Production Control of a Flexible Manufacturing System in a Job Shop Environment

KARL-WERNER HANSMANN
Department of Industrial Management, University of Hamburg, 20146 Hamburg

MICHAEL HOECK
Department of Industrial Management, University of Hamburg, 20146 Hamburg

Abstract:

This paper provides an application oriented analysis of local search procedures for Operation Scheduling and Shop Floor Management of a major German manufacturer of cigarette machines. The heuristics applied are the Threshold- and Simulated Annealing-Algorithm considering Job Shop as well as embedded FMS production features. In this context a new neighbourhood search technique is developed, which is based on a small set of local neighbourhoods and is flexible with respect to the performance measurements of production control. By this approach the scheduling, loading and workload allocation problems of a production facility consisting of an embedded FMS and a conventional Job Shop can be solved simultaneously.

Keywords: Approximation Algorithms, Job Shop Scheduling, FMS, Loading

1. Problem Statement

The conventional **Job Shop Scheduling Problem** can be described as follows (Adams et al. 1988). Given a set $J = \{1(1)n\}$ of jobs, which are to be processed on a set $M = \{1(1)m\}$ of machining centres with the objective of e.g. minimizing the total flow time, subject to the constraints that (a) the sequence of machines for each job is prescribed; (b) a machine can process only one job at a time, and (c) the processing of a job on a machine (operation) can not be interrupted. Let $N_j = \{1(1)k\}$ denote the set of operations of job j, P_j a set of pairs of operations constrained by the precedence relations representing condition (a) as well as O_m a set of operations to be processed on machine $m \in M$. Further, let p_{jo} denote the given processing time and t_{jo} the variable start time of operation o of job j. The problem can then be stated as follows:

(P1) min

s.t.

(1) $t_{j1} \geq 0 \qquad j \in J$

(2) $t_{ji} - t_{jo} \geq p_{jo} \qquad o, i \in N_j,\ (o,i) \in P_j,\ j \in J$

(3) $t_{hi} - t_{jo} \geq p_{jo} \ \vee\ t_{jo} - t_{hi} \geq p_{hi} \qquad h, j \in J,\ h \neq j,\ o \in N_j,\ o, i \in O_m,\ m \in M$

The objective is minimizing the total flow time of the jobs. The processing of the given order stock starts at the starting point 0 (constraint (1)). While constraint (2) specifies the precedence relations of the operations, constraint (3) ensures that operations on a machining centre do not overlap in time.

This classical formulation of the Job Shop Scheduling Problem is based on the assumption that for each part type or production order (job) there is only one processing plan, which prescribes the sequence of machines *and* operations. However, with modern manufacturing equipment and shop floor control systems usually alternative processing plans are available, especially since the versatility of the machining centres has increased. A highly automated form of these multi-purpose production facilities are **Flexible Manufacturing Systems** (FMSs), which can be defined as integrated systems of computer numerically controlled machine tools, each with automatic tool-changing capabilities, connected by automated material handling, all controlled by a "Leitstand" (Stecke 1983). The utilization of an FMS in a Job Shop environment can basically be characterized by three kinds of flexibility, which are closely related to each other:

- **Process flexiblity**

  The process flexibility describes the range of part types or jobs, which can be processed on the FMS. For *dedicated* FMSs, which are designed to manufacture a small part type spectrum almost completely, the Job Shop Scheduling is rather simple, because the Job Shop and the FMS can be scheduled separately. The main FMS scheduling problem is to determine an appropriate part input sequence into the system (Stecke 1992). For *integrated* FMSs the allocation and sequencing of production orders is quite complex, since the scheduling of the FMS and the Job Shop should be done simultaneously considering several ways of producing a given set of jobs or part types. In general the process flexibility of an FMS is determined by the tools and the machines available.

- **Routing flexibility**

  The routing flexibility measures the ability to perform operations by more than one machining centre in order to alleviate bottleneck machines or to handle machine breakdowns. Routing flexibility occurs within an FMS, if machining centres are tooled to a certain extent identically, as well as within a Job Shop, if machines with similar capabilities exist.

- **Machine flexibility**

  The machine flexiblity represents the versatility of the machining centres as well as the capability of performing many different operations. As a consequence of automatic tool interchange modern machining centres are able to process several operations with virtually no set-up times between operations. This versatility in conjunction with the automatic material handling of an FMS allows a considerable flexibility in assigning operations along with associated required tooling among the machines.

