ELSEVIER

Int. J. Production Economics 84 (2003) 115–131

international journal of
production
economics

www.elsevier.com/locate/dsw

Optimization approximations for capacity constrained material requirements planning

Alistair R. Clark*

Faculty of Computing, Engineering and Mathematical Sciences, University of the West of England, Coldharbour Lane, Bristol BS16 1QY, UK

Received 22 January 2001; accepted 9 September 2002

Abstract

This paper develops three mixed integer programming (MIP) models and solution methods to assist in identifying a capacity feasible master production schedule (MPS) in material requirements planning (MRP) systems. The initial exact model takes into account sequence-dependent setup times of both end-items and components, but is optimally solvable only for small product structures. A first approximate model and solution method, to be used with larger product structures, suboptimally schedules setups and lots on a period-by-period basis, estimating the capacity usage of future setups through the use of linear rather than integer variables. A second model and method, developed from the first, greatly accelerates computing time by sequencing setups gradually within each period, but again suboptimally. The trade-offs between schedule quality and computing time are analyzed in computational tests. The second model is able to schedule setups of up to 100 products on 10 machines over 5 periods in reasonable computing time. The tests show that this complex production scheduling problem can be practicably and successfully simplified both in terms of modelling and of solution method.

Keywords: MRP; Sequencing; Rolling horizons; Heuristic; Setups

1. Introduction

In recent years, enterprise resource planning (ERP) systems have been implemented in many industrial firms (Wortmann, 1998; Kennerley and Neely, 2001). Several companies selling ERP software now include optimization facilities that make use of powerful Operational Research approaches, such as mixed integer programming (MIP), to help improve the quality of operations planning and scheduling (Robinson and Dilts, 1999). Prominent examples include SAP's Advanced Production Optimiser (SAP-APO, 2002) and i2's Trade Matrix software (Trade Matrix, 2000). Such advances stem from increased competitive pressures to improve supply chain and manufacturing performance, the development of high level mathematical modeling languages such as AMPL (Fourer et al., 1993) and OPL (Hentenryck, 1999), and the cheap availability of powerful computing technology. Indeed, so exciting are the possibilities that the term *advanced planning and scheduling* (APS) has become a new buzzword, as

*Tel.: +44-(0)-117-344-3144; fax: +44-(0)-117-344-3155.
E-mail address: alistair.clark@uwe.ac.uk (A.R. Clark).

0925-5273/03/$ - see front matter
PII: S0925-5273(02)00400-0

witnessed by several recent publications and seminars (Kruse, 2000; IOM, 2000a, b). Some companies have specialized in the provision of APS software, such as OPL Studio (ILOG, 2000) which incorporates powerful modelling and optimization tools.

The onus is, however, still on the user to *formulate* an optimizing model that suitably reflects the organization's planning objectives and constraints. The formulation of an appropriate model is by no means a simple task and if carried out naïvely can result in models that are inaccurate or impossible to optimize. In addition, even a well-formulated model, if too large, can still take an impossible amount of computing time to optimize and will need to be solved via a heuristic method (Kuik et al., 1993; Sait and Youssef, 1999). A related approach is to use approximate models that are simpler to solve, but still reflect the objectives and constraints.

As a case in point, this paper reports research into the approximate modelling and optimization of capacity utilization in material requirements planning (MRP) multi-level systems. A comprehensive mathematical model is formulated and then solved using two related approaches based on model approximation and sequential decomposition. Computational tests show that the second approach produces reasonable solutions in viable computing time for medium-to-large sized product structures.

2. Capacity planning and optimization in MRP systems

In MRP systems, the master production schedule (MPS) represents a plan for the production of all end-items over a given planning horizon. It specifies how much of each end-item will be produced in each planning period, so that future component production requirements and materials purchases can be calculated using MRP component-explosion logic. As such, the MPS has to be feasible so that components can be produced within the capacity available in each time period. It is clear that there is a role here for a planning tool that efficiently takes capacity and the MRP explosion into account at the same time, a point made by Shapiro (1993). This paper proposes just such a tool in the form of a very general mixed integer programming (MIP) model that allows for sequence-dependent setups, unlike the formulation in Shapiro (1993). As it stands, the model cannot be solved quickly so two approximate models are developed to permit faster specification of efficient MPS/MRP plans.

