A review of capacity planning techniques within standard software packages

Citation for published version (APA):

Wortmann, J. C., Euwe, M. J., Taal, M., & Wiers, V. C. S. (1996). A review of capacity planning techniques within standard software packages. *Production Planning & Control*, 7(2), 117-128.
<https://doi.org/10.1080/09537289608930335>

DOI:

[10.1080/09537289608930335](https://doi.org/10.1080/09537289608930335)

Document status and date:

Published: 01/01/1996

Document Version:

Publisher's PDF, also known as Version of Record (includes final page, issue and volume numbers)

Please check the document version of this publication:

- A submitted manuscript is the version of the article upon submission and before peer-review. There can be important differences between the submitted version and the official published version of record. People interested in the research are advised to contact the author for the final version of the publication, or visit the DOI to the publisher's website.
- The final author version and the galley proof are versions of the publication after peer review.
- The final published version features the final layout of the paper including the volume, issue and page numbers.

[Link to publication](#)

General rights

Copyright and moral rights for the publications made accessible in the public portal are retained by the authors and/or other copyright owners and it is a condition of accessing publications that users recognise and abide by the legal requirements associated with these rights.

- Users may download and print one copy of any publication from the public portal for the purpose of private study or research.
- You may not further distribute the material or use it for any profit-making activity or commercial gain
- You may freely distribute the URL identifying the publication in the public portal.

If the publication is distributed under the terms of Article 25fa of the Dutch Copyright Act, indicated by the "Taverne" license above, please follow below link for the End User Agreement:

www.tue.nl/taverne

Take down policy

If you believe that this document breaches copyright please contact us at:

openaccess@tue.nl

providing details and we will investigate your claim.

A review of capacity planning techniques within standard software packages

J. C. WORTMAN, M. J. EUWE, M. TAAL and V. C. S. WIERS

Authors: Johan C. Wortmann, Graduate School of Industrial Engineering and Management Science, Eindhoven University of Technology, PO Box 513, NL-5600 MB, Eindhoven, The Netherlands; Mark J. Euwe, Graduate School of Industrial Engineering and Management Science, Eindhoven University of Technology, PO Box 513, NL-5600 MB, Eindhoven, The Netherlands/Andersen Consulting, Pastoor Petersstraat 162, NL-5612 LV Eindhoven, The Netherlands, e-mail: m.euwebdk.tue.nl/mark.j.euweac.com; Martin Taal, Baan International B.V., PO Box 250, NL-6710 PG Ede, The Netherlands; Vincent C. S. Wiers, Graduate School of Industrial Engineering and Management Science, Eindhoven University of Technology, PO Box 513, NL-5600 MB, Eindhoven, The Netherlands, e-mail: v.c.s.wiersbdk.tue.nl.

VINCENT C. S. WIERS holds a masters degree in industrial engineering and management science from the Eindhoven University of Technology. Since 1993 he has been a PhD student at the same university. His PhD project integrates knowledge in the field of organizational psychology, information technology and production planning and control. His research interests include human factors in production scheduling, the applicability of OR/AI scheduling techniques in practice, and the interaction between these techniques and human schedulers. His PhD thesis is to appear in 1997.

JOHAN C. WORTMANN studied industrial engineering and management science at Eindhoven University of Technology. He has been active in the development of information systems for production/inventory control since 1973 and wrote a doctoral thesis on the subject. He has been involved in a number of practical applications, both in component manufacturing and in assembly operations. He worked in the first half of 1985 as visiting professor at Rutgers University, New Jersey on databases. Dr Wortmann is currently employed at Eindhoven University of Technology as professor of information systems for production management.

MARK J. EUWE holds a bachelors degree in computer science from the Venlo Institute for Technology and a masters degree in industrial engineering and management science from Eindhoven University of Technology. He is currently working as a management consultant for Andersen Consulting and specializes in the fields of production control and logistics. Besides this he is also working on a PhD research project concerning intelligent manufacturing planning at the Eindhoven University of Technology. His research interests are currently focused on the integration of knowledge-based systems and regular production planning systems.

MARTIN TAAL holds a masters degree in computer science and a post-masters degree in logistical control systems, both from Eindhoven University of Technology. He is currently employed as a Software Engineer for Baan Research B.V., a part of Baan Company which is the supplier of the MRP-system TRITON. His professional interests include finite capacity planning and scheduling for different types of production companies and planning levels.

Keywords capacity planning, scheduling, sequencing, manufacturing resources planning, standard software packages

Abstract. This paper gives a review of capacity planning techniques from which today's standard software packages for production control make their choice. The following techniques are discussed in the paper: four variants of the rough cut capacity check, capacity requirements planning with infinite and finite loading, input/output planning without and with individual work orders, and a number of sequencing techniques. An important issue throughout the paper is the concept of robustness and nervousness of planning techniques. Aspects of interaction between techniques and human planners are given. The human planner is still an important factor in capacity planning.

1. Introduction

This paper presents an overview of capacity planning techniques that are found in standard software packages for production control. Because of the fact that these packages are in most cases based on MRP-II, we will mainly discuss the capacity planning concepts of MRP-II. However, for some parts of the discussion it is worthwhile to take a broader view. For example, standard software packages which are mainly used for make-to-order manufacturing companies (see Wortmann *et al.* 1994, for an overview of this field) and even packages used in the construction industry (see Melles and Wamelink 1993, for an overview of this field) use basically the same techniques.

