

SLSP - SIMULTANEOUS LOTSIZING AND SCHEDULING IN A JOB SHOP ENVIRONMENT

Hoeck M.*

Abstract: This paper provides an application oriented analysis of a multiple constraint scheduling procedure called SLSP, which is designed to batch and sequence production orders simultaneously. The Simultaneous Lotsizing and Scheduling Procedure (SLSP) is easy to implement in a Shop Floor Control System and leads to good results for finite loading problems. Dependent on the data available and the goal of production control SLSP can be used to minimize production costs or any other objective function, like minimizing the mean flow time or tardiness of the jobs. The approach is primarily based on a combination of regular dispatching rules and local search heuristics, such as Simulated Annealing, Threshold Accepting or Tabu Search. Additionally the procedure contains a special routine to calculate lot sizes using the Aspired Machine Time (AMT) as a control parameter. (JEL: E23,M11,P42)

Key Words: scheduling, batching, flexible routing, local search algorithms

Introduction

Manufactures of low to medium volume parts frequently face difficulties handling changes in the order stock or machine breakdowns, since most commercial Manufacturing Planning and Control Systems (MPC-Systems) are lacking of efficient scheduling procedures. The successive expansion of standard MPC-Software - starting with a central MRP-Unit, which was extended by additional features like Capacity Requirements and Master Production Planning - is one reason why process control is often not supported as necessary. Another reason is that production scheduling has become more complex.

Due to technological improvements in the machine tools the versatility of workcenters, especially the ability to produce a wide variety of part types using different cutting tools, has increased. This machine versatility provides scope for several routes of a part type and can be utilized to alleviate bottlenecks. Furthermore additional constraints, e.g. the limited number of tool slots at each machining center, have to be considered while scheduling production orders in a modern Job Shop environment.

In order to reduce the complexity of production scheduling a number of advanced MPC-Systems include decentralized Shop Floor Control Systems (Bauer et al. 1991), such as a 'Leitstand'. A Leitstand is an interactive Information- and Scheduling-System, implemented on a Workstation or a PC, to monitor and control the material flow of one or more workcenters. Order data can either be inserted directly into the Leitstand or loaded automatically from the central

*Prof. Dr. Höck Michael, TU Bergakademie Freiberg, Department of Industrial Management

MPC-System. It is also used to maintain data regarding operations, precedence constraints as well as the NC program library and production environment data including machine, shift and calendar information. The Scheduling System of a Leitstand usually provides hooks for all necessary dispatching rules and sometimes even more sophisticated optimization routines that can be used to improve an existing schedule.

Problem Statement

In shops where considerable routing flexibility exists, batch sizing and routing may significantly affect production throughput and work-in-process inventory. Several research studies and experimental investigations have analyzed the effect of batching and routing decisions in a Job Shop using Queuing Network Models [12, 5]. These models show the potential improvement that can be achieved by simultaneous lot sizing and scheduling. On the other side, Queuing Network Models define performance in terms of long-run, steady-state measures whereby the current state of the production facility as well as precedence constraints are not considered. For operational lotsizing and scheduling decisions a number of Dynamic Programming approaches and Branch-&-Bound-Procedures have been proposed [17, 10]. These optimization techniques are restricted to single facility problems, if more than one item has to be scheduled. Recently a heuristic scheduling algorithm underlying the GOAL SYSTEM Software (formally known as OPT) was revealed, which has been applied successfully to real-world problems and is based on a „drum-buffer-rope“ logic [16]. Similar to the Shifting Bottleneck Method [2] the procedure uses a sequential approach to build detailed constraint resource schedules (drum schedules), while nonbottleneck resources are scheduled roughly allowing a sufficient lead time (buffer) for the operations. However, the scheduling procedure does not consider important characteristics of a modern shop, such as alternative process plans as well as multiple resources per operation, e.g. machine and tool magazine capacity.

In the following sections we will introduce an iterative improvement approach to batch and sequence production orders in a Job Shop considering multiple constraints. In § 3 the basic steps of the scheduling procedure are presented in detail. Afterwards the procedure is tested by a simulation program that replicates the performance of a real-world production facility which is described in § 4.

Simultaneous Lotsizing and Scheduling Procedure

The Simultaneous Lotsizing and Scheduling Procedure (SLSP) should be applied rolling through time and comprises three steps:

- In *step 1* the „urgency“ of arriving jobs is determined by subtracting the estimated lead time from the due date. Only those orders are released

to the shop floor whose planned starting date is within a previously defined time horizon.

- In *step 2* the transfer batches of the production orders are calculated using the Aspired Machine Time (AMT) as a control parameter.
- In *step 3* the process batch sizes of the jobs are determined by sequencing the transfer batches on the machining centers. The scheduling procedure is based on a combination of dispatching rules and iterative improvement methods.

