Hypergraph Partitioning for Bibliometric Term Blocking: An Application of the KaHyPar Framework to IEEE Xplore Data

Boris N. Chigarev *

Posted Date: 27 March 2026

doi: 10.20944/preprints202603.2224.v1

Keywords: bibliometric analysis; term partitioning; KaHyPar framework; IEEE Xplore terms; hypergraph techniques

 doi:10.20944/preprints202603.2224.v1

Article

Hypergraph Partitioning for Bibliometric Term Blocking: An Application of the KaHyPar Framework to IEEE Xplore Data

Boris N. Chigarev

Institute of Oil and Gas Problems Russian Academy of Sciences 3, Gubkina str., 119333 Moscow, Russian Federation; bchigarev@ipng.ru; ORCID: 0000-0001-9903-2800

Abstract

The prevalence of graph-based approaches in bibliometric analysis is limited to considering only pairwise relationships—such as co-authorship, citation lists, or keywords—which artificially simplifies the structure of these relationships. Hypergraphs allow for the direct modeling of numerous relationships, thereby improving the accuracy of the analysis. This study demonstrates the application of the KaHyPar framework to partition sets of IEEE Xplore terms into blocks, treating them as hyperedges. This is the first in a series of articles detailing hypergraph-based term blocking techniques. The study utilized bibliometric data from 2021 to 2025, exported from the IEEE Xplore database using the terms "Artificial Intelligence," "Blockchain," "Data Science," "Deep Learning," "Image Processing," "Internet of Things," "Anomaly Detection," and "Machine Learning." After removing duplicates and excluding 16 records with empty "IEEE Terms" fields, 38,157 records were used for the study. While partitioning IEEE Terms records with KaHyPar is effective, the process requires rigorous data preparation for the .hgr format. This complexity explains why hypergraphs remain underutilized in scientometrics compared to more accessible tools like VOSviewer. The proposed significance criterion—based on a term's occurrence frequency within hyperedges associated with the block—yielded easily interpretable results. Future studies should investigate the impact of KaHyPar parameters and evaluate alternative frameworks such as HYPE and Mt-KaHyPar, alongside other metrics for term significance.

Keywords: bibliometric analysis; term partitioning; KaHyPar framework; IEEE Xplore terms; hypergraph techniques

Introduction

Relevance and Expediency of Using Hypergraphs in Bibliometric Analysis

Let us present a concise historical overview of the development of the field of scientometrics and bibliometrics. In 1961, Eugene Garfield initiated a project to develop the first citation index, which was tested using genetic literature. The full-scale commercial edition, titled Science Citation Index (SCI), was published in 1964. In 1997, the Web of Science platform was introduced, originally known as Web of Knowledge and built upon the Science Citation Index created by Eugene Garfield

The Scopus platform, launched by Elsevier on March 5, 2004, serves as a comprehensive abstract database of peer-reviewed scientific literature.

In Soviet literature, the term "scientometrics" was officially proposed and explained in the monograph by [1].

The international scientific journal Scientometrics was founded in 1978 by the Hungarian chemist and scientometrician Tibor Braun [2,3], who became its first editor-in-chief. The first editorial board included prominent figures in the field, such as Derek de Solla Price [4], Eugene Garfield, and

 doi:10.20944/preprints202603.2224.v1

Soviet scientist Vasily Nalimov, recognized as key figures in establishing the foundations of scientometrics.

The bibliographic coupling method was proposed in 1963 in the publication "Bibliographic Coupling Between Scientific Articles." [5]. Bibliographic coupling analysis is a bibliometric method that allows the identification of thematic proximity among scientific publications based on common references in their bibliographies. The issue that the list of references is a hyperedge was not addressed.

The co-citation method was proposed in 1973 independently by two researchers: Henry Small [6] and Irina Marshakova [7]. The essence of the method: two documents are considered "co-cited" if they are referenced in the same third document. The frequency of such matches serves as a measure of the semantic proximity of such works.

An crucial distinction between bibliographic coupling and co-citation methods. Bibliographic coupling is defined using a classic "Object — Feature" matrix (Document × Reference). It is characterized by staticity, as the link created upon publication remains unchanged, making the method basic and reliant on the fixed data of the document itself. Co-citation refers to a dynamic structure, represented as a Reference × Reference matrix or adjacency matrix. It evolves over time as new articles are written, causing the co-citation metrics of older works to rise. The Document × Reference Matrix (citation matrix) is a raw dataset from which both methods can be derived. Moreover, the Document × Reference Matrix is actually an incidence matrix of a hypergraph.

