Available online at www.sciencedirect.com

ScienceDirect

Procedia Computer Science 180 (2021) 778–786

Procedia Computer Science

www.elsevier.com/locate/procedia

International Conference on Industry 4.0 and Smart Manufacturing

Parallel Metaheuristics for Shop Scheduling: enabling Industry 4.0

Pedro Coelho[a*], Cristovão Silva[a]

[a]*Univ Coimbra, CEMMPRE, Department of Mechanical Engineering, Pinhal de Marrocos, Coimbra 3030-788, Portugal*

Abstract

Production scheduling is one of the most critical activities in manufacturing. Under the context of Industry 4.0 paradigm, shop scheduling becomes even more complex. Metaheuristics present the potential to solve these harder problems but demand substantial computational power. The use of high-performance parallel architectures, present in cloud computing and edge computing, may support the develop of better metaheuristics, enabling Industry 4.0 with solution techniques to deal with their scheduling complexity. This study provides an overview of parallel metaheuristics for shop scheduling in recent literature. We reviewed 28 papers and classified them, according to parallel architectures, shop configuration, metaheuristics and optimization criteria. The results support that parallel metaheuristic have potential to tackle Industry 4.0 scheduling problems. However, it is essential to extend the research to the cloud and edge computing, flexible shop configurations, dynamic problems with multi-resource, and multi-objective optimization. Future studies should consider the use of real-world data instances.

Keywords: Industry 4.0; production scheduling; metaheuristics, parallel processing

1. Introduction

Production scheduling is one of the most critical activities of a company at the operational level for it to remain competitive in demanding consumer markets. According to Pinedo [1], scheduling is a decision-making process that deals with the allocation of resources (e.g. human, machine, money) to tasks in a specific sequence and over given periods. This process is complex and aims to optimise operational activities by using available production data, which may also include previous scheduling results (i.e. scheduling and rescheduling). In recent years, a new manufacturing

* Corresponding author. Tel.: +351 239790733
E-mail address: pedro.coelho@dem.uc.pt

10.1016/j.procs.2021.01.328

paradigm was introduced: Industry 4.0 (I4.0). This paradigm is considered a fourth industrial revolution characterised by increased flexibility, productivity, efficiency, and sustainability, ultimately ensuring competitiveness in the global market. I4.0 is promoting the emergence of smart manufacturing environment system supported by novel and emerging technologies. These technologies include cloud computing, Mobile devices, IoT platforms, Fog/Edge computing, location detection technologies, advanced human-machine interfaces, Authentication & fraud detection, 3D printing, smart sensors, big data analytics and advanced algorithms, multilevel customer interaction and customer profiling and augmented reality/wearables [2]. According to Oztemel and Gursev [2], the I4.0 components are: Cyber-physical systems (CPS), Cloud systems (cloud computing and cloud manufacturing), Machine to machine (M2M) communication, Smart factories, Augmented reality and simulation, Data mining, Internet of things, Enterprise resource planning (ERP) and business intelligence, Virtual manufacturing and Intelligent robotics.

This new paradigm changes also impacts the scheduling request that needs a holistic perspective; they should be online, real-time, and reactive [3]. Decentralisation and autonomous decision-making are pointed as strategies to tackle the elevated production flexibility and complexity requirement [4]. The impact of this decentralisation in shop scheduling implies the use of local autonomous agents to solve less complex optimisation problems. Potential techniques that may contribute to address these new scheduling requirements may be metaheuristics, machine learning, and hyper-heuristics [3]. Meantime, the implementation of those techniques required high-performance computing systems due to the complexity of the problems.

The present work was driven by a preliminary study of the authors focused on the Flexible Job Shop Scheduling Problem – one of the base models used to research the flexible shop environments under I4.0 [5]. This study concludes that metaheuristics are the most used approach used to tackle this problem but none of the new technologies related to I4.0. outstand. Cloud computing is one of the key components of this new paradigm; the use of their high-performance parallel architectures can be an opportunity to develop new parallel metaheuristics and outperform the sequential ones. These high-performance algorithms would enable I4.0 with the solution techniques needed to deal with their scheduling complexity.

