A Comprehensive Review of Evolutionary Algorithms for Multiprocessor DAG Scheduling

Eduardo C. da Silva and Paulo H. R. Gabriel*

Faculty of Computer Science, Federal University of Uberlândia, Uberlândia, MG 38408-100, Brazil; eduardo.iftm@gmail.com
* Correspondence: phrg@ufu.br

Received: 18 March 2020; Accepted: 2 April 2020; Published: 10 April 2020

Abstract: The multiprocessor task scheduling problem has received considerable attention over the last three decades. In this context, a wide range of studies focuses on the design of evolutionary algorithms. These papers deal with many topics, such as task characteristics, environmental heterogeneity, and optimization criteria. To classify the academic production in this research field, we present here a systematic literature review for the directed acyclic graph (DAG) scheduling, that is, when tasks are modeled through a directed acyclic graph. Based on the survey of 56 works, we provide a panorama about the last 30 years of research in this field. From the analyzes of the selected studies, we found a diversity of application domains and mapped their main contributions.

Keywords: DAG scheduling; evolutionary algorithms; systematic literature review; computational environment; optimization criteria

1. Introduction

The multiprocessor task scheduling problem (MTSP) consists of mapping a task graph onto a target platform [1]. In this model, the graph represents the parallel application, that is, vertices denote computational tasks and edges model precedence constraints between tasks. The objective is to find a mapping of tasks over processors in order to optimize some performance criteria [2].

Formally, tasks and their respective relations are modeled using a directed acyclic graph (DAG), $G = (T, E, w)$, where the (finite) set $T = \{t_1, \ldots, t_n\}$ represents the tasks (vertices), and set $E$ (edges) represents precedence constraints between tasks, that is, $(t_i, t_j) \in E$ if and only if $t_j$ depends on the execution of $t_i$. Moreover, the weight function $w: T \to \mathbb{N}^*$ gives the number of processing instructions required by each task [1]. A task $t_i$ may send (communicate) data to a successor task $t_j$; this situation is represented by weights on the DAG edges, indicate the communication cost. When the same processor executes both tasks, this cost is negligible. Figure 1a [3] illustrates a DAG with $n = 8$ tasks and its execution costs (for instance, $t_1/3$ means that task $t_1$ has a cost of 3 million of processing instructions). In this same graph, edges represent a precedence relationship, and its weights are the communication costs.

The computational environment consists of a set $P = \{p_1, \ldots, p_m\}$ of widely interconnected processors. In a homogeneous environment, all processors have the same processing capacity (given in millions of instructions per second—MIPS). On the other hand, in heterogeneous environments, the computational cost of each task may vary depending on the processor where it is attributed. Finally, a solution to the MTSP can be evaluated in several ways: execution time of application, resource utilization, and others. Figure 1b [3] shows a scheduling solution, represented as a Gantt Chart for two identical processors. In this solution, tasks $\{t_1, t_3, t_4, t_7\}$ are allocated over processor $p_1$ and $\{t_2, t_5, t_6, t_8\}$ over processor $p_2$. In this example, the execution time of the application consists of 12-time units.
Similar to many optimization problems in the real world, the search for the optimal MTSP solution has a high computational complexity [1,2] and cannot be solved promptly by brute force techniques. One of the most investigated approaches to this type of problem is the use of techniques that, in many cases, discover a set of potential solutions, from which one can find the global optimum or an approximate solution to that problem. In this context, evolutionary algorithms (EAs) have been investigated as a search method due to their flexibility to solve complex optimization problems.

EAs [4] can be adapted to various types of problems. Inspired by the evolutionary paradigm, these techniques work with the evolution of several solutions (population) simultaneously, ensuring a significant advantage when compared to other techniques. There are several studies involving EAs applied to the scheduling problem, and this number increases every year. For this reason, it is necessary to systematically organize these works in order to summarize their main characteristics, understand contributions, authors, and institutions that maintain relevant publications. With such information, it is possible to identify domains that may receive attention to additional research in the future.

One way to investigate a significant amount of studies, and synthesize their contributions, is through a Systematic Literature Review (SLR) [5,6]. SLR consists of a rigorous procedure to select and analyze works that fit a research question without following a bias on the part of the researchers. Moreover, the results of SLR may be replicated and extended as needed.

This paper proposes the application of SLR techniques to find a collection of works that proposes EAs to MTSP. In this way, we present a comprehensive survey of these works through a meta-analytic approach, examining authors, institutions, and other aspects. We organize works considering task characteristics, computational environments, and optimization criteria. This survey covers an interval of 30 years (from 1990 to 2019) of research papers; thus, we also map the historical evolution of studies in this scientific field. Our investigation considered four digital libraries that index computing studies; after the three stages of papers analysis, we created a collection composed of 56 studies (This paper is an extension of previous work by the same authors [3]. The original work focused only on journal articles and was limited to studies based on genetic algorithms. Moreover, that study was less rigorous in terms of inclusion and exclusion criteria and considered works until 2018.). The next section of this paper details our research protocol, presenting research questions, search key, and inclusion/exclusion criteria. Afterward, Section 3 presents the collection of 56 papers and provides a meta-analysis of them. Finally, we draw our concluding remarks (Section 4) and expose the references list.

2. Research Protocol

According to Kitchenham [5], the Systematic Literature Review (SLR) is a process of interpreting and evaluating relevant studies related to a research question, topic, or phenomenon. Consequently, we consider the SLR as an outstanding tool for aggregating experiences from a collection of different works [7].
The execution of an SLR follows some formal steps, named research protocol: (i) limit the research question; (ii) compose a search key; (iii) choose the databases; (iv) establish the inclusion/exclusion criteria; and (v) report the stages of the review.

