DOI: 10.7763/IPEDR. 2012. V46. 7

Considering Complex Sequence Constraints in Production Scheduling – Results of a Practical Implementation in a German Trailer Company

Richard Lackes[1], Esther Awiszus[1]

[1] Department of Business Information Management, Technische Universität Dortmund
44221 Dortmund, Germany

Abstract. In this article we will consider the sequencing problem of a trailer company. The initial situation is the hand-made planning of the schedule for the final assembly. The purpose of our study is to develop a computer-aided way to solve the scheduling problem that is well known in the literature as the mixed-model assembly line sequencing problem. Unlike the assembly of cars, for which a lot of heuristic were developed, we have not only to deal with maintaining a constant usage rate of all parts on the line. We also have to consider sequence constraints. With a sequence constraint is meant that two units, unit one with a part i and unit two with a part j, should not be sequenced one after another. This kind of constraints is used to prevent line stoppages. We will present the formulation for these sequencing constraints and introduce a heuristic to solve the scheduling problem.

Keywords: Scheduling, Mixed-Model Assembly Line, Just-In-Time, Sequence Constraints

1. Introduction

Starting point of the analysed statement of problem was the sequencing planning for a company which produces trailers for trucks. Field of study was the final assembly where several types of trailers were built which is based on the principle of assembly line production [1].

Within the assembly process of freight trailers there are different stations. Unlike production systems with conveyor belts the stations in this system are not interlinked through any kind of continuous transport system. The various stages are arranged within the assembly sequence but the transport from one stage to another is done by cranes or other transportation systems. In the literature this production system is called system with closed stations [2]. The assembly process at the several stations is effected in cycle time. That means the assembling of a carrier at a certain station has to be accomplished within a certain time window. After the lapse of cycle time the carrier is moved to the next station. Time frame exceeding results work overload or a line stoppage. With the start of the assembly, after putting a vehicle on the assembly line, a rework is not possible. Vehicles, once put on the assembly line have to pass through all stations. Further, there are no buffers prior to the stations of the assembly line. After the cycle time has expired, each carrier is transferred to the next station.

To keep the storage costs at a minimum level, deliveries of materials are made on basis of the 'Just-In-Sequence' principle [3]. This principle is a further development of the 'Just-In-Time' principle [4]. In Addition to the precisely timed delivery, the precisely sequenced delivery of materials is necessary. Like in the JIT principle parts and materials are delivered as closely as possible to the time that they're required. The number of parts in stock corresponds to the amounts needed for a short production time. The supply takes place either by direct delivery by the external supplier or from external stores. The advantage of this method is the minimum use of store capacity for storage at the assembly line.

2. Problem Description

In the following we will deal with the problem of sequencing jobs in the examined production system.Each trailer that has to be assembled is called a job P_n. These jobs are several models which differ from each other with respect to used material or equipment. For every job P_n we will define later a binary variable $b_{n,k}$ that will show if a part k out of a fixed number of parts K should be assembled in a job or not.

The number of jobs that had to be assembled is fixed for one work day. That means that we had to find a schedule for a fixed set of jobs $P=\{P_1,P_2,...,P_N\}$.

The way the problem is solved at the trailer company is hand-made. Every day the production planner prints a list with the set of jobs and some of the associated parts for these jobs. Not all parts were regarded because the problem would become too large to handle with all components. Based on this list the planner first determines the quantity of each part. Afterwards for each part he estimates the number of jobs without this part that had to be assembled in the schedule between two jobs that have this part assembled, so called “part-rules”. There are also rules that are not related to the quantity of each part but we will explain later the meaning of these rules. With the list and the estimated and the fixed rules the planner begins to schedule the jobs. He starts taking one part and distributes the jobs that contain this part equally in the schedule. Afterwards he takes the next part and proceeds the same way. In the end he gets a schedule that he examined again to see if there are jobs that can be shifted to another position to get a “better-looking” schedule. The procedure takes minimum 30 minutes but at some days it could take up to one and a half hour. Therefore the company wanted to have a data processing-based method to deal with this problem.

The considered practical example is known as a mixed-model assembly line. This type of production assembles several variants of different models with lot size one [5]. The sequencing problem of this production is assigned to a short planning period and a fixed production program. We also have a fixed number and length of stations, a constant speed rate and a fixed rate launching. The jobs are "stationary" jobs and the set-up times are negligible. The working at each station starts as soon as possible and the number utility worker is sufficient like in the mixed-model assembly lines.

We also have to specify the objective of the problem. An often taken objective is the minimization of the costs that are caused by work overload. These costs consist of costs for utility worker, costs for postprocessing or line stoppage costs [1]. The previous literature considers these costs by trying to avoid schedules in which work intensive jobs follow each other. Another main cost factor is the storage costs. On purpose to minimize these kinds of costs the manufactures often use the Just-In-Time concept [4]. According to this concept the objective of the sequencing problem is to keep a constant part usage rate.