Proud (1999) argues that master schedulers must be "optimizers", balancing conflicting goals such as low inventories and efficient utilization of capacity. In this spirit, the MIP model aims to minimize the total cost of component & end-item inventory and backorders while keeping within available production capacity. The decision variables are primarily the MPS production quantities, but an MRP plan is also identified at the same time. The capacity requirements of the MPS depend upon the MRP component production plan which in turn depends on the lot-sizes used in the MRP explosion. These are all taken into account in the model.

The model developed represents setup times that are sequence-dependent and permits multiple setups within a planning period. This makes for a combined lot sequencing and sizing problem, a thorny topic on which there has been limited research (Potts and van Wassenhove, 1992; Meyr, 2000, 2002). The resulting huge number of binary (0/1) variables causes great computational intractability for non-trivial problems. Such complexity is overcome by substituting the vast majority of the binary variables and constraints with continuous variables and constraints, as explained below.

Substantial computing effort is needed to optimally solve complex lot-sizing models. Faced with sequence-dependent setup times and dynamic job arrivals, Ovacik and Uzsoy (1995) struck a compromise between the impractical computational effort needed by perfectly optimizing procedures and the poor solution quality of myopic methods. They achieved this by breaking a large problem into a number of smaller semi-myopic ones which were solved exactly by branch-&-bound. This paper takes a similar approach using a series of branch-&-bound solutions to

determine a sequence of lot setups. The poor quality of short-horizon models is partly avoided by taking future demand and capacity into account using an auxiliary MIP to increase unit production times to factor in typical setups.

Clark (1998) developed a very fast myopic rule-based heuristic for rolling-horizon lot-sizing and sequencing on a set of parallel machines with sequence-dependent setup times, the single-level particular case of the problem studied in this paper. However, unlike the models developed in this paper, it assumed just a single setup at the beginning of each period. A subsequent paper (Clark and Clark, 2000) formulated an exact MIP that allowed multiple setups per planning period. Fast approximate models and solution methods were developed and found to produce solutions of reasonable quality. The current paper extends this approach to multi-level systems with a revised method of estimating key parameters and refined additional approximate model.

Meyr (2000) developed a model for simultaneous lot-sizing and scheduling on a single production line with sequence dependent setup times and recently generalized it to parallel machines (Meyr, 2002). For each machine he divided the planning periods into a predetermined number of micro-periods which contain at most one setup. The model in the current paper implicitly follows a similar approach, as shown below. However, Meyr did not permit backlogging (unlike this paper) and solved the model using a local search over the setup sequences, coupled with dual reoptimization of lot sizes, for single level problems with up to 19 products on 2 machines over 8 planning periods. In contrast, this paper uses approximate models solved as a series of MIPs to determine setup sequences, with less emphasis on optimality and more on good quality suboptimal solutions, for up to 100 products on 5 machines over 5 periods in a multi-level MRP structure.

3. A capacity-optimizing MRP model

Many authors (Baker, 1993; Shapiro, 1993; Thomas and McClain, 1993; Drexl and Kimms, 1998; Silver et al. 1998) have presented lot-sizing models for multistage MRP systems. However, these have been simpler than the model now presented which includes explicit representation of sequence-dependent setup times.

Capacity is represented by the key work centres that process the products (i.e., the end-items and their components). A product is processed on just one of a number of alternative (i.e., parallel) work centres. A product that needs to be processed on several work centres in series must be modelled as several separate products.

