



**HAL**  
open science

# Hyper-heuristics applications to manufacturing scheduling: overview and opportunities

Wassim Bouazza

► **To cite this version:**

Wassim Bouazza. Hyper-heuristics applications to manufacturing scheduling: overview and opportunities. IFAC World Congress, IFAC, Jul 2023, Yokoama, Japan. pp.935 - 940, 10.1016/j.ifacol.2023.10.1685 . hal-04329421

**HAL Id: hal-04329421**

**<https://hal.science/hal-04329421>**

Submitted on 7 Dec 2023

**HAL** is a multi-disciplinary open access archive for the deposit and dissemination of scientific research documents, whether they are published or not. The documents may come from teaching and research institutions in France or abroad, or from public or private research centers.

L'archive ouverte pluridisciplinaire **HAL**, est destinée au dépôt et à la diffusion de documents scientifiques de niveau recherche, publiés ou non, émanant des établissements d'enseignement et de recherche français ou étrangers, des laboratoires publics ou privés.

# Hyper-heuristics applications to manufacturing scheduling: overview and opportunities

BOUAZZA Wassim\*

\* Nantes Université, École Centrale Nantes, CNRS, LS2N, UMR 6004, F-44000 Nantes, France (e-mail: [Wassim.Bouazza@ls2n.fr](mailto:Wassim.Bouazza@ls2n.fr))

**Abstract:** modern industrial challenges require solutions that are both efficient and reactive. Heuristic-based approaches allow production systems to react quickly to unexpected events and disturbances. Thus, the term Hyper-Heuristic (HH) covers a wide variety of techniques that allow the selection or generation of heuristics. To conciliate the speed of heuristics with the necessary global performance, a number of mechanisms have been proposed in the literature. This paper first presents the HHs. Afterward, inspired from previous works, a classification is given to categorize them. In addition, an enhancement is proposed. Indeed, a third type of HHs is added (i.e. to classic selection HHs and generation HHs). This new category is named “mixed” HHs. Then, different contributions from the literature to dynamic scheduling are highlighted. Before concluding, a number of trends and future directions such as the use of Machine Learning and simulation are explored.

Copyright © 2023 The Authors. This is an open access article under the CC BY-NC-ND license (<https://creativecommons.org/licenses/by-nc-nd/4.0/>)

**Keywords:** Manufacturing; Scheduling; Hyper-Heuristics; Optimization; Reactive Control; Simulation;

## 1. INTRODUCTION

Modern manufacturing environments are as complex as they are dynamic. The dynamic control of manufacturing systems is a hard issue that mobilizes both researchers and industrialists. Emerging paradigms such as Industry 4.0 allow generating a large amount of data with the aim of improving performance and agility (Gilchrist, 2016; Xu et al., 2021). The ever-increasing automation tries to create a control architecture able to adapt quickly to changes (e.g. unexpected order arrivals, machine breakdowns, etc.). To ensure efficiently the crucial task of manufacturing scheduling, a number of methods are based on heuristics. This type of algorithm proposes to define the fast behavior of the system, to made it able to react quickly to disturbances. Although they are highly reactive, their global performance is often poor. Therefore, the neologism Hyper-Heuristic (HH) gathers all the methods that allow to generate or select heuristics. Many contributions have been published lately, which are not limited to industrial scheduling.

This article focuses on the HH application to manufacturing scheduling. After defining HHs, the opportunities they offer are described. A classification is given to categorize HHs with an extension proposed adding a third type referred as “mixed” HHs. The latter combines heuristic generation and selection. A number of trends and future directions, such as the Machine Learning (ML) application to HHs are discussed.

## 2. WHAT IS A HYPER-HEURISTIC?

“Hyper-heuristic” is a neologism, firstly used in the very early 00s. If the term is relatively new, the idea it formalizes is not. Indeed, we can say that the term HH designates a particular type of algorithm to solve computational problems, not by operating directly on the solutions search space, but on the

rules, which allow generating these end solutions. In other words, the main characteristic of these methods is that they explore a search space of heuristics, rather than a search space of solutions of the problem (Ochoa & Özcan, 2010). Thus, one of the main objectives of the HHs is to raise the level of generality at which optimization systems can perform (E. Burke et al., 2003).