Although extend literature review had been devoted to the topics of I4.0 and Scheduling, no comprehensive review has been conducted to clarify how parallel metaheuristics are used in shop scheduling. This clarification would allow to identify current trends, significant objective functions and solutions methods, and to suggest meaningful directions for further research. Luo and El Baz [6] addressed the subject but limited to parallel genetic algorithms. Therefore, in this paper, we provide a review of parallel metaheuristics applied to shop scheduling and indicate further research potential. To the best of our knowledge, this is the first attempt to provide a systematic overview of these approaches. This study may guide researchers and practitioners towards the efficient implementation of these methods. Within this context, we attempt to answer the following research question: How are parallel metaheuristics being used to solve shop scheduling problems? What have characterised the most recent studies, namely, their parallel architectures, shop configuration, metaheuristics and optimisation criteria?

The following sections present the methodology applied (Section 2). Section 3 presents the results and discussion of the information abstracted from the articles. Lastly, the final conclusions of the study are presented (Section 4).

2. Research Methodology

This study seeks to provide an overview of works related to metaheuristics implementations in parallel computing systems to solve shop scheduling problems. The main goal is to overview the extension of those approaches in the current literature, expose gaps and unveil research opportunities. For that purpose, it follows a methodology based on the one proposed by Zupic and Carter [7], which can be summarised in five-steps: i) study design, ii) data collection, iii) data analysis, iv) data visualisation, and v) interpretation.

The literature review analysis has followed a two-stages cascade approach. On the first stage, data collected from the database were selected according to their title and abstract. This rough selection was followed by a more in-depth analysis where the selected papers were read in full, and their characteristics systematised.

The Web of Science (WoS) database was used for the data collection. Being one of the primary bibliographic sources of information, WoS uses the Web of Science Core Collection database. The database has more than 21.100 peer-reviewed high-quality scholarly journals published worldwide in over 250 disciplines. The access to the database took place in June 2020, and a general search by topic was performed using the keywords: "scheduling" AND

"parallel" AND "shop". The general search by topic retrieved all the published documents with the keywords on the title, abstract, author keywords, or Keywords Plus. The WoS search resulted in 1209 documents classified as follows: 914 of the documents were classified as articles, and 335 as proceedings papers. The remaining minor categories were review (27), book chapter (20), early access (10), editorial material (2), meeting abstracts (1) and books (1). Some of the documents have more than one classification. For this study, were only considered the English language articles. The study focused on parallel algorithms implemented on parallel computing architectures. Those architectures had extensive developments in recent years. To expose that novelty only documents published between 2010 and 2020 were considered. Thus, of the 1209 documents retrieved, only 563 have been included in the analysis.

The initial set of articles had the first stage review, based on their title and abstract and became reduced to 57. This significative reduction was due to bias in the search caused by the use of the word "parallel" to describe some shop configurations - the ones that use parallel machines. After a more in-depth analysis with a full paper reading, the papers collection was reduced to 28. Most of those exclusions were works using parallel algorithms but with a sequential implementation or lack of clarity in the implementation. That collection of 28 papers was the base for abstracting and processing information.

The gathered information was organised on a table, and the main characteristics synthesised statistically and represented with graphs. The next section presents the main findings and discusses them.

3. Results and Discussion

The objective of this review is to provide an overview of the use of parallel metaheuristic approaches in shop scheduling problems. The collection of 28 articles, selected according to the protocol defined in Section 2, were organised in chronological order, and their main characteristics summarised in Table 1. For each paper, the authors, year, parallel architecture, shop configuration, metaheuristic and optimisation criterion are presented. In the following sections, details for each characteristic are shown with a brief discussion under the context of I4.0.

3.1. Parallel Architecture

In the last years, parallel computing architectures systems experienced an evolution. This parallelism is achieved by architectures based on shared or based on distributed memory. Distributed memory was the first architecture developed and consisted of the use of clusters of a single central processing unit (CPU) computers communicating through a network [8]. In this work, this system is referred to as "Distributed CPU". At the beginning of this millennium, shared memory systems have been introduced and are now widely used. Shared memory architectures consist of multiple CPU cores on the same integrated circuit having access to the same global memory. According to the number of cores, these architectures may be classified as Multi-core processors in the case of a lower number of cores processors (two, four, eight, twelve, sixteen...) or Many-core processors, in the case of a larger number of cores. Xeon Phi is considered an example of a many-core processor and may have till 72 cores [8]. Those systems are here referred to "Multi-core CPU" and "Many-Core CPU" respectively. One particular case of many-core CPU is the Graphics Processing Unit (GPU). GPU hardware has a particular architecture and memory management. For that reason, it will be separated into another category and here referred to as "GPU". Programming parallel computing systems, especially heterogeneous ones, is more difficult than sequential programming processors because it depended on the number of cores and communication technologies. In order to take advantage of parallel architectures, algorithms must be adapted and redesigned to allow task and data parallelism.