At the same time, the use of adjacency matrices remains predominant. The most common workflow in academic articles is as follows: Scopus/WoS data → VOSviewer → standard graph → standard article. This cycle ignores n-ary relationships (where a specific combination of 5 authors or 10 keywords forms a unique set of objects that cannot be represented using simple pairwise links) — this is the price that must be paid for simplifying the analysis.

A hyperedge represents a specific set of entities, while its transition to a graph converts it into a clique, disrupting the original unity and reducing entities to individual pairwise relationships. Hypergraph methods are beneficial for identifying system patterns and properties that traditional simple graph methods may overlook.

The VOSviewer has been successful because it has made bibliometrics visual and accessible, but this user-friendliness has not encouraged the development of alternatives. Therefore, specific case studies that make the analysis of hypergraphs intuitive could help to create applications for using hypergraphs in bibliometric analysis that are just as user-friendly as VOSviewer.

A Brief Literary Review

As noted above, scientometrics began to take shape in the first half of the 1960s—at a time when researchers engaged in bibliometric analysis had neither extensive abstract databases nor accessible computing resources. At the same time, the value of the hypergraphical approach had been recognized for some time.

As noted above, scientometrics began to take shape in the first half of the 1960s, a time when bibliometric researchers did not have access to extensive abstract databases or computational resources. However, the value of the hypergraphical approach had been recognized for some time.

Thus, the article [8] quantitatively investigates the social and socio-semantic patterns of forming academic teams for collaboration. Formally, this requires the use of hypergraphs and n-ary interactions, rather than traditional dyadic interactions, such as graphs that connect only pairs of agents. The paper [9] presents a new model using hypergraphs to show how citation networks change over time. The model combines preferential attachment and aging effects. Simulation results indicate that this model effectively represents the citation distribution found in real systems.

One of the first works that is not directly related to scientometrics but is very close in meaning notes that anthropologists and sociologists have focused on studying social networks formed by relationships between pairs of people. However, these networks can be distorted if viewed solely through pairwise relationships. The article [10] proposes a new way to study structures arising from

 doi:10.20944/preprints202603.2224.v1

non-pairwise relationships using ideas from hypergraph theory, which can aid in the investigation of various structural issues.

In more recent studies, there are also publications by Russian authors, for example, the work [11] introduces a model for co-authorship networks, focusing on relationships among co-authors. It uses a hypergraph, where authors are the nodes and publications are the hyperedges. The text explains a method for building this co-authorship network hypergraph using data from a journal archive.

Despite the fact that the first articles appeared quite a long time ago and the significance of using hypergraphs in bibliometric analysis is clear, the number of scientific publications on this subject remains small.

For example, a query to the open abstract database scilit with Subject: Scientometrics & Research Ethics OR Information & Library Science, AND Publication Type: JOURNAL-ARTICLE, AND Language: English, AND 2023–2025 Years, resulted in only 7 publications containing the term Hypergraph in Title, Abstract, and Keyword, while a query without restrictions on publication years resulted in only 33 papers. In total, 836,590 publications were indexed in these two sections. However, when using Hypergraph AND partition (a topic related to community detection, a typical bibliometric task), no publications were found in this database or the specified Subjects.

Thus, a search of the open abstract database Scilit with the following filters: Subject: Scientometrics & Research Ethics OR Information & Library Science, AND Publication Type: JOURNAL-ARTICLE, AND Language: English, AND 2023–2025, yielded only 7 publications containing the term “Hypergraph” in the Title, Abstract, Keyword, while without restrictions on publication years, 33 papers were found. A total of 836,590 publications are indexed in these two sections. When using the search terms “Hypergraph AND partition” (a topic related to community detection, a typical task in bibliometrics), no publications were found in this database under the specified Subjects.

By comparison, a search using the same restrictions on VOSviewer returned 5,795 publications. However, no publications were found for one of the most advanced programs for hypergraph partitioning—KaHyPar [12]—nor for the more widely used HyperNetX (Python package for hypergraph analysis and visualization) [13].

The situation is no better when searching the Dimensions.ai database. When limiting the search to the research field “Library and Information Studies,” the publication type “article,” and the term “hypergraph” in the title and abstract, only 31 publications were found over all years. Meanwhile, a search on VOSviewer using the same restrictions returned 7,479 publications.

Since this study did not aim to provide a comprehensive literature review, only a few publications that are most relevant to the objectives of the bibliometric analysis will be listed.

