

Drexl, Andreas; Kolisch, Rainer

Working Paper — Digitized Version

Model-based assembly management in machine tool manufacturing

Manuskripte aus den Instituten für Betriebswirtschaftslehre der Universität Kiel, No. 346

Provided in Cooperation with:

Christian-Albrechts-University of Kiel, Institute of Business Administration

Suggested Citation: Drexl, Andreas; Kolisch, Rainer (1994) : Model-based assembly management in machine tool manufacturing, Manuskripte aus den Instituten für Betriebswirtschaftslehre der Universität Kiel, No. 346, Universität Kiel, Institut für Betriebswirtschaftslehre, Kiel

This Version is available at:

<https://hdl.handle.net/10419/155419>

Standard-Nutzungsbedingungen:

Die Dokumente auf EconStor dürfen zu eigenen wissenschaftlichen Zwecken und zum Privatgebrauch gespeichert und kopiert werden.

Sie dürfen die Dokumente nicht für öffentliche oder kommerzielle Zwecke vervielfältigen, öffentlich ausstellen, öffentlich zugänglich machen, vertreiben oder anderweitig nutzen.

Sofern die Verfasser die Dokumente unter Open-Content-Lizenzen (insbesondere CC-Lizenzen) zur Verfügung gestellt haben sollten, gelten abweichend von diesen Nutzungsbedingungen die in der dort genannten Lizenz gewährten Nutzungsrechte.

Terms of use:

Documents in EconStor may be saved and copied for your personal and scholarly purposes.

You are not to copy documents for public or commercial purposes, to exhibit the documents publicly, to make them publicly available on the internet, or to distribute or otherwise use the documents in public.

If the documents have been made available under an Open Content Licence (especially Creative Commons Licences), you may exercise further usage rights as specified in the indicated licence.

No. 346

**Model-Based Assembly Management
in Machine Tool Manufacturing**

Drexl, A. and R. Kolisch

June 1994

Abstract: In machine tool assembly several make-to-order products have to be manufactured simultaneously in a multi-stage production system. Due to customer requirements batches are very small. A process plan specifies the necessary machines, tools, auxiliary devices as well as the operations to be performed. We consider the small batch machine tool assembly problem with alternative process plans and multiple resource-constraints. This problem basically relates to nonpreemptive multi-mode resource-constrained project scheduling. Two models, each with a related class of methods, are considered to be most important. The first minimizes the makespan with respect to precedence constraints and a given set of renewable resources. It is designed for (a-)periodic (re-)scheduling as well as to help the decision maker in determining appropriate schedules with associated release dates and deadlines for machine tool orders in acquisition. The second, a new type of model, minimizes renewable resources required in addition to the available stock in order to satisfy deadlines. Among other purposes, it should help to deal with several kinds of interferences (machine breakdowns, etc.). Beyond these scheduling components the system provides an ordering component as well as a layout component.

Keywords: Machine tool manufacturing, assembly of small batches, project scheduling models, layout component, ordering component, leitstand

1. Introduction

Recently the so-called "leitstand-concept" has been introduced in order to combine attractive shop scheduling features [cf., e.g., **Adelsberger and Kanet (1991)**, **Kanet and Sridharan (1990)**]. Regarding the DDM-paradigma [cf., **Watson and Sprague (1989)**], a leitstand is a decision support system with dialog capabilities, a database as well as a model / method base. In general, leitstands developed so far deal with operational scheduling for make-to-stock production of large batches and thus are not suitable for small batch make-to-order production [cf., e.g., **Chryssolouris et al. (1991)**, **Speranza and Woerlee (1991)**].

In machine tool assembly several make-to-order products have to be manufactured simultaneously in a multi-stage production system. Due to customer requirements batches are very small. A process plan specifies the operations to be performed as well as the necessary machines, tools and auxiliary devices [cf. **Ahn and Kusiak (1991)**]. Most traditional scheduling approaches are based on the assumption that for each job one process plan only is available.

Here we consider the small batch machine tool assembly problem with alternative process plans and multiple resource-constraints. This problem basically relates to nonpreemptive multi-project multi-mode resource-constrained project scheduling as follows: Each product corresponds to a project, i.e. we are dealing with multi-project scheduling. Production uses renewable resources. The manufacturing of each product

requires the assembly of parts. Each part requires a number (and sequence) of operations to be performed specified by a process plan. Here we consider the case where alternative process plans are available for each part, one of which has to be chosen. The part level in production scheduling corresponds to the job level in project scheduling and process plans are shortly denoted as modes. For the ease of readability we will use the words project, mode and job here, too. Note that a manufacturing system with alternative process plans is supposed to increase throughput rate via eliminating bottlenecks and handling machine breakdowns [cf. Ahn and Kusiak (1991)].

The focus here is not on the operations, but on the part level, i.e. we simultaneously have to choose one mode per job and the sequence of jobs. Thus we are primarily concerned with medium-term and not with short-term (operational) scheduling. This approach seems to be adequate in manufacturing situations such as small batch machine tool assembly where on the operational level a lot of work has to be done manually.