If a work centre has sequence-dependent setup times, then the utilization of capacity will depend on the sequence in which products pass through the work centre. Two examples are the printing and dying industries where correct ordering of lots using differently coloured inks and dyes is important for the efficient use of capacity. While a master scheduler should not normally have to take product sequencing into account, this is included in the model since sequencing can by itself have a significant impact on capacity utilization. This makes the model too large to solve optimally for anything other than a small number of products and periods. However, as shown below, the problem of model size can be overcome by making some simplifying approximations and then decomposing the model into a series of smaller ones. These models can be quickly solved in series while retaining explicit modelling of setup sequencing.

The initial model, called capacitated material requirements planning (CMRP), is now formulated as a mixed integer linear programme (MIP). Consider the following parameters in an MRP system:

$t = 1, 2, 3, \ldots, T$ are the production planning periods, where T is the planning horizon.
$i = 1, \ldots, P$ are the MPS end-item products.
$i = P + 1, \ldots, Q$ are the MRP component products.
$w = 1, 2, 3, \ldots, W$ are the capacitated work centres.
$P(w)$ is the set of end-items that can be processed on work centre w.
$Q(w)$ is the set of products that can be processed on work centre w.

The decision variables to be optimized are:

$$y^n_{ijwt} = \begin{cases} 1 & \text{if the } n\text{th setup on work centre } w \text{ in period } t \text{ is from product } i \text{ to product } j, \text{ where } i,j \in Q(w); \\ 0 & \text{otherwise.} \end{cases}$$

x^n_{iwt} = Quantity of product i produced between the nth and $n+1$th setups on work centre w in period t (non zero only if the nth setup on work centre w is to product i), where $i,j \in Q(w)$; $\geqslant 0$.

I^+_{it} = Stock of product i at the end of period t; $\geqslant 0$.

I^-_{it} = Backlog of product i at the end of period t; $\geqslant 0$.

To be realistic, model CMRP allows backlogs as many companies face occasional or frequent capacity overloads. An optimal solution to the model will automatically move many overloads forward or backward in time to other periods, but in the face of tight capacity and limited overtime/subcontracting opportunities such possibilities are soon often exhausted and a master scheduler will have no immediate choice but to backlog some of the demand in overloaded periods. Constraint (4), explained below, places limits on the inventory backlogs permitted for MRP components.

The objective function of the model minimizes the total costs associated with stocks and backorders:

$$\text{minimize} \sum_{i=1}^{Q} \sum_{t} (h_i I^+_{it} + g_i I^-_{it}), \tag{1}$$

where h_i is the cost of holding one unit of product i from one period to the next, g_i the Penalty cost of carrying over a backorder of independent demand for one unit of end-item i from one period to the next.

Only product inventory costs and backlog penalty costs are included in (1) since these are our major concerns. Insufficient capacity or poor use of it through inefficient sequences of setups will be reflected in higher backorders. The choice of the penalties attached to backorders depends on market conditions and the importance of the customers for particular products. Since the value of such penalties will partly be based on judgement and imprecise information, there is little point in spending a long time to solve the model optimally rather than approximately. The total costs of the provision of the production capacity are taken as fixed, not depending on the production decision variables x^n_{iwt} and y^n_{ijwt}. Additional setup and direct production costs are not included in the objective function as they are likely to vary little or be negligible in comparison to the penalty costs of the additional backlogs provoked by the lost work centre time that an inefficient sequence of setups would cause. However, if need be, such costs can be incorporated into the objective function without difficulty.

The following constraints link inventory and production to the independent demand for the MPS end-items (2) and demand (both dependent and independent) for components (3):

$$I^+_{i,t-1} - I^-_{i,t-1} + \sum_{w|i \in P(w)} \sum_{n} x^n_{iwt} - I^+_{it} + I^-_{it} = d_{it} \quad \forall t; i = 1, \ldots, P \tag{2}$$

$$I^+_{i,t-1} - I^-_{i,t-1} + \sum_{w|i \in Q(w)} \sum_{n} x^n_{iwt} - I^+_{it} + I^-_{it} = \sum_{w,j \in Q(w)|i \in C(j)} r_{ij} \sum_{n} x^n_{jwt} + d_{it} \quad \forall t; i = P+1, \ldots, Q, \tag{3}$$

where d_{it} is the independent demand for end-item or component i at the end of period t. $C(i)$ is the set of direct subcomponents of each component or end-item i, i.e., at one level lower down in the MRP Bill of Materials. r_{ij} is the number of units of direct subcomponent $i \in C(j)$ required in each unit of component or end-item j.

