TECHNICAL UNIVERSITY OF CLUJ-NAPOCA

ACTA TECHNICA NAPOCENSIS

Series: Applied Mathematics, Mechanics, and Engineering
Vol. 67, Issue Special I, February, 2024

SOLVING THE LOT STREAMING AND SCHEDULING PROBLEMS: A CASE STUDY FROM THE AUTOMOTIVE INDUSTRY

Radu-Constantin VLAD

Abstract: *This paper aimed to determine whether it is worth making the effort of integrating the job shop scheduling and the lot streaming problems. To answer the question a two-stage approach was used. Firstly, the number and the size of sublots were determined with a simple practice inspired heuristic. The schedule would then be developed with the help of a constraint programming model that considered challenging requests from the industry: product dependent setup times and changeovers. The objective function aimed to minimize the makespan, the number of changeovers and the number of orders not delivered on time. 30 test instances have been defined with data taken from a real large scale job shop.*
Key words: *job shop scheduling, lot streaming, constraint programming, large scale job shop.*

1. INTRODUCTION

Scheduling, as a general problem, has been included in the list known as NP-complete as early as 1975 [11]. Shortly after this Lenstra and his colleagues studied the complexity of machine scheduling and have concluded that a large number of them are in fact NP-complete. That is, the solution for these problems cannot be found with an efficient algorithm [7].

In 1976, Garey showed that the “job shop scheduling”, a variant of the general problem, is also NP-complete for any system with three or more machines [4]. Because of the combinatorial nature of the problem the more machines considered the more time is needed to solve the problem. To fully understand the challenge raised by the size of the problem it is worth mentioning the work of Chen and his colleagues who provided optimal algorithms for scheduling related machines [3]. Their work is regarded as an important achievement in the field, but it was published a long time after the first initiatives to solve the scheduling problem and it only referred to two machines.

Despite being a difficult to solve problem, scheduling is a critical activity for a manufacturing company because it allows it to respond to customer requests in a timely and cost-effective manner. For this reason, there are numerous models and algorithms that have been designed to provide a solution to the scheduling problem. To manage the great variety of problems systematic notations have been introduced. In one of the most used, the notation is composed of three fields, $\alpha|\beta|\gamma$, that correspond to the processor (α), to the tasks and resources used in the process (β) while the last field, γ, indicates the criteria used in optimization [5]. To refer to a job shop problem one must set only one value, out of seven, of the first field.

Job shop scheduling, the problem of interest in this paper, is just one of the classes in Graham’s notation and it has been shown to be a NP-hard problem [6]. Once again, being difficult to solve but important for the industry, it has attracted a lot of attention. So much so, that a classification scheme was needed to manage the large number of contributions dedicated to the job shop class alone.

Abdolrazzagh-Nezhad has identified 14 classes of contributions that could be grouped under the name of job shop scheduling (JSS) [1]. To produce all these groups the author has used the following criteria: job arrival process, the inventory policy, processing times and job attributes [1]. It is interesting to note that large-scale problems have been included into a class of their own. At the same time, it is worth pointing out Abdolrazzagh-Nezhad has not used the phrase “lot streaming”. It seems that the author considered that the initiatives that tried to integrate the JSS problem with the task of setting the number and sizes of lots were not numerous or specific enough to form a new class. Since Abdolrazzagh-Nezhad published his work the number of articles aiming to solve the lot streaming problem (LS) has grown motivated perhaps by the advantages that lot streaming could offer to the scheduling process.

A more recent analysis of the contributions to the JSS problem has been performed by Xiong and his colleagues [12]. They have analyzed papers from mid-1960s to 2020s and have classified them based on “attributes, assumptions, basic subtypes, and measures of performance” [12]. Having analyzed almost 300 papers published between 2016 and early 2021 Xiong and his team were able to identify promising research directions in the field of JSS.

Lot streaming, the second topic of interest in this paper, refers to the way lots (batches) are processed and transferred in a manufacturing system. The initial lots are divided into smaller batches called sublots or transfer lots to allow for the simultaneous processing of different sublots over different machines. This approach is supposed to decrease the makespan, production lead times, work-in-process inventory, and improve product delivery.