This paper has no claims on techniques available in any particular package. Rather, it provides an overview on techniques from which today's packages will make their choice. To avoid the impression that particular packages are evaluated in their capacity planning performance, the paper does not make any reference to particular commercially available packages.

Although the material addressed here can be found scattered in several other places, an overview-paper is not easily found. Therefore, companies and engineering students face the problem of getting a general understanding of the field. The purpose of this paper is to fill this gap by providing an overview of capacity planning techniques.

An important point throughout the paper is that, until recently, nearly all software suppliers considered iterative algorithms to be very risky. However, most scientific research has been directed towards relatively small-scale optimization programs that are highly iterative. For builders of software who have to retrieve each single record from a disk there was only one overall guideline: avoid any situation in which a record can be addressed more than once. Clearly, these two points of departure are mutually exclusive, and for several decades, academia has lived in splendid isolation, separated from the outside world.

However, today's technology makes a weakly iterative algorithm possible, and may therefore give practitioners

renewed interest in academic solutions. Vice versa, academic institutions nowadays understand that real problems are probably best suited for testing an academic solution.

First, in Section 2 we discuss reliability, robustness, and nervousness issues of capacity planning systems. The first technique that will be discussed, rough-cut capacity planning, can be found in Section 3. Although the discussion focuses on MRP-II, it should be kept in mind that similar constructs (with limited feedback from reality and from manufacturing engineering) are also encountered in standard software for make-to-order plants and process industries, and in specific software for these companies.

Next, we discuss (detailed) capacity planning (in Sections 4 and 5). We discuss shopfloor control and sequencing in Section 6. In Section 7 we discuss human interaction with these systems, before summarizing our conclusions.

2. Reliability, robustness and nervousness of capacity planning systems

An important issue in the design of each capacity planning concept is the question of what level of detail is applicable for which planning horizon. The vast majority of capacity planning concepts are based on the principle of hierarchical plans. This usually means that capacity planning in the long term is performed at an aggregated level, whereas short-term planning is performed at a detailed level. The implicit assumption underlying this method is that a feasible schedule (at an aggregated level) in the long term guarantees a feasible (detailed schedule) in the short term. Unfortunately, situations can arise where the short-term schedule is not feasible although the long-term schedule suggests the opposite. Consider for instance the situation where three similar capacity resources, available for 40 hours per week, are added up into total figures stating a capacity of 120 hours per week. Now compare this figure to one machine working in three shifts, which amounts also to 120 hours per week. Clearly, the aggregated information about both situations is the same. However, a single order on a single workpiece requiring an operation of 110 hours cannot be processed in one week in the first situation, whereas it can in the second situation. (Consider the simple example of a woman carrying a baby for nine months. It is not possible to split up this biological process into nine women carrying one child for one month.)

Because of these possible contradictions between long-term and short-term schedules, one might want to calculate the longer term schedules on a higher level of detail. Unfortunately, adding detail to longer term schedules creates nervousness: these schedules are sensitive to changes in detailed information and lack robustness. The decision to use detailed information for the longer term is therefore influenced by both the *reliability* of future information and the *robustness* of the planning concept. The higher the reliability

of information, the more sense there is in calculating detailed schedules. The lower the robustness of the planning concept, the more sense there is in calculating detailed schedules.

The requirement of robustness, which can also be interpreted as having a 'memory', means avoiding nervousness. This means that the previous schedule is used when making a new schedule. Capacity planning systems usually suffer from the 'final-state' problem (see also Section 7). This means that a small change in the required final state of the plan results in major changes in the plan itself. Therefore, these systems will continuously 'switch off and on', unless some buffering—damping—is applied. In planning systems, nervous behaviour should be prevented as much as possible. Newly generated planning should be set up between the limits or restrictions that the former planning imposes. The definition of these limits and/or restrictions depends on the choices made by the designer. A simple example is the use of a fixed fence in an MRP-environment. It is up to the designer to set the fence to a month or a week (or to any other time interval). The first choice will decrease nervousness in the first month, however it also decreases the flexibility to react to changes in this period. A time fence of a week leads to more nervousness in the last three weeks when compared to a time fence of a month. The effects of nervousness on planning will be described in more detail in Section 7.

3. Rough cut capacity check

One of the steps in planning production volumes for the longer term is to check if available production capacity is sufficient to produce the required volumes. Such a check could be done by setting up detailed schedules that exactly specify at what date and time specific operations of specific orders are performed in the (far) future. However, intuitively it does not make much sense to perform these detailed calculations, when the future is still unknown in terms of customer orders for specific end products, inventory levels, specific machine availability, individual operator availability, etc.

Therefore, capacity checks that apply to the longer term are usually performed on a higher level, indicated by the term 'rough cut capacity check' (RCCC). A number of different techniques exist to perform an RCCC, each differing in the level of detail that is used in the calculations. The following four types, ordered in level of increasing sophistication, will be discussed:

- (1) RCCC on production volumes
- (2) RCCC using overall factors (CPOF)
- (3) RCCC using RCCC routings
- (4) RCCC using resource profiles

The last three are well described in Vollman *et al.* (1992)

3.1. RCCC on production volumes

In a flow-line production environment, bottlenecks are usually well known. Often the capacity of the production line is expressed in the number of units that can be produced by the bottleneck. If production volumes for different products are expressed in the same unit of measure as the subassemblies that are processed on the bottleneck (e.g. number of rolls), then a capacity check is easy to perform. Total production volume figures may simply not be larger than the bottleneck capacity.

