

Genetic algorithms and multiprocessor task scheduling: A systematic literature review

Eduardo C. da Silva¹, Paulo H. R. Gabriel¹

¹Universidade Federal de Uberlândia – Faculdade de Computação
Av. João Naves de Ávila, 2121 – Uberlândia/MG – Brazil

eduardo.iftm@gmail.com, phrg@ufu.br

Abstract. *This paper reports a systematic review of the literature about genetic algorithms applied to the multiprocessor task scheduling problem. After defining a protocol with the main rules of this review, the research was performed considering journal papers published between 1990 and 2018. At the end of this process, 37 works were recovered and analyzed. By performing a meta-analysis, a variety of information was extracted and summarized, including impact factor, Eigenfactor score, scenarios considered, optimization metrics, volume of citations, and others.*

1. Introduction

The reduction of manufacturing costs of processors and interconnection interfaces has motivated the design of parallel and distributed systems, in order to execute high-performance applications. In these environments, parallel application tasks must be allocated over a set of available computational resources (processors), in order to optimize some performance metrics, such as response time and/or system utilization [Hou et al. 1994, Sheikh et al. 2017]. Besides, the precedence requirements for these tasks must be satisfied. This problem is referred to as a multiprocessor task scheduling problem (MTSP).

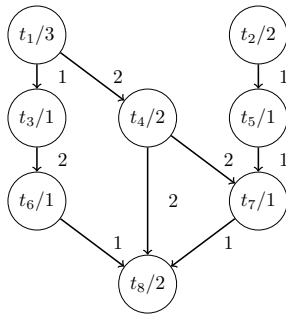
Due to its importance (in fact, the overall performance of the system is highly influenced by the scheduling solution), MTSP has been receiving a large amount of attention. However, it is also a well-defined NP-complete problem [Afrati et al. 1988]. For this reason, several intelligent solutions have been designed in order to find good scheduling solution within adequate processing time. In this context, genetic algorithms (GAs) deserve to be highlighted [Hou et al. 1994, Akbari 2018].

GAs represent a class of evolutionary algorithms (EAs), that can be adapted to various types of problems. Inspired by the evolutionary paradigm, GAs work with the evolution of several solutions (population) simultaneously, ensuring a significant advantage compared to sequential techniques. There are several studies involving GAs applied to the scheduling problem. However, the literature does not present studies that systematically make collections of these works, along with their main characteristics. One way to investigate, summarize, and synthesize these characteristics is through a Systematic Literature Review (SLR) [Levy and Ellis 2006]. This technique consists of a rigorous procedure of selection and analysis of works; consequently, it is possible to answer the questions of research formulated without following a bias on the part of the researchers. Moreover, results of SLR may be replicated and extended.

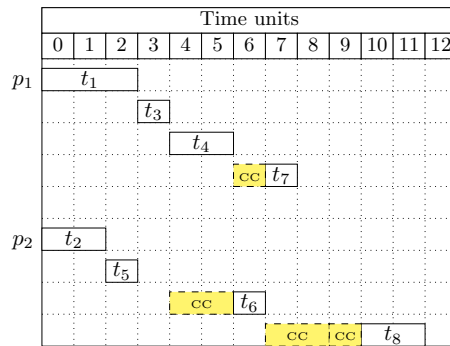
This paper proposes the application of SLR techniques to find a collection of works that proposes GAs to MTSP. We intend to organize these works considering tasks characteristics, system infrastructure, and optimization criteria. Under this collection, we also perform a meta-analysis [Puntel et al. 2018], presenting information such as impact factor, number of citations, and author’s affiliation.

The remainder of this paper is organized as follows. Section 2 gives a description of the MTSP considered here. Section 3 presents the steps of the SLR search protocol applied in this study. The collection of works found is summarized in Section 4 along with the results of our meta-analysis. Finally, conclusions are given in Section 5.

2. Multiprocessor Task Scheduling Problem



(a) Task DAG with $n = 8$.



(b) Gantt chart for $m = 2$.

Figure 1. A MTSP instance. In 1a, we represent a task graph with eight nodes, representing tasks and its execution costs; moreover, edges represent precedence relationship, and its weights are the communication costs. In 1b a solution represented as a Gantt Chart for two processors (“cc” represents the communication cost).