Considering these benefits, it makes sense to combine, to integrate the two problems and solve them in a unified manner. However, solving LS problem is difficult because it can be thought of as a “reversed knapsack” problem. In this case one does not add items to a knapsack with a finite capacity, but it splits a lot into smaller ones and it has to decide on the number of sublots and their sizes. Therefore, combining the JSS problem, for which there is no efficient solution, with another one, just as difficult, may not be best approach, especially in the case of large-scale job shops.

In line with the above idea, the present paper aimed to determine how the benefits of the LS concept could be used in finding a better solution for the JSS problem. The project described here did not aim to solve the integrated JSS-LS problem but followed an iterative approach. The JSS problem has been solved several times considering different solutions of the LS problem. The number and sizes of sublots were not “optimally” determined. Their values have been chosen based on a practice inspired idea. The machines in the manufacturing system for which the schedule was to be developed had productivities that varied significantly from one machine to another. For this reason, the management has already decided to split the large lots into smaller ones but large enough to ensure the continuous operation of some of the machines for 4 to 8 hours.

The practice inspired idea seemed appropriate because the project had to provide a solution for the daily operation of a company. It is well known that the industry has to operate in a timeline set by customers and therefore it needs good solutions and quick. This preference has shaped the scheduling model presented in this paper. It did not include all the features mentioned in the literature but focused instead on the ones of more importance for the company: makespan, lead times and delivery.

There were two important specifications the company insisted upon. First, the model had to provide a schedule for a large job shop in a reasonable amount of time. The company did not expect to get the solution instantly as it needed several days to set the schedule for the next week. Secondly, the model was supposed to be easily updated to future needs.

For these reasons it has been decided to use constraint programming in formulating the model and the use of the CPLEX Optimization Studio for solving the model.

2. LITERATURE REVIEW

Lot streaming has been a research subject for quite some time. A review of the early basic techniques has been performed by Trietsch and Kenneth [10]. They have presented models in which the size of the sublots was continuous and discrete, models that considered the possibility of machines becoming idle, and models with consistent and variable sublots.

A common feature of most of the research papers is that the JSS problem is modeled using a mathematical programming formalism. Most of the time mixed-integer linear programming models are used. If the models are not solved by "brute force" then optimization approaches are used (genetic algorithms, tabu search or particle swarm optimization) to get the best schedule. A common feature could also be mentioned the fact that most mathematical models have the objective function formed of a single item, namely the makespan.

The approach presented in this paper it is related to those that:

- use a two-stage procedure to find the best schedule;
- consider customer orders to define the lots and the sublots to be scheduled;
- use more than one criterion to formulate the objective function of the mathematical model;
- aim to solve the scheduling problem for large scale job shops.

Contributions that are based on a two-stage procedure include the work of Bozek and Werner on one hand and Yegane and his team on the other. In both cases the procedures are more complex than the one used in this paper.

The procedure developed by Bozek and Werner minimizes in the first stage the makespan by using sublots as small as possible [2]. In the second stage the size of the sublots is increased without changing the makespan found in the first stage. Mixed-integer linear programming and constraint programming models are used to find the best schedule.

Yegane and his team designed an approach that entailed the use of a memetic algorithm to get a first instance of the schedule, instance that was later improved with the help of a heuristic like the critical path method [13].

In most research papers the demand of the customers is presented in the form of volumes of products that must be delivered by a certain date. This means that the value of those volumes has to be determined from customer orders before optimization and later when the problem has been solved the same volumes have to be split in order to allocate to each customer the amount it requested. Such an approach may cause mistakes in delivering the products to the customers. To avoid such problems, Liu suggested that an order-based environment be considered [8]. In such approach customer orders are split into sublots. Basically, Liu has investigated if the expected benefits of the LS concept would materialize in a customer order environment.

Few research papers considered more than one criterion in defining the objective function of the optimization models. As an exception, Rooyani and Defersha have considered a 10-item objective function that included the sublot times, machine load along the ever-present makespan [9].

