A COMPARATIVE STUDY OF THE USABILITY OF DIFFERENT PRODUCTION SCHEDULING ALGORITHMS AND RULES

Pavol Jurík[1]

Abstract: Production scheduling optimization is a very important part of a production process. There are production systems with one service object and systems with multiple service objects. When using several service objects, there are systems with service objects arranged in a parallel or in a serial manner. We also distinguish between systems such as flow shop, job shop, open shop and mixed shop. Throughout the history of production planning, a number of algorithms and rules have been developed to calculate optimal production plans. These algorithms and rules differ from each other in the possibilities and conditions of their application. Since there are too many possible algorithms and rules it is not easy to select the proper algorithm or rule for solving a specific scheduling problem. In this article we analyzed the usability of 33 different algorithms and rules in total. Each algorithm or rule is suitable for a specific type of problem. The result of our analysis is a set of comparison tables that can serve as a basis for making the right decision in the production process decision-making process in order to select the proper algorithm or rule for solving a specific problem. We believe that these tables can be used for a quick and easy selection of the proper algorithm or rule for solving some of the typical production scheduling problems.

UDC Classification: 519.8, **DOI:** https://doi.org/10.12955/pns.v2.151

Keywords: logistics, production scheduling, optimization algorithms, flow shop, job shop, open shop, mixed shop

Introduction

Although, the topic of production scheduling optimization is not new, it is still very relevant because of the effort to automate as many production business processes as possible as a part of the Industry 4.0 agenda. Industry 4.0 is considered a new industrial stage in which vertical and horizontal manufacturing process integration and product connectivity can help companies to achieve higher industrial performance (Dalenogare et al., 2016). The whole concept is based mainly on the use of artificial intelligence (AI) and Internet-of-Things (IoT) technology; however, other computational principles may also be included and they too can be very beneficial. Industry 4.0 is a collective term embracing a number of contemporary automations, data exchanges and manufacturing technologies (Fast Technologies, 2021), including new Human-Machine-Interfaces – HMIs (Waschneck et al., 2016). In the field of production scheduling many computational algorithms and rules have been developed to calculate an optimal production plan. These algorithms and rules are ready to be included in the Industry 4.0 solution, since they are time-proven and effective. However, there are too many of them and each of them has its own limitations regarding conditions under which it can be used. Each of them is usable for a slightly different set of attributes, which describes the production scheduling problem. We examined 33 different algorithms and rules in total, however, we are aware that there are many more rules which could be included. Due to the scope of limitations of this article, that we were unable to include more algorithms in our study, we plan to further expand this research in the future.

Basic terminology and context

There are many types of production scheduling problems. In general, the aim of production scheduling is to determine a sequence of performing various operations on one or more service objects (i.e. machines). A service object is a device which is capable of performing one or more operations. Usually, the individual operations are denoted $o_1, o_2, ..., o_n$ and the service objects are denoted $M_1, M_2, ..., M_m$. In other words, there are n operations that need to be performed and m disposable service objects to perform these operations. The term operation means a basic integral activity, which is not divisible into partial operations or one we do not want to divide it into partial operations. Thus, it is an activity which we want to consider as an indivisible unit. For example, it can be the printing of a book, the varnishing of paper packaging, cutting, coloring or performing other forms of material processing on a specialized service object. A sequence of n operations $o_1, o_2, ..., o_n$ forms a job. The goal is the assignment of n operations to one or more service objects in the correct order so that each operation is performed on time or so that the total delay is minimal.

In most production scheduling models, two basic assumptions apply: 1. No service object performs more than one operation at a time. 2. No more than one operation of the same job is performed on more than one service object at a time.

[1] University of Economics in Bratislava, Faculty of Economic Informatics, Department of Applied Informatics, pavol.jurik@euba.sk

Classification of production scheduling models is related not only to the number of the service objects but also to the characteristics of the input data: 1. If all input data are unambiguously known in advance, we speak of *deterministic models*. 2. If the input data is known in advance with only a certain probability, these are *stochastic models*. 3. When a whole set of tasks enters an empty system in which there are no tasks, these are *static models.* In other words, in static models the next set of tasks enters the system only after the processing of all tasks of the previous set is completed. 4. If new tasks enter the system continuously (even if there are some unfinished tasks in the system), these are *dynamic models*.

