network, and cardinality (c_j), which pertains to the the number of nodes that are contained in said hyperedge. While the benchmark results were interesting, since there was no other research that we could compare to, the main attraction was the comparison with a self proposed greedy approach and the real-world application.

Comparison between a Heuristic and GA The validation of the results obtained by the GA should be an important step in validating the usefulness of the algorithm. We proposed a heuristic that similarly used WNDC, by removing the nodes with the highest WNDC. This result in turn gave a good base that we could compare to, since the intuition would say, that if we remove the k nodes with the highest WNDC then the results would be better than with any other combination of k removed nodes. In reality, our GA reached better combinations of removed nodes yet again showing that the whole node set has a larger impact, than the sum of its parts.

4.3 Proposed Applications for the Critical Node Detection Problem Variants

4.3.1 Practical use for the CNDP: stock market analysis

The paper that presented CN-EO was one of the more interesting research that has been done during this doctoral period, with a practical result that proved to be interesting. The practical use found for the CNDP and the CN-EO algorithm was an analysis of the stock market. For economic network analysis, there are a few examples in the literature, banking networks were nodes represent everything from banks banks to people in power in said banks and many more. [15].

Stock market analysis is an important aspect of economic network related work, with one of the first such papers being [12] with [17] also being important, as it analyzed the Chinese stock market from an influence perspective.

We used an unweighted, simlplified version of a stock market graph from [26], that was obtained from the analysis of the correlations between stock in the New York stock market for a period of two years. Using CN-EO, we calculated the most critical nodes. The size of the critical node set started from 3 and went all the way up to 8.

4.3.2 Application for the CNEDP: new network robustness metric

An application was proposed for the combined CNEDP, a new metric which could be used to measure network robustness, and which if produced promising results, would provide a good alternative for measures already present in the literature. In the literature, there exist several robustness measures, that usually try to calculate robustness by looking at different sets of network properties.

We tested our new proposed metric using a set of medium to large sized real-world networks. These included infrastructure networks[31],[36],[24] brain networks [1], power grid networks [31], interaction networks [35] and a computer network [24].

The new network robustness measure was named $NE_{k,l}$, which was based on our (k,l)-CNEDP algorithm which used the pairwise connectivity as its criticality criteria. The new metric has the following form:

$$NE_{k,l} = \frac{2 \cdot (k,l)\text{-CNEDP}}{(n-k-1)(n-k-2)} \in [0,1]$$

.

An interesting observation can be done on the terms of this equation. The equation contains the worst possible results of the pairwise connectivity after the removal of k nodes, meaning that it would always be between 0 and 1 since the CNEDP can at best find this result and realistically only approach it.

$NE_{k,l}$ can qualify as a good robustness metric based on the results, with it giving independent result from other metrics and also working on non-connected graphs. It could also be configured, thanks to its two parameters k and l. These facts made this achievement one of the more important contributions of our research process throughout the years.

4.3.3 Application for CNDP on Hypergraphs: An inflation hypergraph analysis

We propose as a real-world application, the use of our algorithm on a hypergraph constructed from real-world inflation data. Data for about 123 countries were publicly available[1] and contained information about inflation rate from 1960 until 2019. Since then the site has been taken down, but the results stand nonetheless. An analysis was done on the last ten years of data, from 2010 to 2019. With countries eliminated if they had incomplete data, we remained with 98 countries. A hypergraph is then constructed from this data by considering countries as nodes, while hyperedges represent different inflation brackets in a given year, meaning there exists a hyperedge that contains countries that had for example, a negative inflation rate in a studied year, etc. Actually four hypergraphs were obtained for four sets of years from the marked interval (2010-2012,2013-2015,2015-2017,2017-2019). This representation can be useful, since it represents the dynamics of inflation for countries, since a hypergraph contains several values from any given year.

A value of 10 is chosen for k, meaning we want to identify the 10 most critical nodes in the network. The results can provide a glimpse into possible unstable regions or regions with similar rates of change over the years since a critical node in this setting would mean countries that changed inflation rates multiple times during the investigated period.

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

4.3.4 Application for CNDP on Hypergraphs: U.S. Congressional and Senatorial committee data analysis

As a CNDP application on hypergraphs, we investigated two real-world networks, both derived from US congressional data, created by Charles Stewart and Jonathan Woon. These networks were initially used in [13] and we implemented them for our real-world application and proof of algorithm correctness. These networks aggregate memberships in US congressional committees, either in the House or the Senate.

Since the results presented here are real-world data-driven, an interpretation of the results could be useful.

An interesting result of this research shows how much looser the House structure is comparatively to the Senate structure. In the house network, there are a total of 1290 nodes, out of which 51% are present at least once in the critical list. If we instead focus on the nodes that appeared in at least half of the total critical sets, this number lowers to a mere 4.5%, whereas if we looked at 75% of the total critical sets, we would only get 2 critical nodes, which corresponds to an incredibly small 0.015% of all nodes. We can do similar calculations on the Senate network results. Out of the 282 nodes, only 28.7% appear once, 9.9% appear in half of the lists and 3.2% appear in 75% of the total critical sets.

This data is not at all definitive, but the principle idea behind the use of this data can lead to the identification of key players among America's political elite.