In literature the objective of minimizing these costs is pursued by the following approaches. In the "mixed-model sequencing" (MMS) the sequence-related work overload should be reduced by precisely timing the production of the parts, considering the processing time, station length and cycle times [2], [6], [7], [8] and [9]. "Car-sequencing" initiates sequence rules for single parts. For example that out of ten units a maximum of two units contains the same part. On the basis of these rules, sequences are developed. A sequence is considered as "good" when as little sequence rules as possible are violated [10], [11].

There is a third approach in the literature called "level-scheduling"[12]. Objective of this approach is to minor deviation of certain parameters developing from a constant developing. All required production factors can be used as parameters in this process, like parts/material [3], [15], [16], [17], workload or variants. However, the objective for the sequencing does not have to be necessarily based on one single parameter. There are procedures that combine more than one objective statement with each other [1], [13], [14].

3. Sequencing Constraints

As mentioned there exist different approaches for combining the different objectives by integrating both objectives in the objective function. The introduced extension of the problem specifications should be used to combine the objective of the constant part usage and the objective of the workload smoothing without integrating both objectives in the objective function. The purpose of considering both objectives is based by the attempt to prevent line stoppages and to increase the cost for additional work under the aim of keeping the part usage rate constant. In our case we only have sporadic excess of work at some stations which are caused by time intensive installation of some parts. Therefore be beneath the constant part usage rate the objective of the scheduling should be the not-succeeding of time intensive parts. To consider this objective we can use the idea of the car-sequencing. Unlike the car-sequencing rules our rules will prohibit the succession of some parts in the schedule.

We will now introduce sequencing constraints that formulates the rules. These constraints are based on parts and should prevent the overload of the stations. The formulation of these constraints are like "Directly after part k part l should not be assembled" or "Two parts k are not allowed to follow each other in the schedule" or "Between the assembling of part k and the assembling of l there must be two jobs in the schedule without one of this parts". Such sequencing rules $R=\{R_1, ..., R_H\}$ could be specified as follows :

The rule $R_h=(C_{1h}, C_{2h}, s_h)$, h=1,...,H, consists of the conjunctive concept $C_{ih}=(k_{ih_1}, k_{ih_2}, \dots, k_{ih_{Q_i}})$ with $Q_i \geq 1$ parts k_{ih_q}, $q \in \{1,..., Q_i\}$ with i=1,2 and a number $s_h \in \aleph$. This number is the minimal distance between the jobs that are specified by the concepts C_{1h} and C_{2h}. Here the job that is described by the first concept is scheduled before the job that is described by the second concept.

4. Mathematical Formulation

To formulate the mathematical problem we will use two binary variables. The first variable is $b_{n,k}$ which declares if part k is assembled in job P_n. The second variable we need is $x_{n,m}$ which declares if job P_n is sequenced on position m.

We also have to consider some constraints. Every position of the schedule exists only once:

$$\sum_{n=1}^{N} x_{n,m}=1 \quad \text{for } m=1,...,N$$

and every job is scheduled exactly one time

$$\sum_{n=1}^{N} x_{n,m}=1 \quad \text{for } n=1,...,N$$

We also have a binary constraint

$$x_{n,m} \in \{0,1\} \text{ for } n,m=1,...,N$$

For the sequencing constraints we have to make sure that every rule R_h is not violated at every position of the schedule. By the concepts that are formulated in the rules the parts that are assembled in the jobs are specified, so that within a matching with the part-descriptions of the jobs in the vector b can be calculated, if the concerning job satisfy the concept or not. The multiplication of the binary value of the parts of a job in the concept is one, if the job includes all the parts of the concept. The verification of the rules is done for every position at which we have to keep in mind the minimal distance that is specified in the rule. So we get:

$$x_{n,m} \cdot \prod_{k \in C_{2h}} b_{n,k} + x_{p,m-s'} \cdot \prod_{k \in C_{1h}} b_{p,k} \leq 1$$

for all n,p=1,...,N, m=2,...,N, h=1,...,H and $s'=1,...,\min(m-1,s_h)$ for all rules $R_h = (C_{1h}, C_{2h}, s_h)$, h = 1,..., H with $C_{ih}=(k_{ih_1}, k_{ih_2}, \dots, k_{ih_{Q_i}})$, i=1,2

Like mentioned before both objectives, keeping a constant part usage rate and maintain a constant workload, should be considered. We consider the workload smoothing by using the sequencing constraints. The objective of keeping a constant part usage rate we have to consider in the objective function. As objective function we take the square deviation between the actually cumulative part usage and the constant part usage rate. We have to minimize this function. So the objective function is as follows:

$$\min \sum_{m=1}^{N} \sum_{k=1}^{K} \left(\sum_{m'=1}^{m} \sum_{n=1}^{N} x_{n,m'} \cdot b_{n,k} - m \cdot \frac{\sum_{n=1}^{N} b_{n,k}}{N} \right)^2$$

