applied sciences

MDPI

Article

Identifying Hyper-Heuristic Trends through a Text Mining Approach on the Current Literature

Anna Karen Gárate-Escamilla, Ivan Amaya, Jorge M. Cruz-Duarte, Hugo Terashima-Marín and José Carlos Ortiz-Bayliss *

Tecnologico de Monterrey, School of Engineering and Sciences, Monterrey 64849, Mexico
* Correspondence: jcobayliss@tec.mx

Abstract: Hyper-heuristics have arisen as methods that increase the generality of existing solvers. They have proven helpful for dealing with complex problems, particularly those related to combinatorial optimization. Their recent growth in popularity has increased the daily amount of text in the related literature. This information is primarily unstructured, mainly text that traditional computer data systems cannot process. Traditional systematic literature review studies exhibit multiple limitations, including high time consumption, lack of replicability, and subjectivity of the results. For this reason, text mining has become essential for researchers in recent years. Therefore, efficient text mining techniques are needed to extract meaningful information, patterns, and relationships. This study adopts a literature review of 963 journal and conference papers on hyper-heuristic-related works. We first describe the essential text mining techniques, including text preprocessing, word clouds, clustering, and frequent association rule learning in hyper-heuristic publications. With that information, we implement visualization tools to understand the most frequent relations and topics in the hyper-heuristic domain. The main findings highlight the most dominant topics in the literature. We use text mining analysis to find widespread manifestations, representing the significance of the different areas of hyper-heuristics. Furthermore, we apply clustering to provide seven categories showing the associations between the topics related to hyper-heuristic literature. The vast amount of data available that we find opens up a new opportunity for researchers to analyze the status of hyper-heuristics and help create strategic plans regarding the scope of hyper-heuristics. Lastly, we remark that future work will address the limitations of collecting information from multiple data sources and analyze book chapters related to hyper-heuristics.

Keywords: text mining; frequent association rule; hyper-heuristic; optimization

Citation: Gárate-Escamilla, A.K.; Amaya, I.; Cruz-Duarte, J.M.; Terashima-Marín, H.; Ortiz-Bayliss, J.C. Identifying Hyper-Heuristic Trends through a Text Mining Approach on the Current Literature. *Appl. Sci.* **2022**, *12*, 10576. https://doi.org/10.3390/app122010576

Academic Editor: Vincent A. Cicirello

Received: 4 September 2022
Accepted: 17 October 2022
Published: 20 October 2022

1. Introduction

Among the many different strategies proposed for solving optimization problems, hyper-heuristics have arisen as an advantageous alternative given their many practical and successful applications [1–3]. A hyper-heuristic is an optimization method that combines the strengths of individual solvers, most of them of approximate nature, such as heuristics. An initial definition of hyper-heuristics considered them as "heuristics to choose heuristics". However, the term has evolved to also include "heuristics to generate heuristics" and to categorize existing works according to their learning process (online or inline) and type of solving process (constructive or perturbative), to mention a few.

In the last decade, the current state-of-art in hyper-heuristic applications has grown substantially [4]. Therefore, the rise of unstructured data, both in industry and academia, is driving the need to use text mining algorithms in different hyper-heuristic application fields such as social media [5], medicine [6], service management [7], business intelligence [8], and finance [9]. In a broad sense, we may state that hyper-heuristics aim to find the correct sequence of low-level heuristics in a particular situation, obtaining a suitable combination according to the computational problem presented. Several advances in the different problem domains are observed in timetabling [10,11], bin packing [12], vehicle routing [13], and scheduling problems [14,15]. Moreover, recent hyper-heuristic research has proved crucial for specific problems. For example, they are essential for developing novel and quality methods for different optimization problems [4].

As described before, recent years have witnessed an intense growth of interest in hyper-heuristic (HH) methods, which favors new researchers entering this research topic. However, there is plenty of mixed information related to this topic: the problem domains where these techniques have been applied, the methods used to train such techniques, the journals and conferences to present these works, and the most relevant authors in the field, to mention a few. For starters, such a large amount of information can sometimes be overwhelming even for some experienced researchers on the topic. This is why we aim to analyze the current literature in hyper-heuristics from a text mining perspective. In doing so, we pursue a better understanding of what has been done and likely future trends in this field. Although some works have already surveyed the hyper-heuristic-related literature, to the best of our knowledge, there is only one systematic review on this topic, the work conducted by Sánchez et al. [3]. Although some other works —not considered systematic reviews by the community— have surveyed the related literature, no work has previously analyzed the relations between the relevant elements in this topic through text mining. Such relations may be critical, particularly for new researchers willing to enter the field of hyper-heuristics.

It is critical to mention that most of the information about this topic corresponds to unstructured text. Traditional qualitative approaches are unsuitable for processing this kind of data. Therefore, as information in different fields grows, so also does the interest in text mining methods that can extract meaningful information [16]. Some crucial techniques include natural language processing (NLP), information retrieval (IR), text summarization, and supervised and unsupervised learning. Furthermore, this work exhibits an important advantage concerning traditional literature reviews. It requires less human intervention because it includes text mining and machine learning in the data analysis process.

This study reviews 963 scientific documents related to hyper-heuristics in multiple domains. Our goal is to provide a bibliometric literature review on the hyper-heuristic applications and analyze the hot topic domains for researchers, especially those just entering the field. For this reason, we address a simple yet relevant objective: to learn the trends in the literature related to HHs and how they are connected among them. Overall, achieving this objective opens a new opportunity for researchers and academics to focus on high-level strategies to manage hyper-heuristics and their multiple applications.

The remainder of this paper is organized as follows. In Section 2, we describe the fundamental concepts that support this work. Section 3 explains the methodology followed in this work. We present the results and their discussion in Section 4. Finally, Section 5 provides the conclusion of this study and some paths for future work.

2. Fundamental Concepts

Unsurprisingly, we have observed the increased use of computer technologies to enhance decision-making support in recent years. In this regard, HHs have arisen as methods that aim to increase problem solvers' generality by improving how they make their decisions. Overall, a hyper-heuristic can be seen as a method that either (1) selects among existing heuristics or (2) creates new heuristics. However, different types of hyper-heuristics have been defined in the past employing various points of view [1,2,4]. This work employs the term "hyper-heuristic" without considering those classifications.

For starters, a hyper-heuristic is usually powered by a mechanism that learns how to "combine" heuristics. Among the several tools that allow hyper-heuristics to learn, we must highlight metaheuristics (MHs) and machine learning (ML) strategies due to their relevance in the field. Regarding metaheuristics, methods such as genetic algorithms [17,18], genetic programming [19], simulated annealing [20,21], artificial immune systems [22,23], and particle swarm optimization [24,25], to mention a few, have been implemented for this task.

Beyond "traditional" methods that power hyper-heuristics, techniques from the machine learning realm have also attracted the attention of the hyper-heuristic community. For example, some interesting works have incorporated techniques related to classification, such as neural networks and decision trees, while others are related to reinforcement learning [26,27] and clustering [28,29].

Machine learning is an analytical tool employed for decision-making when a task is too large or complex to program, such as transforming medical records into knowledge, predicting pandemics, and analyzing market trends. ML is mainly divided into two categories called supervised learning and unsupervised learning. Supervised learning involves training a model with historical data that knows its output variable's response. The historical dataset is split into training and testing sets. The training set is employed to build the model, while the test set is used exclusively to evaluate the model's performance. The necessary adjustments are made to have an outcome closest to actual values from the historical data. Conversely, unsupervised learning has its training algorithm without a previously determined output response.

In particular, we use two unsupervised learning methods: clustering and association rules. Clustering is a powerful unsupervised machine learning tool for detecting subgroups of individuals with more homogeneous attributes than other subgroups or clusters. This methodology is necessary to handle the interaction of multiple variables. Additionally, it has been used in image processing, document classification, and group creation. Cluster analysis leads to a clearer understanding of a situation, such as customer segmentation [30], fraud detection [31], and heart disease risk factors [32]. The *k*-means procedure, which we use in this article, is the most popular method for cluster analysis. In general, *k*-means splits data into different clusters according to the closeness of observations to their respective centroid. Otherwise, association rules look for the relationship between variables within a database. These methods have become popular with the market basket problem [33]. Here, the goal is to understand consumer habits and create sales strategies and recommendations. There are different algorithms to create association rules, such as Apriori and FP-Growth. In this work, we opted for Apriori, given its simplicity. Apriori is used to identify a collection of items or the probability of consuming a product if another product has been consumed previously [34].