5. Chapter

Conclusions and future work

Influence Maximization For Influence Maximization, the introduction of so many innovations, starting from the EO algorithm, to the combination of EO with Cascade and Game Theory, the introduction of the Shapley value and the subsequent improvements on our process, all led to a final SIM-EO algorithm, that can be considered complete. There is still much to work on, potential future prospects include the analysis of the possible reintroduction of the Inf-EO style improved EO, which was discarded early in our research due to computational constraints. Another option would be to investigate different game theory value, such as the Banzhaf value [6] or different propagation models, such as the linear threshold model. We could also investigate different algorithm types, such as a GA approach to the problem of influence. Finally, the problem of online influence maximization could also be investigated, basically influence maximization on an evolving network, with non-stable k values. All in all, influence maximization was a topic that resulted in exciting results and it should be worth revisiting in the future.

Critical Node Detection In contrast to the influence maximization problem, the critical node detection problem did not have a final unifying algorithm proposed. We diverted our focus on many variants of the problem of criticality, from well known variants such as CNDP to completely new approaches such as WNDC-CNDP on hypergraphs. The future can be interesting for the problem of criticality. Fistly, we could always go on the route of creating the complete collection of improvements, similarly to the influence part, we could also combine already existing parts of our criticality research, or we can investigate new concepts, such as other critical components or new algorithms for existing problems.

The CNDP was and remains an extremely important research topic, because of the whole security aspect of this problem. The results so far provide us with sufficient motivation to continue the research in this field.

Bibliography

[1] K. Amunts, C. Lepage, L. Borgeat, H. Mohlberg, T. Dickscheid, M.-É. Rousseau, S. Bludau, P.-L. Bazin, L. B. Lewis, A.-M. Oros-Peusquens, N. J. Shah, T. Lippert, K. Zilles, and A. C. Evans. Bigbrain: An ultrahigh-resolution 3d human brain model. *Science*, 340 (6139):1472–1475, 2013.

[2] 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.

[3] 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.

[4] 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.

[5] P. Bak. The discovery of self-organized criticality. In *How nature works: The science of self-organized criticality*, pages 33–48. Springer, 1996.

[6] J. F. Banzhaf. Weighted voting does not work: A mathematical analysis. In *Rutgers Law Review*, 1965.

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

[8] S. Boettcher and A. Percus. Nature's way of optimizing. *Artificial Intelligence*, 119(1-2): 275–286, 2000.

[9] S. Boettcher and A. G. Percus. Optimization with extremal dynamics. *Complexity*, 8(2): 57–62, 2002.

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

[11] D. Bucur and G. Iacca. Influence maximization in social networks with genetic algorithms. In *European conference on the applications of evolutionary computation*, pages 379–392. Springer, 2016.

[12] 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.

[13] P. S. Chodrow, N. Veldt, and A. R. Benson. Hypergraph clustering: from blockmodels to modularity. *Science Advances*, 2021.

[14] T. N. Dinh and M. T. Thai. Precise structural vulnerability assessment via mathematical programming. In *2011-MILCOM 2011 Military Communications Conference*, pages 1351–1356. IEEE, 2011.

[15] 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.

[16] N. Fan and P. M. Pardalos. Robust optimization of graph partitioning and critical node detection in analyzing networks. In *International Conference on Combinatorial Optimization and Applications*, pages 170–183. Springer, 2010.

[17] 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.

[18] 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.

[19] F. Gursoy and D. Gunnec. Influence maximization in social networks under deterministic linear threshold model. *Knowledge-Based Systems*, 161:111–123, 2018.

[20] K. Hussain, M. N. Mohd Salleh, S. Cheng, and Y. Shi. Metaheuristic research: a comprehensive survey. *Artificial intelligence review*, 52(4):2191–2233, 2019.

[21] W. Jiang. Graph-based deep learning for communication networks: A survey. *Computer Communications*, 185:40–54, 2022.

[22] K. Kapoor, D. Sharma, and J. Srivastava. Weighted node degree centrality for hypergraphs. In *2013 IEEE 2nd Network Science Workshop (NSW)*, pages 152–155, 2013. doi: 10.1109/NSW.2013.6609212.

[23] 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.

[24] 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. ISBN 9781450320382. doi: 10.1145/2487788.2488173. URL https://doi.org/10.1145/2487788.2488173.

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

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

[27] 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. ISSN 0022-0000.

[28] Y. Li, J. Fan, Y. Wang, and K.-L. Tan. Influence maximization on social graphs: A survey. *IEEE Transactions on Knowledge and Data Engineering*, 30(10):1852–1872, 2018.

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

[30] J. C. Mitchell. Social networks. *Annual review of anthropology*, 3(1):279–299, 1974.

[31] R. A. Rossi and N. K. Ahmed. The network data repository with interactive graph analytics and visualization. In *AAAI*, 2015. URL http://networkrepository.com.

[32] P. Shakarian, A. Bhatnagar, A. Aleali, E. Shaabani, and R. Guo. The independent cascade and linear threshold models. In *Diffusion in Social Networks*, pages 35–48. Springer, 2015.

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

[34] 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.

[35] SocioPatterns. Infectious contact networks. URL http://www.sociopatterns.org/datasets/.

[36] L. Šubelj and M. Bajec. Robust network community detection using balanced propagation. *The European Physical Journal B*, 81(3):353–362, 2011.

[37] R. C. Thomson and D. E. Richardson. A graph theory approach to road network generalisation. In *Proceeding of the 17th international cartographic conference*, pages 1871–1880, 1995.

[38] J. L. Walteros and P. M. Pardalos. *Selected Topics in Critical Element Detection*, pages 9–26. Springer New York, New York, NY, 2012. ISBN 978-1-4614-4109-0. doi: 10.1007/978-1-4614-4109-0_2. URL https://doi.org/10.1007/978-1-4614-4109-0_2.