Decentralized Vs. Centralized Sequencing in a Complex Job-Shop Scheduling

Afshin Mehrsai, Gonçalo Figueira, Nicolau Santos, Pedro Amorim, Bernardo Almada-Lobo

Decentralized vs. Centralized Sequencing in a Complex Job-Shop Scheduling

Afshin Mehrsai*, Gonçalo Figueira, Nicolau Santos, Pedro Amorim, Bernardo Almada-Lobo

INESC TEC, Porto, Portugal
amehrsai@inesctec.pt

Abstract. Allocation of jobs to machines and subsequent sequencing each machine is known as job scheduling problem. Classically, both operations are done in a centralized and static/offline structure, considering some assumptions about the jobs and machining environment. Today, with the advent of Industry 4.0, the need to incorporate real-time data in the scheduling decision process is clear and facilitated. Recently, several studies have been conducted on the collection and application of distributed data in real-time of operations, e.g., job scheduling and control. In practice, pure distribution and decentralization is not yet fully realizable because of e.g., transformation complexity and classical resistance to change. This paper studies a combination of decentralized sequencing and central optimum allocation in a lithography job-shop problem. It compares the level of applicability of two decentralized algorithms against the central scheduling. The results show better relative performance of sequencing in stochastic cases.

Keywords: Decentralization, Centralization, Allocation, Sequencing, Job-Shop scheduling, Industry 4.0.

1 Introduction

In the last decade, studies on distributed data collection and decentralized decision making in manufacturing research has been intensified and promoted [1]. The underlying reason lies on the upcoming surge in e.g., the complexity of operations, individualization of products, and the needs for higher flexibility and responsiveness in manufacturing [1] [2]. Consequently, industries are experiencing a digitalization era of producing products and providing services at the presence of the fourth industrial revolution, namely, “Industry 4.0” (I4.0) [3], see BMBF (Germany). Development in and facilitation of the modern information and communication technology (ICT), being capable of collecting and processing data in a distributed and decentralized manner, supports this ambition. The quick changes in manufacturing necessitate explorations and definition of a comprehensive reference framework for adopting I4.0. This paradigm shift alternates the classical (central and offline) structure of decision making based on aggregated data collection for global optimization. In contrast, the new trend supports a decentralized and distributed structure aiming at simplicity [1]. However, more explorations for applicability is due. Whereas the aggregated data collection and offline decision making assumes authenticity of all assumptions and available data, the process of decentralized data collection in real-time always varies along the time horizon, so that it necessitates a modern approach for dynamic-based decisions. Today, the research question is: where is the best application scenario in manufacturing to adopt this structure?. At this level, a coalition between both structures (hybrid centralized and decentralized) seems practical. Scholars need to introduce ways to practitioners for experiencing this paradigm shift. Generally, the shift toward decentralization of control reflects the needs of improvising higher flexibility and responsiveness at the presence of dynamics, [4]. The facilitation of real-time monitoring and control from (days to seconds) is causing this shift. Among all hierarchical decision making levels (strategic, tactical, and operational), scheduling and control at the shop-floors are very sensitive to real-time data collection [5]. The classical scheduling problems leverage assumptions in theory, which are not realistic in practice. The lack of real-time data collection and relatively few dynamics in the operations has supported the success of the classical solutions, whereas the upcoming circumstances, e.g., cyber-physical systems [6], necessitate scheduling and control decisions closer to real-time. The job scheduling problem can be divided into two parts. First is the allocation of jobs to a set of machines, concerning the specifications of both jobs and machines. Second is the sequencing of jobs for processing on each given machine, accomplished by the permutation of them based on relevant parameters, e.g., processing times, dependent setup times, etc. [7]. Job-shop scheduling problems with sequence dependent setup is severely NP-hard [8]. The broad applicability and variety of job shop scheduling problems in practice makes it an interesting environment for conducting research about decentralized data collection and real-time decision making. Meanwhile, some trade-offs between the pure decentralization of decision (heterarchical structure) and classical central optimization (hierarchical structure) have to be analyzed [9], based on the application level of technologies, dynamic operations, and efficiency of the developed algorithms. This paper studies a combination of central and decentralized approaches in scheduling of a complex job-shop problem at lithography operations. However, the aim is to explore the decentralized vs. central approaches in a three levels of spectrum, as the three options in Fig. 1 (c1-c3). Here only the sequencing task is run by the decentralized structure as in option 1 and the rest are kept for future work. The rest of the paper is structured as follows. An introduction to the scheduling problem and the case study is given. Then the mathematical model for the problem with centralized solution is explained. Later, the decentralized algorithms are discussed and alternative scenarios are experimented. The results are discussed afterwards and the paper ends with the conclusions and research opportunities.