3. Methodology

In this section, we provide the reader with information related to the research methodology we conducted. Overall, the methodology consists of four stages, which are depicted in Figure 1 and summarized as follows:

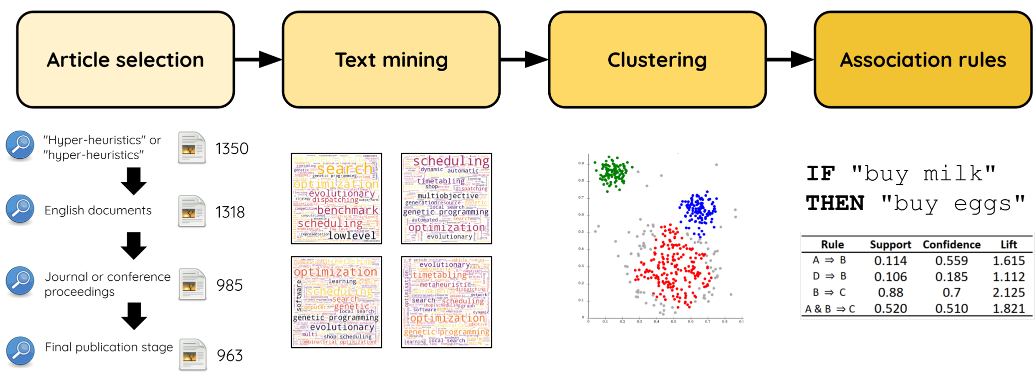


Figure 1. Graphical description of the four-stage methodology followed in this work.

i. Article selection. We selected relevant documents from Scopus, all of them related to hyper-heuristics.
ii. Text mining. We explored high-frequency words in the title, abstract, author keywords, and index keywords of the manuscript selected in the previous stage.

iii. Clustering. We applied a clustering process to those terms extracted from the previous stage to group them according to their similarity.
iv. Association rules. We relied on association rule mining to demonstrate the probability of relationships among the terms within the corpus. In this case, we investigate the probability that two or more terms or concepts are, indeed, related to the others.

More details on each stage are provided below.

3.1. Article Selection

For this study, we used Scopus to collect information about the current literature related to hyper-heuristics. We chose Scopus because it is a citation database recognized worldwide for many peer-reviewed journals in different domains [35,36]. Other works have also considered advanced search and text mining techniques to review the literature indexed in Scopus [7,37,38], albeit in other fields. We extracted the information from the articles considered in this work from Scopus on 5 April 2022. To conduct such a review, we followed four steps:

1. We recovered any document whose title, abstract, or keywords contained the term "Hyper-heuristics" or "Hyperheuristics". Through this query, we retrieved 1350 research articles.
2. We filtered the results by restricting the language to "English", which reduced the number of documents to 1318.
3. We restricted the search by document type, keeping only articles and conference papers, focusing our research on the source type "Journal" and "Conference Proceeding", resulting in 985 documents.
4. Finally, we excluded documents outside the "Final" publication stage. Then, we kept only 963 articles to conduct our study.

3.2. Text Mining

The text mining analysis considered three main steps: preprocessing, word count, and word cloud. The preprocessing consisted of extracting words from the text and using stop-words to remove unimportant words using the `nltk` library. Since common words in English are unlikely to contribute to the analysis, we used stop-words in the abstract, author keywords, index keywords, and title columns. Additionally, we also removed capitalization, punctuation, and special characters. We did not apply any word truncation methods due to the nature of the data and the frequency of the hyper-heuristic word set. We counted the words in the columns abstract, author keywords, index keywords, and title to make a top list, a word cloud, and a set of words. For simplicity, the words count the same in each occurrence. In addition, two- and three-word sets were constructed as shown in Table 1. The top list was made by importing `Counter` from the `collections` library and sorting it from highest to lowest. The word cloud was generated using the `wordcloud` library (word punctuation was removed). To create the two- and three-word sets, we used the `ngrams` library to search for the most-used set of words. In the document, we will use the term "compound words" to refer to the sets of two and three words generated by `ngrams`. Finally, from the library `mlxtend`, we imported `Apriori` and `association_rules` in order to analyze the 90th percentile of the most cited publications in Scopus.

Table 1. Two- and three-word compound words from the abstracts of hyper-heuristic articles.

Two-Word Compounds	Count	Three-Word Compounds	Count
combinatorial optimization	74	particle swarm optimization	45
simulated annealing	72	capacitated arc routing	27
examination timetabling	62	arc routing ucarp	17
job shop	60	ant colony optimization	16
vehicle routing	55	flexible shop scheduling	13
particle swarm	53	dynamic shop scheduling	12
shop scheduling	50	constrained project scheduling	12
multiobjective evolutionary	47	university course timetabling	11
swarm optimization	46	-	-
machine learning	45	-	-
course timetabling	36	-	-
reinforcement learning	35	-	-
nurse rostering	30	-	-

3.3. Clustering

We conducted a k-means analysis to find different clusters using the documents in the dataset. The cluster analysis performed had no missing values. `KMeans` was selected from the `sklearn.cluster` library to perform the analysis. Before clustering, it is necessary to convert the data into a numerical format to perform the analysis. In this work, we use Term Frequency-Inverse Document Frequency (TF-IDF) for such a task. TF-IDF is a technique to vectorize the information and show the importance of a word in the document by assigning it a score. For example, the word *collection* is not important and shows a score of 0.0, while *genetics* may show a score of 0.75. Then, with TF-IDF, k-means creates clusters that are based on the most important terms. The main steps of the k-means algorithm are as follows: (1) first, the initialization used is `k-means++` [39]; (2) then, we calculate the distance between the points using the Euclidian distance; (3) and finally, we set the number of clusters (k) using the elbow method. The centers were chosen using the within-cluster sum of squares (WCSS) criterion, which measures the variance within each cluster. The elbow method aims to find an inflection point where the lowest WCSS is found and a curvature similar to an elbow is made [40]. In this study, we employed only the index keywords variable from our dataset to create the clusters. The reason for this is that the words in index keywords are selected by Scopus and have a more standardized vocabulary, including synonyms and plurals.

3.4. Association Rules

Recall that association rule mining is a machine learning algorithm that finds a confidence relationship among different transactions in a database [41]. The Apriori algorithm is one of the simplest association rules [42]. In this stage, we utilized the Apriori algorithm on the index keywords of the 90th percentile of the most cited articles related to hyper-heuristics. Ninety-four items were used for the association rules, where the algorithm employs words or sets of words separated by commas. As in the case of clustering, we also selected sets of words that were presented in a simplified way for the association rule mining. For this reason, neither the abstract nor the title were considered for the analysis.

Further, we created these association rules using the index keywords and Apriori. Each element that is part of a transaction is known as an item. The set of two or more items together is called an itemset. The *antecedent (X)* has one or more items that represent the `if` component. Hence, the *consequent (Y)* is the `then` component. For example, `if` “buy milk” (*antecedent*), `then` “buy eggs” (*consequent*). In this study, the most important metric selected was *support*, which allows searching for the frequency of the itemset. Therefore, the higher the *support* value, the higher the association between items. Equation (1) yields the *support* rule:

$$support(X \rightarrow Y) = \frac{\text{Transactions containing both } X \text{ and } Y}{\text{Total number of transactions}}. \quad (1)$$

Additionally, the *confidence* metric is the likeliness of a *consequent* given the *antecedent*. Equation (2) describes this metric:

$$confidence(X \rightarrow Y) = \frac{\text{Transactions containing both } X \text{ and } Y}{\text{Transactions containing } X}. \quad (2)$$

Finally, the *lift* evaluates the quality of the rule by assessing the probability of the *consequent* ignoring the presence of the *antecedent*. A *lift* < 1 shows that having an *antecedent* does not increase the change of occurrence of a *consequent*. A *lift* > 1 demonstrates a high association between *antecedent* and *consequent*. The *lift* rule is given by:

$$lift(X \rightarrow Y) = \frac{confidence(X \rightarrow Y)}{\text{Fraction of transactions containing } Y}. \quad (3)$$

4. Results

We conducted a series of analyses to learn about the dominant trends and subjects related to hyper-heuristics in the literature. We divided these analyses into four subsections to better study the results. The first section details the distribution of articles by year and journal. Next, we reveal the most frequent words in abstracts, titles, and keywords. We follow the analysis by showing the clusters of hyper-heuristic-related terms in the index keywords field. Finally, we apply the association rule for text mining on the most pertinent index keywords and relationships among problem domains. For the sake of readability, we show all the words in lowercase, even if some compounds correspond to specific and widely used algorithms or problems.