We also distinguish specific categories of systems with several service objects, especially flow shop, job shop, open shop and mixed shop systems. According to Šeda (2006) *a flow shop system* can be defined as a system in which there are a set of *m* machines (processors) and a set of *n* jobs. Each job includes a set of *m* operations which must be done on different machines. All jobs have the same processing operation order when passing through the machines. There are no precedence constraints among operations of different jobs. Operations cannot be interrupted and each machine can process only one operation at a time. The problem is to find the job sequences on the machines which minimize the makespan, i.e. the maximum of the completion times of all operations. As the objective function, mean flowtime, completion time variance (Gowrishankar et al., 2001) and total tardiness (Pan et al., 2002) can also be used. In a flow shop system the service objects are arranged serially. For example, if there are two service objects M_1 and M_2, then each operation must start on the M_1 object and after that it can be completed on the M_2 object.

Similarly, *a job shop system* is a system in which there are a set of *m* machines (processors) and a set of *n* jobs. Each job includes a set of operations, where each operation means a realization of the job on a different machine. However, unlike the flow shop, in a job shop the order of the operations passing through the machines doesn't have to be the same for each job and each job doesn't have to pass through all machines. The main goal is again to minimize the makespan, i.e. to minimize the total time of performing tasks on the service objects (Brezina, 2003).

In *an open shop system* there are also *m* machines and a set of *n* jobs, where each job includes *m* operations. Each job must pass through all the machines, however, the order is not important. In other words, the operations forming a job are mutually independent and they can be carried out in any order. As Gonzales and Sahni (1976) state: "A car may require the following work: replace exhaust pipes and muffler, align wheels, and tune up. These three tasks may be carried out in any order."

A mixed shop system is a special variant that represents a combination of different types of scheduling systems. The term was first introduced by Masuda et al. (1985) and he defined it as a combination of the flow shop and the open shop system. The aim of the mixed shop system according to Masuda et al. (1985) is to schedule *n* tasks on two machines M_1 and M_2, where the set of all tasks *J* is divided into two disjoint subsets *F* and *O*. The subset *F* is a set of tasks that need to be processed in a *flow shop* manner and the subset *O* is a set of tasks that need to be processed in an open shop manner. In other words, the tasks in the subset *F* must pass through both machines in the same predetermined order, while the tasks in the subset *O* can pass through both machines in any order (i.e. it is not important if the task starts on the first machine and then it goes to the second machine or if the order will be reversed). However, Strusevich (1991) formulated a different kind of a mixed shop as a combination of a job shop and an open shop. A three-machine mixed shop problem was proven to be NP-hard (Shakhlevich et al., 2000). A three-machine mixed shop system consisting of a combination of the flow shop, job shop and open shop was further examined by Liu and Ong (2004) and can be solved using their algorithm.

Data and methodology

In the field of production scheduling many computational algorithms and rules have been developed to calculate an optimal production plan. Due to the scope of such a research as well as scope limitations of this article, we were unable to include all these algorithms and rules in our study. However, we plan to further expand this research in the future and include more algorithms in our analysis. In the study presented in this paper, the following algorithms and rules have been included. They are listed in alphabetical order:

- *Brucker, Jurisch and Siever's branch-and-bound method* – invented by Brucker et al. (1994)
- *Campbel, Dudek and Smith's heuristic* – invented by Campbel et al. (1970)
- *Earliest due date rule (EDD rule)* – described by Hax & Candea (1984)