The first step of an SLR consists of limiting the scope of the research. In the context of this paper, we are interested in mapping the last three decades of EAs applied to the MTSP. Moreover, we desire to characterize the application scenarios of such algorithms, that is, we intend to record which authors are dealing with which particularities of this problem (e.g., presence of communication among tasks, processors heterogeneity, and others).

Having this scope in mind, and concerning the SLR protocol, we formulated the main research question (referred to as MQ), as follows:

MQ: “Between the years 1990 to 2019, what are the central studies on evolutionary algorithms applied to the multiprocessor task scheduling problem?”

In addition to the direction of MQ, it is necessary to prepare other questions (called secondary questions—SQ) to expand the collection with relevant information. These questions are necessary to guide the research process, providing directions to select relevant works. In this paper, we considered nine secondary questions, enumerated here:

SQ.01: What is the period with the most significant amount of research papers?
SQ.02: What are the computational environments considered in these studies?
SQ.03: How do these studies deal with the cost of communication?
SQ.04: What are the main optimization criteria considered by the studies?
SQ.05: Which of these papers has the highest number of citations?
SQ.06: Which regions of the planet have the highest number of authors?
SQ.07: What are the most used expressions in abstracts and keywords?

For survey works corresponding to the questions above, we drew up a search key and applied it to the search mechanisms of the digital databases. For this, we eliminated corresponding keywords and combined different subjects using the logical operators OR and AND, respectively. Since the search key influences the behavior of the search engine, it is necessary to build it verifying each employed term. Table 1 presents the search key constructed in this study.

| Table 1. Combinations used to compose the search key. |
|------------------------------------------------------|
| (“evolutionary algorithm” OR “genetic algorithm”) AND |
| “task scheduling” AND |
| (parallel OR multiprocessor) AND |
| (“directed acyclic graph” OR DAG OR workflow) AND |
| (representation OR encoding) AND |
| “genetic operators” AND |
| “objective function” |

In our research, we considered “evolutionary” and “genetic” algorithms as synonymous. Indeed, several authors do not provide a proper distinction between both terms; for this reason, we opted to maintain both expressions and filter results (according to De Jong [4], EAs are a class of algorithms and genetic algorithms are techniques from such class). We also regarded the words “graph”, “workflow” and “DAG” in conjunction, as suggested by Robert [1]. Besides, we included both “parallel” and “multiprocessor” terms.

Finally, the search key includes the expressions “representation”, “encoding”, “genetic operators”, and “objective function”. All of these are specific for the context of evolutionary algorithms, and we considered them in order to select papers that provide details about algorithm design.

After preparing the search key, it is necessary to select some digital databases. Such databases must contain search engines that reproduce the semantics of the key to find related works. Table 2 shows
the bases used in this study. It is necessary to highlight that intended to consider the ACM Digital Library (https://dl.acm.org/); however, its search engine did not reflect the main aspects contained in the search key, resulting in unsatisfactory outcomes when compared to the other databases.

Table 2. Databases used in the search.

| Database                              | URL                              |
|---------------------------------------|----------------------------------|
| Google Scholar                        | scholar.google.com               |
| IEEE Xplore Digital Library          | ieeexplore.ieee.org              |
| Elsevier-Science Direct               | sciencedirect.com                |
| Portal de Periódicos Capes            | periodicos.capes.gov.br          |

Although the search key encompasses several terms related to research questions, the amount of selected works on digital databases can be pervasive. Thus, the protocol used in the SLR must define some inclusion and exclusion criteria that reflect the format and desired aspects of the investigation. The criteria adopted in this protocol are:

1. Papers must deal with MTSP in its static version and without the use of release times and deadlines (in other words, all information related to the problem is known before scheduling and the execution time is not limited);
2. The research papers must deal with MTSP using EAs solutions, that is, EAs are the focus of the study;
3. Selected studies cannot combine EAs with other meta-heuristics;
4. All the papers must be in the language;
5. Papers must be available for consultation on Internet platforms.

Based on all the rules defined here, we performed the following steps, named search phase, as required by the SLR protocol (we documented each SLR execution phase using annotations and spreadsheets—thus, we may extend this survey or adapt it to answer other research questions.):

1. **Search in search engines**: We perform the search step and collected all resulting works for future analysis;
2. **Results filtering**: we discarded works that did not meet the inclusion/exclusion criteria;
3. **Manual search**: we carried out a new survey of works by looking at the bibliographic references of the first pre-surveyed collection. We also submitted these new studies to the inclusion/exclusion criteria.

3. Analysis of the Collection

This section presents the collection of works recovered with the SLR. Each part answers one or more secondary questions presented in our SLR protocol. Thus, in Section 3.1, we answer the SQ.01; Section 3.2 deals with SQ.02 and SQ.03; Section 3.3 answers SQ.04, and Section 3.4 SQ.05. Sections 3.5 and 3.6 are related to SQ.06 and SQ.07, respectively. Finally, we present a summary in Section 3.7, answering our main research question (MQ).

3.1. Collection and Publication Timeline

This SLR is part of a more comprehensive research project [3,8]. For this reason, we collected the data into separated moments. The first collection took place in April 2018 and returned 430 works distributed as follows:

- Google Scholar: 237 works;
- Elsevier-Science Direct: 157 works;
- Portal de Periódicos Capes: 34 works;
- IEEE Xplore Digital Library: 2 works.
However, we observe a total of 50 works duplicated, that is, they appeared in two or more databases; consequently, after deleting duplicates, our research returned 380 papers. From this collection, we selected 120 papers based on information from their abstracts. Afterward, we applied the inclusion/exclusion criteria (second phase of SLR protocol) and reduced the number of selected papers to 36. Finally, in the third phase, we analyzed the section “References” of such papers, selecting more 27 new works; therefore, our first collection had 63 papers.