It can be summerised that in order to utilise these kinds of flexibilities of modern production facilities the conventional Job Shop Scheduling Problem has to be extended to the so-called *loading problem* (Kusiak 1985), which involves the allocation of operations and tools among the machining centres subject to technological and capacity constraints.

In literature the loading problem has primarily been addressed by a large number of research studies and experimental investigations on isolated FMS production planning problems [Stecke & Solberg 1981; Werra & Widmer 1990; O´Grady & Menon 1987; Liang & Dutta 1993). In this context various models have been proposed considering different loading and tool-managment strategies. Other approaches concerning the allocation of workload between conventional job shops and FMS´s use Linear Programming models or Queueing Networks (Avont & Gelders & Van Wassenhove 1988; Tetzlaff & Pesch 1994; Calbrese & Hausman 1991).

Based on the previous formulation (P1) the **Extended Job Shop Problem** can be described as follows. Contrary to the conventional Job Shop Problem constraint (a) only prescribes the sequence of operations for each job j. Let $x_{jom} \in \{0,1\}$ denote the variable assignment of operation o of job j to a machining centre m with $m \in A_{jo} \subseteq M$ a set of alternative machines for operation o of job j. Let C_m^M represent the given processing time available at machining centre m and C_m^T the capacity of the tool magazin at machine m in number of slots. Further, let d_o denote the number of tool slots needed by operation o and Y a large number. The problem can then be stated as follows:

(P2) min

s.t.

(1) $t_{j1m} \geq a_j$ $\quad j \in J, m \in A_{jo}$

(2) $\quad o, i \in N_j, (o,i) \in P_j, j \in J$

(3) $\quad h, j \in J, h \neq j, o \in N_j,$ $\quad o, i \in O_m, m \in M$

(4) $\quad o \in N_j, j \in J$

(5) $o \in N_j$, $j \in J$, $m \in A_{jo}$

(6) $o \in N_j$, $j \in J$, $m \in M$

(7) $m \in M$

Here the objective is to minimize the total flow time of the jobs considering the routing flexiblity in a Job Shop environment. Constraint (1) means that the processing of a job can start at the release date a_j of the job j and constraint (2) ensures the sequence of operations for each job. Equivalent to the conventional Job Shop Problem constraint (3) specifies that operations on a machining centre do not overlap in time. Constraint (4) ensures that all operation assignments are feasible, meaning that only one out of the set of alternative machining centres is choosen. The available processing time and magazine capacity restrictions are represented by constraints (6) and (7), respectively.

The conventional Job Shop Problem (P1), which was shown to be NP-complete (Garey & Johnson 1979), is among the hardest combinatorial optimization problems. As a consequence many heuristic procedures have been designed in order to produce an acceptable schedule within a reasonable period of time (Pinedo 1995).

2. General Approach

In this paper a new approach, called Acceptance Algorithms, is applied to the extended Job Shop Problem, which are based on the popular heuristic of local neighbourhood search (Laarhoven et al. 1992). In contrast to the iterative improvement approach these procedures also accept inferior solutions for further neighbourhood search, in order to escape local optima and to increase the likelihood of finding the global optimum. The Acceptance Algorithm procedure can be described as follows:

Begin
Generate S_j; - *initial seed schedule generated by a heuristic*
for k := 1 to K do - *number of searches*
begin
Generate S_i from N_j; - *N_j is the neighbourhood of schedule S_j (transition mechanism)*

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

end;

End;

A well known example of this approximation procedure is the Simulated-Annealing Algorithm (Kirkpatrick et al. 1983; Cerney 1985). Similar approaches are the Threshold-, Great Deluge- and Record-to-Record-Travel - Algorithm (Dueck & Scheuer 1989). These procedures have been sucessfully applied to a wide range of combinatorial optimization problems in such diverse areas as computer and VLSI design, facilities layout, distribution planning and production scheduling.

In this context several research studies have emphazised the fact that the performance of Acceptance Algorithms depends on the

1. *initial seed schedule,*
2. *neighbourhood search (transition mechanism) and*
3. *conditions of acceptance (cooling schedule)*,

which will be described in detail in the following sections.

2.1 Initial Seed Schedule

An initial seed schedule can be provided by any good heuristic method. For Job Shop Scheduling Problems various *priority dispatching rules* have been put forward such as FCFS (first come first serve), SPT (shortest processing time), EDD (earliest due date), MWKR (most work remaining) and LWKR (least work remaining). 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 they provide the advantage of low computation time. On the other hand they rarely find the optimum solution, since the set of alternative dispatchable operations is decreasing during the procedure and therefore there are often unfavourable decisions made towards the end. Further, there is no dispatching rule that seems to perfom best for all problem environments (Baker 1984).

