Heuristics for the Influence Maximization Problem on Hypergraphs

Vincenzo Auletta[1[0000−0002−7875−3366]], Francesco Cauteruccio[1[0000−0001−8400−1083]], and Diodato Ferraioli[1[0000−0002−7962−5200]]

Department of Information Engineering, Electrical Engineering and Applied Mathematics (DIEM)
University of Salerno, I84084, Fisciano (Italy)
{auletta,fcauteruccio,dferraioli}@unisa.it

Abstract. While the Influence Maximization (IM) problem has been widely analyzed on graph topologies, with plenty of algorithms and heuristics proposed, very limited interest has been devoted to the study of this problem on more complex, but much more expressive, hypergraph topologies. In this extended abstract, we address this issue by illustrating three families of heuristics for this problem: the first is based on node importance measures appropriately defined on hypergraphs, based both on its topological features, and on cooperative game theory concepts; while the remaining ones cast the problem within the well-known metaheuristic frameworks of Hill Climbing and Evolution Strategy, respectively. We experimentally evaluated these approaches, and we show that they often outperform the best-proposed algorithms in the state of the art.

Keywords: Influence Maximization · Hypergraphs · Shapley Value · Hill Climbing · Evolution Strategies

1 Introduction

The rise of social media and viral marketing has emphasized the importance of understanding how information, behaviors, and innovations spread through networks. Central to this study is the influence maximization (IM) problem, a pivotal concept in network theory. This problem has gained significant attention in viral marketing, personalized recommendations, and information dissemination, aiming to identify a set of key individuals in a network whose collective influence is maximized [13, 5]. It impacts various applications, including viral marketing, opinion formation, combating misinformation, link recommendation, and election manipulation on social networks [3, 7]. The goal is to find the minimal set of nodes (or seeds) in a graph that can maximize information spread under certain propagation models.

Traditionally, the IM problem has focused on ordinary networks with dyadic relationships [5, 15], such as in the seminal study in [13] that provided hardness results and an algorithm with an approximation rate of $(1-1/e)$ under the independent cascade (IC) and linear threshold (LT) diffusion models. Over the years, various approaches have emerged, ranging from algorithms with guaranteed approximations to heuristic solutions [12, 20, 19, 9]. However, recent years have seen the emergence of hypernetwork science, a branch concentrating on higher-order interactions within complex systems [6]. Hypergraphs, the primary modeling tool of this branch, generalize graphs by allowing edges to connect multiple vertices, thus offering a richer representation of complex systems [1]. Despite their potential, the IM problem on hypergraphs remains underexplored. In early studies on IM in hypergraphs, researchers represented multimedia social networks as hypergraphs and transformed them into bipartite graphs for algorithm application [2]. The IM problem's NP-hardness in hypergraphs was demonstrated, with a framework for sandwich approximation and an algorithm preserving a $(1 - 1/e - \epsilon)$-approximation presented [22]. Other studies have proposed heuristics and novel algorithms to address the IM problem on hypergraphs under various diffusion models, including the HyperCascade and SICP models [21, 11].

Our work marks the first attempt to tackle the IM problem on hypergraphs by *(i)* considering centrality measures based on game-theoretic concepts and *(ii)* employing metaheuristic approaches, creating a novel intersection with node properties-based perspectives. Unlike previous methods that transform hypergraphs into simpler forms, our approach operates directly on hypergraphs and is distinguished by its generality. We propose two heuristic families: node properties-based algorithm and metaheuristics-based algorithms. The first involves the SmartPROPS algorithm, which uses centrality values to select seeds while avoiding overlap. This algorithm is specialized by employing different centrality measures, including new ones based on cooperative game theory. The second family includes two metaheuristic algorithms: HC, based on random-restart steepest ascent hill climbing, and ES, based on evolution strategy, which simulates competition among solutions.