The paper [14] suggests using a hypergraph model to represent publication data, treating papers as nodes and authors as hyperedges that connect these nodes. This method is simpler than other models for authorship networks. The paper also introduces a measure of collaboration for authors, which shows how influential an author is on their co-authors’ collaborations.

Literature retrieval helps scientists find past research and create new ideas. However, many models do not consider the differences between search results and why papers are cited. A citation intent graph can show different reasons for citing papers and help researchers with citation decisions [15]. The CitenGL model improves this process by combining citation intent with text matching signals. It includes a special encoder for heterogeneous hypergraphs and a deep fusion unit, resulting in a graph-based output.

Finding the right scholars to work together is important for scientific progress. The study [16] presents a new algorithm designed to identify effective team configurations for collaboration based on a network of scholars. The algorithm assesses the trust and skill levels of scholars by using the collaboration network, and it models relationships using a hypergraph.

It is noteworthy that the publications focus primarily on analyzing collaboration among authors; no similar works dealing with the analysis of index keywords or their equivalents could be found.

 doi:10.20944/preprints202603.2224.v1

The Aim of This Study

Given the relevance of the topic of using hypergraphs in bibliometric analysis, the small number of publications and the dominance therein of co-authorship issues on the one hand, and the availability of well-established methods (KaHyPar, Mt-KaHyPar, HYPE) for forming hyperedge blocks has made it possible to formulate a topical problem that may draw attention to this topic and contribute to its development.

The aim of this study is to conduct a series of proof-of-concept studies demonstrating the potential of the KaHyPar, Mt-KaHyPar and HYPE tools for forming thematic communities of publications using data from the IEEE Terms field in the IEEE Xplore database. This article is the first in a series and uses KaHyPar to form term blocks using hypergraph approaches.

Materials and Methods

The bibliometric data for 2021–2025 used in this study were exported from IEEE Xplore based on the queries "IEEE Terms": Artificial intelligence, Blockchain, Data Science, Deep learning, Image processing, Internet of Things, Machine Learning and Anomaly detection. For the term "Machine Learning," the search was conducted not in the "IEEE Terms" field, but in "All Metadata." For each of the IEEE Terms, bibliometric records were sorted by relevance during export.

A total of 1,000 records were exported for each year. If there were many records, preference was given to journal articles; if fewer records were available upon request, both journal articles and conference proceedings were used. A total of 40,000 records were downloaded; after deduplication, 38,173 remained. Next, records with an empty IEEE Terms field (there were 16 of them) were removed, leaving 38,157 records, which were used in the study.

Why were IEEE Terms chosen for the analysis? 1) IEEE topics are relevant to many engineering problems; access to bibliometric data on IEEE Xplore is free; and the quality of the records is high. 2) IEEE Terms are terms from a controlled vocabulary, which ensures their standardized spelling and expert selection for inclusion in the dictionary; this avoids preprocessing the data, such as lemmatization. 3) As a rule, there are more IEEE Terms in a single record than author keywords, which can also be considered for forming hyperedges. 4) The author of this work is primarily interested in identifying current research topics based on bibliometric analysis, and the selection of index keywords—which are IEEE Terms—is relevant to these interests.

Since one of the objectives of the study was to identify potential issues in partitioning a set of hyperedges into blocks using KaHyPar under fairly typical parameters, this task was somewhat complicated by the fact that, although all the selected IEEE Terms fall under the category of computer science, they are quite different in nature. For example, Deep learning can be considered as subcategory of Artificial intelligence, but one with a large number of publications; Blockchain, Image processing, and the Internet of Things can be viewed as more specialized tasks, whereas Data Science, conversely, is a more general task. Additionally, for the IEEE Term "Machine learning," the search was performed not on the "IEEE Terms" field, but on "All Metadata." The rationale behind this approach was that if the partitioning using KaHyPar yields reasonably interpretable results under such conditions—which are not the simplest—then, on the one hand, we can hope for better results by addressing the identified issues, and on the other hand, we can understand what to focus on when collecting and analyzing bibliometric data. The specific number of IEEE Terms used in the queries determines the number of blocks into which the hypergraph will be partitioned—8 blocks.

The choice of tasks related to computer science was motivated by their wide range of potential applications, from energy to medicine.

