

Modeling and Solving an Extended Parallel Resource Scheduling Problem in the Automotive Industry

Mónika Kulcsár-Forrai, Gyula Kulcsár

Department of Information Engineering, Faculty of Mechanical Engineering and Informatics, University of Miskolc, 3515, Miskolc-Egyetemváros, Hungary,
aitkfm@uni-miskolc.hu, iitkgy@uni-miskolc.hu

Abstract: This paper presents an extended model for solving time-varying resource-constrained scheduling problems. The motivation for our research comes from the automotive industry. The problem is to create fine schedules for a complex manufacturing system to satisfy diverse customer demands. The detailed characteristics of the analyzed scheduling problem and the developed solving approach are described in this paper. To consider the impact of the assistant processes that are connected to the manufacturing primary processes, we elaborated a problem-transformation procedure and a new extended scheduling model that can manage time-varying availability constraints of parallel resources, unit processing times, job-dependent release times and due dates. This paper also presents slack-oriented and JIT-oriented algorithms that can solve the resource-constrained scheduling problems. The research results have been successfully applied and tested in practice.

Keywords: scheduling; resource availability constraint; multi-objective optimization; production planning and control; manufacturing operations management

1 Introduction

Production planning and scheduling systems deal with the allocation of limited resources to production activities to satisfy customer demands over the actual time horizon. Planning and scheduling tasks can be expressed as optimization problems, in which the main goal is to create plans that meet constraints and maximize production performance. These optimization models are very different in practice, corresponding to the characteristics of the real production systems and their business environments.

A hierarchical approach is one of the possible ways for solving such planning and scheduling problems. The hierarchical optimization approach means that the

decision process will run in a layered way by ordering the decisions according to their relative importance. This decomposition technique uses a suitable optimization model at each level of the hierarchical decision-structure. At a given level the applied model extends the decision variables, the constraints and the objective functions of the problem arriving from the upper level.

Production planning and scheduling process typically works according to the rolling horizon principle. This means that an initial plan is created for the actual time horizon, and its first part is executed. The system creates the plan for the next period by considering the previous partially overlapped period in advance. Then, the initial plan may be modified or re-planned by considering the changes and disturbances. The actual plan can also be periodically revised due to the uncertainties that may occur in the business and in the production processes.

Results of production planning and scheduling are usually not applicable to managing operational manufacturing since the created plans are rough and large-scale solutions, and they refer to aggregate resources. The role of fine scheduling (detailed scheduling) is to make a precise executive fine program for a short time horizon (for weeks/days/shifts) that concerns every detail.

To realize the created production fine schedule in practice, the complex decision making has to cover the primary processes and also the most important supplementary (e.g. logistical) and assistant processes (e.g. instrument supply) of the production.

In this paper, we present fine scheduling models and algorithms for solving real-life problems in the automotive industry. We focus on modeling and solving the scheduling problems of vehicle seat element manufacturing. To solve the fine scheduling problem, not only the main manufacturing processes, but the configuration-preparation processes have to be considered.

This paper is organised as follows: Section 2 briefly describes the examined production system. Section 3 reviews fundamental models for production scheduling, while Section 4 introduces a new solution for fine scheduling. Section 5 proposes an extended parallel resource scheduling model and new solving algorithms, and Section 6 shows an application of the theoretical results in practice. Finally, conclusions are given in the last section.

2 The Examined Production System

In the examined vehicle manufacturer workshop, seat elements are made for different types of cars. The customers (vehicle-assembly enterprises) generate product demands (orders) for specified product types and numbers of items. These production orders have to be fulfilled within strictly prescribed time limits.

The plant manufactures the seat elements (end products) on circle-shaped manufacturing systems. The manufacturing systems of the plant complete the orders together. It is possible to produce the specified product type on more than one path. Every single path can perform a given amount of rounds (cycles) in one shift, furthermore it possesses a specified number of attachment points (positions). There are shape carriers in the system. A given type of shape carrier can be connected to a given position of a specified path. The shape carrier can be one or two sided depending on its formation. It is possible to attach tools (molds) to the left and/or right side of the shape carrier. The attachments are determined by technological rules. Strict rules prescribe:

- what kind of products,
- on which path,
- in what kind of position,
- on what type of shape carrier,
- on which side and
- with what kind of other products they can be manufactured.

A path can be considered as a production line that works according to an independently defined calendar. The basic unit of the calendar is the shift. These basic working time intervals are equal (e.g. eight hours) long. In every shift the production lines can be adjusted. This means that it is possible to exchange a given number of shape carriers which are connected to the positions. The unit of one exchange is a given configuration that consists of one carrier and its connected mold(s). A complete exchange happens when we take the current configuration from the position and attach another prepared configuration to the same position.

The constructions of the production lines (paths) are different, so the total number of executable rounds in one shift can differ. In the system, various numbers of molds and carriers are usually available for manufacturing products.

The preparation (assembly and disassembly) of the tool configuration, which is needed for the production, is carried out by skilled workers. The preparation task is time consuming. Therefore, the number of the performable configuration preparations during one shift in the plant is restricted by tight capacity constraints.