The last criteria used to identify similar research initiatives has produced only paper. Zhang and Wu have developed a hybrid approach to solving large-scale JSS problems [15]. Their algorithm used iteratively a decomposition technique to minimize the total weighted tardiness. On each iteration a new subproblem was defined with the help of a simulated annealing approach and later solved with a genetic algorithm. To test the proposed procedure, they have used seven large test instances. All of them were defined in the commonly used format "*n x m*", where "*n*" indicates the number of products and "*m*" the number of machines. The number of products varied between 50 and 1000 while the number of machines varied between 20 and 30. Of the seven test instances only three of them have dealt with a larger number of operations that were to be scheduled. In these cases, Zhang and Wu had to schedule 2, 4, and 5 times more operations than in the case considered in this paper. In other JSS initiatives the largest size of the test instances is "*20 x 20*", that is 10 times less than the size of the case considered in this paper.

3. PROBLEM DEFINITION

This paper presents the outcome of a project that developed a procedure for machine scheduling in a company in the automotive industry. The schedule had to be developed for a week for a set of more than 150 machines /workstations. The operation of the company was driven by the orders placed weekly by customers. The average number of orders per week varied between 50 and 60.

The problem considered in this paper is part of the general class of the JSS problems despite the fact it aimed to develop a schedule in line with the LS concept as well. The model developed did not aim to solve the JSS and the LS problems in an integrated manner, which is it did not aim to simultaneously determine the number, the size, and the sequencing of sublots on various machines. Its purpose was to provide a good schedule in a reasonable amount of time (3 to 5 minutes). It had to do this considering a number of sublots in which a customer order might be split.

The number and the size of those sublots have not been set with the help of a mathematical model but based on a statement made by the production manager overseeing the system's operation. He said that he would not turn on some machines if the size of the lot would not ensure a continuous operation of the machine for 4 to 8 hours. Starting from this statement a simple procedure has been developed to determine the number and the size of all sublots so that no operation on any of the machines would be longer than a limit set arbitrarily (the limit was called *MaxDuration*).

The procedure proposed in this paper comprised of the following steps:

1. a set of values would be chosen for *MaxDuration* – around the 4- or 8-hour limit used in the daily operation;
2. for each value of *MaxDuration* determine the number and the size of the sublots associated with a customer order.
3. develop a schedule for each value of *MaxDuration*;
4. select the best schedule considering criteria like: makespan, number of changeovers, number of orders overdue.

The model mentioned in the procedure above (hereafter called as the SL procedure) has been developed on the following assumptions:

- machines do not breakdown and preemption of operation is not allowed;
- each machine can process only one sublot at a time and each sublot could be processed by only one machine at a time;
- each sublot has a fixed processing route that comprises of a number of machines;
- each sublot has a release date, a due date;
- transportation times between workstations are neglected;
- setup times are product dependent;
- all sublots associated with an order have the same size.

4. MATHEMATICAL FORMULATION

To solve the JSS problem a mathematical model has been used. It has been formulated using the Constraint Programming formalism. Because the text of the model is too large to fit in the format of the page, only the main features of the model are presented in a descriptive manner. The set of decision variables included:

- interval variables associated with every operation that had to be scheduled;
- sequence variables for every machine;
- integer variables to determine the number of changeovers on each machine and the number of orders that could not be delivered before the deadline.

The set of constraints ensured that:

- operations were not scheduled before the release date;
- operations were scheduled only after their predecessors have been completed;
- there will be no overlapping between operations scheduled on the same machine (setup times have been considered).

The expression of the objective function included three items: makespan, the sum of the number of changeovers on each machine and the total number of orders that could not be delivered before the deadline.

5. RESULTS

The mathematical model has been solved with CPLEX 12.10 on a LenovoY700 machine with an Intel(R) Core(TM) i7-6700HQ CPU @ 2.60 GHz processor. A time limit of 90 seconds has been imposed on the duration of the optimization process.

The model has been tested with the help of 30 test instances. Each of these instances has been derived from the original data that included: order volumes, processing times (per operation per machine), processing routes. Test instances differed from one another because they have been generated with different values for the *MaxDuration* parameter. In line with the company's activity *MaxDuration* has been given values from 4 hours to 16 hours, 30 minutes apart.

The "performance" of each test instance has been evaluated against the following criteria: makespan, number of orders completed on time.

Figure 1 shows the variation of makespan as the MaxDuration has decreased from 16 hours to 4 hours. The variation of the makespan values show a small increasing trend. This could be explained by the fact that as *MaxDuration* was reduced the number of interval variables increased making the model more difficult to solve. This does not explain though the significant differences that could be observed between adjacent points. Therefore, it could be concluded that making schedules using only the "hands-on experience" may lead to bad decisions.