This production scheduling approach was implemented in a leitstand within an EUREKA-project. The acronym of this project is PRISMA (Production Improvement of Small Batch Machine Tool Assembly). PRISMA has been initiated by the WZL-Laboratory, RWTH Aachen, in cooperation with several european machine tool manufacturers, software companies as well as research institutes [cf. Drexel et al. (1994)]. The main features of the PRISMA-leitstand are:

- (a) A hierarchical working scheme for either long-term planning based on aggregate data or medium-term scheduling.
- (b) A schedule generation component based on models and methods; graphic components for editing projects and providing several visual representations of schedules; evaluation component for helping to analyze and compare the performance characteristics of generated schedules.
- (c) A layout component (machine tools become quite large as they proceed through the assembly) treating the shop floor area as a scarce resource in the model; graphic visualisation of the results.
- (d) An ordering component providing rules for "just-in-time" parts (expensive parts or parts with high demand for shop floor area), critical parts (received too late in the past), uncritical as well as standard parts.
- (e) Running as a stand alone version or receiving data from an MRP II system; current update of shop floor information (finished jobs, missing parts, machine break down etc.) via an interface to the data collection system.
- (f) C implementation on Sun Sparcstations under UNIX; INGRES relational database system.

Note that the features (c) and (d), quite unusual for scheduling-systems, give regard to essential characteristics of machine tool assembly. Feature (e) in connection with a

rolling horizon for (a-)periodic (re-)scheduling enables the leitstand to treat real life (stochastic) problems via deterministic models.

The PRISMA-project in total is subdivided into two parts:

- The Computer Aided Assembly Planning system (CAAP); the focus of the CAAP is on the (automatic) collection and preparation of the database for construction, engineering, planning and scheduling.
- The Assembly Management System (AMS); the focus of the present paper is on the AMS.

The paper is organized as follows: Section 2 deals with the project scheduling models and methods. In Section 3 the layout component is described. Section 4 is devoted to the ordering component. Additional features are provided in Section 5. A short summary of essentials and an outline of future work is given in Section 6.

2. Scheduling Models and Methods

In the following we present the scheduling models and methods being essential for the PRISMA-system. For the sake of simplicity and ease of notation the models are presented mathematically for the single-project case only. The generalizations to the multi-project case are sketched out below.

2.1 Basic Assumptions

The PRISMA-system relies on the following **assumptions**:

A1. The whole system is designed for production scheduling in a **deterministic environment**. Thus, at a specific point of time, all data (structure of projects, duration and resource usage of jobs, availability of resources etc.) are supposed to be well-known.

A2. A **hierarchical approach**, which distinguishes four levels, is used:

Customer level: Customer-specific products are characterized via functionality, design etc.; with each product a release date and a deadline are associated.

Upper assembly level: Each product is roughly subdivided in, e.g., three phases (initial assembly, main assembly, final assembly); with each phase milestones are associated.

Lower assembly level: Each phase is decomposed by considering the assembly process thoroughly; here (assembly) jobs, their precedence relations (network), requirements for machines, tools, auxiliary devices etc. are identified; each job (assembly task) may be processed based on one of the available modes.

Operational level: Each mode is characterized in detail via a process plan, describing the set of operations to be performed.

- A3. The user may define **two planning horizons**, one for long-term scheduling (usually based on aggregate data / upper assembly level) and one for medium-term scheduling (based on detailed data / lower assembly level).
- A4. The whole system works within a **rolling schedule environment** with part of the horizon to be frozen (fixed partial schedule); this enables the user to (a-)periodic (re-)scheduling based on actual data.

Note that the models and methods to be presented below are designed to support the user on the upper and lower assembly levels only. This implies that on the customer and the operational levels no methods for the (automatic) evaluation of alternatives, rather than a rich functionality as well as a simple and exhaustive access to (large) databases, are provided.

2.2 Critical Path Analysis and Resource Leveling

On the upper assembly level the following **simple methods** are provided to the user:

- (a) Perform a critical path analysis disregarding resource requirements. This method may be either used in order to determine "realistic" corner-dates or in order to analyse criticality based on given corner-dates.
- (b) Associated with a schedule determined according to (a) are (aggregate) resource requirement profiles. These profiles may be either leveled manually "phase by phase" or automatically using a method which shifts phases systematically within their slacks [cf. **Neumann (1975)**].

Resource leveling according to (b) tries to find an "even distribution" of resources required on the aggregate upper assembly level. In the Subsections 2.3 and 2.4 we are going to present two distinct models for the determination of schedules which take into account available resource amounts as well as required resources on the lower assembly level.

2.3 Resource-Constrained Scheduling Model

The situation considered on the lower assembly level may be described by the following **assumptions**:

- B1. The assembly structure of each of the products may be represented as a single project being depicted by an acyclic activity-on-node graph.
- B2. The project consists out of jobs $j = 1, \dots, J$ ($J \equiv$ unique sink). $\mathcal{P}_j(\mathcal{S}_j)$ is the set of all immediate predecessors (successors) of job j .

- B3. Job j may be performed in one of the modes (associated with a process plan) $m = 1, \dots, M_j$. Each job, once initiated in a specific mode, must be processed without interruption (nonpreemptive case).
- B4. Scheduling job j in mode m takes d_{jm} time units (duration).
- B5. There exists a set \mathcal{R} of renewable resources, where resource $r \in \mathcal{R}$ is available with κ_{rt} resource units in period t ; scheduling job j in mode m uses k_{jmr} units per period from resource r .

Note that models and methods with time-varying resource availabilities κ_{rt} are essential prerequisites for a rolling horizon implementation with fixed partial schedules.