With: $\sum_{m'=1}^{m} \sum_{n=1}^{N} x_{n,m'} \cdot b_{n,k}$ = cumulative quantity of part k, k=1,...,K required at position m

$m \cdot \frac{\sum_{n=1}^{N} b_{n,k}}{N}$ = consumption rate of part k, k=1,...,K with smoothed usage for position m

5. Sequencing Procedure

The procedure we will now introduce is based on a simple greedy heuristic. This heuristic estimates priority values to schedule jobs. Such a heuristic consists of three steps for every position of the schedule. After the initialization we have to calculate the priority values for every job $P_n \in P$ that is not scheduled yet. In the next step we select the job P_{n*} with the "best" priority value and assemble this job as the next element in the schedule. We also have to erase the job out of the list of not scheduled jobs P, e.g. P:=P\{P_{n*}}. At least we have to prove if there are still jobs left to schedule, e.g. we will prove if P$\neq\emptyset$. If there are jobs that are not scheduled yet, we will continue with the first step.

Altogether we will introduce two priority values that are well-known in the literature. The first value (GC) is known from the goal-chasing methode [18]. For every position m and for every not yet scheduled job P_{n*} we have to estimate the value:

$$\sum_{k=1}^{K}\left(\sum_{m^*=1}^{m-1}\sum_{n=1}^{N} x_{n,m^*}\cdot b_{n,k} - m\cdot\frac{\sum_{n=1}^{N} b_{n,k}}{N} + b_{n^*,k}\right)^2$$

After we have calculated every priority value we will schedule the job with the minimal value as the next job. We choose the job that will increase the objective function at a minimum.

To compensate the myopic of this procedure we will test another version (GCN). For every position of the schedule we will not only look at the possible jobs for the actual position, we will also consider the best job for the next position. This means that for every position of the schedule we will choose the job that has the "best" priority value together with his next "best" neighbour. For a P_{n*} from all not yet scheduled jobs P the formula for the priority value is the following one:

$$\sum_{k=1}^{K}\left(\sum_{m^*=1}^{m-1}\sum_{n=1}^{N} x_{n,m^*}\cdot b_{n,k} - m\cdot\frac{\sum_{n=1}^{N} b_{n,k}}{N} + b_{n^*,k}\right)^2$$
$$+\min_{P_{n'}\in P\setminus\{P_{n^*}\}}\left\{\sum_{k=1}^{K}\left(\sum_{m^*=1}^{m-1}\sum_{n=1}^{N} x_{n,m^*}\cdot b_{n,k} - (m+1)\cdot\frac{\sum_{n=1}^{N} b_{n,k}}{N} + b_{n^*,k} + b_{n',k}\right)^2\right\}$$

Other priority values can be found in [3] and [12].

We have to extend the greedy heuristic to consider the sequencing rules. We will examine if the sequencing rules are violated in an existing schedule. At first we make a schedule without considering the sequencing constraints. Afterwards we have to exam for every position of the schedule, if there are any sequencing constraints violated. If so, we try to repair the violation by trying to find a new position for the violation causing job without causing any other violation. If we found more than one position, we take the position that leads to a minimal increasing of the objective function. We do not find necessarily a feasible schedule. If no other position can be found for a violation causing job, it stays on its original position and we get a not feasible schedule. But if we found an alternative position we have to check if the new position of the job is before or after the original position. In case of a position before the old position we have to exam the schedule at the next position after the original position. Otherwise we have to start at the original position with the examination.

6. Practical Example

We will now introduce the sequencing rules of the practical problem and explain the causes for these rules. The sequencing constraints are related to parts we will call in the following DOSB, DOS2, ANKE, TRW and, TRW2.

Trailers with the part DOS2 we have to consider that the before scheduled trailers don't have the parts DOS2, DOSB or ANKE. The corresponding sequencing rules are R_1=(DOS2, DOS2, 1), R_2=(DOSB, DOS2, 1) and R_3=(ANKE, DOS2, 1). Trailers who have simultaneous the parts DOSB and ANKE are not allowed to be scheduled after the parts DOSB, DOS2 and ANKE (R_4=(DOSB, (DOSB,ANKE), 1), R_5=(DOS2, (DOSB,ANKE), 1) and R_6=(ANKE, (DOSB,ANKE), 1)). These constraints are caused by the production stage where the parts are produced. This stage is upstream the final assembly and is not regarded in the scheduling procedure. The consecutive production of these parts will cause time delays that will be conveyed to the final assembly.