To avoid cumbersome time indexing notation, the model does not include the lead-time offsets for the components although, strictly speaking, it should. As they stand, constraints (3) allow the production of a component to be scheduled (sequenced) earlier in a period than the production of a required component. However, while this means that a solution may not be strictly implementable, the omission of lead times will have very little impact on the focus of this paper, namely he trade-offs between the solution values

and computing times of the various models tested in Section 6 below. For the interested reader, optimizing MRP models with lead-times are developed in Clark and Armentano (1993).

Any capacity shortage at work centres producing components must be reflected though backlogs of independent demand for those components or backlogs of end-items which require the components. Backlogs of dependent demand cannot be permitted in the model and are prohibited by constraints (4) below which limit any growth in component backlogs to be solely due to independent demand.

$$I^{-}_{i,t} - I^{-}_{i,t-1} \leqslant d_{it} \quad \forall t; i = P+1, \ldots, Q. \tag{4}$$

It would clearly be uneconomic to setup a product twice in a period on a work centre w, and so the number of setups can be limited to at most $|Q(w)|$ per period. Furthermore, at the beginning of each period the work centre will already be configured for the last product produced in the previous period. It makes sense to avoid scheduling a setup to this product in the current period, and to produce what is needed of the product at the beginning of the period. This is enforced by constraints (5)–(8) below which require that the first setup on a work centre in each period is a phantom one from the already-setup product to itself. The first setup in a period on a work centre must occur at the beginning of the period, but the subsequent setups may occur at any time within the period. A limitation of the model, however, is that it does not allow a setup to begin in one period and finish in the next.

$$y^{1}_{ijw1} = 0 \quad \forall w; i, j \in Q(w) \text{ and } i \neq i_{0w}, \tag{5}$$

$$\sum_{j \in Q(w)} y^{1}_{i_{0w}jw1} = 1 \quad \forall w, \tag{6}$$

$$y^{1}_{ijwt} = 0 \quad \forall w; t; i, j \in Q(w) | i \neq j, \tag{7}$$

$$\sum_{i \in Q(w)} y^{1}_{iiwt} = 1 \quad \forall w; t, \tag{8}$$

$$\sum_{i \in Q(w)} y^{n}_{ijwt} = \sum_{k \in Q(w)} y^{n+1}_{jkwt} \quad \forall w; t; j \in Q(w); n \leqslant |Q(w)| - 1, \tag{9}$$

$$\sum_{i \in Q(w)} y^{|Q(w)|}_{ijw,t-1} = \sum_{k \in Q(w)} y^{1}_{jkwt} \quad \forall w; j \in Q(w); t = 2, \ldots, T. \tag{10}$$

Constraints (9) to (10) above ensure that the nth setup on a work centre must and may only occur between a single pair of products, possibly both the same product, and that if a certain product is changed *to*, then it must be changed *from* in the following setup. The equals sign $=$, rather than the $\leqslant$ sign, is necessary so that it is always known for which product a work centre is configured, especially when the work centre is not producing. Thus the combination $y^{n}_{jimt} = 1$ and $x^{n}_{iwt} = 0$ must be allowed. Note that constraints (5)–(10) require that for each triple (n, w, t) there is exactly one pair (i, j) for which $y^{n}_{jimt} = 1$, i.e., there must be precisely $|Q(w)|$ modelled setups in each period on each work centre, even if a setup $y^{n}_{iimt} = 1$ is just from a product i to itself. Since a setup time s_{iim} from a product i to itself is zero, the model does not force a work centre to have exactly $|Q(w)|$ positive-time setups but rather up to $|Q(w)|$ such setups. The remaining zero-time setups are phantoms and do not exist in reality.