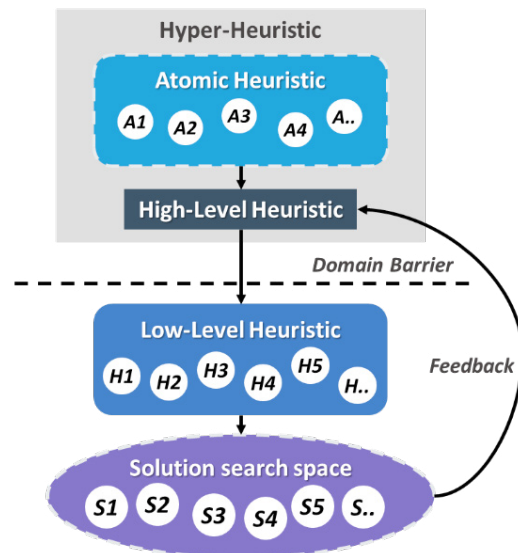


Fig. 1 Main components of a hyper-heuristic in action

As illustrated in Fig. 1 inspired from (E. K. Burke et al., 2019; Swiercz, 2017), there are two main components of HHs:

- The High-Level Heuristics (HLHs) which allow selecting or combining “atomic” heuristics creating new Low-Level Heuristics or to generate new ones “from scratch”.
- The Low-Level Heuristics (LLHs) which allow to “generate” the solutions.