4.1. Distribution of Publications by Year and Journal

Figure 2 shows the distribution by year of publications related to hyper-heuristics between 2002 and 2022. Notice that more than 30% of the research articles analyzed were published in the last three years (312 out of 963). This demonstrates a considerable increase in the publication of hyper-heuristic-related topics in the literature.

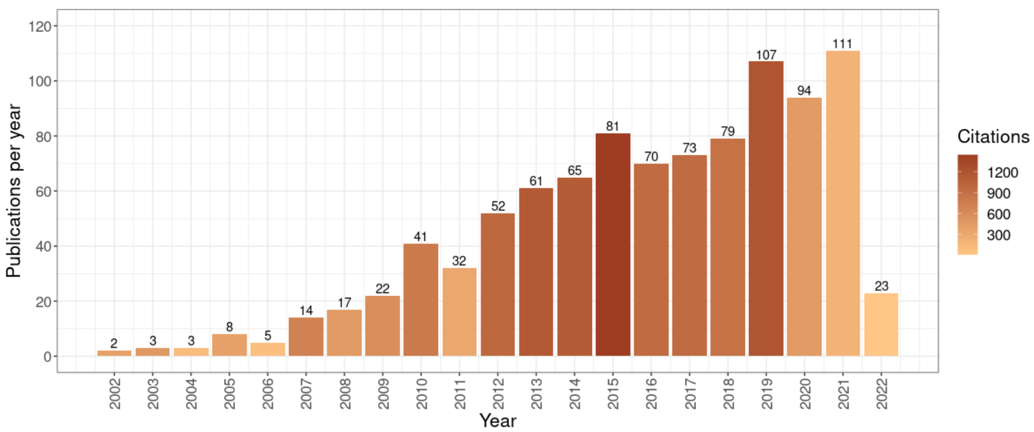


Figure 2. Yearly distribution of articles on hyper-heuristics between 2002 and 2022 (obtained on 5 April 2022).

Figure 2 supports previous publications pointing out the recent growth in the interest in investigating hyper-heuristic-related works [1,3,4,43]. The color of the bars represents the number of citations that took place each year. There is a considerable increase from 2012 onward, highlighting the years with the most citations in 2015 and 2019, with 1440 and 1196 citations, respectively.

Subsequently, Figure 3 illustrates the distribution of the ten most-cited journals given their publications on hyper-heuristics. These ten most-cited journals accumulate close to 44% of the total citations in the field. Among these journals, the European Journal of Operational Research has the largest number of citations to works related to hyper-heuristics, with 8.69%, followed by IEEE Transactions on Evolutionary Computation, with 7.12%. Furthermore, we found no significant difference in the number of articles related to hyper-heuristic per journal. The journals with the highest number of articles in hyper-heuristics are the *Applied Soft Computing Journal* and *Expert Systems with Applications*, both with 20 articles and a significantly lower number of citations than the most cited journal (3.12% and 2.98% of the total citations, respectively).

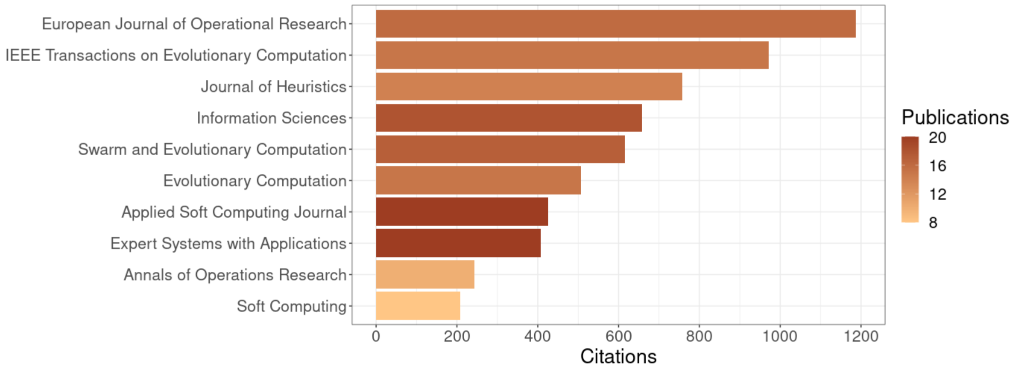
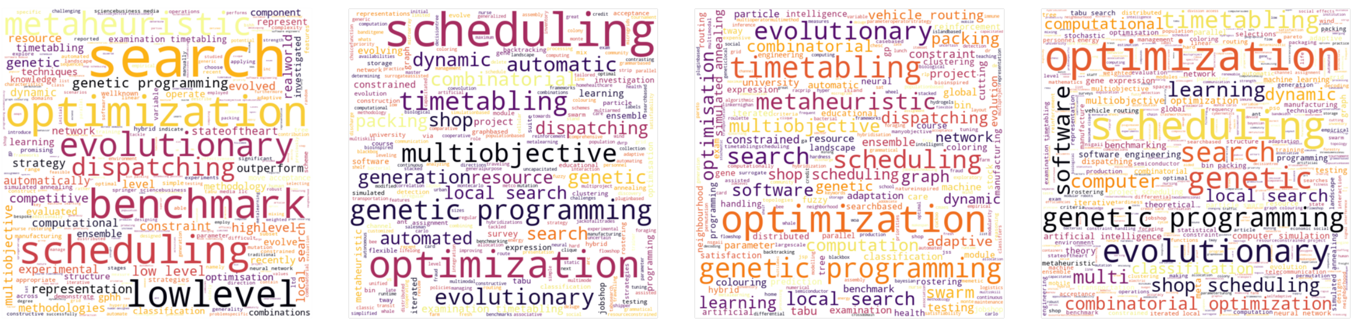


Figure 3. Distribution of citations and number of documents of the top ten journals with the most citations for documents on hyper-heuristics (obtained on 5 April 2022).

4.2. Most Frequent Words Reported

Figure 4 demonstrates the word clouds at the 90th percentile of the most-cited research papers related to hyper-heuristics. We exclude the words heuristic and hyperheuristic, as they are part of the query used to search for the articles. Figure 4a shows the highest-frequency words in the abstract. The most used abstract words are "optimization", "scheduling", "metaheuristic", "search", "evolutionary", and "benchmark". Figure 4b depicts the most popular words in the title, which include "scheduling", "optimization", "timetabling", "evolutionary", "genetic programming", and "multiobjective". Figure 4c shows that the most frequent author keywords are "optimization", "genetic programming", "timetabling", "evolutionary", "scheduling", and "metaheuristic". Finally, in Figure 4d, we can observe that the most popular index keywords are "scheduling", "optimization", "evolutionary", "genetic programming", "timetabling", and "software". Hence, the most frequently mentioned terms in the hyper-heuristic literature involve evolutionary algorithms, optimization problems, and scheduling.



(**a**) Abstract (**b**) Title (**c**) Author keywords (**d**) Index keywords

Figure 4. Word clouds for elements of interest in the publications.

Table 1 displays the most mentioned compound words in the abstracts of selected articles about hyper-heuristics. The two-word compounds most often used in the literature were "combinatorial optimization", "simulated annealing", "examination timetabling", and "job shop". In addition, the most common three-word compounds were "particle swarm optimization", "capacitated arc routing", "arc routing ucarp", and "ant colony optimization".

Therefore, Figure 4 and Table 1 allow us to identify the words and compound words that represent the most dominant subjects in the selected literature related to hyper-heuristics. The main words and topics include "genetic algorithm", "optimization problems", "scheduling", "combinatorial optimization", "evolutionary algorithm", "swarm optimization", "particle swarm", "metaheuristic", "timetabling", and "machine learning".