This paper addresses the multiprocessor task scheduling problem (MTSP) with the following characteristics. Tasks are modelled using a directed acyclic graph (DAG), $G = (T, E)$, that represents the various tasks and their respective precedence constraints. In this way, $T = \{t_1, t_2, \dots, t_n\}$ is a set of tasks and E is a set of edges that represents the precedence relationship, i.e., given $t_i, t_j \in T$, if $e_{ij} \in E$, then t_j can only begin its execution after the completion of the task t_i [Omara and Arafa 2010]. The function $w(t_i) > 0$ quantifies the amount of instructions required to perform the tasks t_i , i.e., its execution cost. Similarly, $c(t_i, t_j) > 0$ delimits the communication cost of a task t_i when sending the data to a task t_j , if t_i and t_j are not allocated to the same processor [Tsuchiya et al. 1998].

Moreover, the computational environment consists of a set $P = \{p_1, p_2, \dots, p_m\}$ of processors fully interconnected. In a homogeneous system, the execution cost of a task is the same for each processor in the set. In contrast, in a heterogeneous environment, this cost depends on the capacity of the processor which the task was assigned to. Finally, a solution to the MTSP can be evaluated in several ways, being the makespan, i.e., the end time of the last stepped task, most common one [Omara and Arafa 2010].

An instance of MTSP is represented in Figure 1, considering a homogeneous scenario composed of two processors ($m = 2$) and eight tasks ($n = 8$). Figure 1a illustrates

a task graph. Inside each node, there is the identification of the task with its execution cost and the weights at edges represent the communication cost. Based on the DAG of Figure 1a, Figure 1b shows a Gantt chart representing a mapping of these tasks over the processors. In this case, the subset $\{t_1, t_3, t_4, t_7\}$ was allocated on processor p_1 , while $\{t_2, t_5, t_6, t_8\}$ was assigned to p_2 . In this example, the solution found has a makespan of 12 units of time.

3. Systematic Literature Review Protocol

In this paper, we perform a Systematic Literature Review (SLR), which is defined as the process of gathering, understanding, analysis, synthesis and evaluation of scientific works in order to create state of the art about a given topic [Levy and Ellis 2006]. In this sense, we need to define a research question and, based on this, we design the search key used to find research articles. After that, we analyze the recovered papers to refine the search process. All these steps are described in this section.

3.1. Research Questions

In the context of this paper, i.e., genetic algorithms applied to the multiprocessor task scheduling problem, we design the main research question (MQ) to guide the systematic review, as follows:

MQ: “Between 1990 and 2018, what were the central studies about genetic algorithms applied to the multiprocessor task scheduling problem?”

In order to extend the applicability of the possible results of this study, we also define the following secondary questions (SQs):

- SQ.01:** In what period have we obtained the greatest number of related works?
- SQ.02:** What are the scenarios considered in these works?
- SQ.03:** How do these works handle communication costs?
- SQ.04:** What are the main performance metrics adopted?
- SQ.05:** Which researchers publish more work on MTSP?
- SQ.06:** Which regions on the planet have the biggest number of authors?
- SQ.07:** Which of these works have the greatest impact factor and Eigenfactor score?
- SQ.08:** Which of these works contains the largest number of citations?
- SQ.09:** Which works are the most cited by the other ones?

3.2. Search Key

We elaborate the search key to find works related to the research questions (Section 3.1) Based on that key, we selected a set of keywords that represent aspects of the problem and solution mechanisms. After that, logical operators **AND** and **OR** are employed to eliminate synonyms and join keywords with distinct meanings in a unique search key. Finally, we perform tests with the search key in research databases.

The final version of the search key, used in this work, is presented in the Table 1. The semantic combination of early versions of the key returned many unrelated works. Therefore, a refinement was employed to make the search key more objective.

Table 1. Search key designed for the SLR.

“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”

3.3. Inclusion and Exclusion Criteria

Due to the inherent nature of each search engines, a research key can return numerous non-topic related works. To select appropriated results, we adopted the following inclusion and exclusion criteria:

- Ct.01:** Works have to address only static MTSP with no deadlines, i.e., all information about problem instances is from the beginning, and it does not cover dynamic time limit;
- Ct.02:** Works must propose GAs or EAs solutions for MTSP, i.e., GAs or EAs are not used only as a baseline;
- Ct.03:** Except for heuristics, works do not combine GAs or EAs with other meta-heuristics;
- Ct.04:** Works must be written in the English language;
- Ct.05:** Works have to be available for consults on Web platforms;
- Ct.06:** Works must be published only on journals that can be evaluated by well-accepted scores.