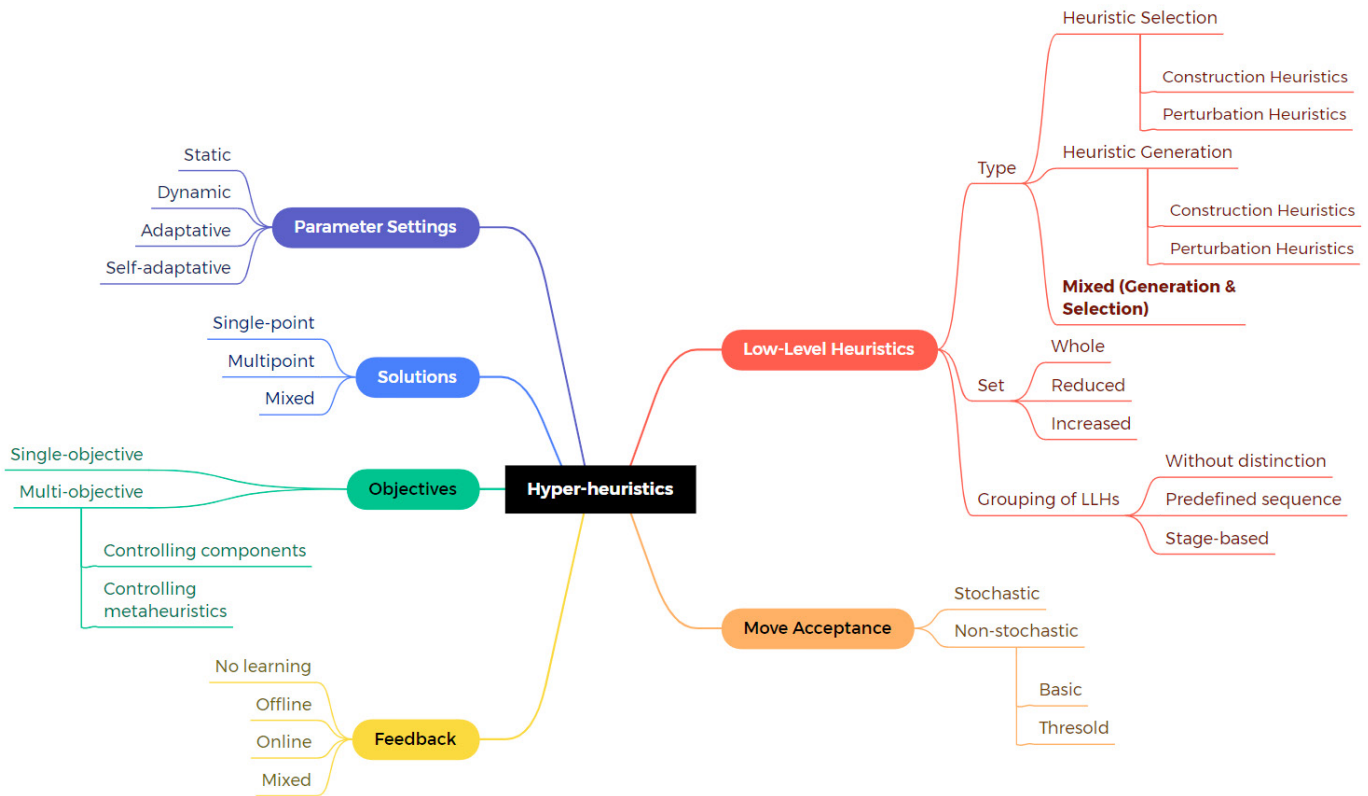


Fig. 2 An extended classification of hyper-heuristics

Thus, possibly using a set of atomic rules, the high-level heuristic (e.g. a metaheuristic like a Genetic Algorithm), combines them generating a set of low-level heuristics specific to the problem domain. It is the execution of these heuristics that generates the final solution to the optimization problem.

As stated above, HHs refer to a wide variety of methods and so many problems dealt with. To better understand this type of algorithm, a number of classifications have been proposed in the literature and further refined over time. Fig. 2 shows the classification inspired from (E. K. Burke et al., 2010, 2019; Drake et al., 2020). Six key criteria are used to categorizes HHs:

- *Low-Level Heuristics (LLHs)*: depends on the set (i.e. whole, reduced or increased), the grouping of LLHs and also their type. Originally, there were only two types i.e. selection or generation. Here we added a third type called “Mixed” HHs. This is discussed further in Section 3.
- *Move Acceptance*: if a probabilistic process is considered when making an acceptance or rejection decision, it then can be classified as stochastic (e.g. using a simulated annealing algorithm), and non-stochastic, otherwise.
- *Parameters Settings*: the search parameters of the HHs can be fixed statically, evolve dynamically (i.e. change value in a predefined way), evolve in an adaptive way (i.e. the value changes reactively during the search process), or with a self-adaptive mechanism (i.e. by searching

simultaneously for the best solution and the best parameterization)

- *Solutions*: three types of population can be distinguished. First, HHs based on “multipoint” populations (i.e. HHs use several solutions when performing a search), HHs based on “single-point” populations i.e. when only the current active solution is processed, and so-called “mixed” populations i.e. that combine both single and multipoint searches, usually in a progressive manner.
- *Objectives*: it is derived from the standard classification of metaheuristics. The latter depends on the type of objective function pursued by the algorithm (i.e. single of multiple).
- *Feedback*: during the search for heuristics, several cases can be distinguished. Either no feedback is used. In this case, the HLHs does not consider the performance of the LLHs. Either, the performance of the LLHs is considered during an off-line phase, prior to the system launch. Either the learning loop is used exclusively during the execution of the system. Or finally, the feedback on the performance of the LLHs, and more generally, on the system state is used in a combination of off-line and on-line learning.

With 6 main criteria and 28 sub-criteria, this classification cannot be detailed too deeply here. However, interested readers can refer to the following publications (E. K. Burke et al., 2010, 2019; Drake et al., 2020).

Moreover, there are also many other applications such as timetable generation, e.g. (Pillay, 2012), machine failure detection e.g. (Navajas-Guerrero et al., 2022) or vehicle routing problems (Tarhini et al., 2022). These are beyond the scope of this paper. In the following section, we will focus on the application of HHs to manufacturing scheduling.

### 3. APPLICATIONS TO MANUFACTURING SCHEDULING

As shown in the previous section, there are many aspects that allow to distinguish HHs. In this section, HHs are divided into three categories that seem to us the most primordial, namely the HHs for generation, for selection and the mixed ones.

#### 1. HHs for Rules selection

A certain number of contributions seek to select the best rule among a set of candidate ones. As mentioned before, the relevance of rule-based methods is particularly motivated by dynamic scheduling problems where the need for reactivity is important. An example of Real-Time Scheduling (RTS) can be found in (Shiue et al., 2018). The authors assume that it is better to use a set of Dispatching Rules (DRs), each related to a specific "Zone" of the production system, than to use a unique DR. They propose an RTS based on reinforcement learning (RL) using two main mechanisms: a first module is in charge of offline learning, while a second RL module operates using a Q-Learning algorithm. Another example of Q-Learning application to DRs selection can be found in (Bouazza et al., 2017). The authors developed a distributed approach based on Smart Products (SPs) with decisional abilities. To solve a partially flexible job-shop scheduling problem, the SPs use a Q-learning effect to select "ad-hoc" both Machine Selection Rules and Sequencing Rules.