In order to illustrate the assumptions B.3 to B.5 let us consider an example with $J = 7$ jobs and $|\mathcal{R}| = 3$ resources, which may be characterized as follows:

- Resource $r = 1$ is a (slow) "manual" milling machine, which is available with $\kappa_{1t} = 1$ units within every period t .
- Resource $r = 2$ is a (fast) numerically controlled milling machine, which is available with $\kappa_{2t} = 1$ units within every period t , too.
- Finally, two assembly workers are associated with resource $r = 3$, i.e. we have $\kappa_{3t} = 2$ units within every period t .

Table 1 provides the data associated with each job in detail, i.e. the number of modes per job, the mode-dependent job durations, and the resource demand of each job, respectively. Note that the process plan, i.e. the set of assembly operations associated with each job, is characterized via the resource demand.

Table 1: Illustrative Example

j	1	2	3	4	5	6	7
\mathcal{P}_j	\emptyset	{1}	{1}	{1}	{2}	{3,4}	{5,6}
m	1	1 2	1	1 2	1	1 2	1
d_{jm}	0	4 2	3	5 2	4	5 3	0
k_{jm1}	0	1 0	1	1 0	0	0 0	0
k_{jm2}	0	0 1	0	0 1	0	0 0	0
k_{jm3}	0	0 0	0	0 0	1	1 2	0

Minimization of the makespan is considered to be the objective function.

Let us assume that an upper bound (corresponding to a *feasible solution*) \bar{T} for the project makespan is known in advance. Then, critical path earliest and latest finish times

EF_j and LF_j may be computed with traditional forward and backward recursion based on $d_j := \min \{d_{jm} | m = 1, \dots, M_j\}$. EF_j and LF_j represent a bound on the maximum time interval within each job j has to be scheduled.

Using binary variables $x_{jmt} = 1$, when job j is scheduled in mode m and finished in period t ($x_{jmt} = 0$, otherwise) the following model arises:

$$\min \quad \sum_{m=1}^{M_j} \sum_{\tau=EF_j}^{LF_j} \tau \cdot x_{jmt} \quad (1)$$

$$\text{s.t.} \quad \sum_{m=1}^{M_j} \sum_{\tau=EF_j}^{LF_j} x_{jmt} = 1 \quad (\forall j) \quad (2)$$

$$\sum_{m=1}^{M_i} \sum_{\tau=EF_i}^{LF_i} \tau \cdot x_{im\tau} \leq \sum_{m=1}^{M_j} \sum_{\tau=EF_j}^{LF_j} (\tau - d_{jm}) x_{jmt} \quad (\forall j, \forall i \in \mathcal{P}_j) \quad (3)$$

$$\sum_{j=1}^J \sum_{m=1}^{M_j} k_{jmr} \sum_{\tau=t}^{t+d_{jm}-1} x_{jmt} \leq \kappa_{rt} \quad (\forall r, \forall t) \quad (4)$$

$$x_{jmt} \in \{0, 1\} \quad (\forall j, \forall m, \forall t) \quad (5)$$

(1)-(5), shortly denoted as **MSM** (= makespan model), has the following characteristics: (1) requires the unique sink to be completed as early as possible and thus minimizes the project's makespan. (2) requires completion of each job j in exactly one of the available modes within the feasible time interval $[EF_j, LF_j]$. (3) states precedence relations between jobs. (4) secures the schedules to be feasible with respect to renewable resources in each period.

Disregarding due dates or deadlines minimization of the makespan is considered to be the appropriate objective function due to the following reasons: (a) Loosely speaking, minimization of makespan reduces "complexity", i.e. manufacturing "one product after the other" is supposed to be less complex than producing two in parallel. (b) In practice nearly the whole contractual price of a product will be paid with delivery; a short makespan thus reduces the cost of the capital tie-up. (c) Short makespan reduces uncertainty of data. (d) The risk of lateness is minimized by providing slack between completion time and deadline. (e) Full usage of renewable resources in "early" periods frees them for make-to-order products which (hopefully) arrive later on (opportunity costs).

In the PRISMA-system, a multi-project generalization of MSM is considered; it is based on the following **additional assumptions**:

B6. P projects have to be scheduled; p ($= 1, \dots, P$) denotes a specific project. Project p has the release date ρ_p and the deadline δ_p ; $w_p := [\rho_p, \delta_p]$ denotes the manufacturing time window of product p .

B7. Project p consists out of J_p partially ordered jobs. W.l.o.g. we assume that each project has one unique source (start job) and one unique sink (finish job).

B8. We assume one additional (super-)source ($j = 1$) and one additional (super-)sink ($j = J$). Furthermore jobs are (re-)labeled consecutively, i.e.

$$j = 1, 2, \dots, J_1 + 1, J_1 + 2, \dots, \sum_{p=1}^P J_p + 2;$$

thus the job number (index) j identifies the related project exactly.

B9. The project-specific time windows are incorporated as follows: A temporal constraint between job 1 and the start job j (with $\mathcal{P}_j = \phi$) of project p is modeled as an arc $(1, j)$ with weight τ_{1j} corresponding to the release date ρ_p (minimum finish-to-start lag). In addition, a temporal constraint between the finish job h (with $\mathcal{S}_h = \phi$) of project p and job J is modeled as an arc (h, J) with weight τ_{hJ} corresponding to $\max \{\delta_\pi \mid \pi = 1, \dots, P\} - \delta_p$, i.e. the "distance" to the greatest deadline, once more denoting a minimum finish-to-start lag [w.r.t. the single-mode case cf. **Bartusch et al. (1988)**].

Thus after adding τ_{ij} on the left hand side of (3), MSM is a correct representation of the multi-project generalization, too.