Constraints (11) below ensure that there must be a setup if a product is produced on a work centre in a period, even if it is just the first (phantom) setup in the period from the product to itself.

$$x^{n}_{jwt} \leqslant M_{jwt} \sum_{i \in Q(w)} y^{n}_{ijwt} \quad \forall w; t; n; j \in Q(w). \tag{11}$$

Constraints (12) below reflect the limited availability of capacity, taking setup and production times into account.

$$\sum_{n, j \in Q(w)} \left[\sum_{i \in Q(w)} s_{ijw} y^{n}_{ijwt} + u_{jw} x^{n}_{jwt} \right] \leqslant A_{wt} \quad \forall w; t. \tag{12}$$

Note that since the parameters for work centre capacity A_{wt} are indexed on t, the planning periods t in the model may be of different lengths. For example, weekend working may be combined into a single planning period.

4. Solving model CMRP

Wolsey (1997) points out that network flow type constraints such as (5)–(10) produce a very tight formulation in that the relaxed solution is the integer solution when no other constraints are present apart from $y^n_{ijwt} = 0$ or 1, thus assisting a branch-&-bound search to converge more rapidly. Even so, the model has up to $T \sum_w (|Q(w)|)^3$ *binary* variables and can only be solved optimally in a reasonable amount of computing time for very small instances, as is now demonstrated.

The computational tests carried out on model CMRP used the following parameters and data:

The bills of materials were randomly generated so that there were twice as many MRP components $i = P+1, \ldots, Q$ as end-items $i = 1, \ldots, P$, i.e., so that $Q = 3P$. An end-item or component i had, with equal probability, one, two or three direct components $j \in C(i)$ randomly selected from the components $j \geqslant \max(P+1, i)$. The number r_{ji} of units of j in each unit i was one, two or three with equal probability.

The set $Q(w)$ of products that can be processed on a given work centre w was generated as follows. The number $|Q(w)|$ of possible products is randomly chosen from $1, \ldots, Q$ so that the expected value of $|Q(w)|$ is $Q/2$. This gives a sensible number of products per workstation for the smaller structures tested ($Q \leqslant 10$), but for the larger structures tested ($Q \geqslant 20$) the number $|Q(w)|$ of possible products is randomly chosen from $1, \ldots, 2.5Q/W$ so that the expected value of $|Q(w)|$ is $1.25Q/W$. Following this, $|Q(w)|$ products are randomly selected from $1, \ldots, Q$. Finally, a check is made that each product is processed on at least one workcentre, a random assignment being made if this is not the case.

The unit inventory holding costs h_i were built up from the unit echelon inventory holding costs e_i (Clark and Scarf, 1960; Clark and Armentano, 1993) using the following recursive expression:

$$h_i = e_i + \sum_{j \in C(i)} r_{ji} h_j \quad \forall i, \tag{13}$$

where $e_i = 1$ unit of value per period $\forall i$.

The backlog cost for product i was $g_i = 100h_i$ so that, although great emphasis is attached to avoiding backlogs, they are not prohibited.

The setup times s_{ijw} for all products and workstations were generated with mean $\bar{s} = 1.0$ h in such a manner that if $i < j$ then $s_{ijw} < 1$ and $s_{jiw} > 1$, simulating, for example, setups between light (i) and dark (j) colours in printing machines. The setup times also obeyed the triangle rule, i.e., it was always quicker to change directly from one product to another than to do so via an intermediate product.

The unit production times u_{iw} for all products were distributed $U(0.005, 0.015)$ with mean $\bar{u} = 0.01$ h.