3.2. RCCC using overall factors (CPOF)

In this method requirements are calculated using planning factors derived from standards or historical data for end products (Vollman *et al.* 1992). In the example in Table 1, a master production schedule (MPS) is used to calculate rough capacity loads for two products:

Historic data shows that the average total capacity load (e.g. labour hours) is 1 hour for item A, and 2 hours for item B. The total requested capacity load for the coming periods is calculated in Table 2.

Historical data shows that the total load is distributed to two workstations X and Y following a 60%-40% distribution. Therefore, requirements for workstations X and Y can be calculated as shown in Table 3.

The next step is to compare these individual workstation requirements with individual workstation availability. It may be obvious that this is a very simple process that can be done manually.

Table 1 Master production schedule.

Item	Period 1	Period 2	Period 3
A	5	6	5
B	10	12	12

Table 2. Total requested load.

	Period 1	Period 2	Period 3
Total load	$(5 \times 1) + (10 \times 2) = 25 \text{ h}$	$(6 \times 1) + (12 \times 2) = 30 \text{ h}$	$(5 \times 1) + (12 \times 2) = 29 \text{ h}$

Table 3 Requirements

Workstation	Period 1	Period 2	Period 3
X	$0.6 \times 25 = 15 \text{ h}$	$0.6 \times 30 = 18 \text{ h}$	$0.6 \times 29 = 17.4 \text{ h}$
Y	$0.4 \times 25 = 10 \text{ h}$	$0.4 \times 30 = 12 \text{ h}$	$0.4 \times 29 = 11.6 \text{ h}$

3.3. RCCC using RCCC routings

A more detailed capacity check can be done using the concept of RCCC routings. This method is named 'capacity bills' in Vollman *et al.* (1992), however, we believe that the term 'routing' is more appropriate than the term 'bill'. This method uses the information contained in production routings and bills-of-materials (BOM). The concept can be summarized as compressing the production routings of all subassemblies into one single RCCC routing for *each* item in the MPS. This will be illustrated by using again the MPS data in Table 1. Suppose that item B is assembled from items C and D. Table 4 shows the routings of the products (these numbers include setup time and process time per item. Setup times per item are calculated by dividing the standard setup time by the average lot size).

Compressing these production routings into a RCCC routing leads to the figures for A and B as depicted in Table 5.

Note that the workload per workstation for MPS item B is aggregated from the production routings for items B, C and D, e.g. workstation Y is needed for 0.2 h to assemble B from C and D, and 0.6 h to produce item C. During the RCCC it is therefore assumed that obtaining one unit of MPS item B will in total require 0.8 h of workstation Y. Capacity requirements can now be calculated as in Table 6, and compared to workstation availability, using the RCCC routings and the item quantities in the master production schedule.

The RCCC routing method is especially preferable to

CPOF in situations with a varying product mix and where different products have very different routings. Using one single workstation distribution in such a case (60–40% in the example) can be completely unreliable.

3.4. RCCC using resource profiles

This method is very similar to the RCCC routing method. The main difference between the two is the inclusion of the time factor when using resource profiles. The RCCC routing method puts the total capacity requirements in the time period in which the MPS-quantity is positioned. However, this method becomes less reliable when lead times go across periods. Resource profiles try to deal with this by incorporating lead-time information that shows in which period requirements are positioned. This is illustrated in the example Table 7.

Table 7 shows that the requirements to complete one unit of B at the end of the period t is one hour of X in period $t-2$ and 0.2 hour in period $t-1$. The requirements for workstation Y is 0.6 hours in period $t-1$ and 0.2 hours in period t . This is the period in which the MPS-quantity needs to be delivered. Capacity requirements can now be calculated by multiplying resource profiles by the quantities from master production schedule, in a similar way to that with RCCC routings.

Table 8 summarizes the four types discussed and briefly shows the calculation of capacity requirements for a workstation X. It should be kept in mind that all these RCCC techniques share the following characteristics:

- Lot sizes are assumed to be fixed. In practice lot sizes vary continuously which leads to different capacity requirements in the short term.
- Inventories on lower-level assemblies are not taken into account during the calculation of capacity requirements. It is assumed that all subassemblies need to be produced. In practice, inventories do exist which means that actual capacity requirements will be lower.
- The actual shop situation is not taken into account. If current work is heavily delayed in the shop, effects for future periods will certainly be present. RCCC techniques do not include this information in their calculations.
- Only the demand of the MPS is included in the checks. Any demand for service parts, additional orders as a reaction to scrap, etc., is not taken into account.

Table 4 Production routings.

Item	Workstation X	Workstation Y
A	0.5	0.5
B	0.6	0.2
C	0.6	0
D	0	0.6

Table 5 RCCC routing

Item	Workstation X	Workstation Y
A	0.5	0.5
B	$0.6 + 0.6 = 1.2$	$0.2 + 0.6 = 0.8$

Table 6. Requirements.

Workstation	Period 1	Period 2	Period 3
X	$(5 \times 0.5) + (10 \times 1.2) = 15.5 \text{ h}$	$(6 \times 0.5) + (12 \times 1.2) = 17.4 \text{ h}$	$(5 \times 0.5) + (12 \times 1.2) = 16.9 \text{ h}$
Y	$(5 \times 0.5) + (10 \times 0.8) = 10.5 \text{ h}$	$(6 \times 0.5) + (12 \times 0.8) = 12.6 \text{ h}$	$(5 \times 0.5) + (12 \times 0.8) = 12.1 \text{ h}$

Table 7. Resource profile for item B.