3 Fundamental Models for Production Scheduling

Discrete manufacturing processes include a large variety of diverse and very distinct technologies that require specific models when creating or improving their efficient control systems. In this regard, there is a special demand for adequate modeling, formulation and solving of scheduling problems [20].

In the professional literature, many books and papers deal with scheduling models and methods. There are well-structured books that focus on manufacturing scheduling (e.g. [16], [17], and [21]). Several review and survey papers can also be found on this topic, for example [1], [5], and [11]. Scheduling plays an important role not only in manufacturing but in many different service industries. Pinedo et al. [18] presented an overview of some of the more important scheduling problems that appear in various service industries.

The scheduling problems lead to optimization tasks. Their complete review and classification exceed the scope of this paper; there is a vast amount of literature that deals with such problems. Therefore, only some of the most important solving approaches are shortly mentioned. The main categories are as follows:

- Mathematical programming approach (e.g. linear, nonlinear, integer programming, disjunctive programming, set assigning, set partitioning, set packing and set covering, etc.).
- Exact optimization approach (e.g. branch and bound methods, dynamic programming, etc.).
- Constraint programming approach (e.g. constraint satisfaction and constraint programming).
- Heuristic approach (e.g. basic scheduling rules and composite dispatching rules, etc.).
- Iterative improvement approach (e.g. beam search, local search and genetic algorithms, etc.).

To solve a scheduling problem in practice, we have to deal with at least three important issues. The first issue is the resource environment. This takes into consideration all features of the resources concerned (machines, workplaces, workers, tools, etc.) and characteristics of the relations among them. We also have to pay special attention to the features of the operations to be executed. The second issue focuses on the job characteristics and constraints. This group includes all the technological rules, manufacturing restrictions, job execution features and alternative process plans. The third issue is the production control policy that specifies the priorities, requirements, objective functions and key performance indices. The possible variants of these groups of issues result in many scheduling problems.

The simplest resource environment of scheduling problems is represented by the single machine type model, which refers to jobs containing one operation to be performed. If the single resource is replaced by a given set or group of resources, the scheduling problems become models of parallel resources. In this case, the jobs are executed simultaneously on different resources (machines or workplaces). Depending on the working resources capability of the parallel models, three basic resource environments are categorized [16], [21]. The first is the model of identical parallel machines, where each job can be processed on any machine, and

the processing time only depends on the dedicated job. The second variant is the model of uniform parallel machines, where the processing time of a dedicated job varies according to the machine speed. The third variant is the model of unrelated parallel machines, where the processing time varies according to the job and the machine. In this case, each machine can work at different speeds on the jobs. Several papers give detailed reviews on parallel machine scheduling, for example [4], [7], and [22]. However, Weng et al. [19] and later Lamothe et al. [13] have shown that the scheduling problems on machines with limited flexibility, setup and secondary resource constraints were poorly studied.

The shop scheduling models involve more machines and more jobs containing more than one operation. However, each operation can be performed on a given machine. According to different prescriptions, specific models can be formed from this general shop, as follows [21]:

- Job shop: the set of operations is job dependent, and each job may have a special precedence chain relation (operation sequence).
- Flow shop: a special case of the job shop in which the number and sequence of operations is fixed for any job.
- Open shop: a special case of the general shop in which there are no precedence relations between the operations.
- Mixed shop: a combination of the above models.

In the above-enumerated models, each operation can be assigned to a given machine (dedicated resource). A further extension of the general shop model is the flexible shop model, where the flexibility feature refers to the machine assignment possibilities. In the flexible models (e.g. flexible flow shop and flexible job shop, etc.) a given operation can be performed on any of the machines of a specified machine group. In this way, the scheduling problem is supplemented with machine selection tasks. The suitable machines from the group can simultaneously work identically or uniformly or even in an unrelated way [6], [23]. These models can be considered as the combined models of the shop models and parallel machine models.

The extended flexible shop models represent a new generation of the scheduling problem class (e.g. extended flexible flow shop and extended flexible job shop, etc.). It often occurs in the manufacturing systems that there are some resource objects (e.g. integrated production lines and cells) that can perform more than one operation as a unit. In this case, some operations can be grouped into larger units such as technological steps or even execution steps. These collecting steps can be considered as basic units for scheduling [8], [9], and [15]. The extended flexible shop models support the usage of alternative technological routings and resources. Parallel machine models are the functional building blocks of these extended models to support the parallel realization of the execution steps.

For applying scheduling models in practice, it is very important to consider the machine eligibility constraints. For example, Lin and Li [24] studied the parallel machine scheduling problem with unit-length jobs in which each job is only allowed to be processed on a specified subset of machines. Many other researchers also have considered the machine eligibility constraints in parallel machine scheduling problems. Lee et al. [14] studied the most general case of machine eligibility constraints as well as special cases of nested and inclusive eligible sets.

From the practice-oriented point of view, flexible scheduling approaches have to pay special attention to the job execution constraints related to the release times and due dates. As is well known, the general case of this problem with only a single machine is NP-hard if the optimization objective is to minimize the maximum lateness. The NP-hard (non-deterministic polynomial hard) property indicates that there is probably no polynomial-time algorithm to reach the optimal solution of the problem. (The precise definition of the NP-hard problem is given in [21].) Consequently, the parallel and extended variants of the problems are also NP-hard. Special cases of the problem have been analyzed in recent years. For example, Lazarev et al. [2] considered the problem with only a single machine and identical processing times for all jobs.