The small number of non-unique (repeated) hyperedges characteristic of our data will result in their contribution to the total cost of the partition being negligible compared to the large number of unique hyperedges. Therefore, in this study, only unweighted hyperedges were considered.

 doi:10.20944/preprints202603.2224.v1

The partitioning into blocks was performed using the KaHyPar command-line client [10.1145/3529090] with fairly standard parameters: ./KaHyPar -h Hyper_IEEE_Terms_No.hgr -k 8 -e 0.03 -o km1 -m direct -p /config/km1_kKaHyPar_sea20.ini -w true > OUTparam.txt

Here: Hyper_IEEE_Terms_No.hgr — a list of unweighted hyperedges in hgr format. "No" means that the terms have been replaced with their indices, ranging from 1 to 4943 (the number of unique IEEE Terms in all records). -k 8 — the number of blocks into which the hypergraph is divided. -e 0.03 — imbalance, reflecting how much the number of nodes varies in each block. -o km1 -m direct -p /config/km1_kKaHyPar_sea20.ini — recommended parameters that perform well in terms of both running time and solution quality. See kahypar.org. -w true — specifies that results be output in the format node number (IEEE Term) → block number.

Note: In short, the primary application of KaHyPar is optimizing task distribution across threads. In this model, tasks are represented as vertices, while shared data or dependencies are hyperedges. The goal is to partition these tasks into groups to minimize hyperedge cuts while maintaining a balanced computational load.

In the context of bibliometrics, these elements (authors, keywords, and references) are similarly partitioned into clusters with minimal hyperedge overlap. This partitioning facilitates various applications, such as systematic cataloging, identifying co-authorship groups with minimal external intersections, and uncovering the cores of cited publications. An analogue for "computational load" in this scenario could be a publication's citation count or the impact factor of the journal.

This article did not set out to investigate the effect of the selected parameters on block partitioning, as it was conducted as a proof-of-concept study.

Notes: At the time of writing this article, in March 2026, KaHyPar could only be compiled on Linux. In this case, Windows 10 WLS was used. It is important to note that when compiling with default settings, the specific features of the processor's instructions are taken into account; therefore, it is not always possible to use a version of KaHyPar compiled on one computer on another. By default, KaHyPar uses compiler optimization flags. The -march=native flag forces the compiler to use all the unique capabilities of that specific processor (for example, AVX2, AVX-512, or BMI extensions). Moving the compiled file to a computer with a different processor may result in the program encountering an instruction that the new processor "does not understand."

Analysis has shown that entries in the IEEE Terms field may contain duplicate terms; for example, there were occasionally entries in which *Artificial intelligence* appeared at both the beginning and the end of the entry. In such cases, KaHyPar issues a warning but does not consider this an error. Small-scale tests showed that removing such duplicates did not affect the result. However, no detailed analysis was conducted. It appears that KaHyPar automatically removes duplicate nodes in hyperedges, since hyperedges are treated as a set of nodes [12].

Minimal data preprocessing is essential. All IEEE terms were converted to lowercase, spaces were replaced with underscores, and to avoid issues with regular expressions, parentheses were replaced with QLLQ and QRRQ, c++ → cplusplus, c# → csharp, web_2.0 → web_2_0, 1/f_noise → 1_f_noise, etc.

No inclusion thresholds were used, neither based on the frequency of terms nor on their co-occurrence. The use of thresholds is a separate task that allows one to focus on more specific issues in the analysis of bibliometric records. Using sets rather than co-occurrence allows for a more flexible approach to selecting thresholds. However, in our case, the goal was to exclude nothing from consideration.

Results and Discussions

Before presenting the results of partitioning the hypergraph consisting of sets of IEEE Terms, let us first examine some features of the IEEE Terms field itself in the 38,157 bibliometric records used.

The total number of IEEE Terms across all entries is 316,698, of which 4,943 are unique; the average number of terms per entry is 8.3.

Table 1 lists the IEEE terms with the highest overall frequency across all records.

 doi:10.20944/preprints202603.2224.v1

Table 1. Top 20 IEEE Terms with the Highest Frequency.

IEEE Term	Count	IEEE Term	Count
deep_learning	6983	accuracy	3217
training	6055	security	3157
feature_extraction	5555	task_analysis	2807
data_models	5518	predictive_models	2709
internet_of_things	5429	real-time_systems	2690
artificial_intelligence	5348	data_science	2557
blockchains	5136	optimization	2036
anomaly_detection	4885	image_processing	1591
machine_learning	4191	signal_processing	1569
computational_modeling	3526	sensors	1534

The data in the table will be useful when analyzing the results of partitioning a hypergraph into blocks.