In order to extend our research and include new articles, covering the years 2018 and 2019, we repeated all those steps, performing a new search. We found 84 new results, but only five of them attempted to the inclusion/exclusion criteria; thus, we generated a final collection of 68 works (this new survey was conducted on February 12, 2020.). Before proceeding with the analysis, we opted to follow an additional exclusion criterion—remove works without the Digital Object Identifier (https://www.doi.org/) (DOI). We took this decision to allow the recovery of detailed information about selected works; after this last step, we eliminated 12 papers, obtained a collection with 56 works.

Table 3 presents all the works that compose our collection in chronological order. When two or more papers were published in the same year, we organize them in alphabetical order. In this table, we also present how the work was published, that is, if it is a journal article (“Jour”) or a conference paper (“Conf”); in this last case, the publication year refers to the publication date of the conference proceedings.

Table 3. Summary of the collection in chronological order. We employ the ID along this paper to referee the works.

| ID | Reference                  | Year | Type | ID | Reference                  | Year | Type |
|----|----------------------------|------|------|----|----------------------------|------|------|
| 1  | Hou et al. [9]             | 1990 | Conf | 29 | Kaur et al. [10]           | 2010 | Jour |
| 2  | Benten and Sait [11]       | 1994 | Jour | 30 | Omara and Arafa [12]       | 2010 | Jour |
| 3  | Hou et al. [13]            | 1994 | Jour | 31 | Chitra et al. [14]         | 2011 | Conf |
| 4  | Wang and Korfhage [15]     | 1995 | Conf | 32 | Daoud and Kharma [16]      | 2011 | Jour |
| 5  | Aguilar and Gelenbe [17]   | 1997 | Jour | 33 | Kang et al. [18]           | 2011 | Conf |
| 6  | Kwok and Ahmad [19]        | 1997 | Conf | 34 | Sathappan et al. [20]      | 2011 | Jour |
| 7  | Tsuchiya et al. [21]       | 1997 | Conf | 35 | Wang et al. [22]           | 2011 | Jour |
| 8  | Tsujimura and Gen [23]     | 1997 | Conf | 36 | Ahmad et al. [24]          | 2012 | Conf |
| 9  | Wang et al. [25]           | 1997 | Jour | 37 | Kaur and Singh [26]        | 2012 | Conf |
| 10 | Woo et al. [27]            | 1997 | Conf | 38 | Panwar et al. [28]         | 2012 | Conf |
| 11 | Tsuchiya et al. [29]       | 1998 | Jour | 39 | Singh and Singh [30]       | 2012 | Jour |
| 12 | Corrêa et al. [31]         | 1999 | Jour | 40 | Xu et al. [32]             | 2012 | Conf |
| 13 | Jezic et al. [33]          | 1999 | Conf | 41 | Dhingra and Gupta [34]     | 2013 | Jour |
| 14 | Zomaya et al. [35]         | 1999 | Jour | 42 | Guzek et al. [36]          | 2014 | Jour |
| 15 | Liu et al. [37]            | 2000 | Conf | 43 | Singh and Pillai [38]      | 2014 | Conf |
| 16 | Topcuoglu and Sevimlis [39]| 2000 | Conf | 44 | Xu et al. [40]             | 2014 | Jour |
| 17 | Zhong and Yang [41]        | 2003 | Conf | 45 | Hassan et al. [42]         | 2015 | Jour |
| 18 | Wu et al. [43]             | 2004 | Jour | 46 | Ahmad et al. [44]          | 2016 | Jour |
| 19 | Yao et al. [45]            | 2004 | Jour | 47 | Amirjanov and Sobolev [46] | 2016 | Jour |
| 20 | Demiroz and Topcuoglu [47] | 2006 | Jour | 48 | Morady and Dal [48]        | 2016 | Conf |
| 21 | Jelodar et al. [49]        | 2006 | Conf | 49 | Sheikh et al. [50]         | 2017 | Jour |
| 22 | Ramachandra and Elmaghraby [51] | 2006 | Jour | 50 | Akbari et al. [52]         | 2017 | Jour |
| 23 | Azghadi et al. [53]        | 2008 | Conf | 51 | Gandhi et al. [54]         | 2017 | Conf |
| 24 | Hwang et al. [55]          | 2008 | Jour | 52 | Pillai et al. [56]         | 2017 | Jour |
| 25 | Bonyadi and Moghaddam [57] | 2009 | Jour | 53 | Saad et al. [58]           | 2018 | Conf |
| 26 | Pop et al. [59]            | 2009 | Conf | 54 | Ahmad and Sheikh [60]      | 2019 | Jour |
| 27 | Chitra et al. [61]         | 2010 | Conf | 55 | Sheikh et al. [62]         | 2019 | Jour |
| 28 | Gupta et al. [63]          | 2010 | Conf | 56 | Wangsom et al. [64]        | 2019 | Jour |

Note that only one paper was published in 1990 [9], and, after this, two others appeared in 1994 [11,13]. Closing the collection, three papers were published in 2019 [60,62,64]. In all, 31 papers (∼55%) were published in journals. Based on Table 3, Figure 2 presents the timeline of publications. Answering the first secondary research question (SQ.01), the period between 2011 and 2012 is the interval with the most significant number of publications.
Another essential characteristic to be highlighted is that our research only returned works based on genetic algorithms, which means that our search key, together with our exclusion criteria, eliminates works that design other evolutionary techniques. This situation was expected since we consider keywords “genetic” and “evolutionary”; moreover, EAs tends to appear with other meta-heuristics, violating our inclusion/exclusion criteria.

3.2. Environments and Communication Cost

As already mentioned, MTSP can present variations in its multiprocessor system. In this context, papers of our collection deal with two primary environments: (i) homogeneous processors, that is, the execution time of a task does not vary depending on where it is executed; and (ii) heterogeneous processors. There are some papers that present solutions for both scenarios as well as papers that model communication costs. Table 4 lists all the works from the collection organized according to their environments.

