Adaptive job-shop control based on permanent order sequencing

M. Niehuesa,*, F. Buschlea; G. Reinharta

*aInstitute for Machine Tools and Industrial Management (iwb), Technische Universität München, Boltzmannstr. 15, 85748 Garching, Germany

* Corresponding author. Tel.: +49 (0)89/289-15471; fax: +49 (0)89/289-15555. E-mail address: michael.niehues@iwb.tum.de

Abstract

The control of today’s production systems is becoming more complex due to an increasing number of product variants, short-time delivery requirements and non-standardized production processes. Especially shop floors organized as job-shops are indispensable for single- and small-series-production with low repetition rates. Current tools for production planning like Advanced Planning and Scheduling (APS) systems support the creation of ideal production plans for this high complex production. However, appropriate methods for production control are not available, mostly due to a lack of transparency concerning the current production status. Hence, decisions to counteract disturbances and deviations from the plan are made locally by foremen or workers based on their experience and know-how. Due to their local field of view, it is often not possible to estimate the impacts on other orders in the production. The consequences are rush orders and high stock levels resulting in long and variable throughput times and at least in a decoupling of real and planned production. This paper presents a new approach for a job-shop control system based on a permanent adaption of the production plan to the current situation. A genetic algorithm for rescheduling the production plan centrally is the focused element of the described control system in this paper.

Keywords: Adaptive control; Algorithm; Decision making; Job-shop production

1. Introduction

Today, manufacturing organizations are confronted with an increasing variety of products while at the same time the production volumes and life-cycles are decreasing [1]. This leads to shorter payback periods for products-specific investments. So the highly flexible job-shop organization will retain its importance especially in the engineer-to-order industry although it has deficiencies concerning modern customer requirements like short delivery times and a high delivery reliability [2; 3], resulting from typical characteristics of job-shop production. Due to the highly flexible product structure, the material flow is undirected and the processing time differs between the products [4]. The resulting complexity of production planning and control (PPC) is typically simplified by a backlog of order to damp existing imbalances [5; 6]. This entails a high work in process (WIP) together with rush orders and a great insecurity concerning the delivery date. Since the production process in single and small-series production is not as predictable in matters of time and stability as in mass production, job shop production systems have also to face deviations from the planned schedule like longer processing times or rework [4]. Therefore, a high production performance has to be reached by an adaptive reaction on deviations instead of highly stable processes [7]. In addition, an adaptive control requires the availability of real-time data. Various factors being listed in [6] constitute the fact that the real-time acquisition of shop-floor data is still a challenge in a manual job-shop production system. This paper presents an approach for an adaptive control to overcome the existing challenges.

2. Research fields to improve the job-shop production

Since the distribution of electronic data processing, the coordination of job-shops has been an important research field. Many studies focus on the job-shop scheduling problem whose task is scheduling a quantity of jobs on a quantity of machines to minimize the makespan [8]. Therefore, solving algorithms have been developed focusing a high solution
quality and average computation times. An overview of these studies in given in [9; 10], but they are still ongoing. This development culminates in advanced planning and scheduling systems (APS), which are established in the industry for creating exact and optimal production schedules [11].

Due to unpredictable events, the exact schedule is usually not realizable at the shop-floor. Thus, an efficient management of countermeasures is necessary. Based on the work of [12], several researchers focus on controlling the WIP level to reach short throughput times and a high delivery reliability [13, 14]. These approaches determine local order sequencing by priority rules, which are not considering the global situation on the shop-floor. So they create new deviances on subsequent job-shops. Moreover, improving effects of priority rules decrease with declining WIP. Multi-Agent Systems (MAS) which act autonomously by decentralized decision making in heterarchical structures use agents as software representatives of e. g. orders, products and resources [15, 16]. MAS also have the lack of central coordination. Due to their high complexity and low transparency, an application in a manual job-shop is not favorable. Several researches focused on approaches of the control theory [17; 18; 19] by recording process data and adapting the input parameters of the shop floor control. Although the aspect of real-time data acquisition is still unsettled and most of the approaches use scenario simulation instead of responding on deviations during the ongoing production, it is the most promising approach for an adaptive production control system.