The evolution from distributed to shared memory systems can be seen in Table 1. The earliest studies made use of distributed CPU systems and had been progressively changing to multi-core and many-core systems over time. Multi-core and many-core systems provide a higher computational power with low latency communication. Meanwhile, in the last years, the work of Dabah et al. [9] uses distributed systems but combined with multi-core processors.

Regarding the use of the architectures, Figure 1(b) details the percentage of papers for each class. The leading architecture is Multi-core CPU, used in 39% of the papers; followed by Distributed CPU (25%) and GPU (15%). Less expressive is the many-core architecture used in two papers (7%). The hybridisation of distributed with multi-core is present in two papers (7%). The multi-core CPU with GPU also receives the attention of two papers (7%).

Table 1. Summary of the information abstracted from the papers collection

Author (Year)	Parallel Architecture	Shop Configuration	Metaheuristic	Optimization Criteria
Defersha and Chen (2010) [10]	Distributed CPU	Flexible job shop	GA	Min makespan
Cruz-Chávez et al. (2010) [11]	Distributed CPU	Job Shop	Hybrid SA-GA	Min makespan
Bozejko et al. (2010) [12]	GPU (CUDA)	Flexible job shop	Hybrid TS- PBM^{2}h	Min makespan
Yusof et al. (2011) [13]	Distributed CPU	Job Shop	GA	Min makespan
Defersha and Chen, (2012) [14]	Distributed CPU Multi-core CPU	Flexible job shop	GA	Min makespan
Huang et al. (2012) [15]	GPU (CUDA)	Flow Shop	GA	Min maximum earliness Min maximum tardiness
Defersha and Chen, (2012) [16]	Distributed CPU	Flexible Flow Shop	GA	Min makespan
Bozejko et al. (2013) [17]	Multi-core CPU	Flexible Flow Shop	TS	Min makespan
Juan et al. (2014) [18]	Multi-core CPU	Flow Shop	ILS	Min makespan
Türkylmaz and Bulkan (2015) [19]	Multi-core CPU	Flexible job shop	Hybrid GA-VNS	Min total tardiness
Defersha (2015) [20]	Distributed CPU	Flexible Flow Shop	SA	Min makespan
Kurdi (2015) [21]	Multi-core CPU	Job Shop	GA	Min makespan
Sun et al. (2016) [22]	Distributed CPU	Job Shop	Hybrid BFOA-PSO	Min makespan
Kurdi (2016) [23]	Multi-core CPU	Job Shop	GA	Min makespan
Bozejko et al. (2016) [24]	Multi-core CPU	Flow Shop	Hybrid TS-SA	Min makespan
Asadzadeh (2016) [25]	Distributed CPU	Job Shop	ABC	Min makespan
Sobeyko and Mönch, (2017) [26]	Multi-core CPU	Flexible job shop	VSN	Min total weighted tardiness
Alekseeva et al. (2017) [27]	Multi-core CPU	Flow Shop	GRASP	Min makespan
Bozejko et al., (2017) [28]	Many-core GPU	Flow Shop	TS	Min cycle-time
Wei et al. (2017) [29]	GPU (CUDA)	Flow Shop	TS	Min makespan
Bożejko et al. (2017) [30]	Many-core CPU	Job Shop	TS	Min cycle-time
Dao et al. (2018) [31]	Multi-core CPU	Job Shop	BA	Min makespan
Luo and El Baz (2019) [32]	Multi-core CPU GPU (CUDA)	Flexible Flow Shop	GA	Min makespan Min total tardiness
Cruz-Chávez et al. (2019) [33]	Multi-core CPU	Job Shop	SA	Min makespan
Luo et al. (2019) [34]	Multi-core CPU GPU (CUDA)	Flexible Flow Shop	GA	Min makespan Min total tardiness
Dabah et al. (2019) [9]	Distributed CPU Multi-core CPU	Job Shop	TS	Min makespan
Kawaguchi and Fukuyama (2020) [35]	Multi-core CPU	Job Shop	Hybrid TS-PSO	Min makespan Min Energy cost
Luo et al. (2020) [36]	Multi-core CPU GPU (CUDA)	Job Shop	GA	Min total tardiness Min total energy cost

ABC – Artificial Bee Colony Algorithm; BA – Bat Algorithm; BFOA – Bacterial Foraging Optimization Algorithm; GA – Genetic Algorithm; GRASP – Greedy Randomized Adaptive Search Procedure; ILS – Iterated Local Search; MA – Memetic Algorithms; PBM^{2}h – Population-Based Meta^{2}heuristics; PSO – Particle Swarm Optimization; SA – Simulating Annealing; TS – Tabu Search; VNS – Variable Neighborhood Search.