4.3. Clustering

As mentioned before, we decided on the number of clusters using the elbow method and the total within-cluster sum of squares of each configuration, which recommended to have nine clusters ($k = 9$). Based on our experience and the nature of the hyper-heuristics, we eliminated two groups, one related to engineering and software and the other to combinatorial optimization. We removed the first cluster because of its lack of relevance to hyper-heuristics and the second cluster because it did not show favorable results in generating similar characteristics within itself. Table 2 illustrates the seven remaining clusters along with their corresponding sizes (number of publications in the cluster), the word counts, and the most critical compound words.

Table 2. Clusters related to hyper-heuristics in the literature. The count of each word or compound word is illustrated between parentheses.

Cluster	Size	Words	Compund-Words
1	84	shop (99) genetic (98) job (67)	shop, scheduling (83) - -
2	369	hyper (92) evolutionary (73)	packing, problem (19) bin, packing, problem (16)
3	56	vehicle (59) genetic (57) capacitated (28) network (25)	combinatorial, optimization (20) capacited, arc, routing (19) routing, policies (13) -
4	140	evolutionary (84) genetic (58) hyper (55) multiobjective (52)	particle, swarm (35) swarm, optimization (30) - -
5	45	genetic (30) resource (24) project (20)	resource-constrained, project (15) resource-constrained, project, scheduling (15) -
6	57	scheduling (37) hyper (29) examination (24) graph (19) - -	combinatorial, optimization (15) university, course (13) simulated, annealing (11) exam, timetabling (10) course, timetabling (8) school, timetabling (7)
7	21	satisfaction (25) hyper (11) genetic (10)	variable, ordering (14) - -

By analyzing the clusters generated, we have identified some exciting patterns in their contents. For example, most clusters involve genetic algorithms as their solution approach. We highlight some particular findings for each achieved cluster as follows.

- Cluster 1 (with 84 documents) deals with scheduling problems with a strong focus on job-shop scheduling. This cluster is characterized by extensive usage of genetic algorithms throughout the solution process.
- Cluster 2 (with 369 reports) is the largest and contains methods mainly using evolutionary algorithms, particularly genetic algorithms.
- Cluster 3 (with 56 records) primarily relates to vehicle routing and capacitated routing problems. It also includes other works on combinatorial optimization. Genetic algorithms also play an important role in this cluster.
- Cluster 4 (with 140 articles) mainly corresponds to optimization problems, including multi-objective ones. Although genetics is one of its main words, the term evolutionary is more relevant in general.
- Cluster 5 (with 45 reports) mostly includes scheduling problems, as was the case for Cluster 1, but it excludes the works on genetic algorithms.
- Cluster 6 (with 57 documents) addresses timetabling and scheduling problems, presenting its applications in different areas of education.
- Finally, Cluster 7 (with 21 articles) deals with constraint satisfaction problems.

4.4. Association Rule for Text Mining

As previously mentioned, we use the Apriori algorithm on the index keywords of the 90th percentile of the most cited articles on hyper-heuristics. Table 3 demonstrates important keywords with Apriori. The most-related areas were "genetic", "evolutionary", "optimization", "combinatorial optimization", "benchmarking", "timetabling", and "artificial intelligence".

Table 3. Apriori's most-related words and their corresponding *support* value using index keywords.

Support	**Itemsets**
0.340	genetic
0.223	evolutionary
0.181	optimization
0.128	combinatorial optimization
0.117	problem solving
0.106	benchmarking
0.085	dispatching rules, genetic
0.085	timetabling
0.085	optimization, evolutionary
0.075	artificial intelligence
0.075	multiobjective optimization
0.053	combinatorial optimization, evolutionary
0.053	combinatorial optimization, optimization
0.053	combinatorial optimization, problem solving
0.053	tabu search

The association rules in Table 4 use the index keywords and the results with a support greater than 0.06. The "scheduling" and "genetic programming" pair is the highest recommended with *support* of 0.128, *confidence* of 0.600, and *lift* of 1.567. The subjects "scheduling" and "evolutionary algorithms" show the most association with other index keywords, such as "genetic programming", "optimization", "combinatorial optimization", and "timetabling". Moreover, we notice that the itemset of "evolutionary algorithms" and "scheduling" is less than one, which means a negative effect; the itemset appears less frequently.

Table 4. Association rules of index keywords in the 90th percentile of the most-cited articles on hyper-heuristic-related research.

Antecedents	*Consequents*	*Support*	*Confidence*	*Lift*
scheduling	genetic programming	0.128	0.333	1.567
genetic programming	scheduling	0.128	0.600	1.567
scheduling	optimization	0.106	0.329	1.088
optimization	scheduling	0.106	0.278	1.088
optimization	evolutionary algorithms	0.085	0.333	1.362
evolutionary algorithms	optimization	0.085	0.348	1.362
scheduling	timetabling	0.085	0.222	2.321
timetabling	scheduling	0.085	0.889	2.321
scheduling	job shop scheduling	0.064	0.167	2.611
job shop scheduling	scheduling	0.064	1.000	2.611
evolutionary algorithms	scheduling	0.064	0.261	0.681
scheduling	evolutionary algorithm	0.064	0.167	0.681

4.5. Discussion

The literature overview presented in this section summarizes the most outstanding subjects in hyper-heuristics. We must remark that this is the first study to use a text mining approach to analyze hyper-heuristic-related literature up to the date this manuscript was submitted.

The analysis we conducted involved different text mining methods previously discussed. Text mining played a crucial role in understanding the issues and keywords presented in the hyper-heuristic-related literature in this review. We analyzed 963 English hyper-heuristic-related publications, consisting of journal and conference articles in their final publication stage. In these documents, the most relevant terms include "optimization", "scheduling", "genetic algorithm", "evolutionary algorithms", "timetabling", and "combinatorial algorithms". Based on these terms, we extracted the problem domains preferred for developing and testing hyper-heuristics and the main techniques used for generating hyper-heuristics for such domains.

Our findings show a preference for evolutionary algorithms to solve optimization problems when using hyper-heuristics [44,45]. Among the evolutionary algorithms, authors seem to prefer genetic algorithms to produce hyper-heuristics for domains such as vehicle routing problems [46], scheduling [15], and school timetabling [18]. Recently, many researchers have used particle swarm algorithms in hyper-heuristic applications [47,48]. The findings also unveil the interest in hyper-heuristics for different variants of scheduling problems, such as education [49], job shop [50], waiting time variance [51], and nurse rostering problems [52].

We now discuss the major areas identified from the systematic literature review methodology. These areas are structured according to the seven areas identified through our cluster analysis.

4.5.1. Cluster 1: Scheduling Problems and Genetic Algorithms

We found evidence that genetic algorithms power most hyper-heuristics used for scheduling problems. In addition to clustering, Table 4 confirms a strong association between scheduling and genetic programming. The first article in this context was published in 2002 [53]. Job-shop scheduling is a popular area among researchers regarding flexible job-shop scheduling [54,55], design of scheduling policies [56], and sequence-dependent setup times [57,58].

Flexible job-shop scheduling allows each job to operate on more than one machine. Different implementations of genetic algorithms have been used to produce hyper-heuristics for this problem. Examples include the neighborhood-based genetic algorithm (NGA) [59] and hybrid versions, such as the HGA-TS, which is a genetic algorithm that incorporates Tabu Search [60]. The study of the design of scheduling policies proposes automatic methods for different problems, their contributions, and challenges [61]. The most important goal in this area is to combine different methods using evolutionary algorithms. Finally, recent publications have focused on setup times and makespan minimization solved by employing genetic programming with hyper-heuristics. Some approaches consider using sequence-dependent setup times (SDST) [57,62,63] to improve manufacturing planning, supply chain management, and logistics.

4.5.2. Cluster 2: Evolutionary Algorithms

As suggested in the previous lines, evolutionary algorithms are the most widely used technique in the literature concerning hyper-heuristics. The publications analyzed focus on demonstrating which genetic algorithm methods are the most effective. We found an important term related to evolutionary algorithms: simulated annealing [64]. Simulated annealing was used with successful results on several benchmark datasets, especially in the scheduling [65] and bin packing [66] problems.

There is a small portion of publications that focus on the comparison of evolutionary algorithms with artificial intelligence, especially in the transportation [67], performance optimization [68], and medical fields [69]. The relationship between evolutionary algorithms and optimization is also evident in the association rules from Table 4.