3.4. Databases and Research Phases

The search key (Table 1) was used in search engines available on the Web. We consult the following databases: Google Scholar, IEEE Xplore Digital Library, Elsevier-Science Direct and Portal de Periódicos Capes. We also considered the ACM Digital Library, however, our search key did not return satisfactory results due to particularities of its search engine; thus, we opted to not consider such database.

From these databases, our study was organized into the following phases:

1. **Search by Web search engine:** The main tool used in this SLR is to scan online repositories using the search key since they offer a big amount of results with high accessibility;
2. **Filter results:** We analyze sections focused on the specification of the problem and the details of the techniques. Thus, we excluded works that do not archive one or more inclusion/exclusion criteria (Section 3.3).
3. **Manual research:** With the selected works, we revised papers’ reference lists to find studies that had not been found by search engines.

4. Results of the Review Through a Meta-Analytic Approach

In this section, we summarize the results of the meta-analysis over the works returned by the SLR. This collection of papers answers the research questions presented in Section 3.1. Several criteria were considered to organize this collection, as presented in the following sections.

4.1. Collection and Publication Time-line

Initially, by executing Phase 1, we found a total of 368 related works. Of these, 120 papers were selected based on an analysis of their abstracts. Subsequently, in the second phase, we consider sections that detail the problem and characteristics of the solution method. As a result, we define the first collection of 36 works.

Then, in the third phase, another collection of 29 papers was built from the references cited by the 36 papers already selected; consequently, we found a total of 65 papers. This step was performed to try to find studies that might not have been reached by the search key. Thus, studies published on conferences were maintained until the end of this phase. Finally, we excluded conference papers from our collection, in respect of the **Ct.06** (Section 3.3). In the end, we obtained a set of 37 works present in several journals.

Table 2 chronologically summarizes the collection of papers obtained by the SLR. This also answer our main research question (**MQ**), defined in Section 3.1. Although we defined the time interval between 1990 and 2018, the first work found here was published in 1994 [Hou et al. 1994].

Table 2. Works from the collection.

ID	Research Work	ID	Research Work	ID	Research Work
1	[Hou et al. 1994]	14	[Bonyadi and Moghaddam 2009]	27	[Guzek et al. 2014]
2	[Benten and Sait 1994]	15	[Omara and Arafa 2010]	28	[Dhingra et al. 2014]
3	[Wang et al. 1997]	16	[Kaur et al. 2010b]	29	[Xu et al. 2014]
4	[Aguilar and Gelenbe 1997]	17	[Kaur et al. 2010a]	30	[Hassan et al. 2015]
5	[Kwok and Ahmad 1997]	18	[Daoud and Kharma 2011]	31	[Ahmad et al. 2016]
6	[Tsuchiya et al. 1998]	19	[Chitra et al. 2011]	32	[Sheikh et al. 2016]
7	[Corrêa et al. 1999]	20	[Sathappan et al. 2011]	33	[Sheikh et al. 2017]
8	[Zomaya et al. 1999]	21	[Wang et al. 2011]	34	[Amirjanov and Sobolev 2017]
9	[Wu et al. 2004]	22	[Mohamed and Awadalla 2011]	35	[Akbari et al. 2017]
10	[Yao et al. 2004]	23	[Singh 2012]	36	[Pillai et al. 2018]
11	[Ramachandra and Elmaghraby 2006]	24	[Singh and Singh 2012]	37	[Akbari 2018]
12	[Demiroz and Topcuoglu 2006]	25	[Amalarethinam and Selvi 2012]		
13	[Hwang et al. 2008]	26	[Awadall et al. 2013]		

Works listed in Table 2 are presented, in Figure 2, which shows the number of papers published per year. Answering the **SQ.01**, we observe an average 1.27 paper published per year. This figure also shows an increase in interest in this problem since 2010.

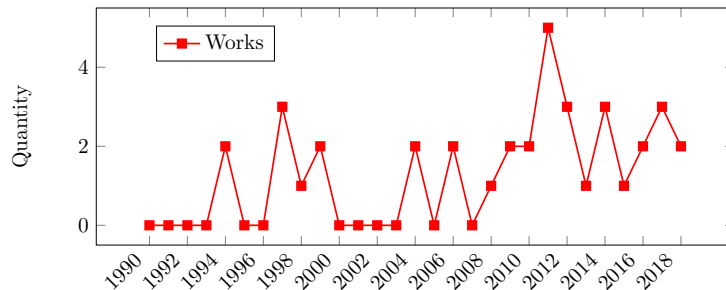


Figure 2. List of works by time-line.