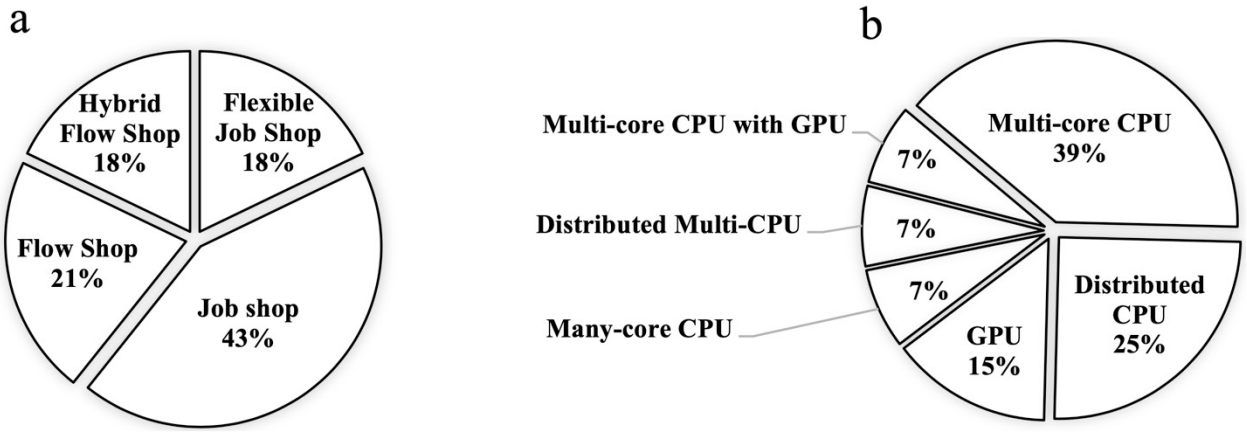

Fig. 1. (a) Percentage of papers by shop configuration; (b) Percentage of papers by parallel architecture

The hybridisation of architecture became an attractive option because shared memory systems have shown limited scalability [8], but that brings a cost. Hybridisation turns the systems more heterogeneous and finding the optimal system configuration that results in the highest performance is challenging. Most of the algorithms were implemented in low-level general-purpose programming languages such as C / C++. These low-level languages allow more granular control over memory transferences. All the works using GPU used Compute Unified Device Architecture (CUDA), an architecture developed by Nvidia for their hardware. The use of CUDA instead of other generic parallel programming architectures like OpenCL, expose the dominance of this hardware company on the GPGPU market. From low cost embedded devices to high-performance GPUs, connected with low latency, their hardware development may have a crucial role in the parallel architectures of I4.0.

Although some of the algorithms run on regular desktop configurations systems, most of them run on scientific grids or in high-end server systems. The need for such specific hardware to run the algorithms may limit their practical application. Such hardware may not be available for most of the companies. Yet, none of the works makes use of Cloud or Edge Computing, technologies that support I4.0. The use of resources from Cloud Computing can lower costs and enable access to the computational requirements. The implementation of the parallel metaheuristics on the cloud should be targeted for further research. As previously referred, decentralised scheduling is a possible mean to tackle the high flexibility of guidance and complexity of requirements that results from the introduction of I4.0 [3]. Bringing the scheduling process to the shop-floor, using a network of low-cost embedded devices may provide the needed solutions. In this case of decentralised scheduling, it is crucial to consider the use of edge computing resources. Meantime, there the need to carry out studies in order to verify the applicability of the algorithms. They must prove able to provide real-time results. Besides, it is essential to study the quality of the required solution. Spending resources on optimal solutions in such dynamic environments is a waste [1].

3.2. Shop Configuration

The shop configurations presented in the papers were classified according to Pinedo [1]. Four shop configurations were identified: Flow shop, Flexible flow shop, Job shop and Flexible job shop. In a Flow Shop, there are m machines in series, and each job has to be processed on each one of the m machines following the same route. In some of the collection paper, this environment is also referred to as a permutation flow shop. The flexible flow shop is a generalisation of the flow shop and the parallel shop configurations. There are c stages in series with at each stage several identical machines in parallel, and each job has to be processed on one of the machines per stage. This configuration is also mentioned as a hybrid flow shop in some papers. The job shop environment has m machines, and each job has its predetermined route to follow on all our just a set of those machines. The flexible job shop is a generalisation of the job shop and the parallel shop configurations. Each job has its route to follow, but, in some stages, there is more than one possible machine where it can be processed.