3. System for an Adaptive Job-shop control

3.1. Requirements

To overcome the existing disadvantages of a job-shop production with focus on the coordination of complex, undirected material flows faced by permanent disruptions, an adaptive job-shop control system has to fulfill the following aspects:

- Detecting deviations from production plan in real-time
- Deriving specific measures to respective disturbances in a short computation time between detection and execution
- Assessing countermeasures centrally to avoid emerging deviations in subsequent schedule periods
- Avoiding re-sequencing the complete production plan permanently to prevent turbulences on the shop-floor

Since the requirements on a system for job-shop planning and adaptive job-shop control are different concerning accuracy, computation time and considered quantity of orders, a separation of these functions in different systems will be necessary. Thus, the developed system for adaptive job-shop control requires an upstream planning system, which is able to commit a realisable schedule to the control system. The regarded time interval of the system is about the doubled sequence of the planning system’s runs. For example, if the planning system creates a new plan every morning before the production starts, the adaptive control system will regard an interval of two days to get more flexibility concerning sequencing measures. Therefore, the horizon of the upstream planning system has to be at least as long as this interval. The situation on the shop-floor at a defined date will be considered in the next planning cycle after each period.

![Fig. 1. Overview of the adaptive Job-shop control system](image)

3.2. Overview

The central idea is to control the job-shop by permanent actualization of the production plan. Therefore the plan is valid at every time to predict the further production flow. The elements of this system are illustrated in figure 1. The first one is the location-based shop-floor data acquisition being the crucial enabler for a real time identification of deviations to the current plan. In the case of deviations, the deviation management will be initiated. It consists of different algorithms to update and repair the outdated production plan to a valid one. All modifications will be evaluated by a cost-based function. If the countermeasures of the deviation management are not effective enough to damp the impacts, the whole production plan will be rescheduled by optimization. This element is a genetic algorithm being developed for a very fast rescheduling. As this paper focused on the algorithm design, the first three elements will be described in the next paragraphs while the next chapter highlights the algorithm.

3.3. Location-based data acquisition

The real-time data acquisition is an important challenge for realizing an adaptive job-shop control. Due to usually manual material flow, state of the art data acquisition technologies like production data acquisition (PDA) or radio frequency identification (RFID) can only collect discrete information about the order’s status. Between these read points, no information about the order’s status and position can be gained. A promising technology to detect all order movements permanently is indoor locating whose first implementations in production systems have been made [20].

This system element uses a shop-floor model based on coordinates to assign the orders to defined areas in each job-shop (e.g. goods receipt, waiting area). Every order has its predetermined sequence of these areas during its production. Hence, every location change of an order which is similar to its plan can be interpreted as a change in its production status.
Changes which are contradictorily to the plan (after a defined fault tolerance) can be seen as a deviation. An additional machine data acquisition (MDA) will extend the accuracy and functionality of the data acquisition but it would not be able to collect real time data without a locating technology.

3.4. Deviation management

Almost every deviation from an optimised production plan affects the production flow in a job-shop production. To reach high delivery reliability as well as an appropriate utilisation of resources and other cost-effecting objectives, it is necessary to survey the resulting consequences on the plan and derive the right measures. Since every deviation decouples the production plan from the real situation on the shop floor, the first measure is to create a valid production plan by shifting all orders being influenced by the disruption along the timeline. A classification of typical job shop disruptions and interferences permits the connection between their indicators and appropriate measures in terms of repair algorithms to counteract them. The following classification of disruptions has been made by their effects on the production plan:

- Deviations between planned and actual times
- Reduction of available capacity
- Increase of capacity demand
- Reduction of capacity demand

While updating the plan is necessary after each significant deviation in the scheduled operations, the repair algorithms are used only when the production flow has been disrupted strongly. If the production objectives cannot be improved next to the former ones by the repair algorithms, a global optimization will result in the best schedule which can be realized in the underlying situation on the shop floor. The concept of evaluating these situations is described in the next chapter.

3.5. Cost-based evaluation of changes

Balancing the different objectives of PPC will lead to a high complexity which is contradictory to short computation times. As all objectives finally result in cost, especially in a short-term view, a function regarding all types of costs being influenced by changes in the production plan is used to evaluate the measures of this system. The elements of the cost function and their application range are listed in Table 1.