From all this works, it is also possible to extract information about characteristics of the problem studied (e.g., heterogeneity of processors, communication costs among

tasks and performance metrics adopted) and about the GA proposed (e.g., chromosome encoding and genetic operators).

4.2. Heterogeneity and Communication Costs

As described in Section 2, the MTSP can present scenario variations that may influence in the complexity of the problem, such as the heterogeneity of the processors or the presence of cost in data interchange between pairs of tasks. Figure 3 presents the inclusion of these peculiarities along the time.

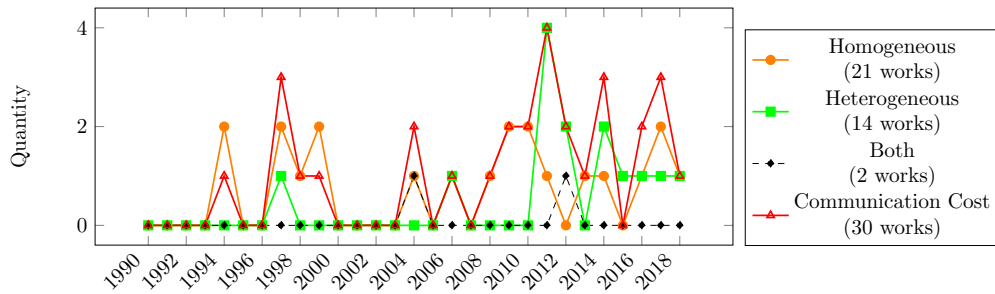


Figure 3. Architecture and communication by time-line.

As we observe, homogeneous computing environments are the most common, reaching a publication average of 0.72 per year, presenting in 21 ($\approx 56.7\%$) papers. However, the interest in heterogeneous scenarios increases in the last decade, being present in 14 ($\approx 37.8\%$) works. Only two papers considered both scenarios [Wu et al. 2004, Singh 2012].

A similar behavior is valid to the presence of communication costs, i.e., this scenario has attract attention in the last years. Moreover, this cost was considered by 30 (81%) works in our collection. In summary, Figure 3 answer both **SQ.02** and **SQ.03** (Section 3.1).

4.3. Performance Metrics

As mentioned in Section 2, several metrics have been used to evaluate different objectives of the MTSP. Based on **SQ.04** (Section 3.1), different metrics were found in this SLR. Table 3 shows the performance metrics considered by each selected paper (such papers are cited considering the ID presented in Table 2).

It is possible to observe the importance given to the minimization of makespan, i.e., the total completion time of tasks (36 papers considered it). Almost all the other metrics were considered together with makespan. Another performance metric that has receiving importance along time is the minimization of the energy consumption of processors.

4.4. Researchers Affiliations

The meta-analysis performed in the collection of studies identified 98 different researchers in 18 different countries. Thus, we answered **SQ.05** and **SQ.06** questions (Section 3.1). In this way, Figure 4 distributes the number of researchers found in each country of a map;

Table 3. Performance metrics and related works.

Performance metrics	Work IDs
Minimize makespan	1, 2, 3, 5, 6, 7, 8, 9, 10, 11, 12, 13, 14, 15, 16, 17, 18, 19, 20, 21, 22, 23, 24, 25, 26, 27, 28, 29, 30, 31, 32, 33, 34, 35, 36, 37
Minimize energy consumption	27, 32, 33, 36
Maximize system reliability	19, 20, 21
Minimize flowtime	16, 17, 28
Maximize load balancing	4, 15, 31
Minimize processor temperature	32, 33
Maximize parallelism	35, 37
Minimize the number of processors	12
Minimize communication cost	4
Maximize resource utilization	25

the nationalities were extracted considering the researchers' affiliation, i.e., it does not represent their home country. USA and India are the countries with the biggest number of researchers (22 and 17, respectively).

The significant amount of researchers and countries involved demonstrates the commitment to better adapt the GAs to the MTSP and in diverse scenarios.