In the table, the terms used to collect the data are highlighted in italics. The terms *training, feature_extraction, data_models* occur more frequently than most of the terms used to search for data, with the exception of *deep_learning*. Although *deep_learning* can logically be considered a subset of *artificial_intelligence*. The term *data_science* has a broad semantic meaning but it is encountered less frequently than *data_models*. It is worth noting the relatively low frequency of the term *image_processing*, which occurs 4.4 times less frequently than *deep_learning*. However, if we refer to the **Top Searches and Matching Documents** page on the IEEE Xplore platform, the data presented in Table 2 shows the following.

Table 2. Browsing Popular Search Terms (IEEE Xplore data, current as of March 10, 2026).

IEEE Term	Count	IEEE Term	Count
Data Science	955,058	AI	147,750
Image Processing	526,770	IoT	134,101
Machine Learning	482,005	VLSI	82,474
Antenna	401,910	Cybersecurity	57,238
Deep Learning	378,274	Blockchain	45,930
Cloud Computing	163,615	Sonar	18,444

A comparison of the data in the two tables shows that bibliometric assessments are highly sensitive to how they are formulated.

Table 3 shows the results of partitioning the hypergraph into blocks using KaHyPar, where the hyperedges are sets of IEEE Terms in each record.

Explanation of the weightings assigned to the terms listed in Table 3. KaHyPar, using the -k parameter, partitions the hypergraph into blocks numbered from 0 to 7. The output (when -w is set to true) provides a list of blocks to which the numbered node values (IEEE Term indices in our case) belong. The following values for the term numbers in blocks (0→7) were obtained: 639, 634, 636, 493, 636, 636, 636, 636. The difference in the number of terms per block is controlled by the -e 0.03 parameter. It is evident that block No. 3 (the fourth in the list, 493) differs significantly from the other blocks. This is worth noting for further analysis of the partitioning results. With these lists, we only know whether a given term is present in a block; no estimates of the significance of the terms themselves can be derived from this. We have to propose our own assessment of significance based on the objectives of the study.

This article proposes using the following metric to measure the weight of a term within a block. The weight of a term in a block can be estimated by its occurrence in all entries of the IEEE Terms field in which at least one of the terms of the block occurs.

 doi:10.20944/preprints202603.2224.v1

To implement this, all instances of "IEEE Terms" in each block were replaced with "IEEE_Terms" (using an underscore instead of a space, and lowercase letters). Next, the ugrep utility [github.com/Genivia/ugrep] was used with the parameters -w -f (-w → only the entire term, -f specifies the file containing the terms for this block). This approach effectively calculates the **local frequency of a term** within the extended context of the block, i.e., it takes into account all hyperedges that "touch" the block, even if they lead to other blocks. If a term from a block frequently appears in hyperedges associated with that block, it serves as its **informational anchor**. Terms with low values will occur less frequently in the hyperedges associated with the block. The applied term weight estimate within the block is easily computable and interpretable. This assessment can be easily upgraded later, for example, by applying a method such as TF-IDF or by incorporating semantic weights that take into account factors such as the citations of an article or the impact factor of the journal in which it was published.

Table 3. Top 10 IEEE Terms for each of the 8 blocks

Top 10 Terms	**N**	**Top 10 Terms**	**N**
IEEE_Term_B#0	Count	IEEE_Term_B#1	Count
deep_learning	1472	deep_learning	1536
training	1136	feature_extraction	1207
feature_extraction	1098	machine_learning	875
anomaly_detection	841	artificial_intelligence	836
machine_learning	803	training	774
artificial_intelligence	790	data_models	670
data_models	771	accuracy	656
imaging	759	predictive_models	525
signal_processing_algorithms	754	convolutional_neural_networks	434
internet_of_things	728	image_segmentation	426
IEEE_Term_B#2	Count	IEEE_Term_B#3	Count
artificial_intelligence	461	machine_learning	202
anomaly_detection	413	blockchains	202
deep_learning	405	deep_learning	200
machine_learning	365	artificial_intelligence	188
training	343	data_models	175
data_models	340	anomaly_detection	171
internet_of_things	319	internet_of_things	151
feature_extraction	293	training	144
predictive_models	222	accuracy	127
real-time_systems	196	feature_extraction	122
IEEE_Term_B#4	Count	IEEE_Term_B#5	Count
artificial_intelligence	1075	internet_of_things	1408
training	504	deep_learning	470
education	482	wireless_communication	461
data_models	478	optimization	434
data_science	467	resource_management	421
deep_learning	448	training	399
blockchains	417	machine_learning	351
machine_learning	413	artificial_intelligence	339
learning_QLLQartificial_intelligenceQRRQ	402	sensors	286
feature_extraction	285	blockchains	265
IEEE_Term_B#6	Count	IEEE_Term_B#7	Count
deep_learning	6983	blockchains	3138
training	6055	artificial_intelligence	2479

 doi:10.20944/preprints202603.2224.v1

feature_extraction	5555	anomaly_detection	2243
data_models	5518	deep_learning	2067
internet_of_things	5429	data_models	1989
artificial_intelligence	5348	internet_of_things	1911
blockchains	5136	training	1770
anomaly_detection	4885	machine_learning	1717
machine_learning	4191	security	1697
computational_modeling	3526	feature_extraction	1612