Due to the combination of standard priority rules and local search techniques SLSP can easily be implemented in a Shop Floor Control System and provides „near optimum“ solutions within a reasonable period of time. Furthermore the approach is flexible regarding the goal of production control. Dependent on the data available SLSP can be applied to reduce the cost of production, e.g. minimizing a combination of inventory costs, set up costs and lateness costs, or any other objective function, like minimizing the mean flow time or tardiness of the production orders.

Setting the release window

By this control parameter the active load of a shop floor is divided into those production orders considered as urgent and other jobs that can be scheduled later on. To reduce the work-in-process inventory only urgent orders whose planned starting date (= due date - lead time) lies within the predefined time horizon are released to the shop floor. The order release window, which starts with the actual period, should be a multiple of the planning horizon of the scheduling system. Enlarging the release window can reduce the tardiness of the production orders because jobs that are not urgent are prereleased, while at the same time the workload and work-in-process inventory of the Job Shop as well as the flow time of the jobs will increase. On the other hand, an insufficient small release window results in high idle times at the machining centers and may not shorten the flow time of the production orders. Setting the release window is therefore a simple instrument to control the *workload* and *tardiness* in a Job Shop.

Calculating the transfer batches

The basic idea of transfer batches was first introduced by KANBAN [18] and afterwards adopted by OPT [11] to control the material flow in Job Shop production. A transfer batch of a part type is defined as the number of parts moved between resources and the *smallest lot size* before a machining center can be set up to a new production order. By sequencing transfer batches of multiple items (jobs) on a machine, which will be described in § 3.3, the process batch of an operation is determined. Therefore the process batch of a part type may differ from the transfer batch and vary from one workcenter to the other. Contrary to the conventional MRP approach, which determines fixed batches for each part type separately minimizing assumed carrying and set up costs, lot sizing is based

here on a *systems approach*, which involves the current state of the shop floor and the overall goal of production control. Further, it provides the advantage that lot sizing decisions are transferred to the shop floor, which usually has more accurate information on constraints.

Because the real holding and set up costs are not known in advance, the aim of transfer batch sizing is to maximize production throughput. A prime factor, next to the potential, yet unknown bottlenecks that retards the material flow in a Job Shop, are the high deviations in the processing time of the jobs. If all batches would be passed from one workcenter to another within a similar cycle time, the queuing time on the floor could be reduced to a large extent. Equivalent to a traffic guidance system an increased throughput can be achieved by a suggested „speed“ of the part types or *Aspired Machine Time (AMT)*. This control parameter defines the „adequate“ processing time of a machining center before it is set up to a new job. If a production order exceeds the AMT, it is split into smaller and therefore faster transfer batches, thus reducing the „traffic jams on the highway“ through the Job Shop. In general, the AMT of a machining center correlates with the average set up time, which includes the time to replace worn-out or broken tools, the time for tool changes to produce a different subset of the given part types, and the time to assemble or mount new fixtures. An „adequate“ AMT will lead to small transfer batches, which shorten the flow time of the production orders. If the AMT is insufficient low, shop time is consumed with nonproductive set ups; the resulting high level of traffic density will cause greatly increased congestion. On the other hand, a high AMT and therefore large transfer batches tie up machines for extended periods of time, thus increasing the unit flow time.

The calculation of transfer batches involves two steps. In a first step the transfer batches are determined independently from the net requirements of the part types. At each machining center the transfer batch size of the part type should be at least as large as the *relative production rate*, which is the ratio of the AMT to the processing time of one part stated in the NC Program. Since a production order usually runs over more than one machining center the transfer batch size of a part type j can be calculated as

$$\begin{aligned} & \text{transferbatchsize of a part type } j = \\ & = \text{Max} \left[\frac{\text{AMT of machining center } m}{\text{Processing time of one part at machining center } m} \quad \forall m \in M_j \right] \quad \forall j \end{aligned} \quad (1)$$

with M_j the set of all machining centers m , where part type j is processed including alternative routes. By this approach *all potential bottlenecks* in a Job Shop which usually require an above average set up time and therefore high AMT are considered.

In a second step the number of batches that have to be produced is determined by dividing the net requirement of a part type by the transfer batch

size. Performing the above division may not result in a integer value. To completely satisfy the requirements of the part types, left overs should either be spreaded over the existing transfer batches or added to one transfer batch.

An Example to Illustrate the Calculation of Transfer Batches

To illustrate the calculation of transfer batches and the effect of the AMT, we will discuss a sample production program of a Job Shop in table 1, assuming that 6 part types are processed on 4 machines. Table 1 contains next to basic data (net requirements and machine time per part) the average processing time of the batches on a machine.