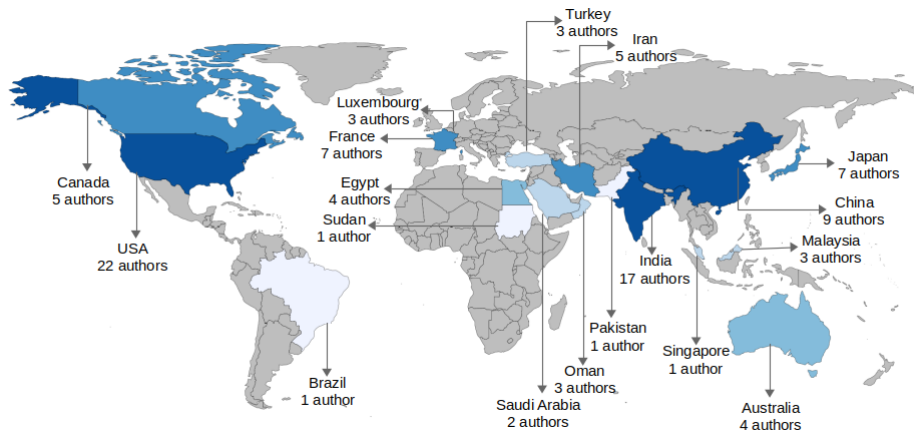


Figure 4. Distribution of researchers around the World.

4.5. Impact Factor and Eigenfactor Score

Scientific articles can have their importance evaluated in several ways, for example, number of citations and journal score. In this paper, we explore two quantitative indices: the Impact Factor¹ and the Eigenfactor score². Impact Factor measures the average number of citations a journal, while the Eigenfactor score quantifies the citations that a journal receives considering the importance of the articles that cite it.

Figure 5a displays the five best-evaluated journals according to the impact factor (*IF*). At the same time, the Eigenfactor score (*EF*) is displayed; its value, however, was

¹Extracted from <https://jcr.clarivate.com>

²Extracted from <http://www.eigenfactor.org/>

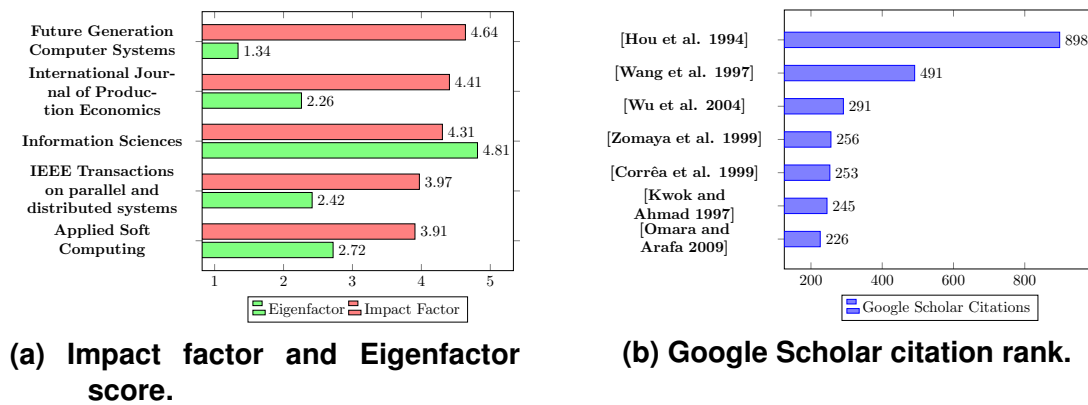


Figure 5. Quality measures of journals.

normalized in order to have the same order of magnitude from the impact factor. Based on this figure, we answer the **SQ.07** (section 3.1).

The journal with the best impact factor is the *Future Generation Computer Systems*, having $IF = 4.639$ points; the work [Wang et al. 2011] as published in this journal. In the second position, we have the *International Journal of Production Economics* with $IF = 4.407$ points (in which the paper [Ramachandra and Elmaghraby 2006] was published). The journal *IEEE Transactions on Parallel and Distributed Systems* appears in the fourth position, but published the largest group of works cited in this paper [Hou et al. 1994, Corrêa et al. 1999, Zomaya et al. 1999, Wu et al. 2004, Sheikh et al. 2016]. When considered the Eigenfactor score, the more relevant journal is the *Information Sciences*, where the two works of this SLR were published [Aguilar and Gelenbe 1997, Xu et al. 2014].

Similarly, addressing the **SQ.08**, Figure 5b quantifies the individual citations of the papers based on the metrics by Google Scholar. In this scenario, the reference [Hou et al. 1994] has the largest number of citations (it is, in fact, one of the seminal works in this research field). It is important to highlight that, despite two works [Wu et al. 2004, Omara and Arafa 2010], all the others that appear in this figure were produced before the 2000 [Hou et al. 1994, Corrêa et al. 1999, Tsuchiya et al. 1998, Zomaya et al. 1999, Kwok and Ahmad 1997]).