After all assumptions have been introduced, Table 2 summarizes the relationship between machine tool manufacturing and multi-project multi-mode resource-constrained project scheduling for the four levels.

Table 2: Fundamental Relationships

Level	Machine Tool Assembly	Multi-Project Multi-Mode Resource-Constrained Project Scheduling
Customer Level	Product	Project
Upper Assembly Level	Phase	Subnetwork
Lower Assembly Level	Assembly of Parts	Job
Operational Level	Process Plan	Mode

The MSM either provides a schedule which is feasible with respect to precedence-, resource- and time window-constraints or proves infeasibility. In the latter case we could either disregard deadlines or - preferable - try to shorten the project length by allotting additional resources in order to meet deadlines.

2.4 Project Compression Model

An important issue within project scheduling are deadlines. As outlined above projects have to be finished at certain deadlines, since otherwise high penalty costs will incur and good will of vendors might get lost. Hence, the furthestmost goal of the project manager is to finish the project in time. By minimizing the makespan the MSM automatically creates a schedule which considers the deadline. A (new) problem arises when the best (or even the optimal) makespan exceeds the deadline. In this case the only way to finish the project in time is to cover the peak resource demands of an in-time schedule by temporarily employing more resources. Additional capacity of such an in-time schedule can be supplied by either (i) working overtime or (ii) subcontracting part of the resource demand [cf., e.g., **Dervitiotis (1981)**].

Assuming that an additional unit of the renewable resource r provided in period t , denoted as κ_{rt}^+ , is available at the cost of c_r and additionally is limited to K percent of the regular capacity κ_{rt} , the objective is to find a feasible project schedule which satisfies the deadline and is associated with the least additional costs. This problem has been termed by **Deckro and Hebert (1989)** as project crashing problem and by **Yau and Ritchie (1990)** as project compression problem. While former authors modelled the single-mode case, herein the multi-mode version shall be focused. The basic idea is to shorten the project length by (i) allowing more parallelism of jobs and/or (ii) accelerating jobs by selecting modes with shorter durations. In both cases, more renewable resources per period are necessary.

More specific, the model is based on the following **assumptions**:

- C1. While like in aggregate production planning [cf., e.g., **Hax and Candea (1984)**] κ_{rt} may be considered to be the capacity (time) of regular labor, κ_{rt}^+ corresponds to overtime labor (capacity of resource r to be provided additionally in period t).
- C2. κ_{rt}^+ may be used up to a maximum amount of $K \cdot \kappa_{rt}$ with $0 \leq K \leq 1$, i.e. overtime is limited in percent of the regular time. Like κ_{rt} overtime κ_{rt}^+ is considered to be integer-valued (measured in resource units).
- C3. The cost of overtime labor are c_r per unit of κ_{rt}^+ .

With $\bar{T} := \max \{\delta_p \mid p = 1, \dots, P\}$ as well as critical path based earliest and latest finish times EF_j and LF_j (like in the MSM-model) the project compression model (PCM) may be formulated as follows:

$$\min \quad \sum_{r \in \mathcal{R}} c_r \sum_{\tau=1}^{\bar{T}} \kappa_{r\tau}^+ \quad (6)$$

$$\text{s.t.} \quad \sum_{m=1}^{M_j} \sum_{\tau=EF_j}^{LF_j} x_{jm\tau} = 1 \quad (\forall j) \quad (7)$$

$$\sum_{m=1}^{M_i} \sum_{\tau=EF_i}^{LF_i} \tau \cdot x_{im\tau} \leq \sum_{m=1}^{M_j} \sum_{\tau=EF_j}^{LF_j} (\tau - d_{jm}) x_{jm\tau} \quad (\forall j, \forall i \in \mathcal{P}_j) \quad (8)$$

$$\sum_{j=1}^J \sum_{m=1}^{M_j} k_{jmr} \sum_{\tau=t}^{t+d_{jm}-1} x_{jm\tau} \leq \kappa_{rt} + \kappa_{rt}^+ \quad (\forall r, \forall t) \quad (9)$$

$$\kappa_{rt}^+ \leq K \cdot \kappa_{rt} \quad (\forall r, \forall t) \quad (10)$$

$$x_{jmt} \in \{0,1\} \quad (\forall j, \forall m, \forall t) \quad (11)$$

$$\kappa_{rt}^+ \in \{\mathbb{N} \cup 0\} \quad (\forall r, \forall t) \quad (12)$$

The objective function (6) minimises the cost procured by the use of additional capacity. Constraint set (9) limits the availability of renewable resources to the amount of regular and additional capacity. Finally, additional capacity is restricted in constraint set (10) to a certain percentage of the number of regular resources.

Two problems related with the PCM are (i) the minimisation of the total cost associated by providing $K \cdot (1 + \max\{\kappa_{rt}^+ | \text{for all } t\})$ units of each renewable resource [cf. Demeulemeester (1992), Möhring (1984)] and (ii) the minimisation of the variance of the resource demand subject to unlimited resources availability [cf. Burgess and Killebrew (1962), Woodworth and Willie (1975)].

PCM defines feasible schedules requiring the least expensive additional amount of renewable resources. Feasibility is achieved based on the (weak) sufficient conditions that mode-dependent shortest job durations satisfy time windows and K is "large" enough in order to satisfy the corresponding resource requirements.