There are several other approaches to implement automatic rule selection. For example, H. Zhang & Roy (2019) propose to select the DRs according to the production objectives. For this purpose, each of the DRs and each of the production objectives are linked to a set of scheduling parameters such as processing time, remaining work, delivery date, delay, etc. Therefore, the developed method is based on the semantic expressions of the selected production objectives and the DRs. The system can be able to select the most pertinent rule, according to the set objectives. Another example of a multi-objective optimization problem is proposed in (Teymourifar et al., 2018). Indeed, to solve a Job Shop Scheduling Problem, the authors propose to dynamically select the DRs according to the state of the scheduling environment. A comparative study with classic rules from the literature has shown the interest of dynamic rules selection.

Simulation is a very useful technique for determining the relevance of the rules to be selected. For example, the discrete event simulation approach integrated with the Flexible and Interactive Tradeoff Compensatory approach is used in (Pergher et al., 2020). The goal is to identify the best combination of scheduling DRs, due date assignment and

release date. One last example of simulation application for dynamic rule selection can be cited. The works presented in (Bouazza et al., 2021, 2022), use simulation-based learning mechanisms. The authors propose the concept of Decision Strategy (DS). This structure is similar to a decision tree whose leaves are DRs (for resource selection and job sequencing). A learning loop using simulation is used to improve the DSs gradually in order to obtain the most efficient combination of rules. The rules are then dynamically selected during the "online" phase. Experimental results support the interest of the developed approaches.

#### 2. HHs for Rules generation

While approaches to select DRs from a finite list can be effective in some cases, it may be interesting to develop more specific rules for other types of problems. However, the man-made rules are highly relevant to the problem, and the results are dependent on the problem itself (Zhou et al., 2019). Thus, among the most commonly used approaches to generate specific DRs is Genetic Programming (GP). GP is a classical metaheuristic inspired by the Genetic Algorithm (GA). Unlike a fixed length gene chain in GA, the individuals in GP are represented by tree structures that lead to a dynamic arrangement of the data. Given this feature, GP is used as an HLH to build a HH scheme capable of dynamically handling multiple LLHs (Zhu et al., 2023).

F. Zhang et al., (2019), address a Dynamic Flexible Job-Shop Scheduling problem (DFJSSP). The solution is built on a Genetic Programming Hyper-Heuristic (GPHH). The proposed method relies on the selection of relevant job features and the exclusion of irrelevant ones. The goal is to reduce the search space of the GPHH. Once this selection of features is done, a second phase uses them to generate a set of rules according to the scenarios. Simulations show that the feature selection phase allows an efficient evolution of both routing rules and sequencing rules for the DFJSSP.

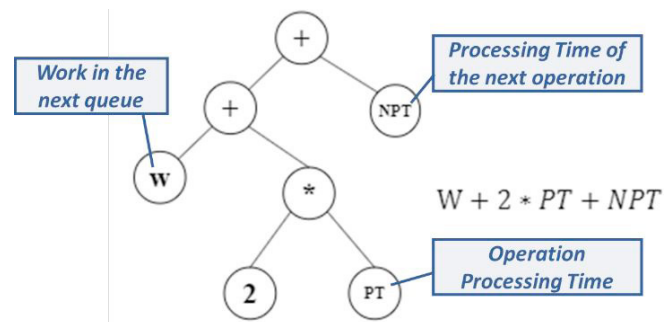


Fig. 3 Example of Genetic Programming structure

The use of ML techniques can help to generate efficient rules. In (Ferreira et al., 2021), an approach applied to the delay-minimizing dynamic job shop scheduling problem is presented (Fig. 3). The approach consists of a guided empirical learning procedure, where reasoning about the problem guides the algorithmic search for GPs in an iterative loop. The results suggest that the method is able to find new efficient rules,



which outperform other GP implementations from the literature in the majority of instances (i.e. by 19% on average).