In addition to these indexes, we analyze the internal citations present in our collection of works (Table 2). Thus, we can observe the connection and influence of some set of works over others. To answer the question **SQ.09**, Figure 6 present a network of citations considering all the works of the collection. Also, the number of citations was represented in the Figure 7.

As expected (based on the previous discussion), the most influential work is *A genetic algorithm for multiprocessor scheduling* [Hou et al. 1994] which is, also, the oldest one. Such paper was cited by other 20 works from our collection.

5. Conclusion

The systematic literature review is an efficient and rigorous mechanism to collect and analyze papers of a specific topic. By consider rules defined in the review protocol, it is possible to ignore search biases and, more critical, reproduce the search results. In

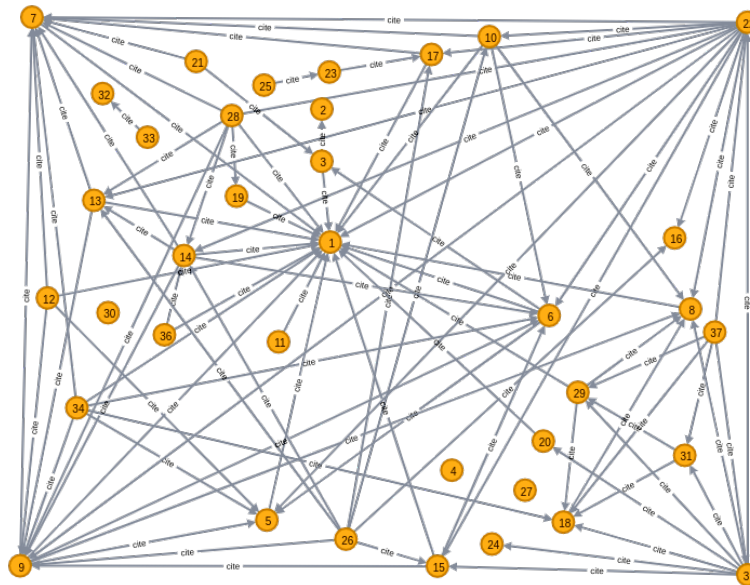


Figure 6. Graph representing the internal citation network.

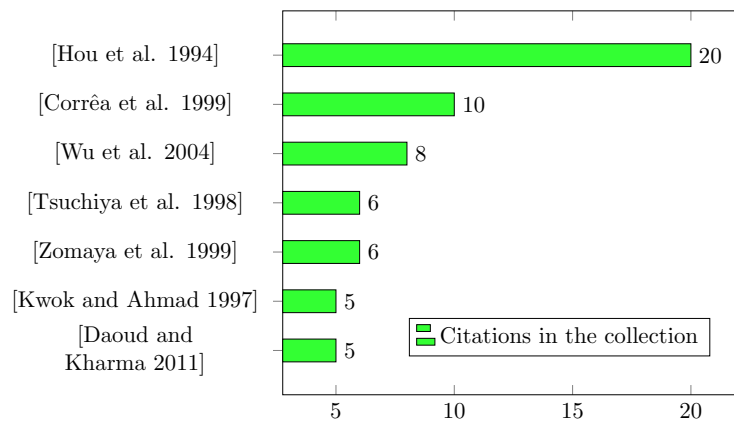


Figure 7. Internal citation rank.

this context, given a publication period, all works that meet the inclusion criteria are selected; besides, the entire process is documented, reporting relevant information that guides reviewing decisions.

This paper reported a SLR about genetic algorithms solutions applied to the multiprocessor task scheduling problem. In this context, it is possible to extract several conclusions:

- Homogeneous systems correspond to the dominant scenario; moreover, heterogeneous models have received interest more recently;
- Communication cost has been explored in the majority of work;
- The main performance metric is the minimization of makespan; also, recent works have explored energy consumption;
- This problem has attracted the interest of several research institutions, but USA and India concentrate the largest number of papers;
- Considering all the journals from the collection, *Future Generation Computer Systems* has the greatest impact factor, while *Information Sciences* has the best

Eigenfactor score;

- The reference [Hou et al. 1994] has the highest number of citations when we consider the Google Scholar rank; in addition, such papers were also the most cited by the others from our collection.

This SLR is part of a more detailed study. We intend to perform a more detailed analysis in terms of characteristics from the proposed GAs. It is possible to evaluate their solution encodes, genetic operators, and objective functions. A survey of these properties allows comparing results in order to map which techniques bring significantly better results. Later, one can design new models to create better GAs proposals.