Workstation	Period $t-2$	Period $t-1$	Period t
X	1.0	0.2	
Y		0.6	0.2

4. Detailed capacity planning

There are two types of detailed capacity planning (DCP) techniques: DCP based on finite loading, and DCP based on infinite loading. The two approaches differ in the fact that with finite loading capacity availability influences the planning process. With infinite loading, the capacity availability is displayed in user-oriented output, but plays no role in planning.

4.1. Detailed capacity planning: infinite loading

Infinite detailed capacity planning (DCP) is used frequently in MRP systems under the name of capacity requirements planning (CRP) (Wemmerlöv 1984, Landvater and Gray 1989). Infinite DCP computes capacity demand per period for each capacity group in a predetermined manner: work orders are planned backwards from their due dates. The operations are planned backwards in a specified order from the due date of the corresponding production order. Queue-, setup-, and transportation time is planned before or after each operation (most software packages assume that setup times are sequence independent). Capacity demand and capacity availability are compared in an infinite DCP-report. The available capacity of a group is computed by summing up the available capacity of each capacity source. The available capacity of each capacity source is predetermined. An example of an infinite DCP report is shown in Figure 1.

Figure 1. A CRP report.

Detailed capacity planning is normally applied to all work orders that are known to the system: not only orders running on the shop floor, but also orders planned to be released at a later stage. In MRP systems, a list of these orders is one of the most important outputs of the MRP-1 algorithm (Landvater and Gray 1989). Such a list provides for each (planned) order the start date, the due date, the item and quantity to be produced, the routing, and data about the required time and capacity. As mentioned, DCP starts with a detailed 'schedule' that produces start and finish dates for each operation.

Infinite DCP does not necessarily consider current time a 'hard' constraint. Therefore it is possible that work orders are planned in the past (as in period -2 and -1 in Figure 1). Many software packages try to overcome this problem by reducing the time between operations. It is assumed that 'late orders' can be rushed through the capacity group. However, these approaches do not take the consequences for other work orders, delayed by the 'rush order', into account.

Table 8. Overview of RCCP types.

RCCC type	Workstation X capacity requirements
Production volumes	$\Sigma(\text{MPS-item quantities})$
Overall factors	$\Sigma(\text{MPS-item quantity} \times \text{total capacity req}) \times \text{workstation X share}$
RCCC routings	$\Sigma(\text{MPS-item quantity} \times \text{workstation X requirement})$
Resource profiles	$\Sigma(\text{MPS-item quantity} \times \text{workstation X}(t) \text{ requirement})$

Many software packages maintain the information concerning work orders and operations separately. This can lead to inconsistencies in lead times of work orders and processing times (per piece) of operations, e.g. when the sum of the operation processing times is larger than the lead time (i.e. due date minus release date) of the work order.

4.2. Consequences of this approach

Infinite DCP is based on the idea that capacity problems should be solved by the planner and not by the system (Landvater and Gray 1989). DCP only identifies capacity problems: corrective actions are taken by the planner. This means that no suggestions or proposals are generated by DCP. The planner must use his own (human) judgement to decide which actions to take. This can lead to problems in complex production systems (e.g. varying capacity demand, many BOM levels) (Bakke and Hellberg 1993, Krupp 1984, Drift 1993, Sum and Hill 1993). These problems will be further described in Section 7.

Another shortcoming of infinite DCP is that it can give an unrealistic picture of the capacity demand and capacity availability. The main cause is the degree of aggregation of the information used by infinite DCP. Infinite DCP uses aggregated information concerning capacity groups, periods and capacity demand. During the aggregation of detailed information from the shop floor to aggregated capacities and periods, important information is lost. So, the aggregated information may give an invalid picture of the actual state of the production system.

However, there is a very good reason for using aggregated instead of detailed information: detailed information is hard to maintain, and changes frequently. Any planning technique that uses this detailed information should consider these aspects and smooth out the frequent changes in the detailed information. Otherwise the planning technique will result in frequently changing and nervous plans.

Infinite DCP is a technique that can give good results in many situations. However, in many production systems infinite DCP is only useful as a very rough capacity demand 'reporter'. The planning itself is done by the planner, based on his or her own human judgement. Infinite DCP should support the manual planning process by presenting capacity demand and availability reports. Infinite DCP does not replace the human, because it gives an unrealistic picture of the capacity demand and availability, and because its output cannot be processed by scheduling or sequencing software.

The lack of realism in infinite DCP output is a well-known problem of the above DCP technique. Many other techniques have been tried to overcome this problem: one of these is DCP with levelling loads. With this approach (aggregated) information concerning capacity availability is used in the planning process. This technique is discussed in Section 4.3.

Another technique is input/output planning, which will be discussed in Section 5.

4.3. Detailed capacity planning: finite loading by levelling

This approach is similar to DCP with infinite loading; requirements and availability of capacity are computed initially in exactly the same manner. However, this method tries to level out overloads in capacity demands: if overloads exist in a certain period, (a part of) this load is transferred to an earlier period with underload. This step is repeated until the overload is resolved or until there are no earlier periods with underload. Consequently, overload is only reported if it is *cumulative* overload.