Table 1. Allocation of cost elements to the different algorithms for rescheduling and global sequencing

<table>
<thead>
<tr>
<th>Elements of the cost function</th>
<th>Rescheduling by deviations</th>
<th>Global re-sequencing</th>
</tr>
</thead>
<tbody>
<tr>
<td>Set-up costs $C_s$</td>
<td>X</td>
<td>X</td>
</tr>
<tr>
<td>Delay costs $C_d$</td>
<td>X</td>
<td>-</td>
</tr>
<tr>
<td>Transport costs $C_T$</td>
<td>X</td>
<td>-</td>
</tr>
<tr>
<td>Machine costs $C_M$</td>
<td>X</td>
<td>-</td>
</tr>
<tr>
<td>Disuse costs $C_A$</td>
<td>X</td>
<td>X</td>
</tr>
</tbody>
</table>

Since the set-up effort mostly depends on the order sequence, the set up costs $C_s$ have to be considered after each change in the sequence. To secure a high adherence to schedules, the costs $C_d$ are the element to evaluate delayed order completions. Orders exceeding the considered interval have to be provided with internal delay costs to avoid shifting them in later periods where they may trigger bottlenecks. These internal delay costs can be interpreted as accruals for penalty risks and derived from the feasibility of causing later delivery delays. An additional benefit of using order-specific cost rates is the possibility of considering differences in their priority (e.g. rush orders or important customers). Measures resulting in additional transportation effort (e.g. if orders are shortly assigned to other machines) are considered by Transport costs $C_T$. The machine costs $C_M$ are based on the machine-hour rate (see [21]). They are changing by measures of allocating operations on alternative machines or changing the processing sequence of an order. Since $C_T$ and $C_M$ are not influenced by changes in the sequence, they are only considered in deviation management but not in the global optimization to save computing time. Since the machine-hour rate is a function of the average machine utilization, it is important to consider situations where the current utilization is on average lower than planned. Thus, additional costs for disuse have to be added as Disuse costs $C_A$. They consist of the fix part of the machine-hour rate (e.g. write offs).

To decide whether it is necessary to request repair algorithms or a global optimization, the costs of the repaired plan $C_{\text{new}}$ are set in proportion to the planned costs before the disturbance event $C_{\text{planned}}$:

$$\frac{C_{\text{new}}}{C_{\text{planned}}} > \alpha \cdot \beta_1$$

As action limit, the factors $\alpha$ for repair algorithms and $\beta_1$ for global optimization are set. They are individual for each use case. To avoid a constant degradation of the cost function under the action limit, the repaired plan is also set in proportion to the last optimized plan:

$$\frac{C_{\text{new}}}{C_{\text{opt.last}}} > \alpha_2 \cdot \beta_2$$

This is based on the assumption that each optimization was initiated by a heavy disturbance disabling any achievement of the cost value of the initial schedule before productions starts.

4. Genetic algorithm for a fast re-sequencing

In cases when countermeasures are not sufficient to reach an adequate cost target in a short-term time frame, a fast rescheduling by optimising the global sequence is the promising way to achieve the best possible production flow in case of preceding occurrences. It requires an optimisation algorithm that is able to calculate a good solution during the proceeding production. Genetic algorithms have been approved for their efficient relation between solution quality and computation time. Several research studies have demonstrated that they are qualified for job-shop scheduling [22; 23; 24]. Hence, they are promising to realize a fast optimization of the complete production plan.
4.1. Representation of the problem

The performance of the genetic algorithm depends highly on the problem representation as it's the genetic code. The presented algorithm represents the production schedule as symbolic representation meaning that the sequence of each machine in the job shop is represented separately. This results by the advantage to calculate the different parts of the cost function separately to accelerate the evaluation process of each generation. In contrast, the symbolic representation form allows the creation of invalid solutions by infringing the processing sequence of an order. Thus, it is important to evaluate the legitimacy of a solution as early as possible to avoid computing with illegal solutions. It was accomplished by a new representation form which had been inspired by the work of [25], who uses position numbers instead of order or machine numbers in a schedule’s representation.

The production plan as the scheduled global sequence is represented by the so-called order sequence matrix (OSM) which includes the order sequence of a machine $i$ as well as the machine sequence of an order $j$. It is realized by numbering all operations depending on their position $p_{i,j}$ in the global start sequence of the initial plan (see figure 2).

$$\text{OSM} = \begin{pmatrix} p_{1,1} & \ldots & p_{1,j} \\ \ldots & \ldots & \ldots \\ p_{i,1} & \ldots & p_{i,j} \end{pmatrix}$$

Fig. 2. Structure of the order sequence matrix (OSM)

Every element of the OSM represents a machine-order-combination. If the processing sequence does not match with a machine, the respective element in the OSM is a zero. With this representation form, a validity check can be made easily by deducing the current machine sequence of an order $j$ by sorting the machines in ascending order of their position $p_{i,j}$ in the column and compare it with the respective processing order in the order’s master data. All solutions with deviations between those sequences will be eliminated.