Acknowledgement

This paper is based upon work supported by FAPEMIG (Minas Gerais Research Foundation), Brazil.

References

- Afrati, F., Papadimitriou, C. H., and Papageorgiou, G. (1988). Scheduling DAGs to minimize time and communication. In Reif, J. H., editor, *VLSI Algorithms and Architectures*, volume 319 of *Lecture Notes in Computer Science*, pages 134–138.
- Aguilar, J. and Gelenbe, E. (1997). Task assignment and transaction clustering heuristics for distributed systems. *Information Sciences*, 97(1-2):199–219.
- Ahmad, S. G., Liew, C. S., Munir, E. U., Ang, T. F., and Khan, S. U. (2016). A hybrid genetic algorithm for optimization of scheduling workflow applications in heterogeneous computing systems. *Journal of Parallel and Distributed Computing*, 87:80–90.
- Akbari, M. (2018). An efficient genetic algorithm for task scheduling on heterogeneous computing systems based on TRIZ. *Journal of Advances in Computer Research*, 9(3):103–132.
- Akbari, M., Rashidi, H., and Alizadeh, S. H. (2017). An enhanced genetic algorithm with new operators for task scheduling in heterogeneous computing systems. *Engineering Applications of Artificial Intelligence*, 61:35–46.
- Amalarethinam, D. I. G. and Selvi, F. K. M. (2012). Grid scheduling strategy using GA (GSSGA). *International Journal of Computer Technology and Applications*, 3(4):1800–1806.
- Amirjanov, A. and Sobolev, K. (2017). Scheduling of directed acyclic graphs by a genetic algorithm with a repairing mechanism. *Concurrency and Computation: Practice and Experience*, 29(5):1–10.
- Awadall, M., Ahmad, A., and Al-Busaidi, S. (2013). Min-min GA based task scheduling in multiprocessor systems. *International Journal of Engineering and Advanced Technology*, 3(2):22–31.
- Benten, M. S. T. and Sait, S. M. (1994). Genetic scheduling of task graphs. *International Journal of electronics*, 77(4):401–415.
- Bonyadi, M. R. and Moghaddam, M. E. (2009). A bipartite genetic algorithm for multiprocessor task scheduling. *International Journal of Parallel Programming*, 37(5):462–487.

- Chitra, P., Venkatesh, P., and Rajaram, R. (2011). Comparison of evolutionary computation algorithms for solving bi-objective task scheduling problem on heterogeneous distributed computing systems. *Sadhana*, 36(2):167–180.
- Corrêa, R. C., Ferreira, A., and Rebreyend, P. (1999). Scheduling multiprocessor tasks with genetic algorithms. *IEEE Transactions on Parallel and Distributed Systems*, 10(8):825–837.
- Daoud, M. I. and Kharma, N. (2011). A hybrid heuristic-genetic algorithm for task scheduling in heterogeneous processor networks. *Journal of Parallel and Distributed Computing*, 71(11):1518–1531.
- Demiroz, B. and Topcuoglu, H. R. (2006). Static task scheduling with a unified objective on time and resource domains. *The Computer Journal*, 49(6):731–743.
- Dhingra, S., Gupta, S. B., and Biswas, R. (2014). Genetic algorithm parameters optimization for bi-criteria multiprocessor task scheduling using design of experiments. *International Journal of Computer, Control, Quantum and Information Engineering*, 8(4):661–667.
- Guzek, M., Pecero, J. E., Dorronsoro, B., and Bouvry, P. (2014). Multi-objective evolutionary algorithms for energy-aware scheduling on distributed computing systems. *Applied Soft Computing*, 24:432–446.
- Hassan, M.-A., Kacem, I., Martin, S., and Osman, I. M. (2015). Genetic algorithms for job scheduling in cloud computing. *Studies in Informatics and Control*, 24(4):387–400.
- Hou, E. S. H., Ansari, N., and Ren, H. (1994). A genetic algorithm for multiprocessor scheduling. *IEEE Transactions on parallel and distributed systems*, 5(2):113–120.
- Hwang, R., Gen, M., and Katayama, H. (2008). A comparison of multiprocessor task scheduling algorithms with communication costs. *Computers & Operations Research*, 35(3):976–993.
- Kaur, K., Chhabra, A., and Singh, G. (2010a). Heuristics based genetic algorithm for scheduling static tasks in homogeneous parallel system. *International Journal of Computer Science and Security*, 4(2):183–198.
- Kaur, K., Chhabra, A., and Singh, G. (2010b). Modified genetic algorithm for task scheduling in homogeneous parallel system using heuristics. *International Journal of Soft Computing*, 5(2):42–51.
- Kwok, Y.-K. and Ahmad, I. (1997). Efficient scheduling of arbitrary task graphs to multiprocessors using a parallel genetic algorithm. *Journal of Parallel and Distributed Computing*, 47(1):58–77.
- Levy, Y. and Ellis, T. J. (2006). A system approach to conduct an effective literature review in support of information systems research. *Informing Science Journal*, 9(1):181–212.
- Mohamed, M. R. and Awadalla, M. H. (2011). Hybrid algorithm for multiprocessor task scheduling. *International Journal of Computer Science Issues*, 8(3):79–89.
- Omara, F. and Arafa, M. (2010). Genetic algorithms for task scheduling problem. *Journal of Parallel and Distributed Computing*, 70(1):13–22.