2 Methods

2.1 Background

A *hypergraph* $H = (V, E)$ is a pair consisting of a set $V = \{v_1, \ldots, v_n\}$ of elements called *nodes*, and a family of sets $E = (e_1, \ldots, e_m)$ called *hyperedges*. A hyperedge represents a relation among a subset of vertices in V, i.e., $e_j \subseteq V$, for all $j = 1, \ldots, m$. The *order* of a hypergraph is its number of nodes, i.e., $n = |V|$, while the *size* of a hypergraph is its number of hyperedges, i.e., $m = |E|$. Instead, the size of a hyperedge is its cardinality, i.e., $|e_j|$. We say that a node $v_i \in V$ belongs to a hyperedge $e_j \in E$ if $v_j \in e_j$. The *degree* of a node v_i, denoted as $d(v_i)$, is the number of neighbors of v_i; a node $v_l \in V$ is the neighbor of node v_i if and only if there exists at least one hyperedge e_j which v_l and v_i both belong to. The *hyperdegree* of a node v_i, denoted as $d^H(v_i)$, is the number of hyperedges to which v_i belongs. The *line graph* of H, denoted as $L(H)$, is the graph on node set $E' = \{e'_1, \ldots, e'_m\}$ and edge set $V' = \{\{e'_i, e'_j\} : e_i \cap e_j \neq \emptyset \text{ for } i \neq j\}$ [1]. Practically speaking, $L(H)$ is the graph where nodes represent the hyperedges, and there is an edge between two nodes if the two hyperedges share at least one common node in H. Furthernore, given a set V of nodes and a function $\nu\colon 2^V \to \mathbb{R}$, that assigns a measure of importance to each subset of nodes, the *Shapley value* of $v \in V$ [17] with respect to ν is defined as the average of the marginal contribution of v to the subsets at which she belongs, i.e., how much she increases the importance of these groups. The Shapley value can be efficiently computed for the following specific choices of ν [16], namely: *(i)* $\nu_{deg}(S)$, that measures the importance of a subset S of nodes as its size and the number of neighbors; (ii) $\nu_{close}(S)$, that measures the importance of a subset S of nodes as the inverse of the minimum distance between nodes outside S from nodes in S. We use these two choices in our approach.

Given a hypergraph $H = (V, E)$, a value $k \in \mathbb{Z}_{>0}$, and a diffusion process model on hypergraph σ_H, the Influence Maximization (IM) problem on hypergraphs consists in finding a subset $S^* \subseteq V$ of k nodes, called *seed node set*, such that the expected number of infected nodes is maximized. Formally, $S^* = \arg\max_{S\subseteq V, |S|=k} \sigma_H(S)$, where $\sigma_H(S)$ indicates the expected influence (i.e., the number of reached nodes) of the seed node set S at the end of the process. This definition depends on a diffusion process model σ_H. Diffusion process models are theoretical frameworks designed to simulate how information, behaviors, or innovations spread through a network. Different diffusion process models have been used for the IM problem [5]. We study the use Susceptible-Infected (SI) model with Contact Process (CP) dynamics on hypergraphs (SICP [21]). In this model, a node can be either in a susceptible (S) or infected (I) state. An S-state node can be infected by each of its neighbors in the I-state with a given infection rate β. The model works as follows: *(i)* Nodes in the seed set are set to be infected (I-state), and the remaining nodes are susceptible (S-state); *(ii)* At each time step t, we find the I-state nodes. For each I-state node v_i, we find all hyperedges E_i containing the node v_i. Then, a hyperedge e_j is chosen from E_i uniformly at random. Then, each of the S-state nodes in e_j will be infected by v_i with probability β; *(iii)* The process terminates after T steps, and we set $\sigma_H(S)$ to be the number of nodes in I-state. We choose this model since it allows us to focus on the relation between the performances of proposed algorithms and the topology of networks, without struggling against the complexity of the diffusion model. However, our approach is independent from the specific diffusion model, and can be immediately tested against more complex models, such as IC and LT.

2.2 Node Properties-based Approaches

Typically, heuristics-based solutions have been proven effective in different variants of the IM problem. A category of heuristic-based solutions is the one focusing on the centrality property of nodes, i.e., nodes are selected as seeds based on their centrality value. Centrality is a well-known measure in network analysis which indicates the importance of nodes in a network. In this section, we propose a general, simple yet surprisingly effective algorithm based on node properties.