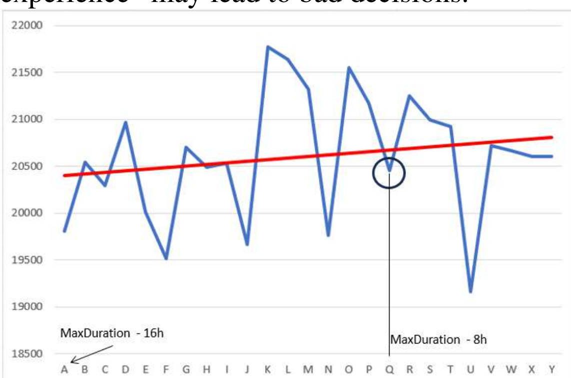

Fig. 1. Variation of the makespan.

With respect to the number of orders completed on time it could be stated that the decrease in the *MaxDuration* parameter led to redistribution of the completed projects. Figure 2 shows that the number of projects completed within 5 days increased at the expense of the projects completed within 12 days.

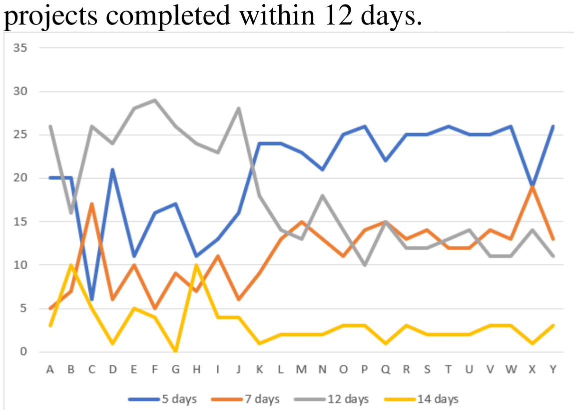

Fig. 2. Evolution of the number of orders completed.

The value of *MaxDuration* and implicitly the number of sublots does affect the utilization of machines as well. The graphs in Figure 3 have been produced considering two utilization indices. The one associated with the Ox axis is the classic utilization index while the one on the Oy axis tells how much time a machine has been used of the total makespan. The left image in Figure 3 shows the utilization of machine when the value of the *MaxDuration* was equal to 16 hours while the right image shows the same thing but for a parameter value of 4 hours.

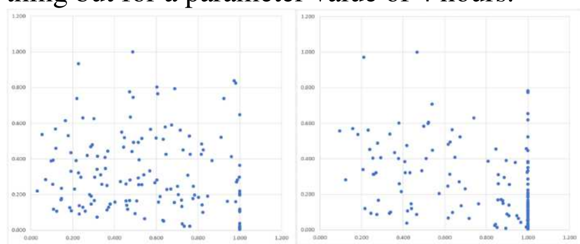

Fig. 3. The impact of the number of sublots on machine utilization.

Figure 3. shows that as the *MaxDuration* value decreased, that is the number of sublots increased, the utilization of some of the machines increased (there are more dots to the right of the right image than in the same area in the left image). In general, increasing the number of sublots should lead to more changeovers and thus to a decrease in utilization. In this case the reasoning is not correct because the number of changeovers was included in the objective function and so it was kept in check by the optimization process.

6. CONCLUSIONS

This paper presented the results of a project that aimed to determine whether it is worth making the effort of integrating the JSS and LS problems. At the end of the project, it was difficult to answer this question because increasing the number of sublots did not have a clear effect on all performance criteria. It was found that a larger number of lots would positively impact the number of orders completed on time and the utilization of machines. Unfortunately, the same is not true for an important criterion, namely the makespan.

Using constraint programming and CPLEX OPL was also a good decision because it was easy to develop and adapt the optimization model to the significant challenges posed by changeovers and setup times.

7. ACKNOWLEDGMENTS

Students of the Management and Economical Engineering Department at the Technical University of Cluj-Napoca are gratefully thanked for making this material available.

8. REFERENCES

[1] Abdolrazzagh-Nezhad, M., Abdullah, S, "*Job Shop Scheduling: Classification, Constraints and Objective Functions*", International Journal of Computer and Information Engineering, Vol. 11(4), pp. 429–434, (2017).