- Pillai, A. S., Singh, K., Saravanan, V., Anpalagan, A., Woungang, I., and Barolli, L. (2018). A genetic algorithm-based method for optimizing the energy consumption and performance of multiprocessor systems. *Soft Computing*, 22(10):3271–3285.
- Puntel, F. E., Charão, A., Franciscatto, M. H., Lima, J. C. D., and da Rocha, C. C. (2018). A systematic review of scheduling algorithms and resource management in context-aware applications: A meta-analytic approach. In *ICEIS*, pages 664–670.
- Ramachandra, G. and Elmaghraby, S. E. (2006). Sequencing precedence-related jobs on two machines to minimize the weighted completion time. *International Journal of Production Economics*, 100(1):44–58.
- Sathappan, O. L., Chitra, P., Venkatesh, P., and Prabhu, M. (2011). Modified genetic algorithm for multiobjective task scheduling on heterogeneous computing system. *International Journal of Information Technology, Communications and Convergence*, 1(2):146–158.
- Sheikh, H. F., Ahmad, I., and Arshad, S. A. (2017). Performance, energy, and temperature enabled task scheduling using evolutionary techniques. *Sustainable Computing: Informatics and Systems*, 22:272–286.
- Sheikh, H. F., Ahmad, I., and Fan, D. (2016). An evolutionary technique for performance-energy-temperature optimized scheduling of parallel tasks on multi-core processors. *IEEE Transactions on Parallel and Distributed Systems*, 27(3):668–681.
- Singh, J. and Singh, G. (2012). Improved task scheduling on parallel system using genetic algorithm. *International Journal of Computer Applications*, 39(17):17–22.
- Singh, R. (2012). Task scheduling with genetic approach and task duplication technique. *International Journal of Computer Applications & Information Technology*, 1(1):7–12.
- Tsuchiya, T., Osada, T., and Kikuno, T. (1998). Genetics-based multiprocessor scheduling using task duplication. *Microprocessors and Microsystems*, 22(3-4):197–207.
- Wang, L., Siegel, H. J., Roychowdhury, V. P., and Maciejewski, A. A. (1997). Task matching and scheduling in heterogeneous computing environments using a genetic-algorithm-based approach. *Journal of parallel and distributed computing*, 47(1):8–22.
- Wang, X., Yeo, C. S., Buyya, R., and Su, J. (2011). Optimizing the makespan and reliability for workflow applications with reputation and a look-ahead genetic algorithm. *Future Generation Computer Systems*, 27(8):1124–1134.
- Wu, A. S., Yu, H., Jin, S., Lin, K.-C., and Schiavone, G. (2004). An incremental genetic algorithm approach to multiprocessor scheduling. *IEEE Transactions on parallel and distributed systems*, 15(9):824–834.
- Xu, Y., Li, K., Hu, J., and Li, K. (2014). A genetic algorithm for task scheduling on heterogeneous computing systems using multiple priority queues. *Information Sciences*, 270:255–287.
- Yao, W., You, J., and Li, B. (2004). Main sequences genetic scheduling for multiprocessor systems using task duplication. *Microprocessors and Microsystems*, 28(2):85–94.
- Zomaya, A. Y., Ward, C., and Macey, B. (1999). Genetic scheduling for parallel processor systems: comparative studies and performance issues. *IEEE Transactions on Parallel and Distributed Systems*, 10(8):795–812.