4.5.3. Cluster 3: Vehicle Routing Problems and Genetic Algorithms

Historically, routing research began in 1959 with the truck dispatching problem proposed by Dantzig and Ramser [70]. The complexity of the traveling salesman problem (TSP) evolved over the years, and the need to solve new optimization problems grew, which opened an opportunity to use hyper-heuristics. We found previous studies using Tabu Search or simulated annealing. In 2003, the first genetic algorithm study for vehicle routing problems showed that genetic algorithms competed favorably with previous algorithms in terms of time and quality [46]. The most prominent areas are capacitated routing [71,72], routing policies [73] and combinatorial optimization [46,74].

4.5.4. Cluster 4: Multi-Objective Optimization Problems with Evolutionary Algorithms

Multi-objective optimization is a form of multi-criteria decision-making involving objective functions to be optimized. Recent years have seen the rise of research in hyper-heuristics for multi-objective problems. Before 2015, there were few publications on this topic, and the researchers mainly focused on single-objective optimization problems.

We found effective results in experiments using the great deluge algorithm as (i) a function choice component in multi-objective problems [75] and (ii) a solver for software module clustering problems to minimize coupling and maximize cohesion [76]. Moreover, we recognized other positive results implementing hyper-heuristics and epsilon greedy selection algorithms [77], multi-objective agent-based hyper-heuristics [78], and reinforcement learning hyper-heuristic schemes with a multi-objective simulated annealing algorithm [79].

4.5.5. Cluster 5: Scheduling Problems

Hyper-heuristics for tackling scheduling problems is a subject of much attention in academia and industry. The goal is to develop an efficient method to improve scheduling heuristics, considering computational time and resource improvement. The most commonly used scheduling problems are related to projects, resource constraints, and workflow scheduling.

A vital scheduling goal is to reduce the cost and duration of projects. An evolutionary hyper-heuristic for the software project scheduling problem was presented to solve operators that would be best suited for process search [80]. Chand et al. implemented a MAP-Elites-based hyper-heuristic for priority rule discovery in a resource-constrained project scheduling problem [81]. The results were superior in diversity and performance compared to previous models, including genetic programming.

Computational advances require new technologies to process large-scale data. Hence, Hadoop MapReduce is a solution for fast and parallel data processing. Panneerselvam and Subbarama employed MapReduce workflows in the cloud for efficient scheduling on massive data [82]. Kenari and Shamsi proposed a new method to improve the hyper-heuristic scheduling for cloud computing using a decision tree algorithm [83]. Their simulation demonstrated a performance improvement compared to existing approaches.

4.5.6. Cluster 6: Timetabling Problems

Timetabling problems in hyper-heuristics have been presented mainly in education-related scenarios. Pillay provided an overview of the effectiveness of using hyper-heuristics to solve educational timetabling problems [84]. This topic focuses on the placement of resources to satisfy different constraints. The most frequently timetabling topics are associated with examinations, university courses, exams, school, and combinatorial optimization.

Timetabling is a recurrent problem of combinatorial optimization, particularly in educational instances. Therefore, it is studied with great interest by researchers and educators. Ahmed et al. utilized 19 benchmark instances of the third international timetabling competition (ITC 2011) [85] to solve high school timetabling problems. They employed a set of nine low-level heuristics in a selection hyper-heuristic [10]. The authors combined the random permutation method with the adaptive great deluge algorithm. Their results proved that the hyper-heuristic selection performed better than most of the algorithms tested in the past. Muklason et al. dealt with the university examination timetabling problem using hyper-heuristics based on the great deluge algorithm [86]. Their manuscript reported a better solution than the metaheuristics proposed in the literature while outperforming simulated annealing and hill climbing algorithms. Another example is exam timetabling. For example, Dewi et al. looked for the optimal solution in the shortest possible time using the Toronto benchmark database [87]. Their study showed that the use of Tabu Search had the best performance with the lowest penalty.

4.5.7. Cluster 7: Constraint Satisfaction Problems

The last cluster focuses on constraint satisfaction problems. Due to its multiple applications, constraint satisfaction is a fundamental problem in hyper-heuristics. In the studies we have examined, the ordering of variables in which constraint satisfaction is presented is also important. The first work on hyper-heuristic for solving constraint satisfaction problems dates back to 2008 when Terashima et al. proposed using a genetic algorithm to produce rules that map instances to heuristics [88]. Afterward, Bittle and Fox proposed a symbolic cognitive architecture, augmented with constraint-based reasoning, to improve the ordering of the variables throughout the search [89]. In a similar approach, Crawford et al. proposed a framework that changes the heuristics already in operation. The substitutions are made based on hyper-heuristics, and the parameters are calibrated by a genetic algorithm, with promising results [90]. More recently, Ortiz-Bayliss et al. implemented existing machine learning techniques to accelerate hyper-heuristic development and outperform some heuristic algorithms in the literature [91]. Only a few works have addressed the importance of the practical applications of constrained problems. For example, Peraza et al. explored differential evolution to generate hyper-heuristics for solving real-world situations [43]. These are modeled as fixed-integer non-linear programming problems.

5. Conclusions and Future Work

As far as we are aware, this study is the first to explore text mining algorithms for the field of hyper-heuristics. We conducted brief research on 963 manuscripts published in journals and conferences written in English to explore potential topics in the area of hyper-heuristics. Traditional literature review systems are time-consuming and lack objectivity in the results. This motivated us to conduct this systematic literature review using text mining techniques, machine learning algorithms, and graphical representations to discover different trends related to hyper-heuristics. Therefore, we provided a practical example of text mining and clustering as an alternative solution to understand the literature and provide researchers with the tools to create strategic plans.

As we stated earlier in this document, the objective of this work was to provide new researchers with valuable information for entering the hyper-heuristic research area. In the following lines, we summarize some relevant ideas that could be useful for starters in this research area:

- This study revealed the increase in hyper-heuristics publications in the last three years (cf. Figure 2). About the analyzed journals, *Applied Soft Computing*, *Expert Systems with Applications*, and *Information Sciences* are the most popular options for publications on hyper-heuristic related works. Considering the number of articles and citations, we found that the most relevant journals regarding hyper-heuristics are the *European Journal of Operation Research* and *IEEE Transactions on Evolutionary Computation* (cf. Figure 3).
- We noticed that publications concerning hyper-heuristics are strongly related to genetic and particle swarm algorithms. These techniques are widely used in other domains and associated with other terms (Tables 1 and 3). In addition, scheduling, when solved with hyper-heuristics, is the domain most often associated with other terms such as optimization, evolutionary algorithms, and timetabling (Table 4). We also observed that scheduling, along with optimization, is the most-used term in the word cloud (cf. Figure 4), which confirms its value for the hyper-heuristic community. It is important to highlight that many works that use hyper-heuristics for scheduling problems have focused on generation hyper-heuristics powered mainly by genetic programming.
- Three problem domains recurrently appeared in our clusters: scheduling (including job shop and timetabling), vehicle routing and constraint satisfaction. All but the last seem to remain active nowadays.

Although this study provides a new perspective on hyper-heuristics, it has some limitations. First, we collected the data only from Scopus. Second, this work focused on journal and conference articles. Third, we mainly focused on the frequency of reports. Still, we did not consider only the most relevant studies for hyper-heuristics. Future work could address the limitations of collecting information for multiple data sources and analyze book chapters related to hyper-heuristics. We also remark on the need for a strategy to select relevant hyper-heuristic publications to better understand the distribution of concepts.

Author Contributions: Conceptualization, A.K.G.-E. and J.C.O.-B.; Formal analysis, A.K.G.-E.; Funding acquisition, A.K.G.-E.; Investigation, A.K.G.-E. and J.C.O.-B.; Methodology, A.K.G.-E., I.A., J.M.C.-D. and J.C.O.-B.; Project administration, H.T.-M. and J.C.O.-B.; Software, A.K.G.-E. and J.C.O.-B.; Supervision, J.C.O.-B.; Validation, I.A., J.M.C.-D. and J.C.O.-B.; Visualization, A.K.G.-E. and J.C.O.-B.; Writing—original draft, A.K.G.-E. and J.C.O.-B.; Writing—review & editing, I.A., J.M.C.-D., H.T.-M. and J.C.O.-B.. All authors have read and agreed to the published version of the manuscript.

Funding: This research was funded by Tecnológico de Monterrey through its Fund for financing the publication of Scientific Articles.

Institutional Review Board Statement: Not applicable.