The parts TRW and TRW2 are dividers. The dividers are putted into the trailer by hinging up. This is a time intensive method because before one can hinge up the dividers he has to affix the divider to a crane. To prevent line stoppages, trailers with dividers are not allowed to follow each other in the schedule. The rules for these constraints are R_7=(TRW, TRW, 1), R_8=(TRW, TRW2, 1), R_9=(TRW2, TRW,1) and R_{10}=(TRW2, TRW2, 1).

7. Testing and Conclusion

Altogether we had 35 real world datasets for testing the scheduling procedure. These datasets corresponds the production program of 35 working days and contains between 39 and 64 jobs a day. The number of parts that should be varied was between 20 and 32 parts. We tested the two priority rules without and with the described repair-algorithm. Without using the repair-algorithm we only get four feasible schedules for each priority rule. But when the repair-algorithm is used, we get feasible schedules for all test cases. We should also mention that the results improve when we consider the next best neighbour in our calculation of the priority rule. The average percent deviation from the best objective function value with the priority value GC was 1.08 (1.29 with repair) and 1.02 (1.25) for the priority value GCN.

As we see in the testing results, we developed a procedure that handled the sequencing problem of the trailer company. We also have to mention that the procedure can handle more parts in the scheduling algorithm than the previous used procedure. The hand-made procedure could handle only a small number of parts (not more than ten different parts), our procedure had no problem with more than 30 different parts. Also the time needed to determine a schedule increases with our procedure. As mentioned the production planner needs more than 30 Minutes to schedule the jobs in best case. The procedure needs less than a minute to get a feasible schedule.

8. References

[1] R. Wild. *Mass production management - the design and operation of production flow-line systems*. Wiley, 1972

[2] J. F. Bard, E. Dar-El, A. Shtub. An analytic framework for sequencing mixed model assembly lines, *International Journal of Production Research* 1992, 30: 35-48

[3] Y. Monden. *Toyota Production System*. Engineering & Management Press, 2002

[4] R. Lackes. *Just-in-Time-Produktion*. Gabler Verlag, 1995

[5] N. Boysen, M. Fliedner, A. Scholl. Produktionsplanung bei Varianten-Fließfertigung: Planungshierarchie und Hierarchische Planung. *Zeitschrift für Betriebswirtschaft* 2007, 77: 759-793

[6] N. T. Thomopoulos. Line balancing - sequencing for mixed-model assembly. *Man. Science* 1967, 14: B59-B75

[7] N. Boysen, M. Kiel, A. Scholl. Sequencing mixed-model assembly lines to minimise the number of work overload situations. *In. Journal of Production Research* 2011, 49: 4735-4760

[8] J. Bautista, A. Cano. Solving mixed model sequencing problem in assembly lines with serial workstations with work overload minimization and interruption rules. *Europ. Journal of Operational Research* 2011, 210: 495-513

[9] L.-H. Tsai. Mixed-model sequencing to minimize utility work and the risk of conveyor stoppage. *Man. Sci* 1995, 41: 485-495

[10] M. Gravel, C. Gagne, W. L. Price. Review and comparison of three methods for the solution of the car sequencing problem. *Journal of the Operational Research Society* 2005, 56: 1287-1295

[11] J. Bautista, J. Pereira, B. Adenso-Diaz. A beam search approach for the optimization version of the car sequencing problem. *Annals of Operations Research* 2008. 159: 233-244

[12] N. Boysen, M. Fliedner, A. Scholl. Sequencing mixed-model assembly lines: Survey, classification and model. *European Journal of Operational Research* 2009, 192: 349-373

[13] W. Zeramdini, H. Aigbedo, Y. Monden. Bicriteria sequencing for just-in-time mixed-model assembly lines. *International Journal of Production Research* 2000, 38: 3451-3470

[14] O. S. Akgündüz, S. Tunali. An adaptive genetic algorithm approach for the mixed-model assembly line sequencing problem. *International Journal of Production Research* 2010, 48: 5157-5179

[15] J. Miltenburg. Level schedules for mixed-model assembly lines in just-in-time production systems. *Man. Sci.* 1989, 35: 192-207

[16] F.-Y. Ding, L. Cheng. An effective mixed-model assembly line sequencing heuristic for just-in-time production systems. *Journal of Operations Management* 1993, 11: 45-50

[17] A. Scholl, R. Klein, W. Domschke. Pattern based vocabulary building for effectively sequencing mixed-model assembly lines. *Journal of Heuristics* 1998, 4: 359-381

[18] R. Sumichrast, R. S. Russell, B. W. Taylor. A comparitive analysis of sequencing procedures for mixed-model assembly lines in a just-in-time production system. *Int. Journal of Production Research* 1992, 30: 199-214

[19] E. A. Duplaga, D. J. Bragg. Mixed-model assembly line sequencing heuristics for smoothing component parts usage: a comparative analysis. *In. Journal of Production Research* 1998, 36: 2209-2224