As observed, most of the works (29 in total) pay attention to the homogeneous environment, that is, when all the processors have the same processing capacity. Besides, 24 papers are related to the heterogeneous environment, and four works [27,28,43,49] deal with both scenarios. Azghadi et al. [53] did not provide sufficient information about where their algorithm is applied; we believe, based on information presented in the publication, that they also considered both.

Based on Table 4, the chart in Figure 3 distributes publications on a timeline (in this chart, we did not consider Azghadi et al. [53]), responding to the SQ.02. In addition to the significant growth between 2010 and 2011, the heterogeneous scenario is also common in the latest studies. This situation may be related to the popularity of MTSP in the Infrastructure as a Service (IaaS) paradigm, that is, virtual machines adjusted by frequency controls.

![Figure 2. Timeline of publications.](image-url)
We also analyzed the publications considering the use of the communication cost, that is, when DAG edges have weights. In this scenario, a task sends data (in bytes) to another, which imposes an additional delay in the execution (without communication, the successor task only waits for the execution of its predecessor). Responding to SQ.03, it is possible to observe that most works employ this scenario (about 80%), representing the most substantial volume of publications. Table 5 lists these works.

### Table 5. Presence of communication costs.

| Communication | Works ID  |
|---------------|-----------|
| No            | 1, 3, 8, 14, 15, 22, 28, 35, 43, 45, 52 |
| Yes           | All the others. |

#### 3.3. Optimization Criteria

There are several ways to measure the performance of an MTSP solution. Such measurement is responsible for driving the search process in an evolutionary algorithm. In practice, the quality of a solution is employed in the design of fitness functions that must be optimized. Table 6 lists all the metrics found in the collection in order to respond to the SQ.04.

### Table 6. Performance metrics for the MTSP.

| Metric                   | Works ID |
|--------------------------|----------|
| Makespan                 | All the works, except 5 |
| Energy consumption       | 42, 49, 52, 54, 55 |
| Load balancing           | 5, 26, 30, 46 |
| System reliability       | 27, 31, 34, 35 |
| Flowtime                 | 27, 29, 53 |
| Communication cost       | 5, 56 |
| Number of processors     | 16, 20 |
| System temperature       | 48, 55 |
| Parallelism              | 50 |
| Financial cost           | 56 |

The most widely used metric is the *makespan* (sometimes referred to as *scheduling length*), which is present in the vast majority of the works surveyed (55 in total). This metric corresponds to the time instant when all the tasks have finished processing (in Figure 1, for example, the makespan value is equated to 12-time units). Besides makespan, the literature in this area also considered other metrics related to the execution time, that is, *flowtime* [10,34,58,61] and *communication cost* [17,64].
Another popular metric is the load balancing \[12,17,44,59\], which consists of the equitable distribution of tasks over processors. This metric is directed related to better resource utilization: if all the processors are overloaded, it is necessary more investments in the computational infrastructure.

It is also possible to observe the increase, in recent years, in the number of works that consider metrics focused on sustainability \[36,48,50,56,60,62\]. These works deal with issues related to the temperature and the energy consumption of computers (processors). Sustainability metrics are related to the popularization of distributed systems, mainly because of the adoption of computational clouds. Another recently adopted metric is the financial cost \[64\], which is related to scenarios that employ IaaS platforms (the performance of virtual machines depends on the frequency values contracted by the client).

All these performance metrics are related to optimization criteria. Most of them (makespan, financial cost, energy consumption, flowtime, temperature, number of processors, and communication cost) must be minimized. On the other hand, system reliability, load balancing, and parallelism are used to design maximization objective functions.

It is essential to highlight that several works consider two or more of these optimization criteria in conjunction. Indeed, 19 papers related to makespan also consider at least one of the other metrics. In most of these cases, works consider a weighted sum to combine these objectives; however, there is also multiobjective evolutionary approaches, based on the concept of Pareto’s front \[14,36,50,60–62,64\]. Finally, only Aguilar and Gelenbe \[17\] do not directly consider the minimization of makespan, having their focus on load balancing and communication cost.

### 3.4. Number of Citations

In this study, we also computed the number of citations in the work collection, answering the secondary question SQ.05. The chart in Figure 4 classifies the ten most cited publications considering the number of citations returned by the Google Scholar platform. We observe that the most referred publication \[13\] appeared in more than 900 related works. In fact, the work by Hou et al. \[13\] is considered one of the most important in this research field, presenting ideas that were extended by several works. Another prominent work is due to Wang et al. \[25\], with about 500 citations.

![Figure 4. Most cited works according to the Google Citation Rank.](image_url)

Additionally, we related which publication vehicles are preferred. The Journal of Parallel and Distributed Computing published a total of six works \[12,16,19,25,44,60\], followed by the IEEE Transactions on Parallel and Distributed Systems, with five works \[13,31,35,43,50\]. After them, Information Sciences \[17,40\] and Microprocessors and Microsystems \[29,45\] published two papers each; the other two works appeared in the same conference, the IEEE International Conference on High-Performance Computing and Communication, in 2012 \[24,32\].
3.5. Researchers Affiliations

The collection of works also allowed the extraction of information regarding the authors and their respective nationalities. In this context, “nationality” reflects the researcher’s country of affiliation; therefore, it does not necessarily represent his country of origin. We identified 145 distinct researchers in 21 countries, responding to question SQ.06.

Figure 5 presents an infographic distributing researchers from their affiliation countries. The countries with the most considerable number of researchers are India and the United States, both with 26 researchers each, followed by China, having 18. Brazil, Singapore, and Turkey have one researcher each.