Many other works are based on simulation for the generation of rules. Freitag & Hildebrandt (2016) propose an application of HH to simulation-based multi-objective optimization (multi-objective GP). The algorithm allows to automatically develop improved DRs specifically for a specific control problem. An application to a complex semiconductor manufacturing scenario is described. The results show that the DRs generated in this way perform significantly better than a number of DRs from the literature. There are other works related to multi-objective optimization of manufacturing scheduling. Zhou et al. (2019) deal with a Dynamic Flexible Job Shop scheduling problem. Three different types of methods are proposed for coevolving both machine assignment rules and task sequencing rules, including multi-objective Coevolution Cooperative Genetic Programming (CCGP) with two subpopulations, multi-objective GP with two subtrees, and another multi-objective GP with two chromosomes. The training results as well as the tests show that the proposed method, named CCGP-NSGAI, is more competitive than other evolutionary approaches.

If most of the DR generation approaches are applied to dynamic contexts, other works focus on their performance on static problems. Đurasević & Jakobović (2021) address the automatic development of DRs specifically adapted to static and off-line conditions. The objective of the paper is to analyze several methods to adapt automatically generated DRs to static manufacturing scheduling conditions. The proposed method is based on new end nodes and the application of a deployment algorithm to adapt DRs. Comparisons with metaheuristics allow positioning them as a viable alternative. DRs can then be used in situations where metaheuristics could not, while offering both, better execution time, and competitive results.

### 3. Mixed HHs (Rules generation and selection)

As previously stated, a third category of HHs is distinguished here. This one does not "only" select rules from a predefined list of DRs (i.e. statically or dynamically), nor does it generate a unique DR. Indeed, all approaches that both generate and select DRs are categorized here under the term "mixed HHs". To illustrate these approaches, an example can be taken from (Baek et al., 1998). In contrast to methods using a single DR, the authors propose a procedure that sequentially generates a DR for each workstation, one by one. The goal is to find the most suitable rule for each workstation, at various decision points in time. The method, named Spatial Adaptation Procedure, is applied to flexible system of semiconductor wafer fabrication. The combination of rules generated allows to reduce the average flow time, significantly compared to reference rules. A few years later, the work proposed by Baek & Yoon (2002), no longer generates DRs sequentially, but makes them evolve simultaneously. Thus, the complexity resulting from the interconnection and the interdependency of the machines is explicitly considered. The chosen approach is a coevolution algorithm (CoEA). Simulations show that the

HH CoEA is more effective and efficient than the sequential procedure suggested earlier. In (Geiger et al., 2006), a novel approach to learning DRs is developed (Fig. 4) for a variety of single-machine environments. The approach has given good results on reducing the Makespan in a two machine-flow shop scheduling problem, comparing to GP algorithm and Evolutionary Algorithm (EA).

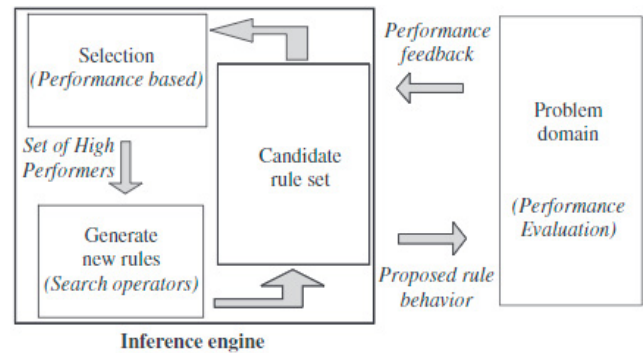


Fig. 4 Scheduling rule learning system (Geiger et al., 2006)

To ensure a RTS problem on a job-shop, Miyashita (2000) combines GP with a Multi-Agent System (MAS). In the control architecture, each agent is in charge of the sequencing of a specific machine with a dedicated DR. Miyashita compares three different HHs, all using GP: the first one develops a single DR for all work-centers, the second one generates a different DR for each work-center, the third one uses a predetermined classification of work-centers into two categories i.e. "bottlenecks" and "non-bottlenecks". Therefore, a specific DR for each of the two categories is developed. The approach is tested on a job-shop consisting of five work-centers with one or two bottlenecks. The HH using classification produces the best results. Another example, building the allocation of rules around a pre-classification of work-centers, is proposed by Jakobović & Budin (2006). The GP HH is built using three trees: two of the trees correspond to a composite dispatch rule, and the third tree is a decision tree that uses attributes related to the workload of a work center to decide which of the two composite rules should be applied. One last example of GP application is proposed by Pickardt et al. (2013). The developed algorithm is a two-stage HH: a first phase allows generating a set of DRs using the GP. Then, a second phase allows assigning them to each work-center composing the production system using an EA. The generated rules are then compared with benchmark rules. They are thus more efficient to reduce the mean weighted tardiness.