Explanation: When using grep -w -f *listofterms.txt* *textfile.txt*, the command treats each line in the list file as a separate pattern and searches for any match using a logical OR (OR) operator. If a single line in the file contains multiple terms from the list, the line will still be output only once. This is why different lists from different blocks yield different sets of lines, and consequently, the same term in the found set of lines may appear differently.

Some results. The term ***deep_learning*** ranks first in blocks #0, #1, and #6; it appears most frequently in block 6, with 6,983 occurrences. According to Table 1, this is the most frequently occurring term among all those for which data was collected. Table 2 also shows the figure of 6,983. ***artificial_intelligence*** appears 1,075 times in block #4 and 5,348 times in Table 1. ***blockchains*** appears 3,138 times in block #7 and ranks first. In Table 1, it appears 5,136 times. The term ***internet_of_things*** ranks first in block #5 and appears 1,408 times; for comparison, it appears 5,429 times in Table 1. ***anomaly_detection*** appears most frequently in block #6—4,885 times, as in Table 1—4,885 times, and ranks second in block #2. It can be assumed that in block #6, this term is "masked" by more frequently occurring terms in all entries, such as ***deep_learning***, ***internet_of_things***, ***artificial_intelligence***, and ***blockchains***. The term ***machine_learning*** ranks first in block #3, but appears more frequently in block #6 alongside other terms commonly found in all records. It should be noted that the search for the term ***machine_learning*** was conducted across "All Metadata," while the others were searched only within IEEE terms. The term ***data_science*** appears significantly less frequently across all records (2,557) and does not rank in the top positions in any block; it appears most frequently in block #4—448 times. The least common term across all records—***image_processing*** (1,591)—does not appear in Table 3 among the top 10 based on the selected criterion.

Next, lets compare the frequency patterns of the terms ***data_science*** and ***deep_learning***—not across all records, but only in those related to records found via the queries: "IEEE Terms": Data Science or "IEEE Terms": Deep learning.

For the query ***"IEEE Terms": Data Science***, here are the top 5 terms by frequency: Data science (2546), Data models (1603), Computer science (1018), Behavioral sciences (927), Data mining (610), Deep learning (357).

For the query ***"IEEE Terms": Deep learning***, here are the top 5 terms by frequency: Deep learning (4954), Feature extraction (1912), Training (1466), Data models (901), Task analysis (755), Data science (1).

As a result, the term ***Data Science*** appears less frequently in its own group than the term ***Deep Learning*** does in its own group. In the opposite group, the term ***Deep Learning*** also appears more frequently than the term ***Data Science*** does in its opposite group. Note that the number of records in both groups is the same.

Thus, 5 out of 8 terms (***deep_learning***, ***artificial_intelligence***, ***machine_learning***, ***internet_of_things***, and ***blockchains***) used in the search ranked first in all blocks; the term ***anomaly_detection*** ranked second in one block and was noted as high-frequency in another. The terms ***data_science*** and ***image_processing***, which were rarely found in all records, did not rank first in any block based on the selected criterion.

Given the considerable overlap in topic meanings—each rooted in computer science and AI—the large volume of hyperedges (38,173) relative to the number of nodes (4,943) led to extensive partitioning of the former. This markedly distinguishes hypergraph partitioning in bibliometric

 doi:10.20944/preprints202603.2224.v1

analysis from its classical application in multi-thread task distribution. In this case, the KaHyPar parameters were not optimized, and results were not compared with other criteria for term relevance in blocks.

To effectively partition IEEE terms into blocks where less frequently occurring terms are positioned at the top, a separate study is necessary. This research could explore various criteria for assessing term importance within blocks, using weights for IEEE Terms, investigate KaHyPar parameter tuning—such as adjusting the -e parameter to create greater imbalance across clusters—or even implement alternative algorithms like HYPE [17]. However, these tasks extend beyond the current study's scope and are intended for a future project.

A note on the relevance of further research on the use of hypergraphs in bibliometric analysis.