In order to illustrate the PCM - a model which so far has not been dealt with in the literature - consider the instance provided in Table 3 and Figure 1 with $|\mathcal{R}| = 1$ renewable resource and a resource availability of $\kappa_{1t} := 4$ units for all t . It is assumed that a deadline exists at the end of period 6. Capacity units in excess of the regular capacity of $\kappa_{1t} = 4$ cost $c_1 := 1$ per period and unit and are limited to 50% (i.e. $\kappa_{1t}^+ := 2$ for all t) of the regular capacity.

Solving (6)-(12) of the example with LINDO to optimality, generates the solution presented in Tables 4 and 5 as well as Figure 2, respectively. $\mu(j)$ denotes the mode assigned to job j and FT_j the finish time of job j , respectively.

Table 3: Data of the PCM Instance

j	1	2	3	4	5	6	7
m	1	1	2	1	1	2	1
d_{jm}	0	4	6	2	3	5	2
k_{jm1}	0	2	1	1	3	1	2

Figure 1: Network of the PCM Instance

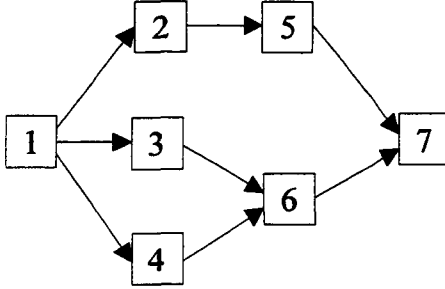


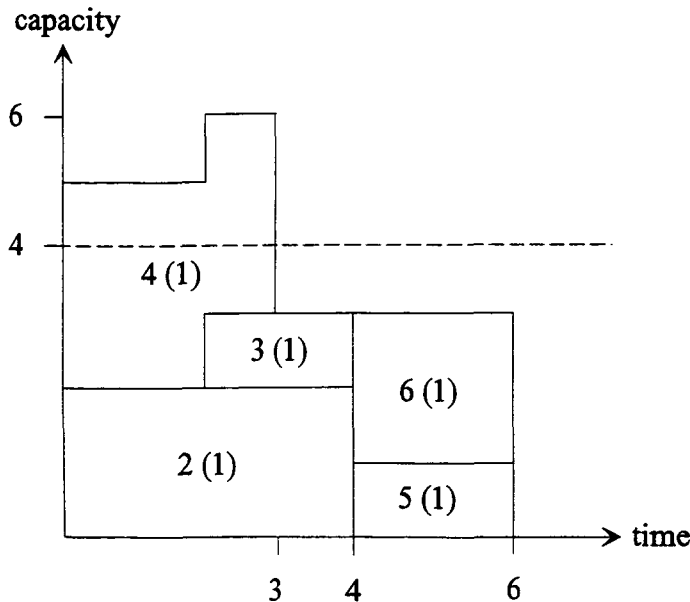
Table 4: Optimal Solution of the Instance – Modes and Finish Times

j	1	2	3	4	5	6	7
$\mu(j)$	1	1	1	1	1	1	1
FT_j	0	4	4	3	6	6	6

Table 5: Optimal Solution of the Instance – Additionally Used Resource Units

t	1	2	3	4	5	6
κ_{1t}^+	1	1	2	0	0	0

Figure 2: Optimal Solution of the Instance – Gantt-Chart



In Figure 2 " $j(m)$ " denotes that job " j " is processed in mode " m ". The optimal objective function value is 4, i.e. an additional costs of 4 units have to be taken into account in order to meet the deadline.

2.5 Scheduling Methods

First we are dealing with methods for solving the MSM. Subsequently we turn to the PCM.

For solving the MSM optimal and heuristic procedures have been proposed in the literature.

Optimal procedures make use of implicit enumeration with branch and bound and have been presented by **Blazewicz et al. (1986)**, **Speranza and Vercellis (1993)**, and **Talbot (1982)**. The latter algorithm was refined subsequently by **Patterson et al. (1989, 1990)**, and **Sprecher (1994)**.

Basically, **Kolisch et al. (1992)** and **Sprecher (1994)** showed that problems with 10 (16) jobs and three modes for each job can be solved to optimality within 0,69 (73,54) seconds on a personal computer with 80486 processor and mathematical coprocessor.

Due to the intractability of the problem, the following *heuristic procedures* for the MSM were developed: Multi-pass priority rule based scheduling [cf. **Boctor (1992)**, **Drexel and Grünewald (1993)**], simulated annealing [cf. **Słowiński et al. (1991)**], decomposition techniques [cf. **Dell'Amico (1990)** and **Kolisch (1994)**], as well as truncated branch and bound [cf. **Słowiński et al. (1991)**].

In the PRISMA-system the following probabilistic multi-pass priority rule based scheduling method - which was successfully used in **Drexel (1991)** to solve assignment type project scheduling problems - has been implemented. The stochastic nature of these method emerges from using some criteria measuring the impact of job selection and mode assignment in a probabilistic way.

More precisely we calculate

$$\omega_{jm} := (\max \{d_{hk} \mid h \in EJ, k = 1, \dots, M_h\} - d_{jm} + \epsilon)^\alpha \quad (j \in EJ, m = 1, \dots, M_j)$$

where EJ corresponds to the set of eligible jobs and job modes $m = 1, \dots, M_j$ are taken appropriately. ω_{jm} measures the worst-case consequence of assigning mode m *not* to job j with respect to job durations. $\epsilon > 0$ secures ω_{jm} to be positive. $\alpha \geq 0$ transforms the term $(.)$ in an exponential way thus diminishing or enforcing the differences between the mode dependent job durations for $\alpha < 1$ or $\alpha > 1$, respectively.