This brief overview shows the richness and potential of HHs for manufacturing scheduling, especially, in dynamic environments. In the following section, trends and possible upcoming challenges are highlighted.

## 4. TRENDS AND FUTURE CHALLENGES

Through the state of the art proposed in the previous section, a trend in the evolution of HHs appears to be emerging. Initially limited mostly to the generation of a single rule, HHs

generating and/or combining several rules have appeared in the literature. Target production systems have thus evolved from single machine systems to increasingly more flexible and complex systems. Also, a growing number of works explicitly consider both types of rules for each of the two sub-problems of manufacturing scheduling, namely the allocation and sequencing of production jobs.

More generally, in many fields, the contribution of Artificial Intelligence technics promises major evolution with increased performance. Hyper-heuristics are no exception. As shown in Fig. 4, the nature of the learning feedback can be a determining element for the quality of the solution to an optimization problem. As discussed above, a number of methods use feedback loops to significantly improve the quality of the generated rules. Others take advantage of this feedback loop to ensure dynamic and efficient rule selection. It seems that this trend will accelerate even more with the integration of more complex techniques such as deep-learning.

As discussed, we have proposed to add a third category in the classification that concerns HHs that allow both rule generation and rule selection. Compared to the two other categories, i.e. rule selection only or rule generation only, these approaches are relatively recent and still few. Moreover, once the DRs have been generated, many are content with allocating the rules in a "definitive" and static way, to work-centers for example. It seems that the decoupling of rule generation and rule selection will allow producing more innovative solutions. Thus, it is possible to take advantage of both very specific rule generation and dynamic rule selection, allowing to change the rule "on the fly" considering the current context.

Another aspect to consider is the integration of HHs into Cyber-Physical Production Systems (CPPS). Their reactivity and their low need in terms of computing resources, makes them particularly interesting, especially for distributed architectures for the control of CPPS. Considering the multiple potential decision nodes in such systems, it seems worthwhile to develop methods that can generate as many rules to apply, depending on the system's state. The behavior is then obtained by emergence. The use of techniques exploiting in particular the learning loop, coupled for example to simulation techniques, allows enhancing the global performances.

## 5. CONCLUSION

The term hyper-heuristics covers a wide variety of algorithms and methods. The concept it refers to is much older than this neologism. HHs allow generating solutions to complex optimization problems, and this, in a quite acceptable time. To counterbalance the poor global performance of the heuristics, many contributions have developed mechanisms that provide a feedback loop to improve the heuristics iteratively. Therefore, it is possible to design reactive systems offering, at the same time, good global performances.

In this paper, the HHs are firstly defined. Then, an extension to the HHs classification in the literature is proposed. This one

concerns an additional category of Low-Level Heuristics type. This new category groups the HHs that allow both the generation and the selection of LLHs. Although still relatively few, this type of HHs seems to hold a lot of potential by taking advantage of both the generation of specific rules, and their selection dynamically, according to the current context of the system. Finally, an overview of the current trends allowed to highlight the benefits of ML techniques, as well as the coupling with computational simulation. Thus, HHs are promising techniques that can combine the reactivity of heuristics with a satisfying global performance obtained by emergence from complex systems.

Works are already ongoing for the development of HHs for dynamic scheduling of flexible manufacturing systems. These are Mixed HHs for both generating allocation and sequencing rules, and dynamically selecting them. Moreover, heuristic selection using deep neural networks is also experimented.