![Figure 5. Distribution of researchers around the world.](image)

3.6. Word Frequency Rank

Finally, in order to identify the main terms adopted in publications, this study also extracted the primary expressions used in abstracts and keywords. In Figure 6a, we quantify the most common keywords found. We also isolate the common expressions found in the abstracts, as presented in Figure 6b.

![Figure 6. Most used expressions in keywords (a) and abstracts (b).](image)

In both charts, we linked singular expressions to their respective plural versions. Moreover, this survey responds to the question SQ.07. The identification of the primary expressions can help in the use of terms more related to researches in this field, consequently helping indexing mechanisms and the search process.

3.7. Summary

In this survey, we perform a meta-analysis and answered seven secondary questions (enumerated in Section 2). As a result, it is possible to respond to the main question (MQ) of this study, that is, what are the central works in this research field.
Here we listed the ten most cited works presented in our collection. In order to summarize our answers, Table 7 related some information about these works: reference (“Ref.”), title, publication vehicle, and publication year. We also present the characteristics of the environments, that is, if the work deals with a homogeneous or a heterogeneous scenario, with or without communication costs (“CC”). Finally, we provide the Scientific Journal Rankings (SJR), retrieved on March 13, 2020.

From these works, seven deal with homogeneous environments [12,13,17,19,25,31,35], two with heterogeneous [22,40], and only one with both environments [43]. Additionally, Hou et al. [13], Zomaya et al. [35], and Wang et al. [22] do not model communication costs. All these works, except Aguilar and Gelenbe [17], consider makespan as primarily optimization metric. Besides, Omara and Arafa [12] and Wang et al. [22] also take into account load balancing and system reliability, respectively, as a secondary metric, and Aguilar and Gelenbe [17] employ load balancing and communication cost (none of them, however, design a multiobjective algorithm). At long last, all these works were published in journals.
**Table 7.** Characteristics of the ten most cited works.

| Ref | Title                                                                 | Published in                                      | Year | Environment   | CC? | SJR  |
|-----|-----------------------------------------------------------------------|---------------------------------------------------|------|---------------|-----|------|
| [13] | A genetic algorithm for multiprocessor scheduling                     | IEEE Transactions on Parallel and Distributed Systems | 1994 | Homogeneous   | No  | 0.88 |
| [25] | Task Matching and Scheduling in Heterogeneous Computing Environments Using a Genetic-Algorithm-Based Approach | Journal of Parallel and distributed Computing     | 1997 | Homogeneous   | Yes | 0.42 |
| [43] | An incremental genetic algorithm approach to multiprocessor scheduling | IEEE Transactions on Parallel and Distributed Systems | 2004 | Both          | Yes | 0.88 |
| [31] | Scheduling multiprocessor tasks with genetic algorithms              | IEEE Transactions on Parallel and Distributed Systems | 1999 | Homogeneous   | Yes | 0.88 |
| [12] | Genetic algorithms for task scheduling problem                       | Journal of Parallel and distributed Computing     | 2010 | Homogeneous   | Yes | 0.42 |
| [35] | Genetic scheduling for parallel processor systems: comparative studies and performance issues | IEEE Transactions on Parallel and Distributed Systems | 1999 | Homogeneous   | No  | 0.88 |
| [19] | Efficient scheduling of arbitrary task graphs to multiprocessors using a parallel genetic algorithm | Journal of Parallel and distributed Computing     | 1997 | Homogeneous   | Yes | 0.42 |
| [40] | A genetic algorithm for task scheduling on heterogeneous computing systems using multiple priority queues | Information Sciences                              | 2014 | Heterogeneous | Yes | 1.62 |
| [22] | Optimizing the makespan and reliability for workflow applications with reputation and a look-ahead genetic algorithm | Future Generation Computer Systems                | 2011 | Heterogeneous | No  | 0.84 |
| [17] | Task assignment and transaction clustering heuristics for distributed systems | Information Sciences                              | 1997 | Homogeneous   | Yes | 1.62 |
4. Conclusions and Future Research

This paper has reported the results of a systematic literature review (SLR) on evolutionary algorithms for the multiprocessor task scheduling problem. Based on four databases, we generated a collection of 56 papers published in journals or presented in conferences between the years of 1990 and 2019, that is, we covered three decades of studies.

From this collection, we presented an overview of general characteristics from the selected works and, consequently, from this research field. We observed that homogeneous computational environments are still the most common, with 29 published works (about 52% of the collection). Moreover, more than 80% of selected works deal with communication costs. All the papers except one look for the minimization of makespan, sometimes in conjunction with other optimization criteria.

In this survey, we focus on genetic algorithms; this means that we did not cover other evolutionary approaches (such as differential evolution, estimation of distribution algorithms, memetic algorithms, swarm intelligence, and others). We took this decision to limit our research to the most common evolutionary technique.

Finally, from this collection, we can now extract a large volume of information about the structure of GAs. For example, it is possible to analyze details about solution encoding (i.e., chromosome), genetic operators (mutation and crossover), as well as mechanisms to deal with variability and convergence. Further studies may include an experimental evaluation of these design decisions.

Regarding the proposed framework for this SLR, it can be extended in future works to grant other scheduling problems (for example, considering independent tasks) or solving methods.

Author Contributions: E.C.d.S. and P.H.R.G. contributed equally to this paper. All authors have read and agreed to the published version of the manuscript.

Funding: This research was funded by FAPEMIG (Minas Gerais Research Foundation), Brazil, grant number APQ-03081-18. Any opinions, findings, and conclusions or recommendations expressed in this material are those of the authors and do not necessarily reflect the views of FAPEMIG.

Acknowledgments: The authors would like to thank the support of the Post-graduation Program in Computer Sciences from the Federal University of Uberlandia (PPGCO/UFU). We are also in debt with FAPEMIG for the master’s scholarship.