The usage of this weighting scheme as described below for stochastic job selection and job-mode assignment tends to produce schedules for $\alpha > 0$ with a relatively small project makespan due to preferring modes with short job durations. Despite this orientation towards the objective function value the repeated application of the method gives the opportunity that schedules will be constructed that are both feasible and "good"

regarding project makespan. In this sense it suggests itself to use ω_{jm} for stochastic job selection and (or) mode assignment with probabilities proportional to ω_{jm} for all $j \in EJ$ and $m = 1, \dots, M_j$.

For convenience we will denote this stochastic job selection and mode assignment procedure with STOCOM/MSM (stochastic construction method for solving MSM). Conceptually STOCOM/MSM may be interpreted as a stochastic generalization of Vogel's method for the transportation problem which uses "regrets" in a deterministic way. A detailed description of the method can be found in **Drexl and Grünewald (1993)**.

STOCOM/MSM produces a schedule $\mathcal{M} := \{(j, \mu(j), FT_j) \mid j = 1, \dots, J\}$ which may be (in-)feasible with respect to deadlines. In case of infeasibility the user is advised to turn to the PCM.

Disregarding the result of one pass through the above procedure (i.e. a feasible or an infeasible schedule) this algorithm should be applied to a specific data set several times (multiple passes) in order to hopefully generate a lot of feasible schedules und thus a near optimal one, too.

It should be emphasized that STOCOM/MSM corresponds to pure random sampling for $\alpha = 0$ (thus the latter is a special case of the former). In pure random sampling the job selection and job-mode assignment probabilities are equal for all eligible jobs and all relevant modes.

In **Drexl and Grünewald (1993)** STOCOM/MSM provided near optimal solutions for the single-project MSM without release dates and deadlines.

Currently, the heuristic of **Kolisch (1994)** seems to generate the best solutions. It decomposes the MSM into a mode-assignment model (MAM) and a (resulting) single-mode makespan model (SM-MSM). Within the latter each job has been assigned exactly one mode. Generally, the SM-MSM can be solved with a variety of heuristics [cf. to **Kolisch (1994)** for a review]. In order to derive near optimal solutions, Kolisch has devised a specially suited heuristic which is basically a multi-pass priority rule based scheduling method. More precisely, a feasible partial schedule is extended in a stage wise fashion. Within each stage a set of jobs which are schedulable w.r.t. precedence and resource constraints is formed and one of the jobs is selected with a priority rule and scheduled afterwards. Biasing the deterministic job selection by a random device, different schedules are obtained by performing multiple passes, whereby the best solution is maintained. Other distinctive features of the algorithm are: An improved priority rule, parameter based choice of the scheduling method and the priority rule, respectively, as well as the application of strong bounding schemes. W.r.t. the best schedule found for the SM-MSM, for each activity, respectively, a slack which takes into account precedence and resource constraints is calculated [cf. **Wiest (1964)**]. This information is then exploited in order to update and re-solve MAM. Hence, the heuristic iterates between the MAM and the SM-MSM for a pre-defined number of steps. Employing 50 steps, 10-job instances with three modes for each job were solved with an average (optimum-based)

deviation of 2.44% within 2.09 seconds CPU-time on a 80386 dx processor with 40 Mhz clockpulse.

For solving the project scheduling problem with given deadline **PCM**, no specialised exact or heuristic solution procedures are known from the literature. Optimal solutions were obtained by **Deckro and Hebert (1989)** for the single-mode version of the **PCM**. They employed a mixed-integer programming formulation in order to solve one example problem (made up by eight activities and two resource types) with a commercial MIP-package. The optimal solution was derived after 6.57 CPU-seconds on a **CYBER 760**. **Burgess and Killebrew (1962)** proposed a heuristic for the related problem of minimising the variance of the resource demand without considering resource constraints.

The only method for solving the **PCM** known so far was proposed by **Kolisch (1994)**. Like his approach for solving the **MSM**, it is a decomposition-heuristic which divides the **PCM** into an **MAM** and a **SM-PCM**. The **SM-PCM** is solved with a tailored heuristic which builds up on the method of **Burgess and Killebrew (1962)** for minimising the variance of the resource demand subject to single-modes and the absence of resource constraints. The idea is to iteratively right- and left-shift the activities within their precedence- and resource-based slack until the objective function cannot be improved. More precisely, the algorithm starts with a schedule where each activity is scheduled as early as possible w.r.t. precedence constraints only. Afterwards, the activities are sorted w.r.t. to 4 lexicographically ordered priority rules and are then right shifted to a - w.r.t. to the objective function of the **PCM** - local optimal finish time such that the deadline is obtained. Based on the resulting schedule, activities are sorted by a (different) priority rule vector for the following left shifting. Right and left shifting are iteratively repeated until no further improvement can be made. The overall cost of the resulting schedule (induced by the use of additional resources) is then accounted to the individual jobs. This information is exploited in order to update and resolve the **MAM**. Again, the heuristic iterates for a pre-defined number of steps between the **MAM** and the **SM-PCM**.

Preliminary tests were done on 5-job instances with 20 iterations performed for each problem. Depending on the hardness of the problem (e.g. tightness of the deadline, availability of regular resources) an average deviation from the optimal solution between 31,0% and 189,2% was reported. The CPU-time was in the range of 2 seconds on a 80386 dx processor with 40 Mhz clockpulse.