In most of the scheduling problems, it is assumed that the resources (machines) are continually available in time, and the number of resources is fixed in each problem instance. These simple types of availability constraints can only limit the number of parallel executable tasks. One other type of availability constraint is connected to calendar elements or shifts. The resource availability constraints specify the time intervals or windows in which the resources can perform the jobs. Ma et al. [25] provided a detailed review of this topic. Kaabi and Harrath [12] reviewed the models and results related to the parallel machine scheduling problem under availability constraints. Nevertheless, it can be seen that parallel resource scheduling problems with due-date related objective functions, resource-availability constraints, distinct release times and due dates are poorly studied in the literature.

Our problem has similarities to the problem addressed by Gharbi and Haouari [3] in the sense that both models have parallel resources, distinct release times and due dates. However, in our model, each resource has its own list of availability time intervals, not only a single availability time window, and the objective functions are also different. Brucker [21] presented the model $P \mid p_i=1; r_i \text{ integer} \mid L_{max}$. This is similar to our scheduling model for configuration preparation in that Brucker's model also uses unit processing time, integer release times and due dates, but we even take into account the distinct availability time constraints of each machine.

In this paper, we study a special parallel resource scheduling problem in which time-varying resource constraints, unit processing times, job-dependent release times and due dates are considered to minimize the maximal tardiness and

earliness. In the literature we have not found any research papers considering this set of constraints and objectives.

4 A New Solution for Production Fine Scheduling

The product-type dependent production intensity of the examined manufacturing system can only be modified slowly because of the defined constraints. Therefore, the service of diverse production orders generates a serious production fine scheduling problem. Based on the necessary information the fine schedule is usually created for a one-week time interval in advance. It is recommended to keep product dependent stock levels adjusted individually due to the long reaction time of the manufacturing system.

During the development, we focused on heuristic and knowledge-intensive searching techniques because the addressed full scheduling problem is NP-hard. At the beginning of the development we started out from our previous models applied successfully in other situations [10].

In our approach, all the issues (batching, assigning, sequencing and timing) are handled simultaneously (Figure 1). The values of the decision variables of the problem are set by a multi-operator and multi-objective searching algorithm. The developed scheduling software modifies the actual schedule iteratively and prepares new solutions with consistent changes (modifications) based on multiple neighboring operators, execution-driven fast simulation, and overloaded relational operators.

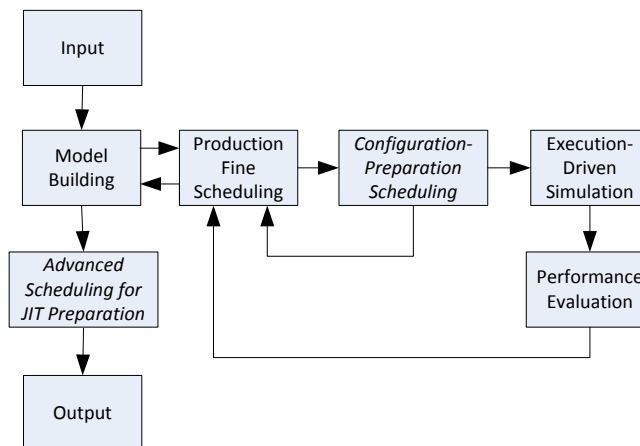


Figure 1

Simplified flow chart of the integrated scheduling approach

Based on the given input data, the model builder component defines the required model objects and initializes them with start values (attributes). Its tasks include the creation of the starting status of resources (paths, positions, carriers, molds), the specification of the internal production orders, and furthermore the definition of the restrictions and objective functions.

After carrying out the availability, applicability, and feasibility studies on the created object model, the builder process defines the currently indexed relationships of the entire system. Using these indices in each decision-making situation (e.g. assignment, selection) of the solving process, the alternatives of choices can easily be retrieved, so the chain of decision making produces feasible solutions.

The core of the implemented solver explores iteratively the space of the feasible solutions and creates neighbor candidate solutions by modifying the decision variables of the fine schedule according to the problem space characteristics. For each shift, the fine schedule specifies the configuration of shape carriers and molds to be run and the kinds of products to be manufactured in each position of each production line.

The candidate schedules are simulated by using an execution-driven fast simulation algorithm that represents the real-world environment with capacity and technological constraints. In this execution-driven simulation, the product units are passive, and they are processed, moved and stored by active system resources such as production lines, material handling devices, and buffers. The numerical tracking of the entities (product units, shape carriers and tools) provides detailed data of the manufacturing.