We address the influence maximization (IM) problem on hypergraphs using a heuristic-based algorithm called SmartPROPS, which leverages node centrality to construct an optimal seed set. The algorithm sorts nodes based on a node property function $\phi \colon V \to \mathbb{R}$, then selects candidate seeds through a sequential process using a threshold function $\rho \colon V \to \mathbb{R}$. This function measures the proportion of overlapping hyperedges between candidate nodes and those already in the seed set. The algorithm operates on the hypergraph H and takes as input the number of seed nodes k. It creates an ordered list of nodes based on ϕ, assuming higher values indicate greater centrality. It iteratively builds the seed set R^*, starting with the highest-ranked node. For each node v_i, the algorithm calculates its threshold value using ρ and assesses the overlap of hyperedges with R^*. Node v_i is added to R^* if the overlap proportion meets or exceeds $\rho(v_i)$; otherwise, it is skipped. ρ helps manage hyperedge overlaps among candidate seeds. In our experiments, we use $\rho(v_i) = d^H(v_i)/m$.

We propose four different variants of the algorithm, namely: *(i)* SMARTDEG, in which the seeds coincide with the top-k highest degree nodes [5], and ϕ is the degree centrality, i.e., $\phi(v_i) = d(v_i)$ for each $v_i \in V$; *(ii)* SMARTHYPERDEG, similar as the previous one, it exploits the hyperdegree of each node; *(iii)* SMARTSHAPDEG, the Shapley Degree value, computed on $L(H)$, is used as the node property; *(iv)* SMARTSHAPCLOSE, here the Shapley Closeness is used as the node property instead.

2.3 Metaheuristics-based Approaches

The second family of algorithms proposed is metaheuristics-based. Metaheuristics are optimization algorithms designed to find, generate, or select a heuristic that may provide a sufficiently good solution to complex optimization problems [10]. We present two algorithms, named HC and ES.

HC: Hill Climbing-based Algorithm Hill climbing (HC) is a simple yet powerful and versatile metaheuristic optimization algorithm that iteratively seeks to improve a solution with respect to a given measure of quality. It has been used extensively in disparate contexts, from graph partitioning [14] to finding optimal matching in advanced sequences comparison algorithms [8].

HC is based on a random-restart steepest ascent hill climbing approach. The algorithm begins by randomly selecting an initial solution R of size k and computing its local expected influence. During each iteration, a neighboring solution R' is generated using a perturbation function, and its expected influence is calculated. If R' has a greater influence than the current solution R, R is updated to R'. Otherwise, the global best solution R^* is updated to R. The algorithm stops when no further improvement is possible, with R^* as the final output. In our experiments, we use four variants: HC_1, where a node from R is replaced with one of its neighbors chosen randomly from the largest hyperedge containing the former; HC_2, similarly as HC_1 but the node to be replace is the one with the smaller degree value; HC_3 and HC_4 replace the node having respectively the smallest Shapley Degree value and the smallest Shapley Closeness value.

ES: Evolution Strategy-based Algorithm Evolution Strategy is a population-based metaheuristic inspired by the principles of biological evolution [4, 18]. An Evolution Strategy algorithm aims to improve the quality of a collection of candidates through the generation of descendants from parents by mutation operators, guided by an objective function [18].

ES is based on a variant of Evolution Strategy called $(\mu+\lambda)$. Here, μ denotes the number of candidate solutions in the parent generation, while λ is the number of candidate solutions obtained from the parent generation. The algorithm considers a number of generations for evolution. At each generation, the best μ solutions are kept from the λ candidates and their parents. At the start, ES initializes a population of μ individual, chosen uniformly at random. An individual is a solution of the problem. In each generation, the algorithms iterates λ in order to generate the same amounts of new individual via a mutation operator. This operator implements the same variations proposed for the HC algorithm, leading to the four different variants ES_1, ES_2, ES_3, and ES_4. The algorithm terminates when the maximum number of generations is reached, and the output is the individual with the largest goodness value in the population, i.e., the greatest expected influence.