Note, that in this method load is shifted without reference to order that can—or cannot—be transferred to an earlier period: the link with individual orders is lost. (There are of course other finite loading systems that keep track of individual orders, and which check precedence relations between operations of orders. These systems will be discussed in Sections 5.2 and 6).

Therefore, levelling DCP is no more than a capacity plan 'reporter'. The characteristics of infinite DCP apply to a large extent also to this DCP method. The major advantage of finite DCP using load levelling over infinite DCP is that it gives a slightly more realistic picture of the capacity load over time.

5. Input/output planning

5.1. Input/output planning technique

When work is 'levelled' in CRP diagrams, the assumption is made that queue times are long enough to allow levelling. Consider for example the situation in Figure 2. Because of capacity problems in period 2, work is levelled backwards into period 1. However, if this work only arrives at the workcentre in period 2, the levelling action is untimely. CRP does not offer the possibility of considering individual work-orders. Furthermore, CRP does not account for detailed queue times.

Figure 2. A CRP diagram

Table 9. Schedule.

Work-order/operation number	Required capacity (h)	Arrival date	Start date	Due date
1234/20	100	1	14	18
6789/30	10	2	14	14
4711/10	50	4	15	17
1994/60	60	6	17	20

To tackle some of these problems, input/output planning (IOP) can be used in many standard software packages. An IOP diagram indicates the arrival times of work at the workcentre. This diagram is calculated by data as depicted in Table 9.

Figure 3 shows an example of an IOP diagram. The cumulative input line in this diagram shows the total amount of work that has arrived at the workcentre. The difference between the cumulative input line and the cumulative output line represents the workload in the workcentre.

A number of situations can be derived from the IOP diagram. First, Figure 3 shows that if the cumulative input line crosses the cumulative available capacity, a certain amount of capacity is lost forever. Hence, the availability line must be adjusted to the right.

Second, Figure 3 shows what happens when there is a shortage of capacity in a certain period (such as the situation in period 2 of Figure 2). This means that the slope of the cumulative output line exceeds the slope of the cumulative availability line. In other words, the cumulative output line approaches the cumulative availability line. When the cumulative output line crosses the cumulative availability line, the workcentre is being overloaded and consequently work cannot be completed in due time.

IOP seems to have some advantages over CRP by considering work arrival times. It is a load-levelling approach that takes planned queue lengths into account. A shortcoming that still stands, however, is that IOP does not consider

individual work orders in its levelling. Furthermore, the approach may be too pessimistic, because it assumes that operations of work orders cannot arrive earlier than planned according to their normative queue times. Thus, IOP is a very similar technique to load levelling. However, where load levelling is too optimistic in its assumptions on moving work earlier, IOP is too pessimistic.

5.2. IOP with individual workorders—'planning across diagrams'

Up to this point in the discussion, we have not mentioned the reasons why links to individual work orders are not maintained during finite loading. The reason is, that load-levelling with individual work orders requires that the loading algorithm jumps from one workcentre to another in an iterative way—which is exactly what most commercial packages want to avoid, as has been explained in the introduction.

However, there is one way in which individual work orders can be scheduled without the implication of an iterative algorithm. This alternative differs from the other finite CRP/IOP approaches, because it does not compute the infinite CRP/IOP before starting levelling. Rather, input/output planning is performed one-by-one for individual work orders, which is closely related to the sequencing method to be discussed in Section 6.2.

For each individual work order, the cumulative input/output graphs of workcentres involved are updated. When one of the exception conditions mentioned in the previous section occurs, not only the load profile is adapted, but also the time schedule of the work order involved. More specifically, if there appears to be a shortage of capacity in a particular period, the operation concerned is moved to an earlier period with underload, and this fact is recognized in the planning of subsequent operations of the same work order. The second part of the previous sentence specifies the difference from 'classical' IOP. A similar condition occurs, if an operation has to be moved late. Note however, that this way of working does not change the planning of work orders that have been planned before this one. Therefore, the algorithm is non-iterative.

The situation can occur that levelling of work backwards in time is not possible. It still may be possible to level work forward in time, while adhering to the work order's due dates. To determine the possibilities of levelling forward,

Figure 3. An IOP diagram

one has to consider slack within succeeding operations. The planner consequently has to consider multiple diagrams and workcentres.

However, the following complication comes into view here. Capacity problems at the workcentre level are only signalled by IOP. When a capacity problem occurs, the human planner has to solve these problems by adapting the MRP-schedule. Consequently, the planner has to use firm-planned-orders (FPOs). Solving one workcentre problem by levelling FPOs may erect problems across two or more workcentres, and is therefore not an efficient approach. A better approach would be to adjust capacity load at the workcentre level by firm-planning of individual operations. However, this is only possible when adequate tools for shop floor control are present.

The DCP and IOP techniques have no other claim than to support capacity planners in a more-or-less intelligent way. The claim is not that planners can be replaced by CRP-systems. Due to the nature of CRP, the planner will normally not be confronted with problems with work orders, but with problems with capacities. In particular, incidental peak loads should be expected.