- *Giffler and Thompson's algorithm for generating active schedules*
- *Gonzales and Sahnis's algorithm for 2 service objects*
- *Gonzales and Sahnis's algorithm for 3 or more service objects*
- *Gupta's heuristic* – invented by Gupta (1971)
- *Heuristic algorithm for non-preemptive operations scheduling on parallel-positioned objects* (we will refer to it as the HNPP algorithm) – described by Unčovský (1991)
- *Hu's algorithm* – invented by Hu (1961)
- *Johnson's algorithm for 2 consecutive service objects* – described by Brezina (2003)
- *Johnson's algorithm for 2 non-consecutive service objects* – described by Brezina (2003)
- *Johnson's algorithm for 3 consecutive service objects* – described by Brezina (2003)
- *Lawler's algorithm* – invented by Lawler (1973)
- *Little's algorithm* as an optimization solution for the travelling businessman problem – described by Hax & Candea (1984)
- *Liu and Ong's algorithm* – invented by Liu and Ong (2004)
- *Longest Alternate Processing Time rule (LAPT rule)* – described by Pinedo (1995)
- *Masuda, Ishii and Nishida's algorithm* – invented by Masuda et al. (1985)
- *McNaughton's algorithm* – described by Brezina (2003)
- *Moore's algorithm* – invented by Moore (1968)
- *Muntz-Coffman's algorithm* – described by Unčovský (1991)
- *Nawaz, Ensocore and Ham's heuristic (NEH heuristic)* – invented by Nawaz et al. (1983)
- *Nearest neighbour algorithm* as a heuristic method for the travelling businessman problem – described by Hax & Candea (1984)
- *Optimization algorithm for non-preemptive operations scheduling on parallel-positioned objects* (we will refer to it as the ONPP algorithm – described by Unčovský (1991)
- *Palmer's heuristic* – invented by Palmer (1965)
- *Parametrized version of the three-phase procedure* – described by Pinedo (1995)
- *Rapid acces heuristic algorithm (RA heuristic)* – invented by Dannenbring (1977)
- *Rothkopf's modification of Smith's* – invented by Rothkopf (1966)
- *Shortest expected processing time rule (SEPT rule)* – described by Hax and Candea (1984)
- *Shortest processing time rule (SPT rule)* – described by Hax & Candea (1984)
- *Shortest remaining processing time rule (SRPT rule)* – described by Hax and Candea (1984)
- *Smith's rule* – invented by Smith (1956)
- *Strusevich's algorithm for the two-machine super-shop scheduling problem* (Strusevich, 1991).
- *Tree-phase procedure* – described by Pinedo (1995)

In total, 33 algorithms and rules were included in the study. Each of them has its specifics and limitations, which we had to analyze. For each of them, our goal was to identify the attributes of the production scheduling problem for which the particular algorithm or rule is appropriate to use. Our secondary goal was to present the result in a practical, easy-to-understand and easy-to-use way. The results are presented in the next chapter.

Results and discussion

In this chapter, we would like to present the results of our study. The results are presented in the form of tables. Please notice that in some cases there are algorithms or rules which appear twice in the same table because they can be used in two different situations.

Table 1: Single machine production scheduling

Name of the Algorithm or Rule	The arrival of operations in the system	Shall the weight (importance) of each operation be the same?	Preemptive (p) or non-preemptive (np) scheduling?	The goal
EDD rule	Static	Yes	np	to minimize mean flow time
SPT rule	Static	Yes	np	to minimize maximum tardiness
Smith's rule	Static	Yes	np	to minimize weighted flow time
Moore's algorithm	Static	Yes	np	to minimize number of delayed operations
Lawler's algorithm	Static	No (each operation may have its individual weight)	np	to minimize the maxi-mum values of weighted delays
Little's algorithm	Static	Yes	np	to minimize the sum of times required to set up the machine
Nearest neighbor algorithm	Static	Yes	np	to minimize the sum of times required to set up the machine
SPT rule	Dynamic deterministic	Yes	np	to minimize mean flow time
SRPT rule	Dynamic deterministic	Yes	p	to minimize mean flow time
EDD rule	Dynamic deterministic	Yes	p	to minimize mean flow time
SEPT rule	Dynamic stochastic	Yes	np	to minimize mean flow time
Rothkopf's modification of Smith's rule	Dynamic stochastic	No (each operation may have its individual weight)	np	to minimize weighted flow time

Source: Author

Table 2: Parallel-positioned machines production scheduling

Name of the Algorithm or Rule	Number of service objects	Preemptive (p) or non-preemptive (np) scheduling?	Are the operations mutually dependent?	Are all operations assumed to have the same processing length?	Optimization (o) or heuristic (h) approach?
McNaughton's algorithm	*m*	p	independent	No	o
HNPP algorithm	*m*	np	independent	No	h
ONPP algorithm	*m*	np	independent	No	o
Hu's algorithm	*m*	np	dependent	Yes	o
Muntz-Coffman's algorithm	*m*	p	dependent	No	o

Source: Author

Table 3: Production scheduling in job shop systems

Name of the Algorithm or Rule	Number of service objects	Are the operations mutually dependent (predecessors and followers)?	Are down-times of the service objects permissible?	Appropriateness
Johnson's algorithm for 2 non-consecutive service objects	2	independent	Yes	For 2 service objects only
Giffler and Thompson's algorithm for generating active schedules	*m*	dependent	No	For small-scale problems
Brucker, Jurisch and Siever's Branch-and-bound method	*m*	independent	Yes	For large-scale problems – 10x10 or more (i. e. 10 service objects and 10 jobs)