2 Problem Framing and Case Study

Job scheduling is a combinatorial optimization problem with varieties [10]. The current problem is a job shop scheduling with alternative routes that require the allocation to machines (choose the route) as well as the sequencing of jobs, according to the set objective function(s), in the most optimized/ efficient way, see also [11]. However, solving the centralized mathematical programming with exact algorithm or heuristics cannot always lead to an optimum solution, because of the underlying complexity. Furthermore, the required assumptions for centralized solutions, e.g., the availability of all jobs at time zero or deterministic parameters, are mostly impractical [12]. A scheduling case out of lithography manufacturing processes for producing various canes with alternative applications, sizes, and content is considered. The complexity of operations and abundant changeover (times) makes it a suitable case for I4.0 experiments. Varnishing for coating and printing for coloring are the major processes to be scheduled in this slice of the manufacturing processes, see Fig.1a.

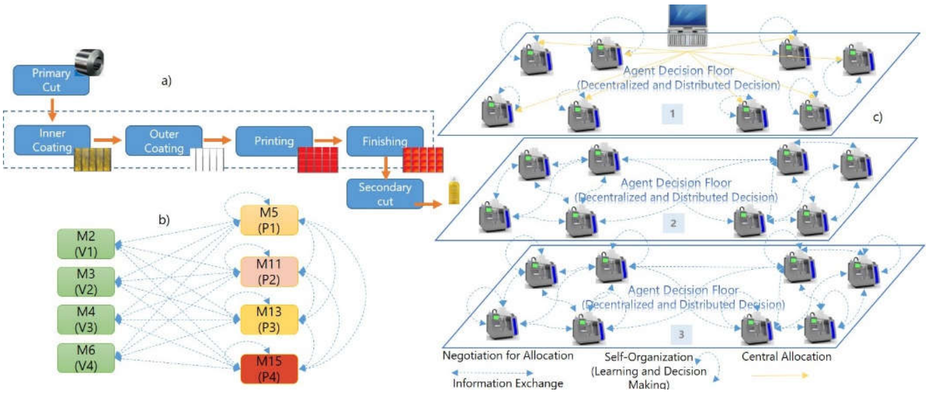


Fig. 1. a) The manufacturing processes for scheduling. b) Arrangement of the flexible job shop with flow possibilities. c1) Decentralized sequencing and central allocation. c2) Decentralized sequencing and allocation based on negotiation and real-time data. c3) Decentralized sequencing and allocation based on real-time negotiation and historical knowledge of agents.