[2] Bożek, A., Werner, F., "*Flexible job shop scheduling with lot streaming and sublot size optimisation*", International Journal of Production Research, ISSN 00207543, (2017).

[3] Chen, X., Lan,Y., Benkő, A., Dósa, G., Han, X., "*Optimal algorithms for online scheduling with bounded rearrangement at the end*". Theoretical Computer Science, Vol. 412, No. 45, pp. 6269–6278, ISSN 0304-3975 (2011).

[4] Garey, M.R., "*The Complexity of Flowshop and Jobshop Scheduling*", Mathematics of Operations Research, 1 (2), pp. 117-129, (1976).

[5] Graham, R.L., Lawler, E.L., Lenstra, J.K., Rinnooy Khan, A., "*Optimization and Approximation in Deterministic Sequencing and Scheduling: A Survey*", Elsevier, pp. (5) 287-326, (1979).

[6] Lageweg, B.J., Lenstra, J.K., Rinnooy Khan, A., "*Job Shop Scheduling by Implicit Enumeration*", Management Science, Vol. 24, No. 4, pp. 441-450, ISSN 0025-1909, (1977).

[7] Lenstra, J.K., Rinnooy, K., Brucker, P., "*Complexity of Machine Scheduling Problems*", Annals of Discrete Mathematics, Vol. 1, pp. 343-362, ISSN 01675060, (1977).

[8] Liu, C.H., "*Lot streaming for customer order scheduling problem in job shop environments*", International Journal of Computer Integrated Manufacturing, Vol. 22, No. 9, pp. 890-907, ISSN: 0951-192X, (2009).

[9] Rooyani, D.; Defersha, F., "*A Two-Stage Multi-Objective Genetic Algorithm for a Flexible Job Shop Scheduling Problem with Lot Streaming*", Algorithms, 15, 246, ISSN: 1999-4893, (2022).

[10] Trietsch, D., Baker, K.R., "*Basic Techniques for Lot Streaming*." Operations Research, vol. 41, no. 6, pp. 1065–1076, 1993.

[11] J.D. Ullman, "*NP-complete scheduling problems*", Journal of Computer and System Sciences, Vol. 10, Issue 3, pp. 384-393, ISSN 0022-0000 (1975).

[12] Xiong, H., Shi, S., Ren, D., Hu, J., "*A survey of job shop scheduling problem: The types and models*", Computers & Operations Research, Vol. 142, ISSN 0305-0548, (2022).

[13] Yegane, B., Y., Kamalabadi, I.,N., Khanlarzade, N., "*Critical Path Method for Lot Streaming Problem in Flexible Job Shop Environment*", International Journal of Engineering, TRANSACTIONS B: Applications Vol. 30, No. 2, pp. 261-269, (2017).

[14] Yu, Y., "*A Research Review on Job Shop Scheduling Problem*", E3S Web Conf., Vol. 253, article no. 02024, (2021).

[15] Zhang, R., Wu, C.A., "*A hybrid approach to large-scale job shop scheduling*", Appl Intell, Vol. 32, pp. 47-59, (2010).

REZOLVAREA PROBLEMELOR DE STREAMING ȘI DE PROGRAMARE A PRODUCȚIEI: UN STUDIU DE CAZ DIN INDUSTRIA AUTOMOBILELOR

Rezumat: Această lucrare și-a propus să determine dacă merită depus efortul de integrare a problemei programării unui sistem de tip "job-shop" cu cea numita "lot streaming problem". Pentru a răspunde la întrebare a fost folosită o abordare în două etape. În primul rând, numărul și dimensiunea subloturilor au fost determinate cu o euristică simplă inspirată de practică. Programul ar fi apoi dezvoltat cu ajutorul unui model de programare cu constrângeri care a luat în considerare cererile provocatoare din partea industriei: timpi de configurare și schimbări în funcție de produs. Funcția obiectiv a vizat minimizarea duratei de pregătire-încheiere, a numărului de schimbări și a numărului de comenzi nelivrate la timp. Au fost definite 30 de instanțe de testare cu date preluate dintr-un sistem de tip "job-shop" de mari dimensiuni.

Radu-Constantin VLAD, Management and Economical Engineering Department, Technical University of Cluj-Napoca, Bdul Muncii no. 103-105, Cluj-Napoca, Romania, Radu.Constantin.Vlad@mis.utcluj.ro