Source: Author

Table 4: Production scheduling in flow shop systems

Name of the Algorithm or Rule	Number of service objects	Strong point	Weak point
Johnson's algorithm for 2 consecutive service objects	2	It finds the optimal solution using combinatorics	It is suitable for two objects only
Johnson's algorithm for 3 consecutive service objects	3	It finds the optimal solution using combinatorics	Very time-consuming calculation
Palmer's heuristic	*m*	Shortest computational time (Modrák et al., 2009)	Result can differ from optimal solution to a high degree (Modrák et al., 2009)
Gupta's heuristic	*m*	More accurate results compared to Palmer's heuristic (Brezina, 2003)	Longer computational time compared to Palmer's heuristic (Modrák et al., 2009)
Campbel, Dudek and Smith's heuristic	*m*	More accurate results compared to Palmer's heuristic (Ghazanfari & Yaghoobi, 2002)	Longer computational time compared to Palmer's heuristic (Modrák et al., 2009)
Rapid access heuristic algorithm	*m*	It gives the best solutions compared to other heuristics for more than 4 objects (Malik & Dhingra, 2013)	Results may be skewed if the number of jobs and machines approaches 10 (Dannenbring, 1977)
NEH heuristic	*m*	It gives the best solutions compared to other heuristics for less than 5 objects (Malik & Dhingra, 2013)	Very demanding on CPU power (Semančo & Modrák, 2012)

Source: Author

Table 5: Production scheduling in open shop systems

Name of the Algorithm or Rule	Number of service objects	Preemptive (p) or non-preemptive (np) scheduling?	Goal
Gonzales and Sahnis's algorithm for 2 service objects	2	np	To minimize makespan
Gonzales and Sahnis's algorithm for 3 or more service objects	3 or more	np	To minimize makespan
LAPT rule	2	np	To minimize makespan
LAPT rule	*m*	p	To minimize makespan
Tree-phase procedure described by Pinedo	*m*	np	To minimize lateness
Parametrized version of the tree-phase procedure described by Pinedo	*m*	p	To minimize lateness

Source: Author

Table 6: Production scheduling in mixed shop systems

Name of the Algorithm or Rule	Number of service objects	Combination of shops
Masuda, Ishii and Nishida's algorithm	2	flow shop and open shop
Strusevich's algorithm	2	job shop and open shop
Liu and Ong's algorithm	3	job shop, flow shop and open shop

Source: Author

Conclusion

The usability of 33 different production scheduling algorithms and rules was analyzed. We believe that this study will be useful for choosing a proper quantitative method for solving some of the typical production scheduling problems.

This work was supported by project VEGA No. 1/0373/18 entitled "Big data analytics as a tool for increasing the competitiveness of enterprises and supporting informed decisions" by the Ministry of Education, Science, Research and Sport of the Slovak Republic.

References

Brezina, I. (2003). *Kvantitatívne metódy v logistike*. Bratislava, Slovak Republic: Vydavateľstvo EKONÓM.

Brucker, P., Jurisch, B., & Sievers, B. (1994). A branch and bound algorithm for the job-shop scheduling problem. *Discrete Applied Mathematics*, 49, 107-127.

Campbell, H. G., Dudek, R. A., & Smith, M. L. (1970). A heuristic algorithm for the `n' job `m' machine sequencing problem. *Management Science,* 16, B630-B637.

Dalenogare, L. S., Benitez, G. B., Ayala, N. F., & Frank, A. G. (2018). The expected contribution of Industry 4.0 technologies for industrial performance. *International Journal of Production Economics,* 204, 383-394. http://dx.doi.org/10.1016/j.ijpe.2018.08.019.

Dannenbring, D. G. (1977). An evaluation of flow shop sequencing heuristics. *Management Science*, 23(11), 1174-1182.

Fast Technologies. Welcome to Fast Technologies. Retrieved March 15, 2021, from https://fasttechnologies.com.

Ghazanfari, M., & Yaghoobi, Z. (2002). A Hybrid GA Model to Solve Regular and Blow Flow-Shop Problems. *Scientia Iranica*, 9(3), 276-282.