6. Sequencing

Work order scheduling has to be followed by work order sequencing. The techniques for scheduling discussed so far make the following, unjustified assumptions:

- Orders can be split indefinitely. The capacity of a workcentre is computed by summing the capacity of machines. The capacity demand of a production order is booked against this total capacity,
- Setups occur between every operation,
- Setups are not sequence dependent,
- Queue times are fixed. In reality, queue times depend on many factors, e.g. the capacity load of a capacity group,
- All capacity demand is available at the beginning of a period,
- There are no capacity sources used by more than one workcentre (for example: tools, personnel)

After scheduling, the focus is on sequencing operations and work orders. The input of the sequencing function is a set of work orders with a start and due date. A work order consists of a set of operations with precedence relations. The sequencing problem can then be stated as follows: for each resource, compute the sequence in which operations are executed under the following constraints:

- finite capacity resources;
- precedence relations;
- start and due dates of work orders.

This sequence should optimize certain goals, for example: minimize setup costs, stock costs, throughput times. When operations can be assigned to multiple resources, the sequencing problem is extended with the capacity allocation problem. The following methods for solving the sequencing problem are discussed here:

- (1) *Priority dispatching methods.* The factory floor is simulated by using a time clock in the sequencing process, and sequencing decisions are based on priority rule.
- (2) *Schedule-generation.* All operations of one work order are scheduled at once, then the next work order is scheduled, etc. These methods do not use a time clock. During scheduling, the operations can be placed anywhere in the schedule between the release and due date of the work order.
- (3) *Optimization techniques.* These methods find an optimal schedule by (implicitly) enumerating all possible schedules, such as in branch and bound approaches, or by using mathematical programming approaches.
- (4) *Bottleneck methods.* A distinction is made between bottleneck and non-bottleneck resources. The bottleneck resources are scheduled before the non-bottleneck resources.

6.1. Priority dispatching

Priority dispatching techniques simulate the activities on the factory floor and the decisions of the planner. Priority dispatching operates as follows: there is a set of operations, ready to be sequenced, and a set of resources. Operations are sequenced when a resource becomes available. When more than one operation can be sequenced on one resource, a priority rule is used. Examples of priority rules are: shortest processing time, work order due date, least slack per operation. These priority rules influence the result of the scheduling process: different priority rules serve different scheduling goals. For example: shortest processing time—under certain conditions—minimizes the average throughput time (Haupt 1989).

Priority dispatching techniques have as major advantages that they are fast, simple, robust and well understood by the planner (Morton and Pentico 1993). These techniques often generate schedules that try to maximize resource utilization in the short term, leaving space in later periods to recover from disturbances.

However, a disadvantage of priority dispatching rules is that they are not very 'smart', and they generate schedules that can be far from optimal. Priority rules in general only use local information, which leads to local optimization, i.e. insensitive to the current overall state of the production system (e.g. workload). In many cases operations are

scheduled too early, and synchronization of the material flow is very difficult.

Another disadvantage of priority dispatching rules is their incompetence in handling partly filled-in schedules: it takes an empty schedule as input. Therefore, decisions made in the past cannot be taken into account during scheduling. In other words: there is no 'memory' function available (refer to the 'final-state' problem that has been introduced in Section 2).

Some of these disadvantages are overcome by using more advanced priority rules based on factory conditions, possible bottlenecks, and due date problems. Instead of using priority rules it is also possible to use partial enumeration (see below: beam search).

6.2. Schedule generation

Schedule generation is closely related to IOP with individual work orders (see Section 5.2) and operates as follows: work orders are ordered according to a priority rule. All operations of the work order with the highest priority are scheduled first. Then all operations of the next work order are scheduled. This continues until all work orders are scheduled. The operations are scheduled between the release and due date of the work order. Operations are normally scheduled backwards from the due date or forwards from the release date of the work order. Scheduling can be done subject to different kinds of constraints: throughput time, finite capacity, workload, etc. For example: minimizing workload in a certain period results in a scheduling procedure that tries to avoid scheduling operations in that particular period. When capacity problems occur, alternative routings or resources can be used. In this 'iterative' way, different schedules can be obtained and evaluated.

Schedule generation is fast and has possibilities for 'smart' scheduling: when a work order is sequenced, the state of the complete schedule between the release and due date is taken into account (in contrast to priority dispatching that only considers one moment in the schedule). This gives possibilities for global instead of local decision making. The technique can start with a schedule that is already partly filled. Changes can then be incorporated in an existing schedule by a 'net-change' procedure. Now decisions made in the past are remembered and taken into account, when computing a new schedule. This 'memory' function can be helpful in smoothing out frequent changes in a dynamic production environment.

This method has a major drawback in that the final schedule can have large gaps. There is also a danger that work orders, sequenced later in the process, cannot be sequenced according to the constraints because of a 'filled up' schedule (Morton and Pentico 1993).

6.3 Optimization techniques

Well-known optimization techniques are mathematical programming and implicit enumeration, such as branch and bound methods. The enumeration techniques have in common that a large number of schedules (possibly all feasible schedules) are computed. The schedule with the lowest cost (or the best performance on another measurement) is chosen (Morton and Pentico 1993).

Mathematical programming uses the simplex method to solve the sequencing problem. Linear and integer programming is only useful for small problems (less than 50 operations), because of the extreme computational cost.

Branch and bound methods use search trees to define and evaluate all possible schedules. An example of a search tree is shown in Figure 4. On a machine three operations can be executed: 1, 2, and 3. All possibilities are tried. The first choice is among three operations, so three nodes are created on the first level. For the second decision two operations remain, leading to two nodes on the second level for each branch, etc.