## 6. REFERENCES

- Baek, D. H., & Yoon, W. C. (2002). Co-evolutionary genetic algorithm for multi-machine scheduling: Coping with high performance variability. *International Journal of Production Research*, 40(1), 239–254. <https://doi.org/10.1080/00207540110079419>
- Baek, D. H., Yoon, W. C., & Park, S. C. (1998). A spatial rule adaptation procedure for reliable production control in a wafer fabrication system. *International Journal of Production Research*, 36(6), 1475–1491. <https://doi.org/10.1080/002075498193129>
- Bouazza, Wassim., Sallez, Y., & Beldjilali, B. (2017). A distributed approach solving partially flexible job-shop scheduling problem with a Q-learning effect. *IFAC-PapersOnLine*, 50(1), 15890–15895. <https://doi.org/10.1016/j.ifacol.2017.08.2354>
- Bouazza, W., Sallez, Y., & Trentesaux, D. (2021). Dynamic scheduling of manufacturing systems: a product-driven approach using hyper-heuristics. *International Journal of Computer Integrated Manufacturing*, 34(6), 641–665. <https://doi.org/10.1080/0951192X.2021.1925969>
- Bouazza, W., Sallez, Y., & Trentesaux, D. (2022). Toward Efficient FMS Scheduling Through Rules Combination Using an Optimization-Simulation Mechanism. In *Studies in Computational Intelligence* (Vol. 1034, pp. 559–571). Springer Science and Business Media Deutschland GmbH. [https://doi.org/10.1007/978-3-030-99108-1\\_40](https://doi.org/10.1007/978-3-030-99108-1_40)
- Burke, E., Kendall, G., Newall, J., Hart, E., Ross, P., & Schulenburg, S. (2003). Hyper-Heuristics: An Emerging Direction in Modern Search Technology. In *Handbook of Metaheuristics* (pp. 457–474). Kluwer Academic Publishers. [https://doi.org/10.1007/0-306-48056-5\\_16](https://doi.org/10.1007/0-306-48056-5_16)
- Burke, E. K., Hyde, M., Kendall, G., Ochoa, G., Özcan, E., & Woodward, J. R. (2010). A Classification of Hyper-heuristic Approaches. In M. Gendreau & J.-Y. Potvin (Eds.), *Handbook of metaheuristics* (pp. 449–468).