An MPS end-item's demand d_{it} per period was distributed $U(50, 150)$ with mean $\bar{d} = 100$ units. Thus the mean work centre time consumed by a given product would be 1 h per period. This is the same as the mean setup time so that, if production were on a just-in-time basis and setups were randomly sequenced, then setups and actual production of MPS end-items would on average consume equal amounts of work centre time. However, the whole point of lot-sizing is to schedule setups less frequently than just-in-time if this minimizes the objective function (1), so that the setups will tend to consume less time than actual production.

The MRP components had zero independent demand d_{it}, so that their demand is entirely dependent on the production of end-items whose bills of materials they are part of.

Capacity was representative of practice in the following sense: the models add most value if they can reduce inventory and particularly backlogs under tight capacity. The tests were carried out under conditions of "very tight" and "moderately tight" capacity, defined and calculated as follows. The basic capacity at a given work centre was that allocated by a simple linear program that minimized the total capacity, over all work centres, needed to satisfy both dependent and independent demand on a just-in-time (lot-for-lot) basis, ignoring setup times. Extra capacity totalling $Q\bar{s}/2$ was then added, evenly allocated among the workstations, thus allowing an average of $\bar{s}/2$ capacity time to setup each product once on a single

workcentre. In other words, capacity is very tight and so, even in an optimal schedule, there would probably be backlogs. However, this permitted the computational tests to compare how efficiently the various models and solution approaches manage to use capacity and thus reduce MPS backlogs. Moderately tight capacity was defined as adding a second amount of $Q\bar{s}/2$ of capacity, totalling $Q\bar{s}$ of extra capacity, again evenly allocated among the workstations, so allowing an average of $\bar{s}$ capacity time to setup each product once on a single workcentre. As will be shown below in the computational tests, where an optimal solution is achievable, "moderately tight" capacity is sufficient for model CMRP to be able to zero inventory and backlog, whereas the allocation of half this amount under "very tight" capacity is generally not sufficient. This provides a base to compare the performance of models developed in Section 5.

All the models in this paper were implemented using AMPL (Fourer et al., 1993), a powerful mathematical programming language that permits rapid formulation and testing of models, and solved using CPLEX 6.5 (ILOG, 1999), an established industrial-strength mathematical programming solver now available in many APS systems. Modelling tools such as AMPL are not only invaluable in the specification and rapid prototyping of appropriate models, but also facilitate revisions as a client's environment and modelling needs inevitably change, a point which is repeatedly cited by Maes and van Wassenhove (1988) in support of mathematical programming based approaches. The computational tests were performed on a Sun Enterprise 450 workstation with a 400 MHz Ultrasparc CPU and 510 MB of RAM.

Initial tests confirmed that model CMRP can be solved optimally in viable computing time only for certain very small problem instances. For example, with 5 products (i.e., end-items and components) on 2 machines and a planning horizon of 5 periods under both very tight and moderately tight capacity, Table 1 shows the number of binary y-variables (after AMPL preprocessing) in ten random instances of model CMRP and the CPU time needed to optimally solve them. Observe that the solution time is longer in the case of very tight capacity (and with slightly fewer binary variables after preprocessing in two instances). Note that, while the solution times for the smaller instances are possibly tolerable, for the largest instances an optimal solution still had not been reached and confirmed after 10 CPU hours of branch-&-bound searching, showing that the model as it stands is just not practical for operational use, even with the use of a heavy-duty MIP solver such as CPLEX.

What can be done about this? As the backorder penalties are often estimates and the demand forecasts are updated as the planning horizon is rolled forward, a previously optimal plan for the immediate period (the former period 2) will have become suboptimal, often seriously so (de Bodt and van Wassenhove, 1983). Thus there is little point spending a lot of computing time trying to optimally solve the model if instead a near-optimal solution can much more quickly found. A possibility is to let a MIP solver identify a (hopefully good) solution to model CMRP in a practical amount of computing time, but experience (Clark and Clark, 2000) suggests that this will not be effective for anything other than small problems. A more proactive alternative is to develop a solution approach that takes into account the fact that any solution to model CMRP, optimal or not, will almost certainly be applied on a rolling horizon basis, period by period.