3. The Layout Component

In most machine tool assembly companies the manufacturing area (shop floor) is an utmost important resource due to the following reasons: Assembly of machine tools takes a long period (months or years) and machine tools offently become quite "large" (like houses).

Currently the space and layout needed by one or several machine tools as well as the available shop floor are drawn manually on "wallpapers". Update is a difficult and time-consuming task and therefore quite often is not done very pretty. The consequence is that shop floor becomes the limiting resource (overlapping, collisions etc.).

In this situation an **integrated layout component** was designed with the following features:

- The space requirements occurring during the assembly process are linked to jobs at the lower assembly level.
- A graphics component provides primitives (lines, polygons, rectangles, circles) with attributes (width of line, color, spacefilling etc.) in order to edit shop floor supply (machines, cranes, ways etc.) and machine tool space demand efficiently. There exists an interface to CAD- and DTP-systems.
- Simulation / animation capability of the layout over time supports the user with important information.

Note that currently there is no kind of automatic evaluation of alternatives like in layout planning [cf., e.g., **Domschke and Drexl (1990)**, **Francis et al. (1992)**, **Mirchandani and Francis (1990)**].

4. The Ordering Component

The ordering component is based on a rough **classification of parts** as follows:

- "Just-in-time" parts (high volume or expensive components; ordered when needed with respect to the schedule)
- Critical parts (availability problems in the past; ordered with respect to the phases of the upper assembly level)
- Uncritical parts (no availability problems in the past)
- Other parts (standard parts; ordered based on forecasts)

Note that there is no kind of method for the evaluation of alternatives with respect to their economic impact [cf., e.g. **Hax and Candea (1984)**, **Silver and Peterson (1985)**]. However, an interface to MRP-system exists where lotsizing or inventory management may be performed.

5. Additional Features

Beyond the components described above, the PRISMA-system contains the following features:

- Interactive graphics-based resource-, network-, and layout-editor; interactive Gantt-charts with high functionality
- Hierarchical description of staff characteristics (with respect to qualification and organization)
- Database-management of products, resources (staff, machines), and layouts
- Interfaces to MRP-systems and factory data collection-systems
- UNIX-implementation in C on Sun Sparcstation under OPEN LOOK
- INGRES relational database with the query language SQL (for further evaluation of schedules etc.)
- Statistics-, calendar-, print- / plot-functions

6. Summary and Future Work

Recently DSS-approaches to project scheduling have gained much attention [cf., e.g., **Anthonisse et al. (1988)**, **Bartusch et al. (1989)**, **Speranza and Vercellis (1991)**]. This paper presented the components of a state-of-the-art model-based DSS-leitstand for production scheduling in small batch machine tool assembly.

The MSM-related models and methods presented were already implemented in a system with the acronym MLS.¹ The PCM-related methods should be elaborated in more detail before being implemented in the MLS-system.

Unlike in the single-mode case [cf., e.g., **Bartusch et al. (1988)**], in the multi-mode case it is not possible to formulate the temporal constraints in its most general form. In the single-mode case the digraph of temporal constraints based on one unique representation of arc weights can be derived, and then the time bounds can be computed via longest path methods [cf., e.g., **Gallo and Pallottino (1989)**]. Note that this is impossible in the multi-mode case due to job durations unknown in advance. To work with dummy jobs (with known duration zero) makes this concept applicable, but is not very elegant.

The integration of an evaluation algorithm into the layout component (based on the quadratic assignment problem, graph-theoretic approaches to layout planning) is an interesting, worthwhile, and challenging task and should be investigated in the future.

¹ Montage-Leitstand; registered trademark of DAT Informationssysteme GmbH.