Another heuristic method in order to obtain a feasible starting solution for conventional Job Shop Scheduling Problems is the *Shifting Bottleneck Procedure*, which was used by Matsuo et al. for their Controlled Search Simulated Annealing Method (Matsuo et al. 1988). The Shifting Bottleneck Approach, which was developed by Adams, Balas and Zawack is strongly tailored to the Job Shop Scheduling Problem and is the best greedy construction method minimizing the makespan known so far (Adams et al. 1988). This approximation approach schedules one machine at a time, consecutively, taking each time the machine identified as a bottleneck among the machines not yet sequenced. Once a bottleneck machine is scheduled, the machines previously scheduled are reoptimized one by one. This initial heuristic procedure provides the advantage of good seed schedules by low computational time, but also the disadvantage of being caught in a local optima environment, especially if the initial acceptance probability is low.

Therefore other authors (Laarhoven et al. 1992) use *randomly generated inital schedules*, arguing that the approach should be independent from the starting solution. This independence is primarily achieved by a relatively high initial acceptance probability, which is lowered during the procedure. In comparison to the Shifting Bottleneck Method described before, this approach requires more computational time, but yields better results if running time is of no concern (Blazewicz et al. 1993; Aarts et al. 1994).

In the following we will start with initial seed schedules contructed by dispatching rules. These dispatching rules are also used for further neighbourhood search, which will be described below.

2.2 Neighbourhood Structure

For the performancce of Acceptance Algorithms the definition of *neighbourhoods* of seed solutions is critical. A neighbourhood NS_i of a seed schedule i is defined as a set of schedules that can be reached from i by exactly one transition. As several research studies have shown it is useful to aim for reasonable small and simple neighbourhoods to improve the performance of these algorithms (Reeves 1993).

The most common strategy to construct neighbouring solutions for Job Shop Scheduling Problems is to *pairwise interchange* operations on a machine. In order to minimize the makespan some authors suggests, 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 (Balas et al. 1995). A more restricted neighbourhood search is used by Laarhoven et al., swapping only *adjacent* operations on a critical machine (Laarhoven 1992). An additional limited neighbourhood 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 (Matsuo et al. 1988).

The major disadvantage of these neighbourhood structures is, that the search procedure is based on the large set of semi-active schedules. Further, the described neighbourhood search procedures are restricted to the objective function of minimizing the makespan. Therefore we will apply a neighbourhood search, which is based on a small set of schedules and is flexible regarding the performance measurement of production control.

The applied *Neighbourhood Search* interchanges alternatively dispatchable operations, where the number of alternative operations depends on the data structure of the scheduling problem as well as on the dispatching rule used. Let S be a schedule created by a priority dispatching rule and let Q_{mt} denote a set of operations waiting in the queue to be processed on machining centre m in period t. Further, let τ denote the set of periods, where more than one job is to be processed on a machinining center or a job-predecessor of an operation is finished. The Neighbourhood Search procedure can then be stated as follows:

S = *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 $S_{new}{}'$;

end;

End;

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

$S_0 := S_{new}$;

The local neighborhood search can be truncated to nondelay schedules by focusing only on periods, where a machining centre becomes idle. This procedure leads not necessarily to an optimum schedule but decreases the number of local neighbourhoods to a large extent. In order to find the optimum solution, periods where a job predecessor of an operation is finished also have to be considered, enlarging the set of possible neighbourhoods to semi-active schedules.

As stated before, there are basically two kinds of flexibility, which have to be considered scheduling an FMS in a Job Shop environment. On the one hand the machine flexibility can be restored by implementing a transition mechanism that changes the sequence of jobs on a machining center (see figure 1).

Figure 1 shows the scheduling of 3 jobs in period t, which are ready for manufacturing, on machining center m by the SPT rule as well as neighbourhood search based on the machine flexibility.

On the other hand the routing flexibility can be restored by moving a job to an idle machine, where the operation can be performed, shown in figure 2.

Therefore both transition mechanisms (TM) are implemented in our neighbourhood search :

TM1 : Interchange operation o_1 originally scheduled by a dispatching rule and o_n waiting in queue, if o_n is an operation of a different job that can be processed on the same machine at the same starting point.

TM2 : Move one of these operations to an idle machining center, which can process the operation at the same starting point.