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

Abbreviations

The following abbreviations are used in this manuscript:

- DOI Digital Object Identifier
- EA Evolutionary algorithm
- GA Genetic algorithm
- IaaS Infrastructure as a Service
- MTSP Multiprocessor task scheduling problem
- SJR Scientific Journal Rankings
- SLR Systematic literature review

References

1. Robert, Y. Task graph scheduling. In Encyclopedia of Parallel Computing; Padua, D., Ed.; Springer: Boston, MA, USA, 2011; pp. 2013–2025.
2. Afrati, F.; Papadimitriou, C.H.; Papageorgiou, G. Scheduling dags to minimize time and communication. In VLSI Algorithms and Architectures; Reif, J.H., Ed.; Springer: New York, NY, USA, 1988; Volume 319, pp. 134–138.
3. Silva, E.C.; Gabriel, P.H.R. Algoritmos genéticos e escalonamento de tarefas em multiprocessadores: Uma revisão sistemática da literatura. In Anais do Encontro Nacional de Inteligência Artificial e Computacional (ENIAC); Bianchi, R.A.C., Rios, R.A., Eds.; Sociedade Brasileira de Computação—SBC: Porto Alegre, RS, Brasil, 2020; pp. 250–261.
4. De Jong, K. *Evolutionary Computation: A Unified Approach*; The MIT Press: Cambridge, MA, USA, 2006; p. 268.
5. Kitchenham, B.A. Procedures for Performing Systematic Reviews; Technical Report TR/SE-0401; Department of Computer Science, Keele University: Keele, UK, 2004.
6. Levy, Y.; Ellis, T.J. A Systems Approach to Conduct an Effective Literature Review in Support of Information Systems Research. *Informing Sci. Int. J. Emerg. Transdiscipl.* 2006, 9, 181–212, doi:10.28945/479. [CrossRef]
7. Budgen, D.; Brereton, P. Performing systematic literature reviews in software engineering. In Proceedings of the 28th International Conference on Software Engineering—ICSE ’06, Shanghai, China, 20–28 May 2006; pp. 1051–1052.
8. Silva, E.C. Representações de Algoritmos Genéticos para o Problema de Escalonamento Estático de Tarefas em Multiprocessadores. Mather’s Thesis, Universidade Federal de Uberlândia, Uberlândia, MG, Brazil, 2020.
9. Hou, E.S.H.; Hong, R.; Ansari, N. Efficient multiprocessor scheduling based on genetic algorithms. In Proceedings of the 16th Annual Conference of IEEE Industrial Electronics Society, Pacific Grove, CA, USA, 27–30 November 1990; pp. 1239–1243.
10. Kaur, K.; Chhabra, A.; Singh, G. Modified Genetic Algorithm for Task Scheduling in Homogeneous Parallel System Using Heuristics. *Int. J. Soft Comput.* 2010, 5, 42–51, doi:10.3923/ijscsom.2010.42.51. [CrossRef]
11. Benten, M.S.T.; Sait, S.M. Genetic scheduling of task graphs. *Int. J. Electron.* 1994, 77, 401–415, doi:10.1080/00207219408926072. [CrossRef]
12. Omara, F.A.; Arafa, M.M. Genetic algorithms for task scheduling problem. *J. Parallel Distrib. Comput.* 2010, 70, 13–22, doi:10.1016/j.jpdc.2009.09.009. [CrossRef]
13. Hou, E.S.H.; Ansari, N.; Ren, H. A genetic algorithm for multiprocessor scheduling. *IEEE Trans. Parallel Distrib. Syst.* 1994, 5, 113–120, doi:10.1109/71.265940. [CrossRef]
14. Chitra, P.; Venkatesh, P.; Rajaram, R. Comparison of evolutionary computation algorithms for solving bi-objective task scheduling problem on heterogeneous distributed computing systems. *Sadhana* 2011, 36, 167–180, doi:10.1007/s12046-011-0014-8. [CrossRef]
15. Wang, P.C.; Korfhage, W. Process scheduling using genetic algorithms. In Proceedings of the Seventh IEEE Symposium on Parallel and Distributed Processing, San Antonio, TX, USA, 25–28 October 1995; pp. 638–641.
16. Daoud, M.I.; Kharma, N. A hybrid heuristic–genetic algorithm for task scheduling in heterogeneous processor networks. *J. Parallel Distrib. Comput.* 2011, 71, 1518–1531, doi:10.1016/j.jpdc.2011.05.005. [CrossRef]
17. Aguilar, J.; Gelenbe, E. Task assignment and transaction clustering heuristics for distributed systems. *Inf. Sci.* 1997, 97, 199–219, doi:10.1016/s0020-0255(96)00178-8. [CrossRef]
18. Kang, Y.; Zhang, Z.; Chen, P. An activity-based genetic algorithm approach to multiprocessor scheduling. In Proceedings of the Seventh International Conference on Natural Computation, Shanghai, China, 26–28 July 2011; Volume 2, pp. 1048–1052.
19. Kwok, Y.K.; Ahmad, I. Efficient scheduling of arbitrary task graphs to multiprocessors using a parallel genetic algorithm. *J. Parallel Distrib. Comput.* 1997, 47, 58–77, doi:10.1006/jpdc.1997.1395. [CrossRef]
20. Sathappan, O.L.; Chitra, P.; Venkatesh, P.; Prabhu, M. Modified genetic algorithm for multiobjective task scheduling on heterogeneous computing system. *Int. J. Inf. Technol. Commun. Converg.* 2011, 1, 146–158, doi:10.1504/ijitcc.2011.039282. [CrossRef]
21. Tsuchiya, T.; Osada, T.; Kikuno, T. A new heuristic algorithm based on GAs for multiprocessor scheduling with task duplication. In Proceedings of the 3rd International Conference on Algorithms and Architectures for Parallel Processing, Melbourne, Australia, 10–12 December 1997; pp. 295–308.
22. Wang, X.; Yeo, C.S.; Buyya, R.; Su, J. Optimizing the makespan and reliability for workflow applications with reputation and a look-ahead genetic algorithm. *Future Gener. Comput. Syst.* 2011, 27, 1124–1134, doi:10.1016/j.future.2011.03.008. [CrossRef]
23. Tsujimura, Y.; Gen, M. Genetic algorithms for solving multiprocessor scheduling problems. In Proceedings of the Asia-Pacific Conference on Simulated Evolution and Learning, Taejon, Korea, 9–12 November 1996; pp. 106–115.
24. Ahmad, S.G.; Munir, E.U.; Nisar, W. PEGA: A Performance Effective Genetic Algorithm for Task Scheduling in Heterogeneous Systems. In Proceedings of the 2012 IEEE 14th International Conference on High Performance Computing and Communication & 2012 IEEE 9th International Conference on Embedded Software and Systems, Liverpool, UK, 25–27 June 2012; pp. 1082–1087.
25. Wang, L.; Siegel, H.J.; Roychowdhury, V.P.; Maciejewski, A.A. Task Matching and Scheduling in Heterogeneous Computing Environments Using a Genetic-Algorithm-Based Approach. *J. Parallel Distrib. Comput.* 1997, 47, 8–22, doi:10.1006/jpdc.1997.1392. [CrossRef]