The Job shop is the shop configuration most used, in 12 papers, 43% of the total as it can be seen in Figure 1(a). Next, the flow shop systems in six cases or 21% of the papers. The hybrid Flow Shop has the same number of cases as the Flexible job shop; both with five cases, or 18% of the total.

Considering that scheduling environment, under I4.0, is mainly described by shop configurations like Flexible job shop or Flexible job shop problems [5], 36% of the studies may provide useful insight to the application of parallel metaheuristics under this context. Further research should be made do adapt the approaches used in the flow shop and job shop environments. Besides flexibility, I4.0 brings more dynamism. Jobs arrive after the initial scheduling [4]. And not only can a job be allocated to several machines but sometimes also request to be addressed by several resources simultaneously. Only Luo et al. [34] considers a dynamic model with uncertain new arrival jobs after the scheduling execution time; all the other papers focus on static problems. Considered as one of the new scheduling perspectives under I4.0 [4], multi-resource constrains (the use of other resources beside machines) is not considered in any of the papers. This highlight another critical gap that should be reduced in future research.

3.3. Metaheuristics

In operation research, a metaheuristic is a general solution method that provides both a general structure and strategy guidelines for developing a specific heuristic method to fit a particular kind of problem [37]. Due to the number of steps, and the amount of data involved, the implementation of metaheuristics algorithms requires computational resources. And the more complex the problem, the higher the computational need. Metaheuristics can have a sequential process implementation, but parallel computing brings the opportunity of performance-enhancing by running several tasks of the algorithm in parallel.

Figure 2(a) shows that the Genetic Algorithm (GA) is the most present metaheuristic in the collection. As a single procedure, it is present in 11 of the analysed papers, 40%. The Tabu Search (TS) is the second single procedure most used, present in five papers (18%). The hybrid metaheuristics are used in six studies (22%). Most of those hybrid algorithms have GA or TS as a base. Showing only in one case as single metaheuristics there are Greedy Randomized Adaptive Search Procedure, Iterated Local Search, Simulating Annealing, Variable Neighbourhood Search and Artificial Bee Colony Algorithm. GA is the most used metaheuristics on shop scheduling. Besides academic works, this metaheuristic is also used in commercial scheduling software's and his base procedure is used to solves some of the biggest problem size instances [1]. It is a population-based metaheuristic and due to is natural parallelism is a suitable candidate for parallel implementations [6]. GA also allow a solution representation beneficial to data parallelism. Beside task parallelism, algorithms that use data parallelism may take full advantage of the many-core architectures. TS is a single solution-based metaheuristic considered as also one of the most effective solution methods for shop scheduling problems [17], and that explains it vast use in the collection papers. The most time-consuming step of TS is the objective function value calculation; therefore, it is on this step of the algorithm that parallelisation improves the TS performance [28].

The hybridisation of these algorithms allows to overcome local optimum traps and expand solution search space while maintaining or improving the solution quality.

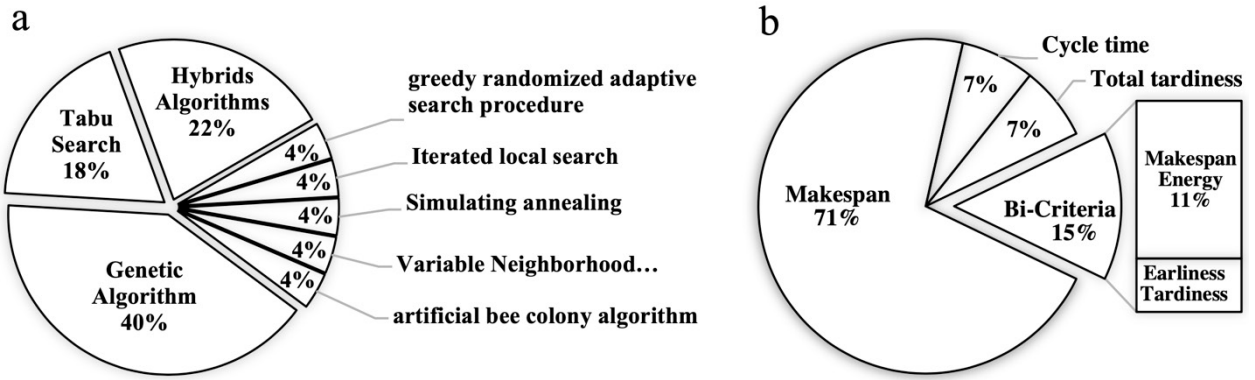

Fig. 2. (a) Percentage of papers by meta-heuristics; (b) Percentage of papers by optimization criteria