Table 1. Master production program of a Job Shop

Part no.	Net requirements [Units]	Processing time per part at machining center [Minutes]				Ø Processing time per batch [Minutes]
		M1	M2	M3	M4	
1	100	5	2	1	4	300
2	420	2	1	2	3	840
3	200	5	2	4	3	700
4	50	20	10	2	20	650
5	10	60	20	30	20	325
6	300	10	5	4	2	1575

The source: authors' calculus, program of a Job Shop

An AMT of 200 minutes for all four machining centers will lead to the following relative production rates and transfer batch sizes of the part types (see table 2). The number of transfer batches results from the division of the net requirement by the *maximum relative production rate* of a part type whereby left overs are spreaded equally over the existing transfer batches.

Table 2. Calculation of transfer batch sizes

Part no.	Relative production rate of the part types [Units/AMT]				Transfer Batch sizes [Units]	Ø Processing time per batch [Minutes]
	M1	M2	M3	M4		
1	40	100	200	50	1 x 100	300
2	100	200	100	66,67	2 x 210	420
3	40	100	50	10	2 x 100	350
4	10	20	100	10	1 x 50	650
5	3,33	10	6,67	10	1 x 10	325
6	20	40	50	100	3 x 100	525

The source: authors' calculus, program of a Job Shop

Here part types 1, 4 and 5 are produced with their original net requirements, while items 2, 3 and 6 are transferred through the Job Shop in smaller batches. In this example an AMT of 200 minutes reduces the standard deviation in the processing time of the batches by 71 %, which will shorten the waiting time of the jobs at each machining center.

As shown in table 2, not all part types may reach the AMT, because of low net requirements. The mean processing time of the transfer batches in a Job Shop is therefore a hyperbolic function that decreases the smaller the AMT is (see figure 1). At the same time the average number of transfer batches expands exponentially resulting in higher set up times. The actual set up times are determined by sequencing the transfer batches, which will be described in the following.

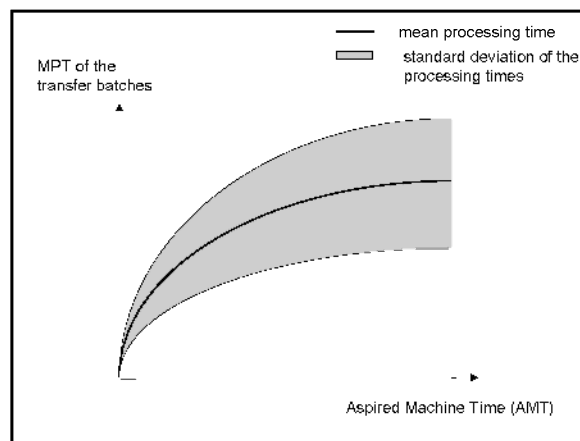


Figure 1. Relation between Minimum Machine Time (AMT) and Mean Processing Time (MPT) of the batches

Sequencing the transfer batches

In a final step of SLSP the process batches of the orders are determined by allocating and sequencing the transfer batches on the machining centers. In many shops accurate cost data is not available, therefore scheduling is usually based on time-oriented objectives, e.g. minimizing the maximum lateness or mean lead time of the jobs that correlate with the cost goals. In practice, these performance measures often change from one planning period to the other: One week the tardiness (lateness costs) is the more important objective, whereas another week the lead time (inventory holding costs) of the jobs is the objective to more pay attention to. Hence, an application oriented scheduling procedure needs to be flexible regarding the objective function so that it can be adapted to the priorities of the scheduler [15].

As mentioned in the previous section, scheduling of transfer batches provides the advantage that process batches of a part type can vary from one workcenter to the other. In order to reduce the flow time of the jobs large lot sizes should be placed on the bottleneck resources, while nonbottlenecks should

produce small batches. Further, the process batches of a part type may overlap in time (see figure 2), which is also a common approach to reduce lead time in a production facility. If two identical machining centers (M3 and M4) exist, it is also possible to parallelize the process batches of a part type.