By using the results of the simulation, the actual values of the key performance indicators (KPIs) can be calculated. To express the shop floor management's goals, we use the following objective functions to be minimized in a multi-objective optimization problem:

- The maximum product shortage at the due dates of the production orders;
- The sum of product shortages at the due dates of the production orders;
- The number of tardy production orders;
- The number of set-up activities;
- The maximum number of set-up activities in one shift;
- The number of product types with surplus;
- The sum of product surpluses;
- The maximum product shortage at the end of the time horizon;
- The sum of product shortages at the end of the time horizon;
- The sum of the priorities of tardy production orders;

- The maximum priority of tardy production orders;
- The number of product types with tardiness;
- The maximum product shortage (compared to zero);
- The sum of product shortages (compared to zero);
- The maximum tardiness of production orders;
- The sum of the tardiness of production orders;
- The number of configuration preparations;
- The sum of unused capacity of the production lines.

Every objective function has dynamically changing importance and sets of values to be taken into consideration. The relative quantification of the currently examined solution can be performed by comparing it to the best solution found so far. The mathematical model of this qualification was described in [8].

In every iteration of the searching algorithm, the actual solution has to be examined to decide whether the candidate fine schedule is feasible from the point of view of configuration preparation. Each candidate fine schedule requires well-defined preparatory activities for the tool configurations to be used. These tasks have to be scheduled on time-varying capacity-constrained parallel resources (skilled workers) to achieve zero tardiness. If this is possible, then the production fine schedule is feasible, otherwise it cannot be executed. Successful adaptation of this approach to practice is highly influenced by the efficiency of the solving algorithm of this sub-problem.

When the final solution of the production fine scheduling is being prepared, it is very important to pay special attention to the robustness of the configuration preparations. In the searching iterations, the algorithm focuses on the minimization of the maximal lateness. In the end, the final solution can be made more sophisticated by using an advanced JIT-oriented scheduling algorithm that adjusts the release time of the preparatory tasks to be as close as possible to the due dates, considering the necessary preparation times and the prescribed safety time intervals.

The complexity of the concrete industrial scheduling problem is described in Section 2. The full (complete) production scheduling problem is handled by using an advanced multi-objective and multi-operator searching algorithm. In each iteration of the searching algorithm, the built-in scheduling sub-problem of the configuration preparations has to be solved. These two scheduling problems constitute a two-level decision hierarchy, in which each level uses its own specific optimization model. The solution of the problem created at the higher level gives input data, constraints and criteria to the lower level. The given set of constraints is extended with the new constraints of the built-in problem at the lower level of the hierarchy.

The built-in sub-problem is a special scheduling problem. The “job” means one preparatory task at the lower level. We have to schedule the preparatory tasks (assembly the configurations of part-adequate tools and shape carriers) required by manufacturing primary (main) processes. The current production schedule (the solution of the overall problem) generates dynamically the jobs to be scheduled and their release times and due dates for the built-in sub-problem (a given set of configurations is used on the production lines and then a given subset has to be re-assembled by due dates). In the following parts of the paper, we focus on modeling and solving this sub-problem.

5 Modeling and Solving the Extended Parallel Resource Scheduling Problem of Configuration Preparations

5.1 The Description of the Problem

The scheduling problem of the configuration-preparations (jobs) can be summarized as follows:

- There are n jobs J_i ($i=1, 2, \dots, n$). They are independent of each other and bound by the earliest starting and the latest completion times.
- The availability time intervals of the resources are defined by a calendar. This calendar consists of lists. Each resource has its own list, which is built up from time intervals. Each interval means a shift. The shifts do not overlap, and they are sorted by the starting times.
- The sets of available resources are classified by the shifts. Each set consists of uniform parallel resources. The resources mean skilled workers, who are able to prepare a prescribed number of configurations separately in shifts.
- The shift calendar is an input data structure and the maximal number of the executable preparatory tasks (jobs) can differ according to the calendar.
- The goal is to create a schedule to minimize the maximal tardiness of the configuration preparations by keeping the restrictions.

5.2 Problem Transformation

The problem described in the previous section is difficult to solve in its original form. Many papers can be found in the literature on parallel machine scheduling, but there is no suitable model for the examined time-varying resource availability features of the current problem. Therefore, we elaborated a problem-transformation procedure. Using this procedure we transform the problem to an advanced parallel machine scheduling problem, in which the actual number of the available machines depends on the time.

The essence of the transformation is the following:

- We give serial numbers in the form of decimal integers to the shifts in the global system joint to the plant. These serial numbers are called slots (s). The slots create a connected series that replaces the time axis. The last slot is denoted by s_{max} .
- The processing time of the configuration preparation cannot be longer than one shift. This fact comes from the applied technology. Each processing time takes one (unit value) slot ($p_i = 1$).
- The time data of the jobs are also transformed to slots: the earliest starting time is converted into the serial number of the next shift (r_i), the due date is converted into the serial number of the target shift (d_i), and we search for the completion time in a form that expresses the serial number of the assigned shift (C_i).
- The lateness of the job is also measured in slots: $L_i = C_i - d_i$. The tardiness is also calculated in slots: $T_i = \max(0, L_i)$.
- The group of assembly workers changes with the set of parallel virtual machines (resources). Each virtual machine can work on one job at one time, and each job can only be executed on one virtual machine at a particular time. The number of the available virtual machines can differ according to the slot. This time-variable number of machines is denoted by $P(s)$. The original limitation in the number of executable jobs in the shift gives the concrete number of the available virtual machines in the specific slot.