Informed Consent Statement: Not applicable.

Data Availability Statement: Not applicable.

Conflicts of Interest: The authors declare no conflict of interest.

References

1. Burke, E.K.; Gendreau, M.; Hyde, M.R.; Kendall, G.; Ochoa, G.; Özcan, E.; Qu, R. Hyper-heuristics: A survey of the state of the art. *J. Oper. Res. Soc.* **2013**, *64*, 1695–1724. [CrossRef]
2. Özcan, E.; Bilgin, B.; Korkmaz, E.E. A Comprehensive Analysis of Hyper-Heuristics. *Intell. Data Anal.* **2008**, *12*, 3–23. [CrossRef]
3. Sánchez, M.; Cruz-Duarte, J.M.; Ortíz-Bayliss, J.c.; Ceballos, H.; Terashima-Marin, H.; Amaya, I. A Systematic Review of Hyper-Heuristics on Combinatorial Optimization Problems. *IEEE Access* **2020**, *8*, 128068–128095. 3009318. [CrossRef]
4. Drake, J.H.; Kheiri, A.; Özcan, E.; Burke, E.K. Recent advances in selection hyper-heuristics. *Eur. J. Oper. Res.* **2020**, *285*, 405–428. [CrossRef]

5. Akundi, A.; Tseng, B.; Wu, J.; Smith, E.; Subbalakshmi, M.; Aguirre, F. Text Mining to Understand the Influence of Social Media Applications on Smartphone Supply Chain. *Procedia Comput. Sci.* **2018**, *140*, 87–94. [CrossRef]
6. Luque, C.; Luna, J.M.; Luque, M.; Ventura, S. An advanced review on text mining in medicine. *WIREs Data Min. Knowl. Discov.* **2019**, *9*, e1302. [CrossRef]
7. Kumar, S.; Kar, A.K.; Ilavarasan, P.V. Applications of text mining in services management: A systematic literature review. *Int. J. Inf. Manag. Data Insights* **2021**, *1*, 100008. [CrossRef]
8. Ishikiriyama, C.S.; Miro, D.; Gomes, C.F.S. Text Mining Business Intelligence: A small sample of what words can say. *Procedia Comput. Sci.* **2015**, *55*, 261–267. [CrossRef]
9. Gupta, A.; Dengre, V.; Kheruwala, H.A.; Shah, M. Comprehensive review of text-mining applications in finance. *Financ. Innov.* **2020**, *6*, 1–25. [CrossRef]
10. Ahmed, L.N.; Özcan, E.; Kheiri, A. Solving high school timetabling problems worldwide using selection hyper-heuristics. *Expert Syst. Appl.* **2015**, *42*, 5463–5471. [CrossRef]
11. de la Rosa-Rivera, F.; Nunez-Varela, J.I.; Ortiz-Bayliss, J.C.; Terashima-Marín, H. Algorithm selection for solving educational timetabling problems. *Expert Syst. Appl.* **2021**, *174*, 114694. [CrossRef]
12. Paquay, C.; Limbourg, S.; Schyns, M.; Oliveira, J.F. MIP-based constructive heuristics for the three-dimensional Bin Packing Problem with transportation constraints. *Int. J. Prod. Res.* **2018**, *56*, 1581–1592. [CrossRef]
13. Feld, S.; Roch, C.; Gabor, T.; Seidel, C.; Neukart, F.; Galter, I.; Mauerer, W.; Linnhoff-Popien, C. A Hybrid Solution Method for the Capacitated Vehicle Routing Problem Using a Quantum Annealer. *Front. ICT* **2019**, *6*, 13. [CrossRef]
14. Alkhanak, E.N.; Lee, S.P. A hyper-heuristic cost optimisation approach for Scientific Workflow Scheduling in cloud computing. *Future Gener. Comput. Syst.* **2018**, *86*, 480–506. [CrossRef]
15. Lin, J.; Zhu, L.; Gao, K. A genetic programming hyper-heuristic approach for the multi-skill resource constrained project scheduling problem. *Expert Syst. Appl.* **2020**, *140*, 112915. [CrossRef]
16. Talib, R.; Hanif, M.K.; Ayesha, S.; Fatima, F. Text Mining: Techniques, Applications and Issues. *Int. J. Adv. Comput. Sci. Appl.* **2016**, *7*. [CrossRef]
17. Domović, D.; Rolich, T.; Golub, M. Evolutionary hyper-heuristic for solving the strip-packing problem. *J. Text. Inst.* **2019**, *110*, 1141–1151. [CrossRef]
18. Raghavjee, R.; Pillay, N. A genetic algorithm selection perturbative hyper-heuristic for solving the school timetabling problem. *ORiON* **2015**, *31*, 39–60. [CrossRef]
19. Burke, E.K.; Hyde, M.R.; Kendall, G.; Ochoa, G.; Ozcan, E.; Woodward, J.R., Exploring Hyper-heuristic Methodologies with Genetic Programming. In *Computational Intelligence: Collaboration, Fusion and Emergence*; Mumford, C.L., Jain, L.C., Eds.; Springer: Berlin/Heidelberg, Germany, 2009; pp. 177–201. [CrossRef]
20. Lim, K.C.W.; Wong, L.P.; Chin, J.F. Simulated-annealing-based hyper-heuristic for flexible job-shop scheduling. *Eng. Optim.* **2022**, *0*, 1–17. [CrossRef]
21. Wu, C.C.; Bai, D.; Chen, J.H.; Lin, W.C.; Xing, L.; Lin, J.C.; Cheng, S.R. Several variants of simulated annealing hyper-heuristic for a single-machine scheduling with two-scenario-based dependent processing times. *Swarm Evol. Comput.* **2021**, *60*, 100765. [CrossRef]
22. Sim, K.; Hart, E. An Improved Immune Inspired Hyper-Heuristic for Combinatorial Optimisation Problems. In Proceedings of the 2014 Annual Conference on Genetic and Evolutionary Computation, Vancouver, BC, Canada, 12–16 July 2014; Association for Computing Machinery: New York, NY, USA, 2014; pp. 121–128. [CrossRef]
23. Sim, K.; Hart, E.; Paechter, B. A Lifelong Learning Hyper-heuristic Method for Bin Packing. *Evol. Comput.* **2015**, *23*, 37–67. [CrossRef] [PubMed]
24. Koulinas, G.; Kotsikas, L.; Anagnostopoulos, K. A particle swarm optimization based hyper-heuristic algorithm for the classic resource constrained project scheduling problem. *Inf. Sci.* **2014**, *277*, 680–693. [CrossRef]
25. Nguyen, S.; Zhang, M. A PSO-based hyper-heuristic for evolving dispatching rules in job shop scheduling. In Proceedings of the 2017 IEEE Congress on Evolutionary Computation (CEC), Sebastián, Spain, 5–8 June 2017; pp. 882–889. [CrossRef]
26. Choong, S.S.; Wong, L.P.; Lim, C.P. Automatic design of hyper-heuristic based on reinforcement learning. *Inf. Sci.* **2018**, *436-437*, 89–107. [CrossRef]
27. de Santiago Júnior, V.A.; Özcan, E.; de Carvalho, V.R. Hyper-Heuristics based on Reinforcement Learning, Balanced Heuristic Selection and Group Decision Acceptance. *Appl. Soft Comput.* **2020**, *97*, 106760. [CrossRef]
28. Tsai, C.W.; Song, H.J.; Chiang, M.C. A hyper-heuristic clustering algorithm. In Proceedings of the 2012 IEEE International Conference on Systems, Man, and Cybernetics (SMC), Seoul, Korea, 14–17 October 2012; pp. 2839–2844. [CrossRef]
29. Yates, W.B.; Keedwell, E.C. Clustering of Hyper-Heuristic Selections Using the Smith-Waterman Algorithm for Offline Learning. In Proceedings of the Genetic and Evolutionary Computation Conference Companion, Berlin, Germany, 15–19 July 2017; Association for Computing Machinery: New York, NY, USA, 2017; pp. 119–120. [CrossRef]
30. Leilei, S.; Guoqing, C.; Hui, X.; Chonghui, G. Cluster Analysis in Data-Driven Management and Decisions. *J. Manag. Sci. Eng.* **2017**, *2*, 227. [CrossRef]
31. Carneiro, E.; Dias, L.; Cunha, A.; Mialaret, L. Cluster Analysis and Artificial Neural Networks A Case Study in Credit Card Fraud Detection. In Proceedings of the 2015 12th International Conference on Information Technology-New Generations, Las Vegas, NV, USA, 13–15 April 2015. [CrossRef]