The size of the tree increases rapidly with the number of operations (e.g. 50 operations results in a tree with at least 3.04×10^{64} nodes). Therefore, these methods are not computationally feasible for practical problems. To overcome this problem a lower bound on the performance measure is maintained. With this lower bound parts of the search tree can be discarded, decreasing the size of the search space. However, finding a good lower bound is difficult and computationally very expensive. Another approach is to try a subset of all possible schedules. At each level of the search tree only a small number of nodes are chosen by some 'smart' heuristic (for example a priority rule). This method is called beam search.

Mathematical programming and enumeration techniques compute complete new schedules. So, there is no 'memory' function available, i.e. scheduling decisions made in the past can not be taken into account (see also Section 2).

Neighbourhood search (Morton and Pentico 1993) starts with a schedule and computes a 'nearby' schedule (a neighbour) to find a better solution. A neighbour is, for

Figure 4 An example of a search tree

Figure 5 The difference between neighbourhood search and tabu search.

example, obtained by interchanging two operations. If the interchange leads to a better performance, the change is maintained and a new interchange is tried. This method finds a local optimum from a given schedule (as illustrated in Figure 5). Tabu search differs from neighbourhood search in that small decreases in performance are excepted. In this way small 'hills' can be overcome to find a better local optimum. This is also illustrated in Figure 5.

Neighbourhood and tabu search can give good results if the start schedule is 'fairly' good. These methods are not very good in improving very 'bad' schedules. For both methods it is very important to use a 'smart' heuristic to choose a neighbour, especially if a large number of neighbours are possible.

6.4. Bottleneck methods

Two techniques are discussed here: optimized production technology (OPT) and the shifting bottleneck procedure. OPT was developed in the late 1970s, and distinguishes bottleneck and non-bottleneck resources. All operations succeeding operations on bottlenecks are called critical. The critical operations are scheduled forward, while the operations on the bottleneck resource are scheduled 'optimal'. The non-critical operations are scheduled backwards from due dates derived from the forward pass.

A major disadvantage of OPT is that it suffers significantly from the final-state problem. Every time OPT is run, a complete new schedule is computed. Successive schedules can differ very much, therefore, leading to very nervous scheduling. This major drawback of OPT stems from a lack of 'memory' (see also Section 2).

The shifting bottleneck procedure (Adams and Balas 1988) starts with a schedule. Bottlenecks are identified and sequenced 'optimally'. Each bottleneck is added to a set and then iterative all the resources in the set are sequenced. Then a new bottleneck is identified, etc. This continues until all the resources are added to the bottleneck set.

This method gives very good results on the job-shop make-span problem (schedule a set of operations so that the completion time of all operations is minimized). However, this method is not very flexible, because it is not able to take into account alternative resources, and it is only useful for the make-span problem.

7. Human scheduling and sequencing

The above gives an overview of scheduling and sequencing techniques within the context of MRP-II. One of the conclusions that can be drawn is that humans are nevertheless indispensable within the MRP-II environment. The techniques described cannot be applied without human intervention. The human role in scheduling and sequencing is discussed below.

7.1. The human role in scheduling

While the capacity planning techniques RCCC, CRP and IOP give useful results in many situations, they only serve as 'reporters', and do not offer the possibility for automatically adjusting schedules when necessary. Techniques that allow manual levelling actions (i.e. CRP with finite loading and IOP) have drawbacks that make straightforward usage impossible. Hence, human schedulers are needed to perform the following scheduling actions:

- (1) Humans have to adjust schedules manually by means of FPOs when capacity problems occur.
- (2) Humans have to verify possible actions of levelling work backwards in addition to the proposed levelling possibilities of capacity planning techniques.
- (3) When levelling work backwards is not possible and levelling work forward may still be, a human scheduler must consider multiple diagrams.

It will be clear that these tasks can become very complex for human schedulers. There are a number of factors that account for this complexity (Campbell 1988, 't Hart and Wiers 1994, Wood 1986):

First, task decomposability is poor, i.e. there is a large amount of contradicting factors to be taken into account simultaneously. Second, the planner has to cope with factors that have low analysability, i.e. there is uncertainty about some factors (e.g. demand level, operator sickness, machine breakdown). Third, human schedulers usually do not receive appropriate feedback about their actions, resulting in poor task controllability. An additional disadvantage of a lack of feedback is that planners are not able to improve their decision behaviour in time (Brehmer 1980, 1987, den Boer 1994). Fourth, planning actions in period t influence the

system in periods t , $t+1$, $t+2$, ... (and theoretically until $t+\omega$), resulting in high task dependency.

7.2. The human role in sequencing

Every person with some practical experience can confirm that sequencing techniques frequently are not applied in practice as described here. Especially in complex and dynamic production control situations, sequencing techniques do not seem to be straightforwardly applicable (Graves 1981). Therefore, in practice, humans are a crucial factor in sequencing. The following reasons for human sequencing can be given:

- (1) Sequencing techniques are designed from a mathematical perspective. These techniques answer questions about sets of orders, to be produced with a set of (capacity) resources. But sequencing in practice is a process of communicating, negotiating, compromising, and learning. A mathematical perspective only covers a small part of the real world.
- (2) Sequencing techniques are often too nervous to follow exactly. Because of the final-state problem, a sequence may change significantly if something relatively small occurs. In dynamic production control systems, the situation changes constantly. For example: machines go down, people get sick, customer demand is not known long beforehand, etc. This nervousness results in a low usability and credibility of the sequence in the eyes of shop floor schedulers and workers. In the short term people need some sort of fixed sequence to work with. When this sequence is not provided by the system, they will create their stability, thereby circumventing the system (Kleinmuntz 1990).
- (3) Sequencing techniques do not allow for every relevant factor on the shop floor. A shop floor control system is too complex, dynamic and unique to model in an economically feasible way. Hence, some factors fall outside the scope of being modelled. However, these factors often cannot be ignored. Therefore, every computer-generated sequence has to be 'translated' by a human scheduler.
- (4) Humans are able to cope with 'fuzzy' production rules. For example: it is preferable to sort moulding products at decreasing temperature. More important considerations can overrule this principle. Sequencing techniques cannot allow for these fuzzy rules. However, current research in fuzzy control systems may lead to future inclusions of fuzzy reasoning in standard software packages.

Mathematical design paradigms cannot overcome the problems mentioned. At an early stage, human factors should be taken into account when designing scheduling and

sequencing techniques and information systems. Humans, techniques and information systems have to work together to create a workable shop floor control system. They all have specific strengths and weaknesses, which should be balanced and harmonized (Bi and Salvendy 1994, Sanderson 1989, 1991).

8. Conclusions

This paper reviewed capacity planning techniques as encountered in current standard software packages for production planning and control. Nowadays, these techniques are characterized by hierarchical plans at different levels of detail. Consistency between levels is an important matter of concern.

Planning problems in practice cannot be seen as mathematical problems only. For their solution, negotiation, communication, risk-assessment and other human activities are essential. This means, that plans should show some robustness, in order to avoid nervousness.

Current techniques try to avoid iterative algorithms. Therefore, the vast bulk of operations research results is not yet used in practice. Due to progress of information technology, the near future will probably lead to more mutual influence of theory and practice.

References

- ADAMS, J., and BALAS, E., 1988, The shifting bottleneck procedure for job shop scheduling. *Management Science*, **43**, 391–401.
- BAKKE, N. A., and HELLBERG, R., 1993, The challenges of capacity planning. *International Journal of Production Economics*, **30–31**, 243–264.
- BI, S., and SALVENDY, G., 1994, Analytical modeling and experimental study of human workload in scheduling of advanced manufacturing systems. *International Journal of Human Factors in Manufacturing*, **4**, 205–234.
- DEN BOER, A. A., 1994, Decision support for material planners. *Production Planning and Control*, **5**, 253–257.
- BREHMER, B., 1980, In one word: not from experience. *Acta Psychologica*, **45**, 223–241.
- BREHMER, B., 1987, Models of diagnostic judgement. In J. Rasmussen, K. Duncan and J. Leplat (eds) *New Technology and Human Error* (Chichester: John Wiley & Sons), pp. 87–95.
- CAMPBELL, D. J., 1988, Task complexity: a review and analysis. *Academy of Management Review*, **13**.
- DRIFT, J. H. M., 1993, Architecture of a Factory Controller, PhD thesis, Delft University of Technology, Delft, The Netherlands.
- GRAVES, S. C., 1981, A review of production scheduling. *Operations Research*, **29**, 646–675.
- HART, M. W., and WIER, V. C. S., 1994, A framework for improving performance in complex tasks. In *Proceedings of the First European Conference Cognitive Science in Industry*, Centre de Recherche Public — Centre Universitaire, 28–30 September 1994, Luxembourg.

- HAUPT, R., 1989, A survey of priority rule-based scheduling. *OR Spektrum*, **11**, 3-16.
- KLEINMUNTZ, B., 1990, Why we still use our heads instead of formulas: toward an integrative approach. *Psychological Bulletin*, **107**, 296-310.
- KRUPP, J. A. G., 1984, Why MRP systems fail: traps to avoid. *Production and Inventory Management*, **3**, 48-53.
- LANDVATER, D. V., and GRAY, C. D., 1989, *MRP-II Standard System* (Essex Junction, VT: Oliver Wight Publications).
- MELLES, B., and WAMELINK, J. W. F., 1993, Production Control in Construction. PhD thesis, Delft University of Technology, Delft University Press, Delft.
- MORTON, T. E., and PENTICO, D. W., 1993, *Heuristic Scheduling Systems, with Applications to Production Systems and Project Management* (New York: John Wiley & Sons).
- SANDERSON, P. M., 1989, The human planning and scheduling role in advanced manufacturing systems: an emerging human factors domain. *Human Factors*, **31**, 635-666.
- SANDERSON, P. M., 1991, Towards the model human scheduler. *International Journal of Human Factors in Manufacturing*, **1**, 195-219.
- SUM, C. C., and HILL, A. V., 1993, A new framework for manufacturing planning and control systems. *Decision Sciences*, **24**, 739-760.
- VOLLMAN, T. E., BERRY, W. L., and WHYBARK, D. C., 1992, *Manufacturing Planning and Control Systems* (Homewood: Richard D. Irwin).
- WEMMERLOV, U., 1984, *Capacity Management Techniques (for manufacturing companies with MRP systems)* (Fallschurch, Va.: APICS).
- WOOD, R. E., 1986, Task complexity: definition of the construct. *Organizational Behavior and Human Decision Processes*, **57**, 60-82.
- WORTMANN, J. C., MUNTSAG, D. R., and TIMMERMAN, P., 1994, *Customer-driven Production* (London: Chapman and Hall).