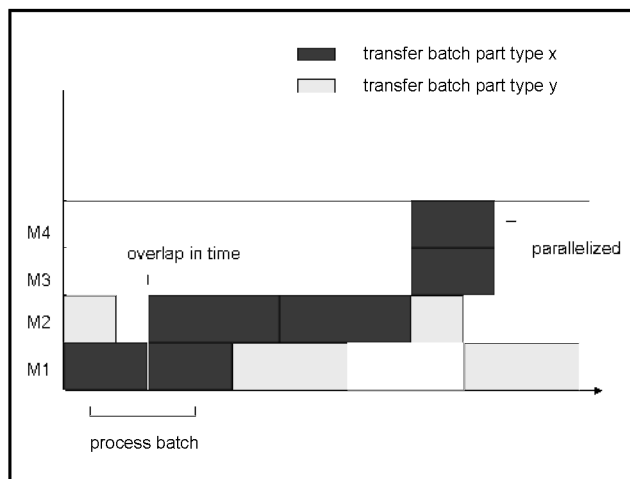


Figure 2. Scheduling transfer batches in a Job Shop

As a result of the routing flexibility, scheduling in a modern Job Shop environment is rather complex, especially if additional constraints, e.g. shift or tool magazine capacities, need to be considered. For this reason we designed an efficient heuristic approach, which combines regular dispatching rules and local search procedures, such as Simulated Annealing [13, 6], Threshold Accepting [8] or Tabu Search [9]. The general approach of local search can be described as follows :

Begin

Generate S_j ; - initial seed schedule constructed by a heuristic

for $k := 1$ to K do - number of searches

begin

Generate S_i from N_j ; - N_j is the neighborhood of schedule S_j

If S_i is accepted then $S_j := S_i$; - conditions of acceptance

end;

End;

The basic idea is to generate - in an iterative process - new solution proposals based on a feasible seed solution, which are accepted under certain conditions for further neighborhood search. Contrary to conventional iterative improvement techniques these procedures also accept inferior solutions for further

neighborhood search, in order to escape local optima and to increase the likelihood of finding the global optimum. A local search procedure involves three steps:

- *Generating an initial seed schedule*

In a first step an initial seed schedule S_j is generated, which can be provided by any good heuristic method. For Job Shop scheduling problems various dispatching rules have been put forward such as FCFS (first come first serve), SPT (shortest processing time), EDD (earliest due date), LWKR (least work remaining) or SST (shortest set up time), if sequence-dependent set up times exist. These single pass heuristics construct a schedule through a sequence of decisions on what seems locally best and the decisions once made are final. In comparison to other approaches priority rules provide the advantage of low computation time and can be easily adapted to constraints. On the other hand they rarely find a „near optimum“ solution, since the set of alternative dispatchable operations is decreasing during the procedure and therefore there are often unfavorable decisions made towards the end.

- *Neighborhood Search*

To improve an initial seed schedule neighborhood search techniques, such as a *pairwise interchange* of operations or batches on a machine, can be applied. Several research studies [1, 21] have shown that the definition of a neighborhood structure N_j is critical to the performance of local search. In literature local search procedures are often applied to the classical Job Shop Problem (JSP), which is to minimize the makespan in a conventional shop. In this context some authors suggest to move an operation u right before or right after an operation v on a machine, such that both operations are on the *critical path* [3]. A more restricted neighborhood search is used by Laarhoven et al., swapping only *adjacent* operations on a critical machine [19]. An additional limited neighborhood structure is applied by Matsuo et al., interchanging only successive operations, where the *job-predecessor of u or the job-successor of v* also belongs to the critical path [14]. All described search techniques have in common that they diminish the large set of possible neighboring solutions in order to increase the speed of search. On the other hand they are all restricted to the objective function of minimizing the makespan.

In the following we will apply a neighborhood search technique, which is also based on small neighborhoods, but flexible regarding the performance measures of production control. The neighborhood search implies a priority dispatching rule and interchanges alternatively dispatchable transfer batches (operations). Each time a schedule is constructed by a dispatching rule the set of alternatively dispatchable transfer batches o_n is recorded. Let then S_j be a seed schedule and let Q_{mt} denote the set of transfer batches waiting in queue to be processed on machining center m in period t . Further, let τ denote the set of periods where more than one job is to be processed on a machine or a job-predecessor of an operation is finished. The neighborhood search procedure can then be stated as follows:

S_i := seed solution

Begin

select randomly $t \in \tau$

if in t $o_n \in Q_{mt}$ exist

begin

select randomly o_n ;

apply transition mechanism resulting in S'_{nzw} ;

end;

End;

for $t := t+1$ to T do priority rule dispatching

$S_i := S_{new}$;

Here neighborhood search is focused on good heuristic solutions. Further, there are two transition mechanisms implemented (see figures 3 and 4).