References

- Adelsberger, H.H. and J.J. Kanet (1991): "The leitstand - a new tool for computer-integrated manufacturing", *Production and Inventory Management J.*, Vol. 32, pp. 43-48.
- Ahn, J. and A. Kusiak (1991): "Scheduling with alternative process plans", in: *Modern production concepts*, Fandel, G. and G. Zäpfel, (eds.), Springer, Berlin, pp. 386-403.
- Anthonisse, J.M.; K.M. Van Hee and J.K. Lenstra (1988): "Resource-constrained project scheduling: an international exercise in DSS development," *Decision Support Systems*, Vol. 4, pp. 249-257.
- Bartusch, M.; R.H. Möhring and F.J. Radermacher (1988): "Scheduling project networks with resource constraints and time windows", *Annals of Operations Research*, Vol. 16, pp. 201-240.
- Bartusch, M.; R.H. Möhring and F.J. Radermacher (1989): "Design aspects of an advanced model-oriented DSS for scheduling problems in civil engineering", *Decision Support Systems*, Vol. 5, pp. 321-344.
- Blażewicz, J.; W. Cellary; R. Słowiński and J. Węglarz (1986): *Scheduling under resource constraints - deterministic models*, Baltzer, Basel (*Annals of Operations Research*, Vol. 7).
- Boctor, F.F. (1992): "Heuristic for scheduling projects with resource restrictions and several resource-duration modes", Working Paper, Faculté des Sciences de l'Administration, Université Laval, Québec, Canada.
- Burgess, A.R. and J.B. Killebrew (1962): "Variation in activity level on a cyclical arrow diagram", *The J. of Industrial Engineering*, Vol. 18, pp. 76-83.
- Chrysosouris, G.; J. Pierce and K. Dicke (1991): "An approach for allocating manufacturing resources to production tasks", *J. of Manufacturing Systems*, Vol. 10, pp. 368-382.
- Deckro, R.F. and J.E. Hebert (1989): "Resource constrained project crashing", *OMEGA*, Vol. 17, pp. 69-79.
- Dell'Amico, M. (1990): Un algoritmo euristico per la pianificazione delle risorse nei progetti software, in: Proceedings of the Annual Conference of the Operational Research Society of Italy, pp. 211-226.
- Demeulemeester, E. (1992): *Optimal algorithms for various classes of multiple resource-constrained project scheduling problems*, Ph.D. Dissertation, Leuven.
- Dervitiotis, K.N. (1981): *Operations management*, McGraw Hill, Tokyo.
- Domschke, W. and A. Drexl (1990): *Logistik: Standorte*, 3rd edition, Oldenbourg, München-Wien.
- Drexl, A. (1991): "Scheduling of project networks by job assignment", *Management Science*, Vol. 37, pp. 1590-1602.
- Drexl, A. and J. Grünewald (1993): "Nonpreemptive multi-mode resource-constrained project scheduling", *IIE Transactions*, Vol. 25/5, pp. 74-81.
- Drexl, A.; W. Eversheim; R. Grempe and H. Esser (1994): "CIM im Werkzeugmaschinenbau - Der PRISMA-Montageleitstand", *Zeitschrift für betriebswirtschaftliche Forschung*, Jg. 46, pp. 279-295.
- Francis, R.L.; L.F. McGinnis and J.A. White (1992): *Facility layout and location: an analytical approach*, 2nd edition, Prentice Hall, Englewood Cliffs, New Jersey.
- Gallo, G. and S. Pallottino (1989): "Shortest path algorithms", *Annals of Operations Research*, Vol. 13, pp. 3-79.
- Hax, A.C. and D. Candea (1984): *Production and inventory management*, Prentice Hall, Englewood Cliffs/New Jersey.

- Kanet, J.J. and V. Sridharan (1990): "The electronic leitstand - a new tool for shop scheduling", *Manufacturing Review*, Vol. 3, pp. 161-170.
- Kolisch, R. (1994): *Project scheduling under resource constraints - Efficient heuristic for several problem classes*, Ph.D. Dissertation, Kiel.
- Kolisch, R.; A. Sprecher and A. Drexel (1992): "Characterization and generation of a general class of resource-constrained project scheduling problems", Working Paper, University of Kiel (under review for *Management Science*).
- Mirchandani, P.B. and R.L. Francis (1990) (eds.): *Discrete location theory*, Wiley, New York.
- Möhring, R.H. (1984): Minimizing costs of resource requirements in project networks subject to a fixed completion time. *Operations Research*, Vol. 32, pp. 89-120.
- Neumann, K. (1975): *Operations Research Verfahren. Vol. III: Graphentheorie und Netzplantechnik*, Hanser, München-Wien.
- Patterson, J.; R. Słowiński; F.B. Talbot and J. Węglarz (1989): "An algorithm for a general class of precedence and resource-constrained scheduling problems", in: *Advances in project scheduling*, Słowiński, R. and J. Węglarz (eds.), Elsevier, Amsterdam, pp. 3-28.
- Patterson, J.; R. Słowiński; F.B. Talbot and J. Węglarz (1990): "Computational experience with a backtracking algorithm for solving a general class of precedence and resource-constrained scheduling problems", *European J. of Operational Research*, Vol. 49, pp. 68-79.
- Silver, E.A. and R. Peterson (1985): *Decision systems for inventory management and production planning*, 2nd edition, Wiley, New York.
- Słowiński, R.; B. Somiewicki and J. Węglarz (1991): "DSS for multiobjective project scheduling subject to multiple-category resource constraints", Discussion Paper, University of Poznań.
- Speranza, M.G. and C. Vercellis (1991): "A hierarchical multiobjective approach to project management", in: Korhonen, P. et al. (eds.), *Multiple criteria decision support*, Springer, Berlin, pp. 191-204.
- Speranza, M.G. and C. Vercellis (1993): Hierarchical models for multi-project planning and scheduling, *European J. of Operational Research*, Vol. 64, pp. 312-325.
- Speranza, M.G. and A.P. Woerlee (1991), "A decision support system for operational production scheduling", *European J. of Operational Research*, Vol. 55, pp. 329-343.
- Sprecher, A. (1994): *Resource constrained project scheduling - Exact methods for the multi-mode case*, Ph.D. Dissertation, Kiel.
- Talbot, F.B. (1982): "Resource-constrained project scheduling with time-resource trade-offs: the nonpreemptive case", *Management Science*, Vol. 28, pp. 1197-1210.
- Watson, H.J. and R.H. Sprague (1989): "The components of an architecture for DSS", in: *Decision support systems - putting theory into practice*, Sprague, R.H. and H.J. Watson (eds.), 2nd edition, Prentice Hall, Englewood Cliffs/New Jersey, pp. 107-130.
- Wiest, J.D. (1964): Some properties of schedules for large projects with limited resources, *Operations Research*, Vol. 12, pp. 395-418.
- Woodworth, B.M. and Ch.J. Willie (1975): "A heuristic for resource leveling in multi-project, multi-resource scheduling", *Decision Sciences*, Vol. 6, pp. 525-540.
- Yau, C. and E. Ritchie (1990): "Project compression: A method for speeding up resource constrained projects which preserve the active schedule", *European J. of Operational Research*, Vol. 49, pp. 140-152.