Moving from an incidence matrix to an adjacency matrix transforms a cohesive group event into a fragmented set of pairwise links, degrading **structural fidelity**: algorithms begin to perceive direct connections where, in reality, there was only indirect participation in a large group. This also compromises **explainability**, as the link to the specific original source (such as a shared document or project) that unified the participants is lost, turning a meaningful data structure into an anonymous "cloud" of contacts. While graph projections offer computational convenience and weighted edges can partially recover some information, they cannot preserve higher-order context. Further research into scalable hypergraph methods is therefore essential for bibliometric tasks where group structure and provenance matter.

Why KaHyPar Tuning Requires Further Study

The following results demonstrate how the imbalance parameter (e) fundamentally alters the logical structure of a term-based hypergraph.

Brief Description & Analysis

The results show a hypergraph partitioning of 4,943 terms (nodes) and 38,157 hyperedges (short text contexts) into 8 blocks using **KaHyPar**. The analysis compares three levels of allowed imbalance (e):

1. **Low Imbalance ($e = 0.03$):** Blocks are nearly identical in size (~636 nodes). However, this "forced" symmetry results in the highest **Hyperedge Cut** (29,072), meaning many natural thematic connections are broken to maintain equal partitions.
2. **Medium Imbalance ($e = 0.3$):** Connectivity metrics improve as the cut drops to 25,965. The partitions begin to vary, with one block shrinking significantly (19 nodes), suggesting the algorithm is starting to isolate smaller, highly specific sub-topics.
3. **High Imbalance ($e = 0.9$):** This setting achieves the best clustering quality (lowest **Cut** of 19,061 and highest **Absorption**). However, it leads to extreme structural decay where 4 blocks become tiny (3–217 nodes), while the bulk of the data collapses into the remaining 4 large clusters.

Main Conclusion

There is a clear trade-off between partition balance and cluster quality. The high-imbalance results ($e = 0.9$) suggest that the underlying terminology in AI, IoT, and Data Science does not naturally form 8 equal-sized groups. While $e = 0.9$ offers the most mathematically "pure" clusters, $e = 0.3$ serves as the best practical compromise, significantly improving connectivity metrics without completely depleting the smaller blocks.

Summary Tables

Table 4. Partitioning Objectives Comparison

Metric (Optimization)	$e = 0.03$ (Strict)	$e = 0.3$ (Moderate)	$e = 0.9$ (Loose)
Hyperedge Cut (min)	29,072	25,965	19,061
SOED (min)	69,241	59,148	41,256

 doi:10.20944/preprints202603.2224.v1

(k-1) Cut (min)	40,169	33,183	22,195
Absorption (max)	32,229.6	33,259.4	34,902.6
Actual Imbalance	0.029	0.299	0.899

Table 5. Block Size Distribution (Node Counts)

Block ID	$e = 0.03$	$e = 0.3$	$e = 0.9$
Part 0	636	19	3
Part 1	634	803	1174
Part 2	636	262	34
Part 3	493	803	75
Part 4	636	802	1173
Part 5	636	800	1131
Part 6	636	803	217
Part 7	636	651	1136
Std. Deviation	~50.5	~297.8	~558.1

Here:

Hyperedge Cut → The number of hyperedges that have vertices in more than one block.

SOED (Sum of External Degrees) → For each vertex, its external degree is the number of blocks it is connected to via hyperedges that span multiple blocks. SOED sums these external degrees over all vertices.

The $(k-1)$ metric defines the total cost as $\sum_{e \in E}(\lambda(e)-1)$, where $\lambda(e)$ is the number of blocks spanned by hyperedge e.

Imbalance → The relative deviation of the largest block weight from the average block weight.

Absorption → Absorption is a measure of how well the partition "absorbs" hyperedges inside blocks.

In this section, the results are presented solely as an illustration of the impact of e on hypergraph partitioning and to demonstrate the need for further research, which may include, among other things, adding weights to nodes (terms) so that general terms such as AI and ML do not overshadow more specific ones like anomaly detection or blockchain.

Conclusions

The approach of interpreting individual entries of the IEEE Terms field in bibliometric records exported from IEEE Xplore as hyperedges, followed by partitioning the resulting hypergraph into blocks using KaHyPar, has demonstrated a certain degree of effectiveness.

Using the hypergraph approach requires a lot of data preparation to create a file in .hgr format. It is not complicated, but it lacks the conveniences offered, for example, by VOSviewer for analyzing bibliometric data using a graph-based approach. This is one of the reasons why the hypergraph approach is rarely encountered in scientometric studies, even when the advantages of the hypergraph approach are listed in them.