Another advantage is the direct representation of the machine’s sequence by sorting the orders in ascending order of their position $p_{i,j}$ in each line representing a machine. Thus, the set-up cost can be derived directly without calculating the complete schedule. An example of the first step of the algorithm, transforming the repaired production plan into the OSM, is illustrated in figure 3.

4.2. Genetic Operators

The process of the genetic algorithm and the embedded operators of mutation, crossover and selection are illustrated in figure 4. Mutation operators are used to create an initial population as well as to modify the chromosomes in each generation, in combination with crossover operators. Selection operators are used to evaluate the fitness of the chromosomes and to select those for the population of the next generation. Since the genetic operators have a high effect on the performance of the algorithm as well as they depend on the problem’s representation, this system requires a new design of the genetic operators. Due to the problem representation of this algorithm, each number of $p_{i,j}$ exists only once. So mutation and crossover operations only work by swapping numbers, not by modifying them.

The mutation operators (see fig. 5) are derived from typical sequencing activities in a job-shop. Mutation within a line corresponds to preferring a single operation in a machine sequence compared to another one. If an order is preferred to a second, this operation will be represented by the mutation of complete columns. The consequence is a modified sequence on all machines. The latter operator depends highly on the similarity of the order’s processing sequence. High variances between them will result in probable illegal solutions. To avoid them, the share of column mutation in mutation has to be minimized to zero with growing variances between the respective sequences. Both kinds of mutation prohibit the mutation of zeros. Since every zero stands for a not existing machine-order-combination, any kind of mutation will create an invalid solution.

The matrix provides also the possibility of line mutation, mutation within a column or diagonal mutation. All other kinds of mutation being provided by the matrix structure are not reasonable because they will create invalid solutions with a high probability. To achieve a higher variance with a mutation step, the operators will be applied on several positions in the OSM.

Crossover operators combine the features of two parent chromosomes. In this case, they join parts of one OSM with those of another OSM. The matrix structure enables three types of crossover operations (see fig. 5): line, column and
diagonal crossover. The line crossover combines lines or pairs of lines from different chromosomes. An advantage of this crossover operator is the perpetuation of partially good machine sequences because they do not change during this operation. The column crossover combines the machine sequence per order from different chromosomes by combining columns or a set of columns of the OSM. Assuming that similarities exist in the machine sequence of the orders, the creating procedure of the OSM has the effect, that generally the operations in the top left field of the OSM are scheduled short-time while later operations are at the bottom right field of the OSM. The diagonal crossover operator is derived from this fact by joining the short-time sequence of a chromosome with the long-term sequence of another one.

![Fig. 5. Operators of the genetic algorithm](image)

Within all crossover operations, prohibited solutions may emerge in two ways: Either by creating an invalid processing sequence or by creating a matrix with numbers for \( p_{ij} \) existing twice. Thus, a validation check and in the second case, for column and diagonal crossover, a recovery algorithm follows the crossover operator. The recovery algorithm identifies twins of \( p_{ij} \) and modifies \( p_{ij} \) in a way, that each number only exists once again.

To realize a short computation time, an efficient design of the genetic algorithm is indispensable. The selection operators are the elements to choose the best chromosomes for the next generation’s population. To avoid sorting out potentially good solutions, the algorithm uses two kinds of fitness values \( F \) to evaluate OSM of each generation \( n \) and each chromosome \( x \):

The lowest cost fitness \( F_{opt} \) and the proportional cost fitness \( F_{pro} \). \( F_{opt} \) is based on the cost function by setting the cost in relation with those of the best chromosome of the previous generation:

\[
F_{opt}(ORM_{n,x}) = \left( \frac{C_{opt}(ORM_{n-1,x,opt})}{C_{opt}(ORM_{n,x})} \right)^2
\]

The proportion is squared to increase the interval between better and worse fitness values. The proportional cost fitness roots in the intention of retaining a solution with a favourable set-up sequence or minor disuse times. Thus, the set-up, delay and disuse cost have to be weighted with the factor \( w \).