3.4. Optimization Criteria

In the decision-making process, an optimisation criterion must be defined to guide the search for the solution best suitable for the organisation. Regarding the Optimisation criteria, Figure 2(b) shows that 85% of the sampled papers consider only one criterion—most of them, 71%, makespan. The use of makespan is an essential objective because it is strongly related to resource utilisation. The other criteria represented are the cycle time and total tardiness. Cycle time also relates to the resource utilisation and tardiness is critical to ensure on-time delivery to meet customer demands. Bi-objective criteria are present only in three papers; two papers (11%) consider makespan with energy consumption and one paper (4%) consider Earliness and Tardiness. Currently, energy is a significant social and economic concerning; it may be expected that the presence of this criterion became more present in future studies.

The bi-objective optimisation presented used simples linear weighted sum approach to evaluate the solutions. Lue [36] points the immaturity of the parallel model to manage a Pareto approach. Meanwhile, Bozejko [38] proposes a parallel objective function determination method. Research that takes the algorithm parallelism to the objective function determination deserves attention; it may have a substantial impact on the reduction of the multi-objective algorithms run time. Taking a closer look at the performance evaluation of the algorithms it is noticeable the absence of case studies. All papers used literature benchmarks or generated their data. Tackling real-world data with these metaheuristics may allow fine-tuning them and improve their performance. Research to provide new sets of realistic data would promote the development of better approaches and bring models close to real-world problems.

4. Conclusion

With the ongoing introduction of I4.0, shop scheduling problems are changing and becoming even harder. Due to the last technological advances in parallel computing, parallel metaheuristics may play a vital role in tackling those complex problems. This work provides a review of the literature on parallel metaheuristics for Shop Scheduling produced in the last years. We analysed and classified 28 peer-reviewed paper according to their parallel architecture, shop configuration, metaheuristic and optimisation criteria. In-depth considerations of the parallel architecture indicate that multi-core technology is taking the lead over distributed processing. Despite being the newest technology, GPGPU is also standing out. A trend to the hybridisation of more than one architecture is apparent. Job shops are the target of most studies. However, these methods are also being implemented to solve problems in more flexible shop configurations like flexible job shops or flexible flow shops. Genetic algorithm and Tabu Search are metaheuristics which, due to their flexibility, efficiency and parallelism potential, are most frequently implemented in parallel computing systems. Hybridisations of GA and TS with other metaheuristics are also getting attention. Minimising the makespan as an optimisation criterion dominates the literature, while multi-objective have hardly been considered.

This study support that parallel metaheuristics have the potential to solve the Industry 4.0 required scheduling problems, but it is essential to extend the work already developed. Directions to further research include studying: the use of cloud computing and edge computing, shop configurations more flexible, problems more dynamic and with multi-resource constrains; redesign algorithms to apply with task and data; parallel multi-objective optimisation criteria that besides makespan addresses tardiness and energy consumptions issues. Besides literature benchmarks, those studies should consider the use of real-world data instances.

This work did not intend to be an exhaustive review of the literature but certainly has limitations. The final papers collection was generated by searching only one scholarly database. Although we tried to be as comprehensive as possible, the selected keywords may not intercept all the relevant publications. The exclusion of conference proceedings may also have left out some significant contributions. Also, the 10 years limitation may have excluded conceptual models that, with the actual processing power, may turn out good proposals. We recommend an extended version of this work focused on more in-depth content analysis and description of the metaheuristics, covering a broader literature period. Relevant conference proceeding should be selected based on a literature snowballing procedure.

Acknowledgements

This research is sponsored by FEDER funds through the program COMPETE – Programa Operacional Factores de Competitividade – and by national funds through FCT – Fundação para a Ciência e a Tecnologia –, under the project UID/EMS/00285/2020 and the doctoral grant to P.C. (SFRH/BD/129714/2017).