- Springer US. [https://doi.org/10.1007/978-1-4419-1665-5\\_15](https://doi.org/10.1007/978-1-4419-1665-5_15)
- Burke, E. K., Hyde, M. R., Kendall, G., Ochoa, G., Özcan, E., & Woodward, J. R. (2019). A classification of hyper-heuristic approaches: Revisited. In M. G. J.-Y. Potvin (Ed.), *International Series in Operations Research and Management Science* (2nd ed, Vol. 272, pp. 453–477). Springer. [https://doi.org/10.1007/978-3-319-91086-4\\_14](https://doi.org/10.1007/978-3-319-91086-4_14)
- Drake, J. H., Kheiri, A., Özcan, E., & Burke, E. K. (2020). Recent advances in selection hyper-heuristics. *European Journal of Operational Research*, 285(2), 405–428. <https://doi.org/10.1016/j.ejor.2019.07.073>
- Đurasević, M., & Jakobović, D. (2020). Automatic design of dispatching rules for static scheduling conditions. *Neural Computing and Applications*. <https://doi.org/10.1007/s00521-020-05292-w>
- Ferreira, C., Figueira, G., & Amorim, P. (2021). *Effective and interpretable dispatching rules for dynamic job shops via guided empirical learning*. 1–30. <https://doi.org/10.1016/j.omega.2022.102643>
- Freitag, M., & Hildebrandt, T. (2016). Automatic design of scheduling rules for complex manufacturing systems by multi-objective simulation-based optimization. *CIRP Annals - Manufacturing Technology*, 65(1), 433–436. <https://doi.org/10.1016/j.cirp.2016.04.066>
- Geiger, C. D., Uzsoy, R., & Aytug, H. (2006). Rapid Modeling and Discovery of Priority Dispatching Rules: An Autonomous Learning Approach. *Journal of Scheduling*, 9(1), 7–34. <https://doi.org/10.1007/s10951-006-5591-8>
- Gilchrist, A. (2016). *Industry 4.0*. Apress. <https://doi.org/10.1007/978-1-4842-2047-4>
- Jakobović, D., & Budin, L. (2006). Dynamic Scheduling with Genetic Programming. In P. Collet, M. Tomassini, M. Ebner, S. Gustafson, & A. Ekárt (Eds.), *Genetic Programming* (pp. 73–84). Springer. [https://doi.org/10.1007/11729976\\_7](https://doi.org/10.1007/11729976_7)
- Miyashita, K. (2000). Job-shop scheduling with genetic programming. *Proc. of the Genetic and Evolutionary Computation Conference (GECCO-2000)*, 505–512. <https://doi.org/10.1145/3321707.3321790>
- Navajas-Guerrero, A., Manjarres, D., Portillo, E., & Landa-Torres, I. (2022). A hyper-heuristic inspired approach for automatic failure prediction in the context of industry 4.0. *Computers & Industrial Engineering*, 171, 108381. <https://doi.org/10.1016/j.cie.2022.108381>
- Ochoa, G., & Özcan, E. (2010). Special issue on hyper-heuristics in search and optimization. *Journal of Heuristics*, 16(6), 745–748. <https://doi.org/10.1007/s10732-010-9147-x>
- Pergher, I., Frej, E. A., Roselli, L. R. P., & de Almeida, A. T. (2020). Integrating simulation and FITradeoff method for scheduling rules selection in job-shop production systems. *International Journal of Production Economics*, 227, 107669. <https://doi.org/10.1016/j.ijpe.2020.107669>
- Pickardt, C. W., Hildebrandt, T., Branke, J., Heger, J., & Scholz-Reiter, B. (2013). Evolutionary generation of dispatching rule sets for complex dynamic scheduling problems. *International Journal of Production Economics*, 145(1), 67–77. <https://doi.org/10.1016/j.ijpe.2012.10.016>
- Pillay, N. (2012). Evolving hyper-heuristics for the uncapacitated examination timetabling problem. *Journal of the Operational Research Society*, 63(1), 47–58. <https://doi.org/10.1057/jors.2011.12>
- Shiue, Y.-R., Lee, K.-C., & Su, C.-T. (2018). Real-time scheduling for a smart factory using a reinforcement learning approach. *Computers & Industrial Engineering*, 125, 604–614. <https://doi.org/10.1016/j.cie.2018.03.039>
- Swiercz, A. (2017). Hyper-Heuristics and Metaheuristics for Selected Bio-Inspired Combinatorial Optimization Problems. In *Heuristics and Hyper-Heuristics - Principles and Applications*. InTech. <https://doi.org/10.5772/intechopen.69225>
- Tarhini, A., Danach, K., & Harfouche, A. (2022). Swarm intelligence-based hyper-heuristic for the vehicle routing problem with prioritized customers. *Annals of Operations Research*, 308(1–2), 549–570. <https://doi.org/10.1007/s10479-020-03625-5>
- Teymourifar, A., Bahadir, O., & Ozturk, G. (2018). Dynamic Priority Rule Selection for Solving Multi-objective Job Shop Scheduling Problems. *Universal Journal of Industrial and Business Management*, 6(1), 11–22. <https://doi.org/10.13189/ujibm.2018.060102>
- Xu, X., Lu, Y., Vogel-Heuser, B., & Wang, L. (2021). Industry 4.0 and Industry 5.0—Inception, conception and perception. *Journal of Manufacturing Systems*, 61(October), 530–535. <https://doi.org/10.1016/j.jmsy.2021.10.006>
- Zhang, F., Mei, Y., & Zhang, M. (2019). A two-stage genetic programming hyper-heuristic approach with feature selection for dynamic flexible job shop scheduling. *Proceedings of the Genetic and Evolutionary Computation Conference, 2022-March*, 347–355. <https://doi.org/10.1145/3321707.3321790>
- Zhang, H., & Roy, U. (2019). A semantics-based dispatching rule selection approach for job shop scheduling. *Journal of Intelligent Manufacturing*, 30(7), 2759–2779. <https://doi.org/10.1007/s10845-018-1421-z>
- Zhou, Y., Yang, J. J., & Zheng, L. Y. (2019). Hyper-Heuristic Coevolution of Machine Assignment and Job Sequencing Rules for Multi-Objective Dynamic Flexible Job Shop Scheduling. *IEEE Access*, 7, 68–88. <https://doi.org/10.1109/ACCESS.2018.2883802>
- Zhu, L., Zhou, Y., Sun, S., & Su, Q. (2023). Surgical cases assignment problem using an efficient genetic programming hyper-heuristic. *Computers & Industrial Engineering*, 178, 109102. <https://doi.org/10.1016/j.cie.2023.109102>