In contrast to the approaches described before, these transition mechanisms have the advantage of small neighbourhoods, which are independent of the chosen objective. Furthermore this method provides the advantage, that the local neighbourhood search is focused on good heuristic solutions using priority dispatching rules.

Apart from the definition of neighbourhood structures there are two strategies to select neighbouring solutions. The standard approximation algorithm procedure samples randomly from the neighbourhood of solutions, which will be done in the following. Other approaches suggested that sampling should be cyclic rather than random, to ensure that all neighbours are tried once before any are considered a second time (Reeves 1993). Whether the new solution proposal is the basis for further neighbourhood search, a so called seed solution, depends on the conditions of acceptance, which are described in the next section.

2.3 Conditions of acceptance

The conditions of acceptance control neighborhood search by defining the space of feasible cost-increasing transitions. In general these conditions can be defined either stochastically or deterministically. In the following, the stochastic Simulated Annealing approach (Kirkpatrick et al., 1983) is compared to the deterministic procedure of Threshold Accepting (Dueck & Scheuer 1989).

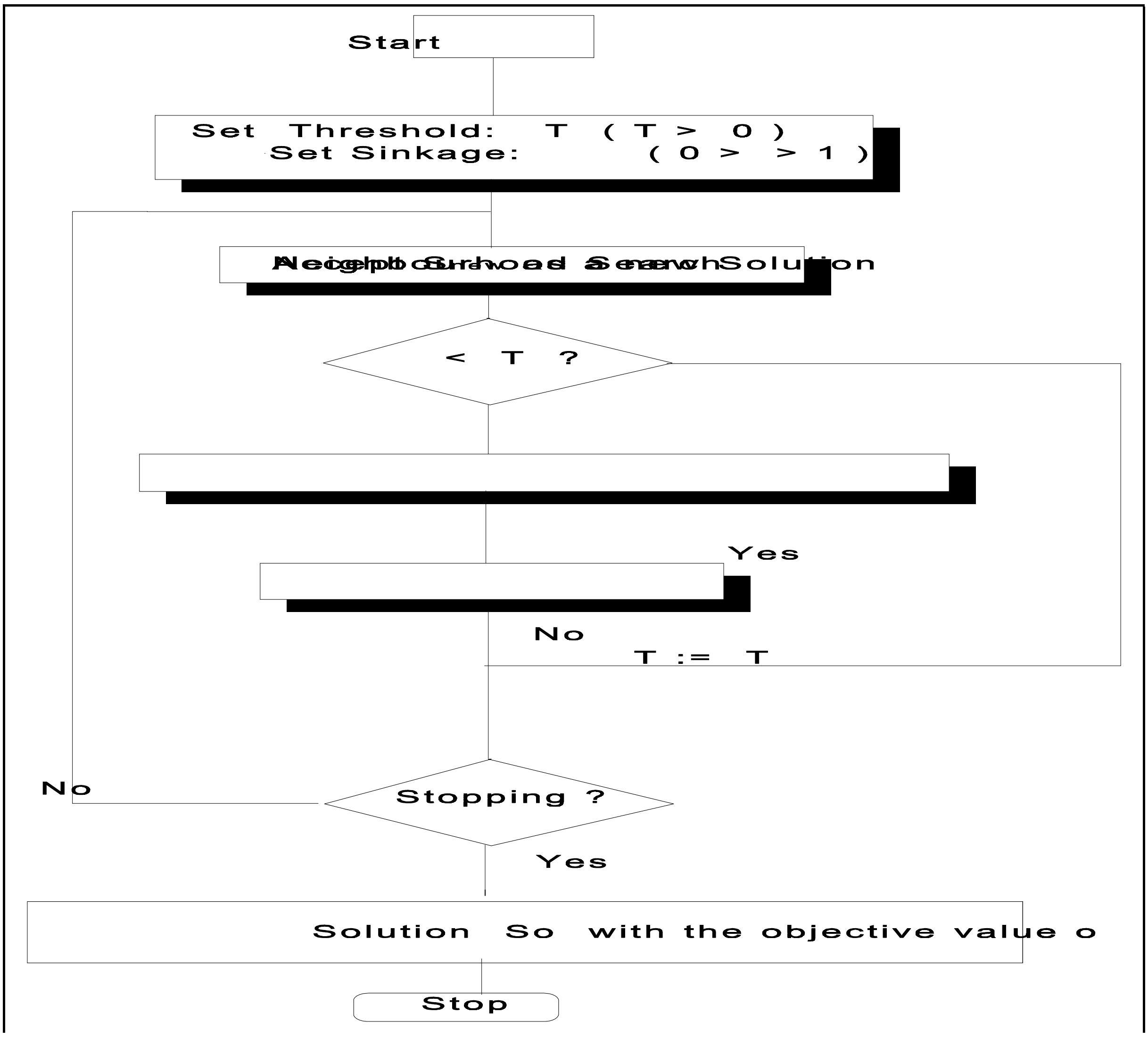
A Simulated Annealing Algorithm (SA) for a minimization problem with solution space S, objective function f, and neighbourhood structure NS can be stated as follows:

Select an initial Solution $S_0 \in S$;
Select initial control parameters $c > 0$; $0 \leq \beta \leq 1$;
Repeat
    Randomly select $S_{new} \in NS(S_0)$;
    $\Delta := f(S') - f(S_0)$;
    if $\Delta < 0$ *then* $S_0 := S_{new}$
        else generate random p uniformly in the range (0,1) ;
            if $p < e^{-\Delta/c}$ *then* $S_0 := S_{new}$;
    if S_{new} *is accepted then* $c := \beta \cdot c$;
Until stopping condition = true ;
S_0 *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 solution 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 to the data of the scheduling problem. If the final solution is to be independent of 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 neighbouring solutions at the beginning.

The Threshold approach (TA) resembles the desribed SA in its structure. In contrast to the SA, the TA employes a deterministic Threshold (T) and Sinking rate (β) as control parameters.

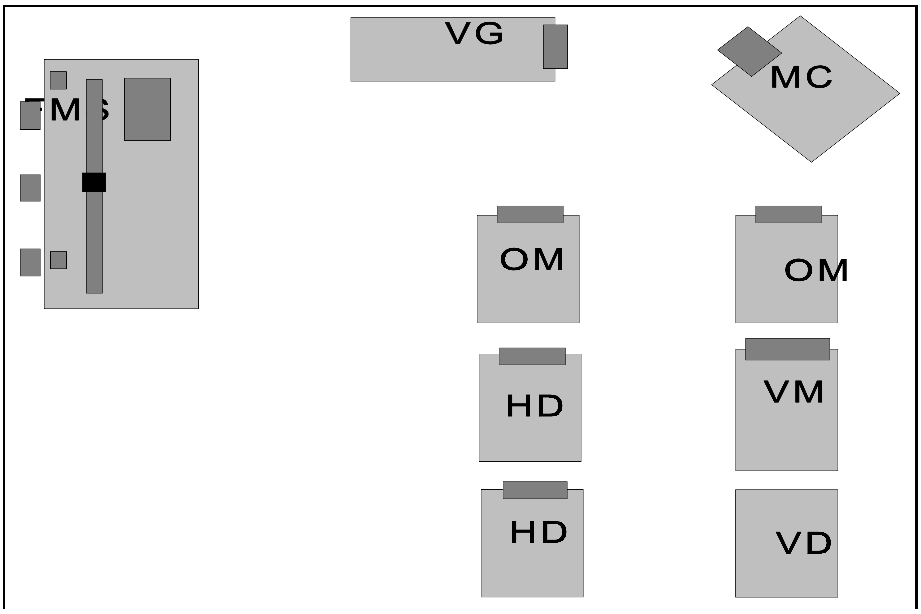
While the Threshold describes the area of cost-increasing transitions, the sinking-rate indicates the speed of narrowing this area.



The TA as well as the SA-Algorithm have been tested on several reference data, especially on Job Shop (JSP) and Traveling Salesman Problems (TSP) (Aarts et al. 1994). As a result the TA approach barely outperformed the SA-Algorithm for TSP´s, because in the deterministic version the conditions of acceptance can be calculated much faster and therefore more neighborhood searches can be performed in the same time. For more complex problems, such as the JSP, the SA, with its sophisticated acceptance probability, leads to much better results.

3. Industrial application and computational results

The described approach has been applied to a production facility of a major German manufacturer of cigarette and packaging machines, consisting of an FMS, which is embedded in a job shop production for heavy parts (see figure 4).



While most of the machine tools of the investigated Job Shop production are milling and drilling machines, there are two identical horizontal 3-axis CNC drilling machines (HD), each with a tool magazine capacity of 12 - 24 tools. Further there are two identical 5-axis CNC Omnimills (OM) tooled with an average set of 24 tools. In addition the production facility consists of a conventional vertical drilling machine (VD), a CNC vertical milling machine (VM) and a CNC vertical grinding machine, which is used to finish the parts. Extreme large part types are processed on a special CNC milling and drilling machining centre (MC), while smaller and regular parts can be be processed on the FMS.