32. Guo, Q.; Lu, X.; Gao, Y.; Zhang, J.; Yan, B.; Su, D.; Song, A.; Zhao, X.; Wang, G. Cluster analysis: A new approach for identification of underlying risk factors for coronary artery disease in essential hypertensive patients. *Sci. Rep.* **2017**, *7*, 43965. [CrossRef]
33. Ünvan, Y. Market basket analysis with association rules. *Commun. Stat. Theory Methods* **2020**, *50*, 1–14. 610926.2020.1716255. [CrossRef]
34. Hegland, M. The Apriori Algorithm—A Tutorial. *Word Sci.* **2008**, *11*, 209–262. [CrossRef]
35. Burnham, J.F. Scopus database: A review. *Biomed. Digit. Libr.* **2006**, *3*, 1–8. [CrossRef]
36. Mongeon, P.; Paul-Hus, A. The journal coverage of Web of Science and Scopus: A comparative analysis. *Scientometrics* **2016**, *106*, 213–228. [CrossRef]
37. Bhanot, N.; Singh, H.; Sharma, D.; Jain, H.; Jain, S. Python vs. R: A Text Mining Approach for analyzing the Research Trends in Scopus Database. *arXiv* **2019**, arXiv:cs.CY/1911.08271.
38. Sinoara, R.A.; Antunes, J.; Rezende, S.O. Text mining and semantics: A systematic mapping study. *J. Braz. Comput. Soc.* **2017**, *23*, 1–20. [CrossRef]
39. Arthur, D.; Vassilvitskii, S. K-Means++: The Advantages of Careful Seeding. In Proceedings of the Eighteenth Annual ACM-SIAM Symposium on Discrete Algorithms, New Orleans LA, USA, 7–9 January 2017; Society for Industrial and Applied Mathematics: Philadelphia, PA, USA, 2007; pp. 1027–1035.
40. Saputura, D.M.; Saputura, D.; Oswari, L.D. Effect of Distance Metrics in Determining K-Value in K-Means Clustering Using Elbow and Silhouette Method. *In Proceedings of the Sriwijaya International Conference on Information Technology and Its Applications (SICONIAN 2019);* Atlantis Press: Paris, France, 2020; pp. 341–346. [CrossRef]
41. Agrawal, R.; Imieliński, T.; Swami, A. Mining Association Rules between Sets of Items in Large Databases. *SIGMOD Rec.* **1993**, *22*, 207–216. [CrossRef]
42. Agrawal, R.; Mannila, H.; Srikant, R.; Toivonen, H.; Verkamo, A.I. Fast discovery of association rules. *Adv. Knowl. Discov. Data Min.* **1996**, *12* , 307–328.
43. Peraza-Vázquez, H.; Torres-Huerta, A.M.; Flores-Vela, A. Self-Adaptive Differential Evolution Hyper-Heuristic with Applications in Process Design. *Computacióny Sist.* **2016**, *20* , 173–193. [CrossRef]
44. Pappa, G.L.; Ochoa, G.; Hyde, M.R.; Freitas, A.A.; Woodward, J.; Swan, J. Contrasting Meta-Learning and Hyper-Heuristic Research: The Role of Evolutionary Algorithms. *Genet. Program. Evolvable Mach.* **2014**. *15*, 3–35. [CrossRef]
45. de Sá, A.G.C.; Pappa, G.L. A Hyper-Heuristic Evolutionary Algorithm for Learning Bayesian Network Classifiers. In *Ibero-American Conference on Artificial Intelligence* ; Springer: Cham, Switzerland, 2014.
46. Baker, B.M.; Ayechew, M. A genetic algorithm for the vehicle routing problem. *Comput. Oper. Res.* **2003**, *30*, 787–800. [CrossRef]
47. Bonab, M.B.; Mohd Hashim, S.Z.; Haur, T.Y.; Kheng, G.Y. A New Swarm-Based Simulated Annealing Hyper-Heuristic Algorithm for Clustering Problem. *Procedia Comput. Sci.* **2019**, *163*, 228–236. [CrossRef]
48. Okulewicz, M.; Mańdziuk, J. A Particle Swarm Optimization hyper-heuristic for the Dynamic Vehicle Routing Problem. *arXiv* **2020**, arXiv:cs.NE/2006.08809.
49. Drake, J.H.; Hyde, M.; Khaled, I.; Özcan, E. A genetic programming hyper-heuristic for the multidimensional knapsack problem. *Kybernetes* **2014**, *43*. [CrossRef]
50. Garza-Santisteban, F.; Sánchez-Pámanes, R.; Puente-Rodríguez, L.A.; Amaya, I.; Ortiz-Bayliss, J.C.; Conant-Pablos, S.; Terashima-Marín, H. A Simulated Annealing Hyper-heuristic for Job Shop Scheduling Problems. In Proceedings of the 20 19 IEEE Congress on Evolutionary Computation (CEC), Wellington, New Zealand, 10–13 June 2019; pp. 57–64. [CrossRef]
51. Mahapatra, S.; Dash, R.R.; Pradhan, S.K. Heuristics Techniques for Scheduling Problems with Reducing Waiting Time Variance. In *Heuristics and Hyper-Heuristics-Principles and Applications;* IntechOpen: Rijeka, Croatia, 2017; pp. 43–64.
52. Anwar, K.; Awadallah, M.A.; Khader, A.T.; Al-betar, M.A. Hyper-heuristic approach for solving nurse rostering problem. In Proceedings of the 2014 IEEE Symposium on Computational Intelligence in Ensemble Learning (CIEL), Orlando, FL, USA, 9–12 December 2014; pp. 1–6. [CrossRef]
53. Nguyen, S.; Zhang, M.; Johnston, M.; Tan, K.C. A coevolution genetic programming method to evolve scheduling policies for dynamic multi-objective job shop scheduling problems. In Proceedings of the 2012 IEEE Congress on Evolutionary Computation, Brisbane, Australia, 10–15 June 2012; pp. 1–8. [CrossRef]
54. Lin, J. Backtracking search based hyper-heuristic for the flexible job-shop scheduling problem with fuzzy processing time. *Eng. Appl. Artif. Intell.* **2019**, *77*, 186–196. [CrossRef]
55. Grobler, J.; Engelbrecht, A. Hyper-heuristics for the Flexible Job Shop Scheduling Problem with Additional Constraints. In *International Conference on Swarm Intelligence* ; Springer; Cham, Switzerland, 2016; Volume 9713, pp. 3–10. [CrossRef]
56. Nguyen, S.; Zhang, M.; Johnston, M.; Tan, K.C. Automatic Design of Scheduling Policies for Dynamic Multi-objective Job Shop Scheduling via Cooperative Coevolution Genetic Programming. *IEEE Trans. Evol. Comput.* **2014**, *18*, 193–208. [CrossRef]
57. Song, H.B.; Lin, J. A genetic programming hyper-heuristic for the distributed assembly permutation flow-shop scheduling problem with sequence dependent setup times. *Swarm Evol. Comput.* **2021**, *60*, 100807. [CrossRef]
58. Burcin Ozsoydan, F.; Sağir, M. Iterated greedy algorithms enhanced by hyper-heuristic based learning for hybrid flexible flowshop scheduling problem with sequence dependent setup times: A case study at a manufacturing plant. *Comput. Oper. Res.* **2021**, *125*, 105044. [CrossRef]
59. Nouri, H.E.; Belkahla Driss, O.; Ghédira, K. Solving the flexible job shop problem by hybrid metaheuristics-based multiagent model. *J. Ind. Eng. Int.* **2018**, *14*, 1–14. [CrossRef]