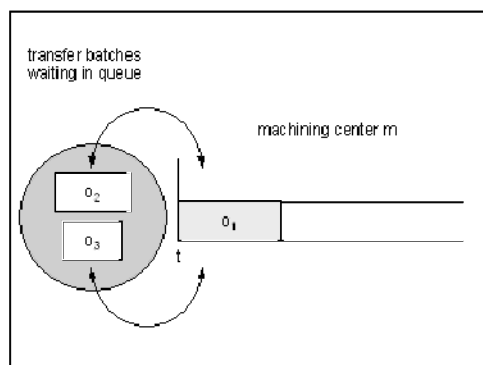


Figure 3. Transition mechanism 1

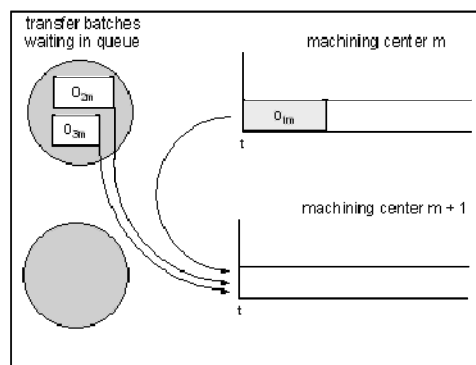


Figure 4. Transition mechanism 2

The first transition mechanism changes the sequence of jobs on a machining center by swapping a transfer batch o_1 - originally scheduled by a dispatching rule - and o_n waiting in queue, if o_n is a transfer batch of a different part type. The second transition mechanism utilizes the routing flexibility of the production orders and moves one of these transfer batches to an idle machining center, which can process the operation at the same starting point. After the pairwise interchange of operations or move of a transfer batch to another machining center a dispatching rule is used to construct the schedule for the rest of the periods.

• *Conditions of acceptance*

Apart from the definition of the neighborhood there are several strategies to control local search, which is done by the conditions of acceptance. In general these conditions can be defined either stochastically or deterministically. The stochastic Simulated Annealing approach for a minimization problem can be stated as follows [20] :

Generate a seed solution S_j ;
 Calculate the value of the objective function $f(S_j)$;
 Select initial control parameters $c > 0$; $0 < \beta < 1$;
 Repeat
 for $k := 1$ to K do
 begin
 Randomly select $S_i \in N_j$;
 $\Delta := f(S_i) - f(S_j)$;
 if $\Delta < 0$ then $S_j := S_i$;
 else generate random p uniformly in the range of $(0, 1)$;
 if $p < e^{-\Delta/c}$ then $S_j := S_i$;
 end;
 $c := c \cdot \beta$;
 Until stopping condition = true ;
 S_j is an approximation of the optimal solution;

In this case the acceptance probability p depends on the difference Δ of the new solution proposal to the value of the objective function of the seed schedule as well as on the control parameter c . The acceptance probability of cost-increasing transitions therefore decreases the higher the deviation or the smaller the control parameter is. The initial control parameter c as well as the reduction β of this parameter are user-specified and have to be tuned according the data of the scheduling problem. If the final solution is to be independent from the starting solution the initial control parameter c should cover a large portion of the total solution space to allow an almost free exchange of neighboring solutions at the beginning.

The Threshold Accepting approach as well as Tabu Search, which uses a so called „tabu list“ of prohibited transitions to control neighborhood search [7], resemble the described Simulated Annealing in its structure. In contrast to Simulated Annealing, Threshold Accepting employs a deterministic Threshold (T) and a sinking rate (β) as control parameters and accepts all uphill moves, which are smaller than the actual Threshold ($\Delta < T$). For the following industrial application we tested the Simulated Annealing Algorithm and Threshold Accepting Method, because they can „climb out“ local optima easier than Tabu Search, which is an advantage in the steep, „cliffy“ solution space of real-world scheduling problems.

Industrial application and computational results

SLSP has been tested at a production facility of a major German manufacturer of cigarette and packaging machines, consisting of an Flexible Manufacturing System (FMS), which is embedded in a Job Shop production of heavy parts. The integrated FMS includes three workcenters, which are 3-axis

drilling and milling machines, connected by a monorail conveyor. Next to the FMS there are 8 CNC machine tools with two identical Omnimills and two identical horizontal drilling machines.

The production program of the Job Shop includes a wide range of part types, such as housings, bearings, holders etc., which are assembled on the next production stage. The parts are made of aluminium, plastic, cast iron and steel with average production requirements of 25 parts.