References

[1] Pinedo ,Michael L (2016) "Scheduling Theory, Algorithms, and Systems". Springer International Publishing

[2] Oztemel ,Ercan and Gursev ,Samet (2020) "Literature review of Industry 4.0 and related technologies". *Journal of Intelligent Manufacturing* **31**: 127–182

[3] Parente ,Manuel, Figueira ,Gonçalo, Amorim ,Pedro, and Marques ,Alexandra (2020) "Production scheduling in the context of Industry 4.0: review and trends". *International Journal of Production Research* **7543**:

[4] Zhang ,Jian, Ding ,Guofu, Zou ,Yisheng, et al (2019) "Review of job shop scheduling research and its new perspectives under Industry 4.0". *Journal of Intelligent Manufacturing* **30**: 1809–1830

[5] Dolgui ,Alexandre, Ivanov ,Dmitry, Sethi ,Suresh P, and Sokolov ,Boris (2019) "Scheduling in production, supply chain and Industry 4.0 systems by optimal control: fundamentals, state-of-the-art and applications". *International Journal of Production Research* **57**: 411–432

[6] Luo ,Jia and El Baz ,Didier (2018) "A survey on parallel genetic algorithms for shop scheduling problems". *Proceedings - 2018 IEEE 32nd International Parallel and Distributed Processing Symposium Workshops, IPDPSW 2018* 629–636

[7] Zupic ,Ivan and Čater ,Tomaž (2015) "Bibliometric Methods in Management and Organization". *Organizational Research Methods* **18**: 429–472

[8] Barney ,Blaise (2010) "Introduction to parallel computing". *Lawrence Livermore National Laboratory* **6**: 10

[9] Dabah ,Adel, Bendjoudi ,Ahcene, AitZai ,Abdelhakim, and Taboudjemat ,Nadia Nouali (2019) "Efficient parallel tabu search for the blocking job shop scheduling problem". *Soft Computing* **23**: 13283–13295

[10] Defersha ,Fantahun M and Chen ,Mingyuan (2010) "A parallel genetic algorithm for a flexible job-shop scheduling problem with sequence dependent setups". *International Journal of Advanced Manufacturing Technology* **49**: 263–279

[11] Cruz-Chávez ,Marco Antonio, Rodríguez-León ,Abelardo, Ávila-Melgar ,Erika Yesenia, et al (2010) "Gridification of Genetic Algorithm with Reduced Communication for the Job Shop Scheduling Problem". *International Journal of Grid and Distributed Computing* **3**: 13–28

[12] Bozejko ,Wojciech, Uchroński ,Mariusz, and Wodecki ,Mieczysław (2010) "Parallel hybrid metaheuristics for the flexible job shop problem". *Computers and Industrial Engineering* **59**: 323–333

[13] Yusof ,Rubiyah, Khalid ,Marzuki, Hui ,Gan Teck, et al (2011) "Solving job shop scheduling problem using a hybrid parallel micro genetic algorithm". *Applied Soft Computing Journal* **11**: 5782–5792

[14] Defersha ,Fantahun M and Chen ,Mingyuan (2012) "Jobshop lot streaming with routing flexibility, sequence-dependent setups, machine release dates and lag time". *International Journal of Production Research* **50**: 2331–2352

[15] Huang ,Chieh Sen, Huang ,Yi Chen, and Lai ,Peng Jen (2012) "Modified genetic algorithms for solving fuzzy flow shop scheduling problems and their implementation with CUDA". *Expert Systems with Applications* **39**: 4999–5005

[16] Defersha ,Fantahun Melaku and Chen ,Mingyuan (2012) "Mathematical model and parallel genetic algorithm for hybrid flexible flowshop lot streaming problem". *International Journal of Advanced Manufacturing Technology* **62**: 249–265

[17] Bozejko ,Wojciech, Pempera ,Jarosław, and Smutnicki ,Czesław (2013) "Parallel tabu search algorithm for the hybrid flow shop problem". *Computers and Industrial Engineering* **65**: 466–474

[18] Juan ,Angel A, Lourenço ,Helena R, Mateo ,Manuel, et al (2014) "Using iterated local search for solving the flow-shop problem: Parallelization, parametrization, and randomization issues". *International Transactions in Operational Research* **21**: 103–126

[19] Türkylmaz ,Alper and Bulkan ,Serol (2015) "A hybrid algorithm for total tardiness minimisation in flexible job shop: Genetic algorithm with parallel VNS execution". *International Journal of Production Research* **53**: 1832–1848

[20] Defersha ,Fantahun M (2015) "A simulated annealing with multiple-search paths and parallel computation for a comprehensive flowshop scheduling problem". *International Transactions in Operational Research* **22**: 669–691