60. Wang, Y.; Zhu, Q. A Hybrid Genetic Algorithm for Flexible Job Shop Scheduling Problem with Sequence-Dependent Setup Times and Job Lag Times. *IEEE Access* **2021**, *9*, 104864–104873. [CrossRef]
61. Branke, J.; Nguyen, S.; Pickardt, C.; Zhang, M. Automated Design of Production Scheduling Heuristics: A Review. *IEEE Trans. Evol. Comput.* **2015**, *20*, 110–124. [CrossRef]
62. Luo, M.; Lin, J.; Xu, L. Solving Flexible Job-Shop Problem with Sequence-Dependent Setup Times by Using Selection Hyper-Heuristics. In Proceedings of the 2nd International Conference on Artificial Intelligence and Advanced Manufacture, 15–17 October 2020, Manchester, UK; Association for Computing Machinery: New York, NY, USA, 2020; pp. 428–433. [CrossRef]
63. He, L.; Weerdt, M.; Yorke-Smith, N. Time/sequence-dependent scheduling: the design and evaluation of a general purpose tabu-based adaptive large neighbourhood search algorithm. *J. Intell. Manuf.* **2020**, *31*, 1051–1078. [CrossRef]
64. Bai, R.; Blazewicz, J.; Burke, E.; Kendall, G.; Mccollum, B. B.: A simulated annealing hyper-heuristic methodology for flexible decision support. *4OR* **2012**, *10*, 43–66. [CrossRef]
65. Kartika, H.M.; Ahmad, M. Self Adaptive and Simulated Annealing Hyper-Heuristics Approach for Post-Enrollment Course Timetabling. *J. Phys. Conf. Ser.* **2020**, *1577*, 012033. [CrossRef]
66. Dowsland, K.; Soubeiga, E.; Burke, E. A simulated annealing based hyperheuristic for determining shipper sizes for storage and transportation. *Eur. J. Oper. Res.* **2007**, *179*, 759–774. [CrossRef]
67. Cheng, J.; Yang, B.; Gen, M.; Jang, Y.; Liang, C.J. "Machine Learning based Evolutionary Algorithms and Optimization for Transportation and Logistics". *Comput. Ind. Eng.* **2020**, *143*, 106372. [CrossRef]
68. Zhang, T.; Liu, Q.; Dan, Y.; Yu, S.; Han, X.; Dai, J.; Xu, K. Machine learning and evolutionary algorithm studies of graphene metamaterials for optimized plasmon-induced transparency. *Opt. Express* **2020**, *28*, 18899–18916. [CrossRef] [PubMed]
69. Dura-Bernal, S.; Neymotin, S.A.; Kerr, C.C.; Sivagnanam, S.; Majumdar, A.; Francis, J.T.; Lytton, W.W. Evolutionary algorithm optimization of biological learning parameters in a biomimetic neuroprosthesis. *IBM J. Res. Dev.* **2017**, *61*, 6:1–6:14. [CrossRef] [PubMed]
70. Dantzig, G.B.; Ramser, J.H. The Truck Dispatching Problem. *Manag. Sci.* **1959**, *6*, 80–91. [CrossRef]
71. Mlejnek, J.; Kubalik, J. Evolutionary hyperheuristic for capacitated vehicle routing problem. In Proceedings of the 15th Annual Conference Companion on Genetic and Evolutionary Computation, Amsterdam, The Netherlands, 6 July 2013; pp. 219–220. [CrossRef]
72. MacLachlan, J.; Mei, Y.; Branke, J.; Zhang, M. Genetic Programming Hyper-Heuristics with Vehicle Collaboration for Uncertain Capacitated Arc Routing Problems. *Evol. Comput.* **2019**, 1–29. [CrossRef] [PubMed]
73. Wang, S.; Mei, Y.; Zhang, M. A Multi-Objective Genetic Programming Hyper-Heuristic Approach to Uncertain Capacitated Arc Routing Problems. In Proceedings of the 2020 IEEE Congress on Evolutionary Computation (CEC), Glasgow, UK, 19–24 July 2020; pp. 1–8. [CrossRef]
74. Pillay, N.; Qu, R. Vehicle Routing Problems. In *Hyper-Heuristics: Theory and Applications*; Springer International Publishing: Cham, Switzerland, 2018; pp. 51–60.
75. Maashi, M.; Kendall, G.; Özcan, E. Choice function based hyper-heuristics for multi-objective optimization. *Appl. Soft Comput.* **2015**, *28*, 312–326. [CrossRef]
76. Alshareef, H.; Maashi, M. Application of Multi-Objective Hyper-Heuristics to Solve the Multi-Objective Software Module Clustering Problem. *Appl. Sci.* **2022**, *12*, 5649. [CrossRef]
77. Yang, T.; Zhang, S.; Li, C. A multi-objective hyper-heuristic algorithm based on adaptive epsilon-greedy selection. *Complex Intell. Syst.* **2021**, *7*, 765–780. [CrossRef]
78. de Carvalho, V.R.; Özcan, E.; Sichman, J.S. Comparative Analysis of Selection Hyper-Heuristics for Real-World Multi-Objective Optimization Problems. *Appl. Sci.* **2021**, *11*, 9153. [CrossRef]
79. Cao, P.; Tang, J. A Reinforcement Learning Hyper-Heuristic in Multi-Objective Single Point Search with Application to Structural Fault Identification. *arXiv* **2018**, arXiv:1812.07958.
80. Wu, X.; Consoli, P.; Minku, L.; Ochoa, G. An Evolutionary Hyper-heuristic for the Software Project Scheduling Problem. In *International Conference on Parallel Problem Solving from Nature* ; Springer: Cham, Switzerland, 2016; Volume 9921, pp. 37–47.
81. Chand, S.; Rajesh, K.; Chandra, R. MAP-Elites based Hyper-Heuristic for the Resource Constrained Project Scheduling Problem. *arXiv* **2022**, arXiv:2204.11162. [CrossRef]
82. Panneerselvam, A.; Subbaraman, B. Hyper Heuristic MapReduce Workflow Scheduling in Cloud. In Proceedings of the 2018 2nd International Conference on I-SMAC (IoT in Social, Mobile, Analytics and Cloud) , Palladam, India, 30–31 August 2018; pp. 691–693. [CrossRef]
83. Kenari, A.R.; Shamsi, M. A hyper-heuristic selector algorithm for cloud computing scheduling based on workflow features. *OPSEARCH* **2021**, *58*, 852–868. [CrossRef]
84. Pillay, N. A review of hyper-heuristics for educational timetabling. *Ann. Oper. Res.* **2016**, *239*, 3–38. [CrossRef]
85. Post, G.; Schaerf, A.; Kingston, J.; Di Gaspero, L. The third international timetabling competition. *Ann. Oper. Res.* **2016**, *239*, 69–75. [CrossRef]
86. Muklason, A.; Syahrani, G.B.; Marom, A. Great Deluge Based Hyper-heuristics for Solving Real-world University Examination Timetabling Problem: New Data set and Approach. *Procedia Comput. Sci.* **2019**, *161*, 647–655. [CrossRef]
87. Dewi, S.; Tyasnurita, R.; Pratiwi, F. Solving examination timetabling problem within a hyper-heuristic framework. *Bull. Electr. Eng. Inform.* **2021**, *10*, 1611–1620. [CrossRef]

88. Terashima-Marín, H.; Ortiz-Bayliss, J.C.; Ross, P.; Valenzuela-Rendón, M. Hyper-Heuristics for the Dynamic Variable Ordering in Constraint Satisfaction Problems. In *Proceedings of the 10th Annual Conference on Genetic and Evolutionary Computation;* Association for Computing Machinery: New York, NY, USA, 2008; pp. 571–578. [CrossRef]
89. Bittle, S.A.; Fox, M.S. Learning and Using Hyper-Heuristics for Variable and Value Ordering in Constraint Satisfaction Problems. In *Proceedings of the 11th Annual Conference Companion on Genetic and Evolutionary Computation Conference: Late Breaking Papers;* Association for Computing Machinery: New York, NY, USA, 2009; pp. 2209–2212. [CrossRef]
90. Crawford, B.; Castro, C.; Monfroy, E.; Soto, R.; Palma, W.; Paredes, F. Dynamic Selection of Enumeration Strategies for Solving Constraint Satisfaction Problems. *Rom. J. Inf. Sci. Technol.* **2012**, *15*, 106–128.
91. Ortiz-Bayliss, J.C.; Amaya, I.; Cruz-Duarte, J.M.; Gutierrez-Rodriguez, A.E.; Conant-Pablos, S.E.; Terashima-Marín, H. A General Framework Based on Machine Learning for Algorithm Selection in Constraint Satisfaction Problems. *Appl. Sci.* **2021**, *11*, 2749. [CrossRef]