The integrated FMS includes three Fritz Werner 2.6 TC CNC machine tools, which are horizontal 3-axis drilling and milling machines, connected by a monorail conveyor. There are three load/unload stations and 15 buffer stations. Also each CNC machine has one local input and output buffer.

The production program of the job shop includes a wide range of part types, such as housing, bearings, holders etc., which are assembled on the next production stage. The part types are made of aluminium, cast iron, steel or plastic, with average production requirements of 15 parts.

The industrial application of Acceptance Algorithms in loading and scheduling an embedded FMS has been analysed by a simulation study. Our simulation program replicates the performance of the described real world production facility, considering the following loading and scheduling constraints:

· *Machine pooling*

The two horizontal drilling machines and the two Omnimills as well as all three CNC machine tools of the FMS are tooled with a set of standard tools, which are frequently used during the operation of the FMS. For the rest of the tools the strategy of difference tools is applied, meaning that only extra tools needed for the upcoming job (part type) are loaded, while the tools, which are not used during the next production orders are unloaded. The *pooling by standard tools*, which enables the machining centres to perform the same standard operations, provides scope for several feasible processing routes within the Job Shop and the FMS.

· *Tool magazine capacity*

Each CNC machine tool of the FMS is equipped with a local tool magazine, that has a capacity of 108 slots. On the other hand the local tool magazines of the stand alone CNC machines have a capacity of 50 slots. Further there is an average number of two sister (duplicate) tools available.

· *Day shift capacity*

The production runs on a daily one-shift basis, with 8 hours per shift. The scheduling process of the investigated Job Shop is strongly connected with the assignment of shifts. For technical and computational reasons a job should therefore be completed within a day shift.

· *Dynamic tool movement*

The tools are transfered from one machining centre to another with an average changeover time (including inspection) of 30 to 60 minutes. Within the FMS the tools are transfered automatically by the monorail conveyor taking about 30 minutes, while for stand alone CNC machining centres the setup is done manually, which takes about 60 minutes.

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 is to reduce the mean flow time of the production orders in consideration of their due dates, which are set by the central MPC-System. Therefore we define a combined objective function (3), which minimizes the mean flow time (1) and the mean tardiness (2) of the production orders, using the following symbols :

p_{jo} = processing time for operation o of job j
w_{jo} = waiting time of job j preceding operation o

d_j = due date of job j

r_j = release time of production order j

n = number of completed jobs

α = weight of the objectives with $0 \leq \alpha \leq 1$

(1) Mean Flow Time (MFT)

(2) Mean Tardiness (MT)

(3) Combined Objective Function (COF)

The input data of the simulation program is based on 4 exact production programs, consisting of 20 to 30 production orders, as well as on the route sheets of the jobs. The number of operations per production order ranged from 2 to 8, while each operation can be performed on 1 to 3 machining centres. On average there were 330 operations to be scheduled within the planning horizon of one week. For the combined objective function, α is chosen according to the decision of the manufacturing firm (see table 1), giving the mean tardiness a significantly higher weight than the mean flow time. Although, since the data has not been normalized, the implicent weight on the mean flow time and mean tardiness in the combined objective function are about equal for the investigated production programs.

Table 1. Initial Solution by the SPT-rule.

Problem	**α**	**MFT [minutes]**	**MT [minutes]**	**COF**
prg01	0.28	1235.17	347.33	600.00
prg02	0.04	1100.76	43.56	85.00
prg03	0.12	493.85	58.40	109.89
prg04	0.07	904.30	63.30	125.88
Mean	0.13	933.52	128.65	230.19

On the basis of these starting solutions the Simulated Annealing- and Threshold-Algorithm were applied. There were 28 different "macroruns" to compare, each representing a different combination of approximation algorithm and parameter configuration. The probabilistic nature of the local search procedures makes it necessary to carry out multiple runs on the same problem instance in order to get meaningful results. In this simulation study each "macrorun" consits of three regular simulation runs for one parameter setting (Tab. 2). To ensure an effective comparison between the Simulated Annealing and Threshold Accepting approach, both approximation algorithms are based on the same cooling schedule. The initial control parameter of the Simulated Annealing procedure (C) as well as the Threshold (T) are defined as a percentage of the objective value of the starting solution ranging from 3 % to 1 %. During the neighbourhood search the control parameters of the Simulated Annealing and Threshold Accepting are lowered by the factor β, as described in the previous section.