26. Kaur, R.; Singh, G. Genetic algorithm solution for scheduling jobs in multiprocessor environment. In Proceedings of the 2012 Annual IEEE India Conference (INDICON), Kochi, India, 7–9 December 2012; pp. 968–973.

27. Woo, S.H.; Yang, S.B.; Kim, S.D.; Han, T.D. Task scheduling in distributed computing systems with a genetic algorithm. In Proceedings of the High Performance Computing on the Information Superhighway, Seoul, Korea, 28 April–2 May 1997; pp. 301–305.

28. Panwar, P.; Lal, A.K.; Singh, J. A Genetic Algorithm Based Technique for Efficient Scheduling of Tasks on Multiprocessor System. In Proceedings of the International Conference on Soft Computing for Problem Solving (SoCProS 2011), Roorkee, India, 20–22 December 2012; Volume 131, pp. 911–919.

29. Tsuchiya, T.; Osada, T.; Kikuno, T. Genetics-based multiprocessor scheduling using task duplication. *Microprocess. Microsyst.* 1998, 22, 197–207, doi:10.1016/s0141-9331(98)00079-9. [CrossRef]

30. Singh, J.; Singh, G. Improved task scheduling on parallel system using genetic algorithm. *Int. J. Comput. Appl.* 2012, 39, 17–22, doi:10.5120/4912-7449. [CrossRef]

31. Corrêa, R.C.; Ferreira, A.; Rebreyend, P. Scheduling multiprocessor tasks with genetic algorithms. *IEEE Trans. Parallel Distrib. Syst.* 1999, 10, 825–837, doi:10.1109/71.790600. [CrossRef]

32. Xu, Y.; Li, K.; Khac, T.T.; Qiu, M. A Multiple Priority Queueing Genetic Algorithm for Task Scheduling on Heterogeneous Computing Systems. In Proceedings of the 2012 IEEE 14th International Conference on High Performance Computing and Communication & 2012 IEEE 9th International Conference on Embedded Software and Systems, Liverpool, UK, 25–27 June 2012; pp. 639–646.

33. Jezic, G.; Kostelac, R.; Lovrek, I.; Sinkovic, V. Scheduling Tasks with Non-negligible Intertask Communication onto Multithreaded Processors by using Genetic Algorithms. In *Artificial Neural Nets and Genetic Algorithms*; Dobnikar, A., Steele, N.C., Pearson, D.W., Albrecht, R.F., Eds.; Springer: Vienna, Austria, 1999; pp. 196–201.

34. Dhingra, S.; Gupta, S.B. Genetic algorithm parameters optimization for bi-criteria multiprocessor task scheduling using design of experiments. *Int. J. Comput. Control Quantum Inf. Eng.* 2013, 8, 661–667, doi:10.5281/ZENODO.2813952. [CrossRef]

35. Zomaya, A.Y.; Ward, C.; Macey, B. Genetic scheduling for parallel processor systems: comparative studies and performance issues. *IEEE Trans. Parallel Distrib. Syst.* 1999, 10, 795–812, doi:10.1109/71.790598. [CrossRef]

36. Guzek, M.; Pecero, J.E.; Dorronsoro, B.; Bouvry, P. Multi-objective evolutionary algorithms for energy-aware scheduling on distributed computing systems. *Appl. Soft Comput.* 2014, 24, 432–446, doi:10.1016/j.asoc.2014.07.010. [CrossRef]

37. Liu, D.; Li, Y.; Yu, M. A genetic algorithm for task scheduling in network computing environment. In Proceedings of the Fifth International Conference on Algorithms and Architectures for Parallel Processing, Beijing, China, 23–25 October 2002; pp. 126–129.

38. Singh, K.; Pillai, A.S. Schedule length optimization by elite-genetic algorithm using rank based selection for multiprocessor systems. In Proceedings of the International Conference on Embedded Systems, Paris, France, 20–22 August 2014; pp. 86–91.

39. Topcuoglu, H.; Sevilmis, C. Task Scheduling with Conflicting Objectives. In *Advances in Information Systems*; Yakhno, T., Ed.; Springer: Heidelberg, Germany, 2002; pp. 346–355.