Our simulation study covers a planning horizon of 5 workdays (4800 Minutes) with two 8-hour-shifts per day. The input data of the simulation program includes 10 master production schedules, each with 50 „urgent“ orders. The net requirements of the part types are determined by a uniform distribution in the interval {5, 55}. Also the arrival as well as the due dates of the production orders are chosen randomly from the discrete periods {0, 960, 1920} and {2880, 3840, 4800} respectively, assuming that starting and mounting dates are set by a central MPC-System on a daily basis. For each part type there are {1, 5} operations to be performed, whereby the workcenters of the FMS as well as the identical drilling and milling machines can be utilized alternatively. Further, the processing time of an operation varies between {3, 60} minutes per part with tool requirements of {1, 10} tools. However, the average change over time of a process batch is 30 minutes on all workcenters in the shop. A total set up of a machining center, meaning that all unneeded tools are removed and new tool sets are loaded onto the magazines, occurs if an operation exceeds the actual tool magazine capacity. Otherwise it is assumed that the tools are loaded in advance, so that the set up time of an automatic tool exchange is 0. All machine tools of the FMS are equipped with local tool magazines that have a capacity of 30 tools, while the stand alone CNC machines have a capacity of 20 tools.

At present, the Job Shop scheduling is performed by a Shop Floor Control System using priority dispatching rules while the FMS is scheduled manually. The global objective of production control is to minimize the mean flow time of the production orders. We therefore apply the SPT-rule to construct an initial seed schedule and for further neighborhood search.

In a first step the impact of the Aspired Machine Time (AMT) on the performance measures of the Job Shop is analyzed. Because of the similar set up times on all workcenters, we apply only one control parameter to calculate the transfer batch sizes of the part types. Figure 5 shows the mean processing time as well as the standard deviation of the processing times of the transfer batches in relation to the AMT, taking one order stock as an example. The AMT is stated here in percent of the shift capacity (480 Minutes) varying from 48 to 240 minutes.

In this example the average number of transfer batches per part type is close to one with a mean processing time of 330 Minutes, if a AMT of 50 % (240 Minutes) is chosen. Reducing the AMT from 50 % to 10 % will lead to smaller transfer batches with an average processing time of 90 minutes on each

machining center. At the same time the average number of transfer batches per part type increases exponentially to 5 batches.

Mean processing time of a transfer batch per machine

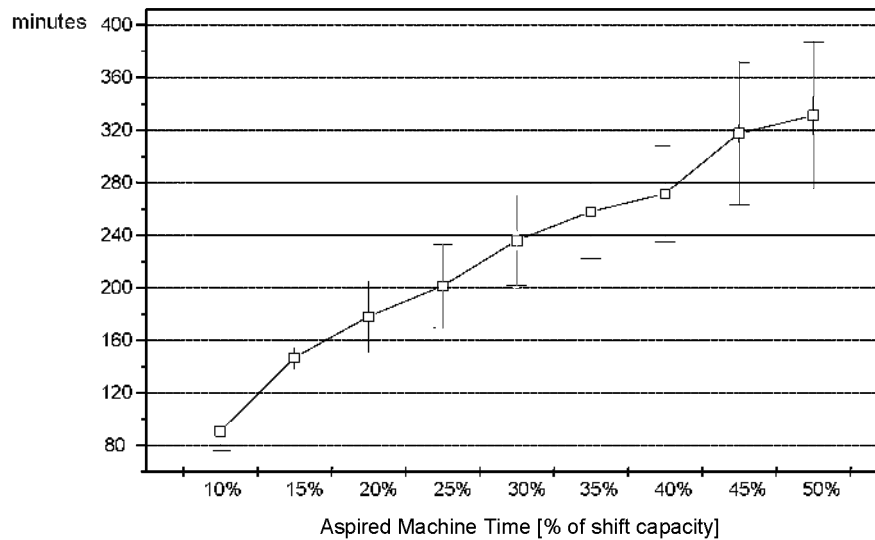


Figure 5. Mean processing time of the transfer batches in relation to the AMT

Table 3 contains the performance measures of the Job Shop using the SPT-rule to schedule the transfer batches. The average flow time (MFT) of the part types is 1.73 days while the mean tardiness (MT) is 221 minutes.

Table 3. Performance measures of the Job shop using the SPT-rule

AMT performance measures	Makespan [min.]	MFT [min.]	MT [min.]
10 %	4848,04	1960,59	243,30
15 %	4922,44	1941,73	226,71
20 %	4601,24	1646,45	154,94
25 %	4789,24	1285,51	122,56
30 %	4368,13	1165,33	118,61
35 %	4274,13	1519,94	216,63
40 %	4689,78	1606,40	288,27
45 %	4875,56	1656,87	299,09
50 %	4925,89	2206,24	319,82

The source: authors' calculus, program of a Job Shop

The results indicate that the shop floor performance is highly dependent on the AMT of the workcenters. It can be observed that the makespan, the mean flow time as well as the mean tardiness can be reduced to a large extent, if production orders are split into smaller transfer batches. Contrary to the conventional approach of scheduling part types with their net requirements or given lot sizes (\approx AMT of 240 minutes), the process batches vary here from one workcenter to the other, overlap in time or are parallelized on identical machining centers, which reduces the mean lead time up to 50 %.