Tab. 2. Cooling schedule of the approximation algorithms.

Parameter configuration	Initial C / T	β
1	3 %	0.95
2	2 %	0.97
3	1 %	0.99

These parameter configurations cover a wide range of the solution space, while configuration (1) accepts major cost-increasing transitions configuration (3) allows only minor uphill moves. Next to the cooling schedule the stopping criteria of the approximation algorithms has to be defined. For the investigated instances a simulation run was aborted after the local neighborhood had been searched randomly for 3 times without any improvement of the best solution. The size of the local neighbourhood of a solution is thereby defined as the number of periodes, where more than one job can be processed on a machining centre or a predecessor of an operation is finished.

The results of the different acceptance algorithms are summarised in Table 3. The performance of the approximation algorithms is measured by the average COF-values [$Mean\_S_{best}$] found by the local search procedures on all instances using the described parameter settings. $Mean\_t_1$ is the mean CPU-time of one simulation run in seconds on a 486 PC (60 Mhz), while $Mean\_t_0$ describes the average time needed to find the best solution. Table 3 also contains the results obtained by the iterative improvement approach, which allows only cost-decreasing transitions during the search procedure and is based on the same neighbourhood structure as the Simulated Annealing and Threshold Accepting approach. The simulation results show that the classical iterative improvement approach is easily outperformed by the approximation algorithms. For instance, the average best solution obtained by the Threshold Accepting Algorithm is 9 % better than the one obtained by iterative improvement with about the same computing time.

Table 3. Performance of the approximation algorithms

Heuristic	$Mean\_S_0$	$Mean\_S_{best}$	$Mean\_t_0$	$Mean\_t_1$	% Dv_S_0
Simulated Annealing	230.19	125.85	1585.43	3569.90	- 45.32
Threshold Accepting		139.46	1688.45	3619.43	- 39.41
Iterative Improvement		161.02	1572.25	3509.65	- 30.04

Comparing the results of the different approximation algorithms it can be observed that for investigated instances the Simulated Annealing approach in combination with the SPT-rule performed best, improving the mean starting solution of the dispatching rule by 45.32 %. The average computation time in order to find an approximation of the optimal solution [$Mean\_t_0$] is 1615.37 seconds.

4. Conclusions

The application of acceptance algorithms improves the mean flow time and tardiness of the orders significantly in comparison to the traditional approach based on priority dispatching rules. In order to obtain an efficient method, which is flexible regarding the objective function, we combined priority rules and approximation algorithms. Further we implemented transition mechanisms to restore the machine and routing flexibility. For the investigated data sets of a Job Shop production with an embedded FMS the Simulated Annealing approach, with its sophisticated conditions of acceptance, performed best.

References

Aarts, E. H.; Van Laarhoven, P. J.; Leenstra, J. K.; Uldner, N.L.; "A Computational Study of Local Search Algorithms for Job Shop Scheduling", *ORSA Journal of Computing*, Vol. 6, No. 2, pp. 118 - 126 (Spring 1994).

Avonts, L. H.; Gelders, L. F.; Van Wassenhove, L. N., "Allocation work between an FMS an a conventional Jobshop: A Case Study, *European Jounal of Operational Research*, Vol. 33, pp. 245-256 (1988).

Adams, J.; Balas, E.; Zawack, D.; "The Schifting Bottlenck Procedure for Job Shop Scheduling", *Management Science*, Vol. 32, No. 3, pp. 391-401 (March 1988)

Baker, K. E., "Sequencing Rules and Due-Date Assignment in a Job Shop", Management Science, Vol. 30, No. 9, pp. 1093-1105 (September 1984).

Balas, E.; Vazacopoulos, A., "Guided Local Search with Schifting Bottleneck for Job Shop Scheduling", *Working Paper* 95-01, Samuel J. Silberman College of Business Administration, Faileigh Dickinson Universtiy, Teaneck, NJ 07666 (1995).

Blazewicz, J.; Ecker, K.; Schmidt, G.; Weglarz, J.; *Scheduling in Computer and Manufacturing Systems*, Springer, Berlin (1993).

Calabrese, J. M., Hausman, W. H., "Simultaneous Determination of Lot Sizes and Routing Mix in Job Shops", Management Science, Vol. 37. No. 8, pp. 1043-1057 (August 1991).