This shop-floor consists of eight machines with four varnishing (*V*) and four printing machines (*P*). For the sake of simplicity, all machines are named as *M={M2, ..., M15}*, see Fig. 1b. The varnishing machines are capable of doing all types of operations, whereas the printing machines have alternative ink capacities not for all types. On this basis, the scheduling environment does neither resemble a pure job-shop nor a pure flexible job-shop. However, it can represent a flexible job-shop problem, with machine eligibility restrictions (M_i), recirculation ($rcrc_i$), setup-dependent (s_{ik}) jobs, and makespan (Cm) as the objective, i.e., machine environment is $FJ_8|M_i, s_{ik}, rcrc_i|Cm$. This lithography scheduling has complex constraints, e.g., precedence severity, drastic setup times, limited combination of colors (the CMYK color model), and recirculation, see [13]. However, this is much simplified to comply with the limitation of a conference. To solve this, the problem is split into the allocation and sequencing sections. A global mathematical model is initially developed to statically model this scheduling and help to its comprehension. However, it is solved by two approaches. First is the centralized solution with allocation and sequencing. This uses the classical optimization with mathematical programming and using the exact algorithm of branch and bound (BB) for allocation [14] and then using constructive heuristic [15] for sequencing. This is solved by the IBM ILOG Cplex solver. Second is the decentralized sequencing in real-time by developing a discrete-event simulation model, using the Anylogic program. The decentralized approach at this level presents a pure decentralization of sequencing in real-time of the centrally allocated jobs.

3 Centralized Allocation and Sequencing

Initially, the MRPs planning derives the jobs (in an aggregated manner and a longer time-horizon) to be scheduled. To do the allocation for this study, the characteristics of the jobs regarding the combination of colors, the capacity of the machines and the coating operations, and the objective function are decisive. These specifications of the jobs are given in a master data as the parameters (inputs) of the mathematical model. The consideration of these simply results in one to few alternative operation-routes for each job as (L_i). A generic mathematical model can represent the problem as follows.

Indices and Sets:		Parameters:	
$i \in I$	Jobs	$pt_q =$	Processing time of operation q
$m \in M$	Machines	$st_{\acute{q}q} =$	Setup time of operation $\acute{q}$ after process of operation q on same machine
$l \in L_i$	Sequence id of job i	$G =$	Big value
$q \in Q_l$	Operation id of sequence l	$gap =$	a gap time (transport time) between machine switch
$n \in N$	Operation number in sequence		
m_q	Operation q on machine m		

Variables:
$Cm =$ makespan
$x_l = 1$ if sequence l is performed, 0 otherwise (allocation)
$y_{\acute{q}q} = 1$ if operation $\acute{q}$ is performed before operation q, 0 otherwise (sequencing)
$S_q =$ Starting time of operation q.

$$Minimize\ (Cm) \tag{1}$$

St.
$$\textstyle\sum_{l \in L_i} x_l = 1;\ \forall\, i \in I \tag{2}$$

$$S_q \ge S_{\acute{q}} + pt_{\acute{q}} + st_{\acute{q}q} - \left(1 - y_{\acute{q}q}\right) \times G - (1 - x_l) \times G - (1 - x_{\acute{l}}) \times G;\ \forall\, \acute{q}, q \in Q: \acute{q} > q, m_q = m_{\acute{q}}, l = l_q, \acute{l} = l_{\acute{q}},\ l_q = l_{\acute{q}} \tag{3}$$

$$S_{\acute{q}} \ge S_q + pt_q + st_{q\acute{q}} - \left(y_{\acute{q}q}\right) \times G - (1 - x_l) \times G - (1 - x_{\acute{l}}) \times G;\ \forall\, \acute{q}, q \in Q: \acute{q} > q, m_q = m_{\acute{q}}, l = l_q, \acute{l} = l_{\acute{q}}, l_q = l_{\acute{q}} \tag{4}$$

$$S_q \ge S_{\acute{q}} + pt_{\acute{q}} + gap - (1 - x_l) \times G;\ \forall\, \acute{q}, q \in Q: n_q = n_{\acute{q}} + 1\,, m_q \ne m_{\acute{q}}, l_q = l_{\acute{q}} \tag{5}$$

$$Cm + (1 - x_l) \times G \ge S_q + pt_q;\ \forall\, i \in I, l \in L_i, q \in\ Q_l \tag{6}$$