The $P(s)$ values (limits) define the time-varying resource availability constraints. For example, the limit is 3 in the second shift, and 5 in the seventh shift. In the transformed model, these constraints define the number $P(s)$ of virtual machines (special skilled workers) in slot s , so each job has a unit processing time (equal to the shift length) without loss of generality. For example, one worker team performs three jobs, or there are three workers and each worker performs separately one job.

Based on the formal description $\alpha \mid \beta \mid \gamma$ commonly used in the literature (e.g. [21]) and the symbols introduced above, the new scheduling problem can be formalized as follows:

$$P(s) \mid p_i = 1; r_i = \text{integer}; d_i = \text{integer} \mid L_{\max} \quad (1)$$

5.3 Slack-oriented Solving Algorithm

For solving the transformed scheduling problem of the configuration preparations (1), we developed a relatively simple algorithm that gives an optimal solution (Figure 2).

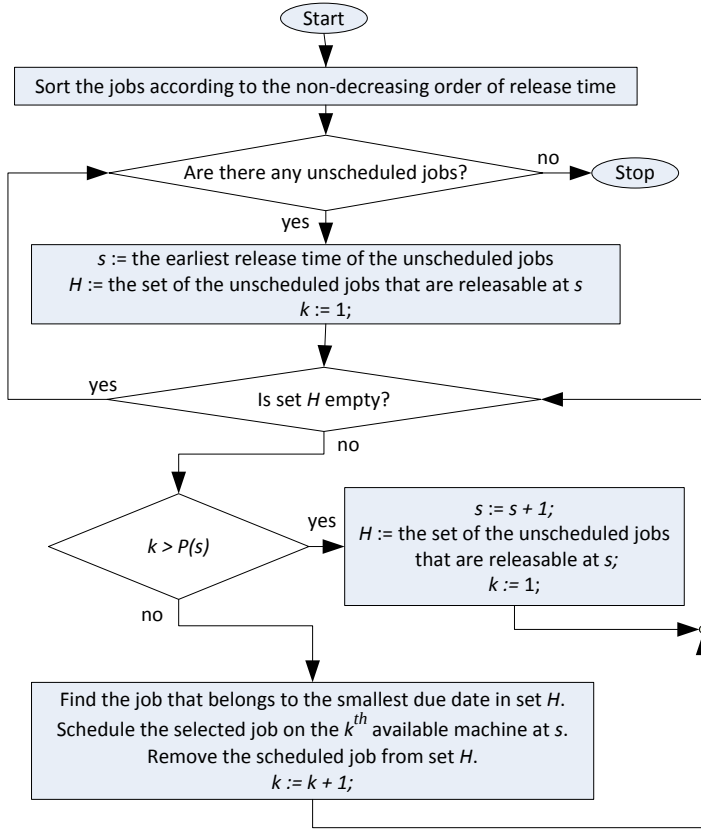


Figure 2

Simplified flow chart of the slack-oriented solving algorithm

The key element of the solving algorithm is that we store the actual number $P(s)$ of the available parallel machines in each slot and schedule the releasable jobs according to the well-known SDS (Smallest Dynamic Slack first) rule. This rule

selects the releasable job that has the smallest dynamic slack time. The slack of a given job is equal to the due date minus the sum of the actual time and the remaining processing time: $slack_i = \max(d_i - p_i - t, 0)$, where t denotes the actual time of the decision. As the processing time p_i of all jobs is one unit time, in each situation the job with the smallest dynamic slack can be achieved by using the EDD (Earliest Due Date first) rule. This means that if we can find a free machine at an intermediate slot s and there are at least one unscheduled and releasable job, then we schedule the job J_i with the nearest due date d_i .

The proposed algorithm creates a solution that minimize the maximum lateness. It produces an optimal solution in polynomial running time. If an ordered input data structure is used for jobs ($r_1 \leq r_2 \leq \dots \leq r_n$), the algorithm runs in $O(n \log n)$ time.

If the maximum lateness (L_{max}) is not greater than zero, then the configuration-preparation schedule is able to fully serve the execution requirements of the examined production fine schedule. The algorithm calculates the value C_i and thereby creates the required solution (schedule) at the same time. The job J_i (configuration preparation) must be performed in the shift of slot C_i .

5.4 Optimality of the Slack-Oriented Algorithm

To prove the optimality of the presented algorithm, we show that all optimal solutions can be transformed into the result of the slack-oriented algorithm by retaining the optimal value of the objective function.

Each job has a serial number (index i) in order of release times ($r_1 \leq r_2 \leq \dots \leq r_n$). Let S_a be the schedule created by the slack-oriented algorithm. Let S_b be an optimal schedule. Two vectors store the completion time of the jobs according to the two solutions (S_a and S_b). Let index x be the smallest job index where the C_x value (completion time) is different in S_a and S_b . This means that the first $x-1$ jobs are processed in the same slot according to S_a and S_b . We suppose that the value of index x is maximal because our assumption covers the relationship between these two solutions.