A simple criterion for assessing the significance of terms (nodes) within a block has been proposed, based on the frequency of a term's occurrence in entries of the IEEE Terms field (hyperedges) that contain at least one term from the block under consideration. The method for evaluating the significance of terms within a block requires further research, but offers flexible options for various tasks.

This study is a proof-of-concept and suggests further research, including: an analysis of the impact of KaHyPar parameters on the results obtained; the possibility of using other programs such as *HYPE* and *Mt-KaHyPar*; and the choice and justification of criteria for determining the significance of terms within a block. It would be advisable to conduct studies in which hypergraphs are generated from other data, such as a bibliography.

 doi:10.20944/preprints202603.2224.v1

Acknowledgments: This work was funded by the Ministry of Science and Higher Education of the Russian Federation, State Assignment no. 125021302095-2.

References

1. V. V. Nalimov and Z. M. Mulchenko, Scientometrics: The Study of the Development of Science as an Information Process (Moscow: Nauka Publishing House, 1969)
2. De Solla Price D. Editorial statements. *Scientometrics*. 1978;1(1):3-8. doi:10.1007/BF02016836
3. Schubert A, Glänzel W, Braun T. Scientometric datafiles. A comprehensive set of indicators on 2649 journals and 96 countries in all major science fields and subfields 1981–1985. *Scientometrics*. 1989;16(1-6):3-478. doi:10.1007/BF02093234
4. Price DJDS. Networks of Scientific Papers: The pattern of bibliographic references indicates the nature of the scientific research front. *Science*. 1965;149(3683):510-515. doi:10.1126/science.149.3683.510
5. Kessler MM. Bibliographic coupling between scientific papers. *Amer Doc*. 1963;14(1):10-25. doi:10.1002/asi.5090140103
6. Small H. Co-citation in the scientific literature: A new measure of the relationship between two documents. *J Am Soc Inf Sci*. 1973;24(4):265-269. doi:10.1002/asi.4630240406
7. Marshakova IV. System of document connections based on references. *Nauch-TechnInform*. 1973;6:3-8.
8. Taramasco C, Cointet JP, Roth C. Academic team formation as evolving hypergraphs. *Scientometrics*. 2010;85(3):721-740. doi:10.1007/s11192-010-0226-4
9. Hu F, Zhao HX, Zhan XX, Liu C, Zhang ZK. Evolution of citation networks with the hypergraph formalism. *arXiv*. Preprint posted online 2014. doi:10.48550/ARXIV.1406.0936
10. Seidman SB. Structures induced by collections of subsets: a hypergraph approach. *Mathematical Social Sciences*. 1981;1(4):381-396. doi:10.1016/0165-4896(81)90016-0
11. Bredikhin SV, Scherbakova NG. Scientific journal co-authorship network model. *Problems of Informatics*. 2023; 3:5–18. doi:10.24412/2073-0667-2023-3-5-18
12. Schlag S, Heuer T, Gottesbüren L, Akhremtsev Y, Schulz C, Sanders P. High-Quality Hypergraph Partitioning. *ACM J Exp Algorithmics*. 2022;27:1-39. doi:10.1145/3529090
13. Praggastis B, Aksoy S, Arendt D, et al. HyperNetX: A Python package for modeling complex network data as hypergraphs. *JOSS*. 2024;9(95):6016. doi:10.21105/joss.06016
14. Lung RI, Gaskó N, Suciu MA. A hypergraph model for representing scientific output. *Scientometrics*. 2018;117(3):1361-1379. doi:10.1007/s11192-018-2908-2
15. Shi K, Liu K, He X. Heterogeneous hypergraph learning for literature retrieval based on citation intents. *Scientometrics*. 2024;129(7):4167-4188. doi:10.1007/s11192-024-05066-4
16. Ghasemian F, Zamanifar K, Ghasem-Aghaee N. Composing Scientific Collaborations Based on Scholars' Rank in Hypergraph. *Inf Syst Front*. 2019;21(3):687-702. doi:10.1007/s10796-017-9773-z
17. Mayer C, Mayer R, Bhowmik S, Epple L, Rothermel K. HYPE: Massive Hypergraph Partitioning with Neighborhood Expansion. In: *2018 IEEE International Conference on Big Data (Big Data)*. IEEE; 2018:458-467. doi:10.1109/BigData.2018.8621968

Boris N. Chigarev, Cand. Sci. (Phys.-Math.), Senior Researcher of Analytical Center for Energy Policy and Security, Oil and Gas Research Institute, Russian Academy of Sciences, Moscow, Russia.