Later, by splitting this model into allocation and sequencing ones, BB chooses the best route (among alternatives) for each job (allocation task). This part is relatively simple and can reach more than 99.9% optimality within 10 min. The sequencing model is an NP-hard problem. For the instances of the case study, it was not even able to solve the linear re-laxation. Therefore, a heuristic approach was used. The sequencing method consists of a multi-start constructive heuristic, which generates thousands of solution and keeps only the best. The procedure to construct a solution is probabilistic, i.e., the next operation to sequence is chosen based on some probability that assigned to each operation. This probability is higher for operations with lower setups, considering the operation that is on the machine at that moment. The level of greediness of the algorithm can be parametrized, by using a scoring function such as $u_q = \left(\frac{1}{st_{\acute{q}q}}\right)^k$, where $\acute{q}$ is the operation at the machine, q is the one being considered, u_q its score and k the greedy factor. The higher the k, the greedier the algorithm will be, since it will be amplifying the differences between operations. The probability is then obtained by dividing each score by the sum of all the scores.

4 Decentralization in Real-Time Sequencing

Decentralization of the sequencing decisions in real-time is an extreme case of the spectrum of decentralization scenarios. At this instance, a combination between central allocation and decentralized sequencing is made to witness the cooperation level of both approaches in practice. Here, each machine decides about the sequence of its own queue in real-time status. Two major decentralized algorithms for sequencing are considered to be compared against the purely central (optimum) solution for the allocation and sequencing. Makespan is the objective function. The first algorithm, Setup rule, purely considers the dependent setup times of the operations in the queue It is equivalent to the constructive heuristic described previously, but in its deterministic form and executed only once (no multi-start). However, this simple process is repeated each time a product is to be processed. The algorithm is as follows:

1. While $\exists\, q$ in the queue

$$F_q = \left(s_{0q}^m \times (sum\ of\ setups)\right)^2 \Big/ \sum_q^Q \left(\left(s_{0q}^m \times (sum\ of\ setups)\right)^2\right);\ \forall \mathrm{q} \in Q$$

2. Sort the queue in descending order of F_q and send the first operation to the machine.

The second algorithm has a broader view to each entire queue and follows the Little´s Law [16] in analyzing the entire queue, by calculating the queue-length (QL) at each new instances. It considers several possible permutations (of the *n!* possibilities) for the arrangement of the products in the queue and, for each of them, it calculates the QL. Given this, the best permutation is selected regarding the least QL. However, this algorithm is not necessarily beneficial in purely sequencing decisions, but in real-time allocation. Since the queue arrangement changes dynamically in every entrance instance, the permutation with the least QL at an instance does not necessarily lead to the least makespan of the same machine or the global makespan. Also, the least QL does not guarantee the product with the least setup time as the next one to be processed. Knowing that the permutation instances can be very large, a simple genetic algorithm (GA) constructively covers a certain number of the permutations. The GA considers only mutation and crossover operators for producing 2 child out of 2 parents. The number of individuals in each generation varies between 5 to 10 and the termination is after 5 generations. The GA is as follows:

1. $Z \leftarrow (popSize - 1)$ random queue permutations

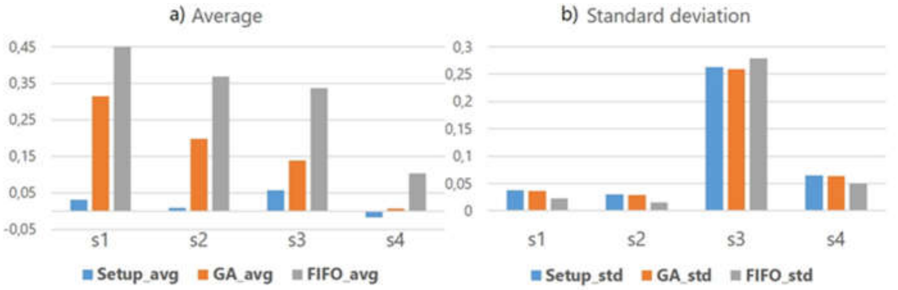
2. Z←Z ∪ {current queue}
3. while $Z > 0$
   - Take Fitness Function $f = QL + (Localavg.throughputtime + globalavgthroughputtime)/2$ of each $indiv \leftarrow Z$, where $QL = \lambda_t^m \cdot \left(\sum_{q_t<\acute{q}_t=1}^{Q_t} \left(s_{\acute{q}_t q_t}^m + pt_{\acute{q}}^m\right) \times y_{\acute{q}_t q_t}^m\right)$ based on Little´s Law
4. Use Roulette Wheel selection $P_t = f_{rt}/\sum_{r=1}^{R} f_{rt}$ for choosing the Best (#) Individuals
5. Select the Best Individuals by Descendingly sorting their P_t
6. Apply GA operators (Mutation and Crossover) to the Best Individuals to breed Children; add the Best Individuals (from parents list) to the Children list in next Generation if (#BestInd<Z)
7. Sort the Best Individuals in the new Generation and Repeat up to the Termination #
8. Take the first Individual of the List BestInd as the queue for performance!

5 Simulation Results and Discussion

Both decentralized algorithms as the Setup rule and GA are evaluated against the centralized solution and FIFO in the simulation model. For comparing their performances, five scheduling instances based on real data are experimented. To integrate the results and save space only the average values (avg.) of all instances are demonstrated. Then, four scenarios (1 deterministic and 3 stochastics) for processing times are comparing the avg. of the other algorithms relative to the avg. of the central solution, as the reference with the value=0, see Fig.2. It is observed that the average performance of the central solution mostly outperforms the decentralized algorithms in makespans. However, in each scenario this relativeness was changing. In the deterministic scenario, this was fully expected, whereas in the scenario 4 central solution was not the best. In s3 the performance deviation between the instances was noticeable. This is because of the broad spread of the normal distribution. Nonetheless, in most stochastic cases the decentralization showed a reasonable performance relative to the central one. It was witnessed that the decentralization in some other measures like average throughput time (ATPT) could hit the record of the central one in several single tests. However, for the makespan it was even not expected to hit the record of the central solution, since it is a global measure. The Setup rule was easy to implement, but showed pretty good results. The GA as expected was not specifically suitable for sequencing, while it is expected to operate much better in case of decentralized allocation in future works. In decentralization the focus is on local awareness, so as the dynamics of the global system may negatively influence the global objective of the system. However, having local overview, while being comparable with the global measure, achieved by the central solution, shows the potential of the decentralization in sequencing.

6 Conclusion

To gradually adopt the I4.0, more explorations in manufacturing and shop-floor operations are required. Scheduling must adapt itself to this opportunities by employing real-time data for planning the flows. This issue was the concern of this study. The outputs are supposed to help practitioner to smoothly experience the transformation phase. The simplicity, yet comparable, of the algorithms in decentralization presents a promising application of that in practice, though for a verification more experiments are required. The central solutions are usually very limited, sensitive, and have less flexibility in several aspects. For instance, the assumptions are fixed at the run of the solutions and global overview of the entire system is necessary. In case of static models, no urgent order can intervene the system and rescheduling is required. Any changes in the model (e.g., changing deterministic processing time to stochastic one) requires several modifications and solution efforts. In contrast, these all dynamics can easily happen in a decentralized and real-time system without requiring any changes in the structure of the running system (no disturbance in the performance of material flow control). In implementing I4.0, (intelligent) agents (e.g., machines) with simple algorithms can deal with the complexities and still deliver good results. The extension of this work with considering all color combinations for machines (routes+ the printing capacities) and the experiments of other decentralization levels and algorithms are still due.