Therefore, the job J_x is carried out in slot C_x according to S_a while the same job J_x is processed in a later slot according to S_b . In this situation, two possible cases may be distinguished.

In the first case, a given machine is free in slot C_x according to S_b . Therefore, the job J_x can be moved to slot C_x on the free machine. This modification does not spoil the maximal lateness because the target job starts earlier, so the schedule S_b remains optimal.

In the second case, there is no free machine in slot C_x according to S_b . Therefore, there exists a job J_y that is carried out in slot C_x according to S_b , but the same job J_y is carried out in a later slot according to S_a . The due date d_y of job J_y is greater

than or equal to the due date d_x of job J_x . This follows from the fact that the slack-oriented algorithm scheduled the job J_x in an earlier slot than the job J_y according to S_a . As the schedule S_b is optimal, we can interchange J_x and J_y in schedule S_b , and the value of the objective function does not increase.

In both possible cases, the modified schedule S_b remains optimal, and the value x has increased after modification. This result contradicts the assumption that value x is maximum, so the modification can continue until the index x is greater than the number of jobs. Finally, each job J_i is completed in the same slot C_i according to schedule S_a and S_b . This proves that the slack-oriented algorithm creates an optimal solution.

5.5 Advanced Scheduling for Just-In-Time Preparation

In practice, one very important expectation of production management is that the manufacturing control system can fulfill the execution conditions of the released production fine schedule according to the paradigm “Just-In-Time”. For example, the needed tools should not be prepared too early, because if some kind of uncertainty or unexpected events occur between the preparation completion time and the actual starting time of the inducing operation, then the released production fine schedule has to be modified, so some preparatory activities will have been done unnecessarily. Consequently, it is appropriate to schedule the implementation of each preparatory task as close as possible to the starting time of the related operation by taking into account the necessary preparation time.

In scheduling the configuration-preparation tasks, the presented slack-oriented algorithm is focused on the minimization of the maximum lateness. If there is a solution in which the maximum lateness is not greater than zero, then we can further refine the schedule so that the JIT principle can be validated. For this purpose, we developed an advanced algorithm that is able to reduce the maximal earliness without violating the due dates. The earliness of the job J_i is measured in slots: $E_i = \max(0, -L_i)$.

The essence of the JIT-oriented algorithm is the following:

1. Start from the schedule S_a generated by the slack-oriented algorithm. Create an LDD (Latest Due Date) list by sorting the jobs in the non-increasing order of due dates.
2. Select the first job from the LDD list.
3. Examine the loading of the slots. In order to find a free machine in a suitable slot, start from the due date of the selected job and go backward in time (slot by slot) until a free machine is found or the original slot of the selected job is reached.

4. If a free machine can be found in a later slot than the original slot, then break the searching loop and move the selected job into the first-found free machine of the latest suitable slot.
5. If there is no free machine in the suitable slots, then the selected job remains in its original place.
6. Delete the selected job from the LDD list. If the LDD list is not empty, then go to 2, otherwise stop.

This JIT-oriented algorithm converts the schedule S_a into a new feasible schedule S_{JIT} , where the maximal tardiness is not increased, and the maximal earliness is reduced. It is easy to see that if a selected job has been moved from its original slot to a later slot, then its earliness is reduced, but the modification does not violate the due date or release time restrictions because of the well-defined boundaries of the searching loop. The modifications are carried out on the jobs in the non-increasing order of due dates. Therefore, jobs can be moved into the places which are freed by one of the previous job movements so the algorithm can achieve the maximum improvement. In the worst case, each job remains in its original place, and thus the tardiness and earliness are not changed.

The presented JIT-oriented algorithm creates a very sharp schedule in each situation. To increase the flexibility of the solution, we extended the algorithm with a set of job-dependent control parameters that specify the safety slack of each job. In this case, the free machine searching loop of the algorithm starts at an earlier (time) slot than the due date. For each job, the initial value of the first examined slot is equal to the difference of the given due date and the given safety slack control value of the job. This technique is well suited to the sophisticated safety requirements of the manufacturing control and to creating not only effective but also robust schedules.

6 Transferring the Theoretical Results into Practice

The scheduling problems outlined in this paper are inspired by a real case study concerning the plant of Fehrer Hungaria Járműipari Kft. specialized in vehicle seat products (Mór, Hungary). The firm produces different types of seat elements with variable series simultaneously. It is typical that the customers set very strict delivery due dates.

Many enterprise resources planning (ERP) and advanced production scheduling (APS) systems can be found in the market. Their functionalities cover a very wide range of different production systems and business environments. However, these general solutions cannot be applied directly to the operational production management in the plant under consideration because the created plans are based

on aggregated resources and they consider only the primary manufacturing processes.

We developed a new fine scheduling software based on the presented approach, models and algorithms. Our software can automatically create short-term execution plans that cover every important detail and process of the analysed production system. The generated schedules specify the tasks that the manufacturing system should perform in the planned time horizon. These detailed solutions can be executable directly at the shop floor level.