The fitness value \( F_{pro} \) is evaluated similar to \( F_{opt} \), also with squaring the cost ratio to enlarge the interval:

\[
F_{pro}(ORM_{n,x}) = \left( \frac{C_{pro}(ORM_{n-1,x,pro})}{C_{pro}(ORM_{n,x})} \right)^2
\]

A population consists of a defined quantity of chromosomes. The selection of the chromosomes for the next generation is made by roulette wheel selection (see [26]). This process based on a random choice of chromosomes with their feasibility of selection is proportional to their fitness value. For optimal and proportional fitness, an amount of positions have been reserved separately in the population. In addition, the respectively best chromosome of the parent population will be taken over in the next population for each fitness type.

### 4.3. Experimental results

To validate the algorithm’s function and performance, it has been implemented in MatLab R2013b. The computing performance of the algorithm was tested by implementing the ft10 benchmark problem of [27]. Since ft10 is designed with regard to short throughput times, the fitness function bases on minimum makespan of the production plan instead of costs. The optimum makespan of ft10 is 930 time units.

The initial OSM was created by scheduling each operation with the highest remaining production time first. Due to the different machining sequences in the ft10 benchmark problem, column crossover had been deactivated.

Table 2. Results of the ft10 implementation

<table>
<thead>
<tr>
<th>Problem</th>
<th>MakeSpan</th>
<th>Time</th>
</tr>
</thead>
<tbody>
<tr>
<td>ft10</td>
<td>Average</td>
<td>Best</td>
</tr>
<tr>
<td>ft10</td>
<td>1115</td>
<td>1091</td>
</tr>
<tr>
<td>Modified opt. ft10</td>
<td>1158</td>
<td>1103</td>
</tr>
</tbody>
</table>

The test series consists of 10 experimental runs with a population of 101 chromosomes. Since the job-shop control system is designed for short-time optimizations to react on disruptions, the ft10 scenario of creating an optimized schedule at once is not realistic. Thus, a second run had been executed with the optimum sequence of ft10, being modified at several positions to simulate disruptions in the realization of an optimized plan.

The results are listed in table 2 showing that the algorithm fulfills the requirements of creating a good solution in a short computation time. The average makespan improvement is about 48.5 percent of the time interval between the initial, priority-rule-based schedule of 1289 time units to the global optimum of ft10. The second test series based on the modified optimum of ft10 results in an average achievement of 124.5 percent of the optimum. Hence, the algorithm is appropriate to optimize a schedule in a few seconds but it does not reach the global optimum.

Further experiments will examine large problems with about 2000 operations (e.g. 20 machines, 100 orders) to evaluate the computing time of realistic job-shop dimensions. Since classical benchmark problems focus on creating an
optimal initial schedule, additional implementations of them will not lead to new findings. Hence, an implementation of the algorithm within the complete system for which it was designed initially is essential for validation. Therefore it is intended to simulate a real use case to compare the system’s performance to classical job-shop control by using local priority rules. Within this simulation, the performance of the rescheduling algorithms will also be validated.

5. Conclusion

Permanent disruptions necessitate an adaptive job-shop control which is able to consider the global production flow to avoid creating new problems with countermeasures. The developed systems will be able to detect deviations by location-based data-acquisition and updates the schedule by shifting the orders along the timeline. To countermeasure on negative effects on the production objectives, a two-tiered procedure avoids turbulences by permanent re-planning as good as possible. Thus, a global re-sequence by optimization is used only when the situation cannot be improved by local countermeasures. Since the optimization algorithm will work in real-time, it is essential to realize a very short computation time. First experiments show that an acceptable computation time can be reached, but with a trade-off concerning the quality of the result. Although the computation time is very short, it cannot be expected to realize the sequencing run in a few second with large-scale job-shops. Especially the calculation of the costs will demand additional time. Thus, a frozen zone must be set for a global optimization to avoid production interruptions while the optimization is running. Significant estimation will be made after the implementation of a real use-case. This scenario will also provide potential for an application of mutation operators on specific positions in the OSM, which is not possible with the existing benchmark problems. In conclusion, it can be estimated that the developed system provides a high potential to solve the current problems of job-shop control while some details and the complete evaluation is still outstanding.

Acknowledgements

The authors would like to acknowledge the funding support from the German Federal Ministry of Economics and Technology (BMWi) within the programme “AUTONOMIK for Industry 4.0”.

References