	Scenarios for Processing Time							
	s1) pt		s2) $N\sim(\mu = pt, \sigma = pt/2)$		s3) $N\sim(\mu = pt, \sigma = pt)$		s4) $Exp(x = pt, \lambda = 1/pt)$	
	Avg.	Std.	Avg.	Std.	Avg.	Std.	Avg.	Std.
Setup	0.0305	0.037	0.0081	0.030	0.0565	0.262	-0.0166	0.064
GA	0.3150	0.036	0.1974	0.029	0.1385	0.259	0.0061	0.063
FIFO	0.4489	0.022	0.3680	0.015	0.3372	0.278	0.1034	0.050

Fig. 2. Comparing the divergence of the 3 Algorithms, in 4 Scenarios, from the Centralized Sequencing as reference 0, a) avg. b) standard deviation.

References

1. Zhuming B., Xu, L., Wang, C.: Internet of things for enterprise systems of modern manufacturing. IEEE Transactions on industrial informatics, 10(2), 1537-1546 (2014)
2. Mehrsai, A., Karimi, H. R., Thoben, K. D.: Integration of supply networks for customization with modularity in cloud and make-to-upgrade strategy. Systems Science & Control Engineering: An Open Access Journal, 1(1), 28-42 (2013)

3. Posada, J., et al.: Visual computing as a key enabling technology for industrie 4.0 and industrial internet. IEEE computer graphics and applications, 35(2), 26-40 (2015)
4. Ivanov, D., Sokolov, B., Kaeschel, J.: A multi-structural framework for adaptive supply chain planning and operations control with structure dynamics considerations. European Journal of Operational Research. 200(2), 409-420 (2010)
5. Huang, G. Q., Zhang, Y. F., Chen, X., Newman, S. T.: RFID-enabled real-time wireless manufacturing for adaptive assembly planning and control. Journal of Intelligent Manufacturing, 19(6), 701-713 (2008)
6. Wang, L., Törngren, M., Onori, M.: Current status and advancement of cyber-physical systems in manufacturing. Journal of Manufacturing Systems. 37(Part 2), 517-527 (2015)
7. Pined Ivanov, D., Sokolov, B., Kaeschel, J. o, M.: Scheduling Theory, Algorithms, and Systems, Springer (2015)
8. Rossi, A.: Flexible job shop scheduling with sequence-dependent setup and transportation times by ant colony with reinforced pheromone relationships," International Journal of Production Economics. 153, 253-267 (2014)
9. Leitão, P.: Agent-based distributed manufacturing control: A state-of-the-art survey. Engineering Applications of Artificial Intelligence, 22(7), 979-991 (2009)
10. Baykasoğlu, A., Hamzadayi, A., Köse, S. Y.: Testing the performance of teaching–learning based optimization (TLBO) algorithm on combinatorial problems: Flow shop and job shop scheduling cases. Information Sciences. 276, 204-218 (2014)
11. Mehrsai, A., Karimi, H. R., Thoben, K. D., Scholz-Reiter, B.: Application of learning pallets for real-time scheduling by the use of radial basis function network. Neurocomputing. 101, 82-93 (2013)
12. Kreipl, S., Dickersbach, J. T.: Scheduling coordination problems in supply chain planning. Annals of Operations Research. 161(1), 103-122 (2008)
13. Govind, N., Bullock, E. W., He, L., Iyer, B., Krishna, M., Lockwood, C. S.: Operations management in automated semiconductor manufacturing with integrated targeting, near real-time scheduling, and dispatching. IEEE Transactions on Semiconductor Manufacturing, 21(3), 363-370 (2008)
14. Baptiste, P., Flamini, M., Sourd, F.: Lagrangian bounds for just-in-time job-shop scheduling. Computers & Operations Research. 35(3), 906-915 (2008)
15. Jungwattanakit, J., Reodecha, M., Chaovalitwongse, P., Werner, F.: A comparison of scheduling algorithms for flexible flow shop problems with unrelated parallel machines, setup times, and dual criteria. Computers & Operations Research. 36(2), 358-378 (2009)
16. Li, L.: An optimistic differentiated service job scheduling system for cloud computing service users and providers," in In Multimedia and Ubiquitous Engineering, 2009. MUE'09. Third International Conference on (2009)