The application has useful graphical user interfaces for supporting user interactions in order to increase the flexibility of the production fine scheduling and control process. For example, when the process engineers want to declare mandatory configurations to be used for test manufacturing in a given time interval, an advanced editor module is available for them to express their exact requirements. The automatic scheduler takes into account these constraints. For this purpose, the solver engine is equipped with blocking techniques that make it possible to manage the modifiable and the protected configuration exchanges. The software offers many formats to show the solutions. The most important results are the fine schedule of the production processes, the schedule of the configuration preparations, and the values of the performance indicators. The fine schedule can be displayed as a list of the configuration exchanges with the corresponding data that specify which carriers and tools have to be attached to which position in which shift and what kind of product types have to be produced. The schedule of the configuration preparations declares exactly what pre-assembly activity has to be carried out in which shift.

The developed production fine scheduling system integrates many new functional components. The most important functions are as follows:

- Multi-objective production fine scheduling;
- Scheduling the configuration preparations;
- Managing the time-varying availability-constrained resources;
- Managing the shared accessible resources;
- Managing the process engineers' requirements;
- Managing the product-type dependent stock levels.

The multi-objective approach and these advanced functionalities of the fine scheduling software effectively help satisfy the requirements of shop floor management in practice.

For testing and evaluating the proposed models and algorithms, we used real industrial problems. Practical experience confirms that the production orders can be fulfilled with minimal tardiness and the manufacturing processes can be realized with a minimal number of configuration preparations and exchanges. The

accumulation of excessive stock can be avoided and safe levels of product-type dependent stock can be maintained. The utilization of the production lines can be increased. The importance of these goals can vary over time, so our software supports the user in expressing the actual importance of the objective functions by adjusting the priorities.

Conclusions

In this paper, we summarized our research results on the practice-oriented modeling and solving of fine scheduling problems related to vehicle seat element manufacturing. Extended models and advanced scheduling algorithms were presented to adapt to the concrete requirements of real-life situations by taking into consideration the specific characteristics of modern manufacturing.

We introduced the full problem and the proposed solving approach (fine scheduling); in addition the main part of the paper presented the concrete model and the solving algorithms of a built-in sub-problem, which in itself is a meaningful and important scheduling problem. This sub-problem focuses on scheduling only the preparatory activities (jobs) required by the manufacturing primary (main) processes.

The full production scheduling problem is NP-hard, so we handle this problem by using an advanced multi-objective and multi-operator searching algorithm to create near-optimal solutions. In each iteration of the searching algorithm, the built-in scheduling sub-problem has to be solved in order to decide whether the current full production schedule is feasible from the point of view of the configuration preparation. The new model $P(s) \mid p_i=1; r_i=\text{integer}; d_i=\text{integer} \mid L_{\max}$ and the proposed algorithms are intended to solve only the built-in sub-problem. This model includes a special resource environment that consists of time-dependent sets of parallel machines, while a set of independent jobs with release time constraints, due dates, and unit processing times is considered. To minimize the maximum lateness, we proposed a new slack-oriented solving algorithm that produces the optimal solution in polynomial running time. For supporting the Just-In-Time paradigm in manufacturing control, we introduced a JIT-oriented version of the scheduling algorithm that is able to reduce the maximal earliness without violating the due dates. To increase the robustness of the released schedule, safety slack control parameters can be used in the scheduling process.

The application of the proposed approach in practice showed that multi-operator and multi-objective fine scheduling based on simulation can determine what the actual manufacturing system should perform in the planned time horizon. To solve scheduling problems in real environments, the primary manufacturing processes have to be considered, and special attention has to be paid to the preparatory processes. The achievements and experiences of the software application have been very positive. The obtained results encourage the application of this

approach to other multi-objective optimization problems in production systems and processes.

The presented new scheduling models and algorithms can be applied effectively to solving different scheduling problems. One of the possible cases is the scheduling of manufacturing processes to which time-dependent resource constraints of assistant processes are connected. Another potential case is the scheduling of direct manufacturing jobs with due dates on time-dependent sets of parallel resources.

Our research and development project highlights the importance of modeling for the treatment of production planning, scheduling and control problems; in addition it emphasizes the interconnection of theoretical results and practical demands. The main purpose of this paper was to share our experiences and results with researchers and industrial practitioners working in the fields of production information engineering.

Acknowledgements

This research was partially carried out in the framework of the Center of Excellence of Mechatronics and Logistics at the University of Miskolc.

This research was partially connected to the TAMOP-4.2.1.B-10/2/KONV-2010-0001 project with support by the European Union, co-financed by the European Social Fund.

Software development and application of the research results in practice is supported by Fehrer Hungaria Jár್ಮűipari Kft. (Mór, Hungary).