For the investigated shop floor an AMT of 30 % provides the best results, meaning that a potential bottleneck resource processes a transfer batch at least 144 Minutes before it's set up to a new job. A further reduction of the AMT expands the mean lead time of the part types, since the number of transfer batches will increase over proportionately and shop time is consumed with nonproductive set ups. The calculation of the transfer batch sizes as well as sequencing the transfer batches using a priority dispatching rule is a matter of seconds on a PC.

A further improvement of the mean lead time can be achieved by applying the described Simulated Annealing (SA) or Threshold Accepting (TA) using the following parameter settings (see table 4). Here the initial control parameter c and T are defined in % of the objective value of the starting solution.

Table 4. Parameter configuration of SA and TA

Parameter	T / c	β
1	4 %	0.95
2	3 %	0.97
3	2 %	0.99

The source: parameter setting

These parameter settings cover a wide range of the solution space, while configuration 1 accepts major cost-increasing transitions and configuration 3 allows only minor uphill moves. Further, the number of searches K per iteration is set to neighborhood size (= number of alternatively dispatchable operations) of each accepted schedule. A simulation run was aborted after a local neighborhood has been searched randomly for three times without any improvement of the best solution.

The results of the different local search procedures are presented in table 5. In comparison to the quality of the initial seed schedules ($\emptyset S_0$) created by the SPT-rule, local search improves the mean flow time (DV_{S_0}) of the production orders by 40 % on average, while the total computation time ($\emptyset CT$) of a simulation run is about 12 Minutes on a 486 PC (60 MHz). Major improvements (48 %) are achieved at a high AMT, while the smallest improvements (36 %) result at an „adequate“ AMT, thus reducing the mean lead time of the production orders to 0.77 days.

Table 5. Performance of different local search procedures

Heuristic	$\bar{\sigma}_{S_0}$ [Min.]	MFT [Min.]	σ_{MFT} [Minuten]	$\bar{\sigma}_{CT}$ [Sec.]	DV_ S_0 [%]
Threshold Accepting	1665.45	1007.9	90.00	700.87	-39.48
Simulated Annealing		981.7	96.45	689.34	-41.05

The source: authors' calculus

In this experiment the Simulated Annealing approach slightly outperforms the Threshold Accepting method, although the quality of the best solutions found as well as the run time do not differ significantly. Also the configuration of the parameters c / T and β had no significant influence on the performance of local search. Hence, SLSP can easily be implemented in practice, because schedulers do not have to spend much time on tuning the local search control parameters.

Summary

In this paper we introduced an application oriented approach of lot sizing and scheduling in a Job Shop environment. The aim of SLSP is not to create a minute-based timetable, but to find a good allocation and sequence of the production orders subject to organizational and technological constraints. At a first stage a rough cut order release is performed to control the workload, work-in-process inventory and tardiness in the Job Shop. Afterwards the „urgent“ production orders are batched and scheduled using a systems approach that can be adapted to the priorities of the scheduler.

The key control parameter of SLSP is the Aspired Machine Time (AMT), which defines the „adequate“ processing time of a machining center before it can be set up to a new job. The „adequate“ processing time of a machining center depends on the overall goal of production control and the current state of the workcenter, whether the machine is a bottleneck or nonbottleneck resource. As a result of the routing flexibility in a modern Job Shop bottlenecks are rarely known in advance or may shift within the planning period. Therefore one should apply a simulation run using a regular dispatching rule to determine the AMTs of the workcenters, which takes only a few seconds on a regular PC.

To improve a given schedule the described scheduling procedure can be applied, which combines regular dispatching rules and local search. SLSP can easily be adapted to additional constraints, such as local buffer and workforce capacities. In general, scheduling constraints diminish the set of alternatively dispatchable

operations, thus increasing the speed of local search. On the other hand, additional availability checks have to be performed, which prolong the computational time of the dispatching rules. Therefore only 'hard' constraints that determine the feasibility of the schedules should be considered in SLSP.