[21] Kurdi ,Mohamed (2015) "A new hybrid island model genetic algorithm for job shop scheduling problem". *Computers and Industrial Engineering* **88**: 273–283

[22] Sun ,Liang, Ge ,Hongwei, and Wang ,Limin (2016) "A coevolutionary bacterial foraging model using pso in job-shop scheduling environments". *International Journal of Grid and Distributed Computing* **9**: 379–394

[23] Kurdi ,Mohamed (2016) "An effective new island model genetic algorithm for job shop scheduling problem". *Computers and Operations Research* **67**: 132–142

[24] Bozejko ,Wojciech, Uchroński ,Mariusz, and Wodecki ,Mieczysław (2016) "Parallel metaheuristics for the cyclic flow shop scheduling problem". *Computers and Industrial Engineering* **95**: 156–163

[25] Asadzadeh ,Leila (2016) "A parallel artificial bee colony algorithm for the job shop scheduling problem with a dynamic migration strategy". *Computers and Industrial Engineering* **102**: 359–367

[26] Sobeyko ,Oleh and Mönch ,Lars (2017) "Integrated process planning and scheduling for large-scale flexible job shops using metaheuristics". *International Journal of Production Research* **55**: 392–409

[27] Alekseeva ,Ekaterina, Mezmaz ,Mohand, Tuyttens ,Daniel, and Melab ,Nouredine (2017) "Parallel multi-core hyper-heuristic GRASP to solve permutation flow-shop problem". *Concurrency Computation* **29**: 1–15

[28] Bozejko ,Wojciech, Chaczko ,Zenon, Uchroński ,Mariusz, and Wodecki ,Mieczysław (2017) "Parallel patterns determination in solving cyclic flow shop problem with setups". *Archives of Control Sciences* **27**: 183–195

[29] Wei ,Kai Cheng, Sun ,Xue, Chu ,Hsun, and Wu ,Chao Chin (2017) "Reconstructing permutation table to improve the Tabu Search for the PFSP on GPU". *Journal of Supercomputing* **73**: 4711–4738

[30] Bożejko ,Wojciech, Gnatowski ,Andrzej, Pempera ,Jarosław, and Wodecki ,Mieczysław (2017) "Parallel tabu search for the cyclic job shop scheduling problem". *Computers and Industrial Engineering* **113**: 512–524

[31] Dao ,Thi Kien, Pan ,Tien Szu, Nguyen ,Trong The, and Pan ,Jeng Shyang (2018) "Parallel bat algorithm for optimizing makespan in job shop scheduling problems". *Journal of Intelligent Manufacturing* **29**: 451–462

[32] Luo ,Jia and El Baz ,Didier (2019) "A dual heterogeneous island genetic algorithm for solving large size flexible flow shop scheduling problems on hybrid multicore CPU and GPU platforms". *Mathematical Problems in Engineering* **2019**:

[33] Cruz-Chávez ,Marco Antonio, Peralta-Abarca ,Jesús del C, and Cruz-Rosales ,Martín H (2019) "Cooperative threads with effective-address in simulated annealing algorithm to job shop scheduling problems". *Applied Sciences (Switzerland)* **9**:

[34] Luo ,Jia, Fujimura ,Shigeru, El Baz ,Didier, and Plazolles ,Bastien (2019) "GPU based parallel genetic algorithm for solving an energy efficient dynamic flexible flow shop scheduling problem". *Journal of Parallel and Distributed Computing* **133**: 244–257

[35] Kawaguchi ,Shuhei and Fukuyama ,Yoshikazu (2020) "Improved parallel reactive hybrid particle swarm optimization using improved neighborhood schedule generation method for the integrated framework of optimal production scheduling and operational planning of an energy plant in a factory". *Electronics and Communications in Japan* **103**: 37–48

[36] Luo ,Jia, El Baz ,Didier, Xue ,Rui, and Hu ,Jinglu (2020) "Solving the dynamic energy aware job shop scheduling problem with the heterogeneous parallel genetic algorithm". *Future Generation Computer Systems* **108**: 119–134

[37] Hillier ,Frederick S and Lieberman ,Gerald J (2015) "Introduction to operations research". McGraw-Hill Science, Engineering & Mathematics

[38] Bozejko ,Wojciech (2012) "On single-walk parallelization of the job shop problem solving algorithms". *Computers and Operations Research* **39**: 2258–2264