3 Preliminary Results

We now present some preliminary results of the proposed algorithms on eight real-world hypergraphs, currently used as benchmark datasets in the literature [21, 11], namely, *Algebra*, *Restaurant-Rev*, *Geometry*, *Music-Rev*, *NDC-classes*, *Bars-Rev*, *iAF1260b*, and *iJO1366*. These experiments have been performed on a 2.3GHz MacBook computer (Intel Core i9) with 16 GB of memory. All algorithms have been implemented in Python 3.10. To assess the performance of our proposed algorithms, we compare them with different baselines, namely, Degree, Greedy, HADP, and Adeff. The first one is a common baselines extensively used in literature [21], based on selecting the top-k nodes with the maximal degree. The second one is inspired by the approach in [13], and selects the node with maximal influence in each iteration. HADP, and Adeff are two recently proposed approaches for the problem [21, 11]. As for parameters, for HC, we set the number of restarts to 25, while for ES, we set $\mu = 4$, $\lambda = 10$, and the maximum number of generations to 25.

The obtained influence spread curve is presented in Figure 1, averaged over 100 runs. Here, we only consider HC_1 and ES_1. The x-axis refers to the value of k, while the y-axis reports the average expected influence $\sigma_H(S)$. Inside each plot, a smaller one reports the obtained value at $k = 25$. We can see how the metaheuristics-based algorithms generally perform better than the remaining ones. Also, they achieve the largest expected influence when $k = 1$. This is somewhat expected, considering that neighbor solutions and mutations are taken into account respectively. Except for some cases where the Greedy reaches similar values, no other algorithms perform in the same manner. As far as $k = 25$

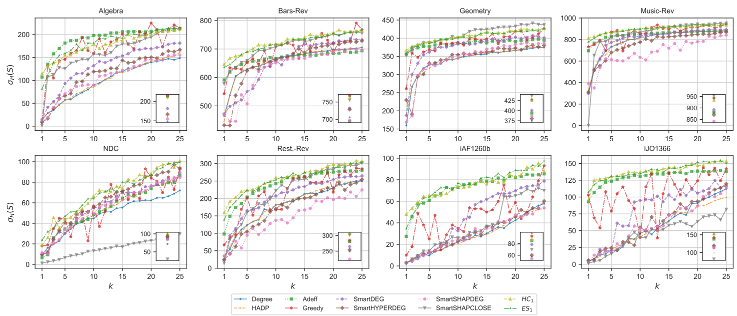

Fig. 1. Expected influence $\sigma_H(S)$ obtained with different values of k (from 1 to 25), averaged over 100 runs, and with the SICP parameters being $\beta = 0.01$ and $T = 25$.

is concerned, we observe an almost identical behavior as the previous one. Here, HC_1 and ES_1 perform effectively, together with the GREEDY one.

4 Conclusion

In this extended abstract, we presented an approach to tackle the influence maximization problem on hypergraphs. We proposed two families of algorithms: *(i)* node properties-based, and *(ii)* metaheuristics-based. The node properties-based family includes a general algorithm called SMARTPROPS, which selects seed candidates based on specific properties. The metaheuristics-based family comprises two algorithms, HC and ES, which use hill climbing and evolution strategy approaches, respectively. We introduced variants for these algorithms, particularly those using game-theoretic centrality measures. Our preliminary experiments demonstrated the effectiveness of our approaches and highlighted the promise of game-theoretic centrality measures in solving this problem.

Acknowledgment

This work is supported by the PNRR project FAIR—Future AI Research (PE00000013) under the NRRP MUR program funded by NextGenerationEU.