References

- [1] Aarts E., Van Laarhoven P., Lenstra J. and Ulder N., *A Computational Study of Local Search Algorithms for Job Shop Scheduling*, ORSA Journal of Computing, Vol. 6, No. 2, 1994.
- [2] Adams J., Balas E. and Zawack, D., *The Shifting Bottleneck Procedure for Job Shop Scheduling*, *Management Science*, Vol. 34, No. 3, 1988.
- [3] Balas E. and Vazacopoulos A., *A Guided Local Search for Job Shop Scheduling*, Working Paper POM WP-01, Fairleigh Dickinson University 1995.
- [4] Bauer A., Bowden R., Browne J., Duggan J. and Lyons G., *Shop Floor Control Systems*, London et al. 1991.
- [5] Calabrese J. M. and Hausmann W. H., *Simultaneous Determination of Lot Sizes and Routing Mix in a Job Shop*, *Management Science*, Vol. 37, No. 8, 1991.
- [6] Cerny V., *Thermodynamical Approach to the Traveling Salesman Problem*, *Journal of Optimization Theory and Applications*, Vol. 45, No. 1, 1985.
- [7] De Werra D. and Hertz A., *Tabu Search Techniques*, *OR Spektrum*, Vol. 11., No. 1, 1998.
- [8] Dueck G., *New Optimization Heuristics*, *Journal of Computational Physics*, Vol. 104, 1993.
- [9] Glover F., *Future paths for integer programming and links to artificial intelligence*, *Computers and Operations Research*, Vol. 13, 1986.
- [10] Fleischmann B., *The Discrete Lot-Sizing and Scheduling Problem*, *European Journal of Operational Research*, Vol. 44, No. 3, 1990, Fry T., Cox J., Blackstone J. and Hoffmann T., *An Analysis and Discussion of the Optimized Production Technology Software and Its Use*, *Production and Operations Management*, Vol.1, No. 2, 1992.
- [11] Karmarkar U. S., "Lot Sizes, Lead Times and In-Process Inventories", *Management Science*, Vol. 33, No. 3.
- [12] Kirkpatrick S., Gelatt C. and Vecchi M., *Optimization by Simulated Annealing*, *Science*, Vol. 220, No. 4598, 1983.
- [13] Matsuo H., Suh C. and Sullivan R., *Controlled Search Simulated Annealing Algorithm for the General Jobshop Scheduling Problem*, Working Paper 3-4-88, Department of Management, Graduate School of Business, University of Texas, Austin 1988.
- [14] Pinedo M., *Commentary on "An Exposition of Multiple Constraint Scheduling as Implemented in the Goal System"*, *Production and Operations Management*, Vol. 6, No. 1, 1997.
- [15] Simons J. V. and Simpson W. P., *An Exposition of Multiple Constraint Scheduling as Implemented in the Goal System*, *Production and Operations Management*, Vol. 6, No. 1., 1997.
- [16] Solomon M., *Deterministic Lotsizing Models for Production Planning*, Berlin et al. 1991.

- [17] Sugimor Y., Kusunoki K., Cho F. and Uchikawa, S., *Toyota Production System and Kanban System Materialization of Just-In-Time and Respect-For-Human System*, International Journal of Production Research, Vol. 15, No. 6, 1977.
- [18] Van Laarhoven P., Aarts, E. And Lenstra, J., *Job Shop Scheduling by Simulated Annealing*, Operations Research, Vol. 40, No. 1.
- [19] Van Laarhoven P. and Aarts E., *Simulated Annealing*, Dordrecht 1987.
- [20] Veassens R., Aarts E. and Lenstra J., *Job Shop Scheduling by Local Search*, Memorandum COSOR 94-5 (revised version), Department of Mathematics and Computing Science, Eindhoven University of Technology, Eindhoven 1994.

USTALANIE WIELKOŚCI ZAMÓWIENIA I PLANOWANIE W SYSTEMIE Gniazdowym

Streszczenie: Artykuł przedstawia praktyczną analizę zastosowania tzw. procedury planowania SLPS z wieloma ograniczeniami, która została zaprojektowana na potrzeby realizacji zamówień produkcji seryjnej oraz sekwencyjnej. W systemie sterowania produkcją (ang. *Shop Floor Control*) można bardzo łatwo zaimplementować procedurę równoległego ustalania wielkości zamówienia i planowania (ang. *Simultaneous Lotsizing and Scheduling Procedure (SLSP)*), co prowadzi do dobrych rezultatów w przypadku problemów z oceną zdolności produkcyjnych (ang. *finite loading problems*). W zależności od dostępnych danych oraz celu kontroli produkcji, SLSP może być wykorzystana do zminimalizowania kosztów produkcji lub innych rzeczywistych funkcji, jak np. zminimalizowanie średniego czasu przepływu lub opóźnień zadań. Zaproponowane w artykule podejście początkowo opiera się na zasadach regularnych reguł priorytetu i heurystykach lokalnego wyszukiwania, takich jak symulowane wyżarzanie, akceptacja progowa, czy przeszukiwanie tabu (ang. *Tabu search*). Ponadto SLSP zawiera specjalną procedurę obliczania wielkości zamówienia z wykorzystaniem aspiracyjnej maszyny czasowej (*Aspired Machine Time – AMT*) w roli parametru kontrolnego.