Gonzales, T., & Sahni, S. (1976). *Open Shop Scheduling To Minimize Finish Time*. Journal of the ACM, 23(4), 665-679. doi:10.1145/321978.321985.

Gowrishankar, K., Rajendran, C., & Srinivasan, G. (2001). Flow Shop Scheduling Algorithms for Minimizing the Completion Time Variance and the Sum of Squares of Completion Time Deviations from a Common Due Date. *European Journal of Operational Research*, 132, 643-665.

Gupta, J. (1971). A functional heuristic algorithm for the flow shop scheduling problem. *Operations Research Quarterly*, 22, 39-47.

Hax, A. C., & Candea, D. (1984). *Production and Inventory Management*. Englewood Cliffs, New Jersey: Prentice-Hall.

Hu, T. C. (1961). Parallel Sequencing and Assembly Line Problems. *Operations Research*, 9(6), 841-848.

Liu, S. Q., Ong H. L. (2004). Metaheuristics for the Mixed Shop Scheduling Problem. *Asia-Pacific Journal of Operational Research*, 21(4), 97-115.

Lawler, E. L. (1973). Optimal sequencing of a single machine subject to precedence constraints. *Management Science*, 19, 544-546.

Masuda, T., Ishii, H., & Nishida, T. (1985). The Mixed Shop Scheduling Problem. *Discrete Applied Mathematics*, 11, 175-186.

Malik, A., & Dhingra, A. K. (2013). Comparative Analysis of Heuristics for Makespan Minimising in Flow Shop Scheduling. *International Journal of Innovations in Engineering and Technology*, 2(4), 263-269.

Modrak, V. Semanco, P. & Kulpa, W. (2009). Performance measurement of selected heuristics algorithm for solving scheduling problems. European Journal of Operational Research, 34, 158–183. doi: 10.1109/SAMI.2013.6480977.

Moore, J. M. (1968). A n job, one machine sequencing algorithm for minimizing the number of late jobs. *Management Science*, 15, 102-109.

Nawaz, M. Enscore, E. & Ham, I. (1983). A heuristic algorithm for the m machine, n job flow shop sequence problem. *OMEGA*, 11, 91-95.

Palmer, D. S. (1965). Sequencing jobs through a multi stage process in the minimum total time - a quick method of obtaining a near optimum. *Operational Research Quarterly*, 16, 101-107.

Pan, J. H., Chen, J. S., & Chao, C. M. (2002). Minimizing Tardiness in a Two-Machine Flow-Shop. *Computers & Operations Research*, 29, 869-885.

Parveen, S., & Ullah, H. (2010). Review On Job-Shop And Flow-Shop Scheduling Using Multi Criteria Decision Making. *Journal of Mechanical Engineering*, ME41(2), 130-146.

Pinedo, M. (1995). *Scheduling: Theory, Algorithms, and Systems*. New Jersey: Prentice Hall.

Rothkopf, M. (1966). Scheduling with random service times. Management Science, 12 (9), 707-713. doi: 10.1287/mnsc.12.9.707.

Semančo, P., & Modrák, V. (2012). A Comparison of Constructive Heuristics with the Objective of Minimizing Makespan in the Flow-Shop Scheduling Problem. *Acta Polytechnica Hungarica*, 9(5), 177-190.

Shakhlevich, N. V., Sotsko Y. N. & Werner, F. (2000). Complexity of mixed shop scheduling problems: a survey. *European Journal of Operational Research*, 120, 343–351.

Smith, W. E. (1956). Various optimizers for single-stage production. *Naval Research Logistics Quarterly*, 3.

Strusevich, VA (1991). Two machine super shop scheduling problem. *Journal of the Operational Research Society*, 42(6), 479–492.

Šeda, M. (2006). Mathematical Models of Flow Shop and Job Shop Scheduling Problems. *International Journal of Applied Mathematics and Computer Sciences*, 4(4), 241-246.

Unčovský, L. (1991). *Modely sieťovej analýzy*. Bratislava, Slovak Republic: Alfa.

Waschneck, B., Bauernhansl, T., Altenmüller, T., & Kyek, A. (2016). Production Scheduling in Complex Job Shops from an Industrie 4.0 Perspective: A Review and Challenges in the Semiconductor Industry. *1st International Workshop on Science, Application and Methods in Industry 4.0, SAMI 2016. Proceedings.*