Cerney, V., "Thermodynamical Approach to the Traveling Salesman Problem : An Efficient Simulation Algorithm", *Journal of Optimization Theory and Applications*, Vol. 45, No. 1, pp. 41 - 51 (January 1985).

Dueck, G. and Scheuer, T.; "Threshold Accepting", *Journal of Computational Physics*, Vol. 90, pp. 161 - 175 (1990).

Dueck, G.; "New Optimization Heuristics", *Journal of Computational Physics*, Vol. 104, pp. 86 - 92 (1993).

Garey, M. R., Johnson, D. S., *Computers and Intractability*, San Francisco 1979.

Hansmann, K.-W. and Kleeberg, K.; "Extension of Capacity Oriented Scheduling to Flexible

Manufacturing Systems", *Paper for the TIMS/ORSA meeting* in Rio de Janeiro, Brazil 1991.

Hansmann, K.-W. and Höck, M., "Application of new Heuristics to Scheduling a Flexible Manufacturing System", *Working Paper No. 1*, Department of Industrial Management, University of Hamburg, 20146 Hamburg 1994.

Jaikumar, R. and Van Wassenhove, L. N.; "A Production Planning Framework for Flexible Manufacturing Systems", *Journal of Manufacturing and Operations Management*, Vol. 2, No. 1, pp. 52 - 79 (1989).

Kirkpatrick, S.; Gelatt, C.D.; Vecchi, M.P.; "Optimization by Simulated Annealing", *Science*, Vol. 220, pp. 671 - 680 (May 1983).

Kusiak, A.; "Flexible Manufacturing Systems: A Structural Approach", *International Journal of Production Research*, Vol. 23, No. 6, pp. 1057-1073 (1985).

Van Laarhoven, P. J.; Aarts, E. H.; Lenstra, J. K.; "Job Shop Scheduling by Simulated Annealing", *Operations Research*, Vol. 40, No. 1, pp. 113 - 124 (January 1992).

Liang, M.; Dutta, S. P., "Solving a Combined Part-Selection, Machine-Loading, and Tool-Configuration Problem in Flexible Manufacturing Systems", Production and Operations Management, Vol. 2, No. 2, pp. 97-113 (Spring 1993).

Matsuo, H., Suh, C. J., Sullivan, R. S., "A controlled Search Simulated Annealing Method for the general Jobshop Scheduling Problem", *Working Paper 3-4-88*, Department of Management, Graduate School of Business, The University of Texas Austin, Austin, TX 78712 (1988).

O´Grady, P.J. and Menon, U.; "Loading a Flexible Manufacturing System", *International Journal of Production Research*, Vol. 25, No. 7, pp. 1053 - 1068 (1987).

Pinedo, M., *Scheduling*, Englewood Cliffs 1995.

Reeves, C. (Ed.); *Modern Heuristic Techniques for Combinatorial Problems*, Blackwell Scientific Publications, New York, NY (1993).

Stecke, K. E.; "Planning and Scheduling Approaches to Operate a Particular FMS", *European Journal of Operational Research*, Vol. 61, pp. 273-291 (1992).

Stecke, K. E.; "Formulation and Solution of Nonlinear Integer Production Planning for Flexible Manufacturing Systems", *Management Science*, Vol. 29, No. 3, pp. 273 - 288 (March 1983).

Stecke, K. E., "Procedures to Determine Part Mix Ratios for Independenet Demands in Flexible Manufacturing Systems", IEEE Transactions on Engineering Managment, Vol. 39, No. 4, pp. 359-369 (November 1992).

Stecke, K. E. and Solberg, J. J., "Loading and Control Policies for a Flexible Manufacturing System", *International Journal of Production Research*, Vol. 19, pp. 481-490 (1981).

Tetzlaff, U. W., Pesch, E., "Optimal Workload Allocation between a Job Shop and an FMS", *Working Paper*, George Mason University, School of Business Administration, Fairfax, Virginia 220303-444 (August 1994).

Vakharia, A. J. and Chang, Y. L.; "A Simulated Annealing Approach to Scheduling a Manufacturing Cell", *Naval Reasearch Logistics*, Vol. 37, pp. 559-577 (1990).

Vollman, T. E.; Berry, W. L.; Whybark, D. C., *Manufacturing Planning and Control Systems*, 3rd Ed., Irwin, Homewood, Ill. (1992).

Werra, D.; Widmer, M., "Loading Problems with Tool Management in Flexible Manufacturing Systems", *International Journal of Flexible Manufacturing Systems*, Vol. 3, pp. 71-82 (1990).