References

1. Aksoy, S.G., Joslyn, C., Marrero, C.O., Praggastis, B., Purvine, E.: Hypernetwork science via high-order hypergraph walks. EPJ Data Science **9**(1), 16 (2020)
2. Antelmi, A., Cordasco, G., Spagnuolo, C., Szufel, P.: Social influence maximization in hypergraphs. Entropy **23**(7), 796 (2021)
3. Auletta, V., Ferraioli, D., Greco, G.: On the complexity of reasoning about opinion diffusion under majority dynamics. Artificial Intelligence **284**, 103288 (2020)

4. Back, T.: Evolutionary algorithms in theory and practice: evolution strategies, evolutionary programming, genetic algorithms. Oxford University Press (1996)
5. Banerjee, S., Jenamani, M., Pratihar, D.K.: A survey on influence maximization in a social network. Knowledge and Information Systems **62**, 3417–3455 (2020)
6. Battiston, F., Cencetti, G., Iacopini, I., Latora, V., Lucas, M., Patania, A., Young, J.G., Petri, G.: Networks beyond pairwise interactions: Structure and dynamics. Physics Reports **874**, 1–92 (2020)
7. Castiglioni, M., Ferraioli, D., Gatti, N., Landriani, G.: Election manipulation on social networks: Seeding, edge removal, edge addition. Journal of Artificial Intelligence Research **71**, 1049–1090 (2021)
8. Cauteruccio, F., Terracina, G., Ursino, D.: Generalizing identity-based string comparison metrics: Framework and Techniques. Knowledge-Based Systems **187**, 104820 (2020)
9. Chen, W., Yuan, Y., Zhang, L.: Scalable influence maximization in social networks under the linear threshold model. In: Proc. of the 2010 IEEE International Conference on Data Mining (ICDM). pp. 88–97. IEEE (2010)
10. Gendreau, M., Potvin, J.Y.: Handbook of metaheuristics, vol. 2. Springer (2010)
11. Gong, X., X., H.W., Wang, Chen, C., Zhang, W., Zhang, Y.: Influence maximization on hypergraphs via multi-hop influence estimation. Information Processing & Management **61**(3), 103683 (2024)
12. Goyal, A., Lu, W., Lakshmanan, L.V.: Celf++ optimizing the greedy algorithm for influence maximization in social networks. In: WWW. pp. 47–48 (2011)
13. Kempe, D., Kleinberg, J., Tardos, E.: Maximizing the spread of influence through a social network. In: KDD. pp. 137–146 (2003)
14. Li, H., Liu, Y., Yang, S., Lin, Y., Yang, Y., Yoo, J.: An improved hill climbing algorithm for graph partitioning. The Computer Journal **66**(7), 1761–1776 (2023)
15. Liu, P., Li, L., Wen, Y., Fang, S.: Identifying influential nodes in social networks: Exploiting self-voting mechanism. Big Data **11**(4), 296–306 (2023)
16. Michalak, T., Aadithya, K., Szczepanski, P., Ravindran, B., Jennings, N.: Efficient computation of the shapley value for game-theoretic network centrality. Journal of Artificial Intelligence Research **46**, 607–650 (2013)
17. Shapley, L.: A value for n-person games. Contributions to the Theory of Games pp. 307–317 (1953)
18. Simon, D.: Evolutionary optimization algorithms. John Wiley & Sons (2013)
19. Tang, Y., Shi, Y., Xiao, X.: Influence maximization in near-linear time: A martingale approach. In: Proc. of the 2015 ACM International Conference on Management of Data (SIGMOD). pp. 1539–1554 (2015)
20. Wang, C., Chen, W., Wang, Y.: Scalable influence maximization for independent cascade model in large-scale social networks. Data Mining and Knowledge Discovery **25**, 545–576 (2012)
21. Xie, M., Zhan, X.X., Liu, C., Zhang, Z.K.: An efficient adaptive degree-based heuristic algorithm for influence maximization in hypergraphs. Information Processing & Management **60**(2), 103161 (2023)
22. Zhu, J., Zhu, J., Ghosh, S., Wu, W., Yuan, J.: Social influence maximization in hypergraph in social networks. IEEE Transactions on Network Science and Engineering **6**(4), 801–811 (2018)