References

- [1] A. Allahverdi, C. T. Ng, T. C. E. Cheng, M. Y. Kovalyov, A Survey of Scheduling Problems with Setup Times or Costs, *European Journal of Operational Research*, Vol. 187, pp. 985-1032, 2008
- [2] A. A. Lazarev, D. I. Arkhipov, F. Werner, Scheduling Jobs with Equal Processing Times on a Single Machine: Minimizing Maximum Lateness and Makespan, *Optimization Letters*, pp. 1-13, 2016
- [3] A. Gharbi, M. Haouari, Optimal Parallel Machines Scheduling with Availability Constraints, *Discrete Applied Mathematics*, Vol. 148, pp. 63-87, 2005
- [4] C. Koulamas, The Total Tardiness Problem: Review and Extensions, *Operations Research*, Vol. 42, pp. 1025-1041, 1994
- [5] D. Lei, Multi-Objective Production Scheduling: a Survey, *The International Journal of Advanced Manufacturing Technology*, Vol. 43, Issue 9-10, pp. 926-938, 2009

- [6] D. Quadt, H. Kuhn, A Taxonomy of Flexible Flow Line Scheduling Procedures, *European Journal of Operational Research*, Vol. 178, pp. 686-698, 2007
- [7] E. Mokotoff, Parallel Machine Scheduling Problems: a Survey, *Asia-Pacific Journal of Operational Research*, Vol. 18, pp. 193-242, 2001
- [8] Gy. Kulcsár, F. Erdélyi, A New Approach to Solve Multi-Objective Scheduling and Rescheduling Tasks, *International Journal of Computational Intelligence Research*, Vol. 3, Issue 4, pp. 343-351, 2007
- [9] Gy. Kulcsár, F. Erdélyi, Modelling and Solving of the Extended Flexible Flow Shop Scheduling Problem, *Production Systems and Information Engineering*, Vol. 3, pp. 121-139, 2006
- [10] Gy. Kulcsár, M. Kulcsárné Forrai, Detailed Production Scheduling Based on Multi-Objective Search and Simulation, *Production Systems and Information Engineering*, Vol. 6, pp. 41-56, 2013
- [11] H. Aytug, M. A. Lawley, K. Mckay, S. Mohan, R. Uzsoy, Executing Production Schedules in the Face of Uncertainties: a Review and Some Future Directions, *European Journal of Operational Research*, Vol. 161, pp. 86-110, 2005
- [12] J. Kaabi, Y. Harrath, A Survey of Parallel Machine Scheduling under Availability Constraints, *International Journal of Computer and Information Technology*, Vol. 3, Issue 2, pp. 238-245, 2014
- [13] J. Lamothe, F. Marmier, M. Dupuy, P. Gaborit, L. Dupont, Scheduling Rules to Minimize Total Tardiness in a Parallel Machine Problem with Setup and Calendar Constraints, *Computers & Operations Research*, Vol. 39, Issue 6, pp. 1236-1244, 2012
- [14] K. Lee, J. Y.-T. Leung, M. L. Pinedo, Scheduling Jobs with Equal Processing Times Subject to Machine Eligibility Constraints, *Journal of Scheduling*, Vol. 14, Issue 1, pp. 27-38, 2011
- [15] M. Kulcsárné Forrai, F. Erdélyi, Gy. Kulcsár, A New Extended Model for Solving Flexible Job Shop Scheduling Problems, *Proceedings of the International Conference on Innovative Technologies, IN-TECH*, pp. 325-328, 2013
- [16] M. L. Pinedo, *Planning and Scheduling in Manufacturing and Service*, 2nd ed., Springer-Verlag New York, 2009
- [17] M. L. Pinedo, *Scheduling Theory, Algorithms, and Systems*, 3rd ed., Springer-Verlag New York, 2008
- [18] M. Pinedo, C. Zacharias, N. Zhu, Scheduling in the Service Industries: an Overview, *Journal of Systems Science and Systems Engineering*, Vol. 24, Issue 1, pp. 1-48, 2015

- [19] M. X. Weng, J. Lu, H. Ren, Unrelated Parallel Machine Scheduling with Setup Consideration and a Total Weighted Completion Time Objective, *International Journal of Production Economics*, Vol. 70, pp. 215-226, 2001
- [20] P. Bikfalvi, F. Erdélyi, Gy. Kulcsár, T. Tóth, M. Kulcsárné Forrai, On Some Functions of the MES Applications Supporting Production Operations Management, In G. Bognár, T. Tóth (eds.): *Applied Information Science, Engineering and Technology: Selected Topics from the Field of Production Information Engineering and IT for Manufacturing: Theory and Practice*, (Topics in Intelligent Engineering and Informatics; 7), Berlin: Springer-Verlag, pp. 103-129, 2014
- [21] P. Brucker, *Scheduling Algorithms*, 5th ed., Springer-Verlag Berlin Heidelberg, 2007
- [22] T. C. E. Cheng, C. C. S. Sin, A State-of-The-Art Review of Parallel-Machine Scheduling Research, *European Journal of Operational Research*, Vol. 47, pp. 271-292, 1990
- [23] W. Wang, *Flexible Flow Shop Scheduling: Optimum, Heuristics, and Artificial Intelligence Solutions*, *Expert Systems*, Vol. 22, Issue 2, pp. 78-85, 2005
- [24] Y. Lin, W. Li, Parallel Machine Scheduling of Machine-Dependent Jobs with Unit-Length, *European Journal of Operational Research*, Vol. 156, Issue 1, pp. 261-266, 2004
- [25] Y. Ma, C. B. Chu, C. R. Zuo, A Survey of Scheduling with Deterministic Machine Availability Constraints, *Computers & Industrial Engineering*, Vol. 58, pp. 199-211, 2010