40. Xu, Y.; Li, K.; Hu, J.; Li, K. A genetic algorithm for task scheduling in network computing environment using multiple priority queues. *Inf. Sci.* 2014, 270, 255–287, doi:10.1016/j.ins.2014.02.122. [CrossRef]

41. Zhong, Y.W.; Yang, J.G. A genetic algorithm for tasks scheduling in parallel multiprocessor systems. In Proceedings of the 2003 International Conference on Machine Learning and Cybernetics (IEEE Cat. No.03EX693), Xi’an, China, 2–5 November 2003; Volume 3, pp. 1785–1790.

42. Hassan, M.A.; Kacem, I.; Martin, S.; Osman, I.M. Genetic algorithms for job scheduling in cloud computing. *Stud. Inform. Control* 2015, 24, 387–400, doi:10.24846/v24i4y201503. [CrossRef]

43. Wu, A.S.; Yu, H.; Jin, S.; Lin, K.C.; Schiavone, G. An incremental genetic algorithm approach to multiprocessor scheduling. *IEEE Trans. Parallel Distrib. Syst.* 2004, 15, 824–834, doi:10.1109/tpds.2004.38. [CrossRef]

44. Ahmad, S.G.; Liew, C.S.; Munir, E.U.; Ang, T.F.; Khan, S.U. A hybrid genetic algorithm for optimization of scheduling workflow applications in heterogeneous computing systems. *J. Parallel Distrib. Comput.* 2016, 87, 80–90, doi:10.1016/j.jpdc.2015.10.001. [CrossRef]
45. Yao, W.; You, J.; Li, B. Main sequences genetic scheduling for multiprocessor systems using task duplication. *Microprocess. Microsyst.* 2004, 28, 85–94, doi:10.1016/j.micropro.2003.10.002. [CrossRef]

46. Amirjanov, A.; Sobolev, K. Scheduling of directed acyclic graphs by a genetic algorithm with a repairing mechanism. *Concurr. Comput. Pract. Exp.* 2016, 29, e3954, doi:10.1002/cpe.3954. [CrossRef]

47. Demiroz, B.; Topcuoglu, H.R. Static Task Scheduling with a Unified Objective on Time and Resource Domains. *Comput. J.* 2006, 49, 731–743, doi:10.1093/comjnl/bxl030. [CrossRef]

48. Morady, R.; Dal, D. A multi-population based parallel genetic algorithm for multiprocessor task scheduling with Communication Costs. In Proceedings of the 2016 IEEE Symposium on Computers and Communication (ISCC), Messina, Italy, 27–30 June 2016; pp. 766–772.

49. Jelodar, M.S.; Topcuoglu, H.R. Static Task Scheduling with a Unified Objective on Time and Resource Domains. *Comput. J.* 2006, 49, 731–743, doi:10.1093/comjnl/bxl030. [CrossRef]

50. Sheikh, H.F.; Ahmad, I.; Fan, D. An Evolutionary Technique for Performance-Energy-Temperature Optimized Scheduling of Parallel Tasks on Multi-Core Processors. *IEEE Trans. Parallel Distrib. Syst.* 2016, 27, 668–681, doi:10.1109/tpds.2015.2421352. [CrossRef]

51. Ramachandra, G.; Elmaghraby, S.E. Sequencing precedence-related jobs on two machines to minimize the weighted completion time. *Int. J. Prod. Econ.* 2006, 100, 44–58, doi:10.1016/j.ijpe.2004.10.014. [CrossRef]

52. Akbari, M.; Rashidi, H.; Alizadeh, S.H. An enhanced genetic algorithm with new operators for task scheduling in heterogeneous computing systems. *Eng. Appl. Artif. Intell.* 2017, 61, 35–46, doi:10.1016/j.engappai.2017.02.013. [CrossRef]

53. Pillai, A.S.; Singh, K.; Saravanan, V.; Anpalagan, A.; Woungang, I.; Barolli, L. A genetic algorithm-based method for optimizing the energy consumption and performance of multiprocessor systems. *Soft Comput.* 2008, 37, 3271–3285, doi:10.1007/s10766-008-0107-8. [CrossRef]

54. Pop, F.; Dobre, C.; Cristea, V. Genetic algorithm for DAG scheduling in Grid environments. In Proceedings of the 2009 IEEE 5th International Conference on Intelligent Computer Communication and Processing, Cluj-Napoca, Romania, 27–29 August 2009; pp. 299–305.

55. Ahmad, I.; Sheikh, H.F. A multi-staged niched evolutionary approach for allocating parallel tasks with joint optimization of performance, energy, and temperature. *J. Parallel Distrib. Comput.* 2019, 134, 65–74, doi:10.1016/j.jpdc.2019.05.009. [CrossRef]

56. Chitra, P.; Revathi, S.; Venkatesh, P.; Rajaram, R. Evolutionary algorithmic approaches for solving three objectives task scheduling problem on heterogeneous systems. In Proceedings of the 2nd International Advance Computing Conference, Patala, India, 19–20 February 2010; pp. 38–43.

57. Sheikh, H.F.; Ahmad, I.; Arshad, S.A. Performance, Energy, and Temperature Enabled Task Scheduling using Evolutionary Techniques. *Sustain. Comput. Inform. Syst.* 2019, 22, 272–286, doi:10.1016/j.suscom.2017.10.002. [CrossRef]
63. Gupta, S.; Kumar, V.; Agarwal, G. Task Scheduling in Multiprocessor System Using Genetic Algorithm. In Proceedings of the 2010 Second International Conference on Machine Learning and Computing, Bangalore, India, 9–11 February 2010; pp. 267–271.

64. Wongsom, P.; Lavangnananda, K.; Bouvry, P. Multi-Objective Scientific-Workflow Scheduling With Data Movement Awareness in Cloud. IEEE Access 2019, 7, 177063–177081, doi:10.1109/access.2019.2957998. [CrossRef]