Table 1
CPU solution times for model CMRP with 5 products and 2 work centres

Random instance	Moderately tight capacity		Very tight capacity	
	Number of binary variables	Optimal solution time	Number of binary variables	Optimal solution time
1	192	6 s	192	30 s
2	290	27 s	290	133 s
3	290	102 s	290	709 s
4	290	153 s	290	507 s
5	355	370 s	355	790 s
6	556	713 s	556	101 min
7	556	over 10 h	552	over 10 h
8	556	over 10 h	556	over 10 h
9	1000	over 10 h	996	over 10 h
10	1000	over 10 h	1000	over 10 h

The solution approach initially adopted decomposes the model into a succession of smaller MIPs, each with a more tractable number of binary variables, using the *relax-&-fix* approach (Wolsey, 1998), also known as fix-&-relax (Dillenberger et al., 1992). This involves the solution of a series of partially relaxed MIPs, each with a number of binary variables that is small enough to be readily solved by branch-&-bound search. As the series progresses, each set of binary variables is permanently fixed at their solution values. The procedure is broadly similar to a depth-first identification of an initial integer solution for a MIP model in a large branch-&-bound search. Its big advantage is its speed, but as will be shown below, the resulting solutions are not of good quality unless model CMRP is further developed.

For model CMRP, the relax-&-fix procedure is:

1. Given that constraints (5)–(8) imply that the first (phantom) setup on work-centre w in period 1 is from product i_{0w} to itself, fix the value of $y^1_{i_{0w}i_{0w}w1}$ to be 1 and the values of y^1_{ijw1} to be 0 for all $i,j \neq i_{0w}$.
2. To identify to which product the second setup is on work centre w in period 1, solve the partial linear programming relaxation, where the values of $y^2_{i_{0w}jw1}$ are constrained to be 0 or 1, while the other y-variables may vary continuously between 0 and 1, other than those fixed in step 1.
3. For $n = 3, \ldots, \max_w |Q(w)|$, solve the partial linear programming relaxation, with the y^1_{ijw1} and y^2_{ijw1} fixed at their 0 or 1 solution values from steps 1 and 2, with the values of y^3_{ijw1} to y^{n-1}_{ijw1} fixed at their 0 or 1 solution values from previous applications of step 3, and with the values of y^n_{ijw1} constrained to be 0 or 1, while the remaining y-variables may vary continuously between 0 and 1.
4. For $t = 2, \ldots, T$, repeat steps 1 and 3 for y^1_{ijwt} to $y^{\max_w |Q(w)|}_{ijwt}$, fixing their values at those of the solutions in steps 1 and 3.

Note that each cycle of steps 1–3 involves solving $\max_w |Q(w)| - 1$ problems with just $\sum_w |Q(w)|$ binary variables each, as $\sum_w |Q(w)|(|Q(w)| - 1)$ y-variables in each problem will newly have value 0 due to constraints (9) and (10).

Thus the application of the relax-&-fix approach to model CMRP involves the solution of $T(\max_w |Q(w)| - 1)$ MIPs, each with $\sum_w |Q(w)|$ binary variables. This is still a substantial number of binary variables to solve in a MIP, but far more capable of being solved than model CMRP with its $T \sum_w |Q(w)|^2(\max_w |Q(w)| - 1)$ binary variables. Recall that Table 1 shows that model CMRP can only hope to be solved optimally in reasonable computing time for problems of the order of 5 products and 2 work centres. For instance, planning 10 products on 5 work centres over 5 periods involves solving model CMRP with up to $5(5)(10)^2(10 - 1) = 2500$ binary variables, an impracticable proposition, while the relax-&-fix approach solves a succession of $5(10 - 1) = 45$ MIPs with just $5(10) = 50$ binary variables each. Twenty computational tests carried out for 10 products on 5 machines under both moderately tight and very tight capacity showed it is viable to solve model CMRP using the relax-&-fix method for this size of problem. It took a mean of only 65 s of CPU time to solve model CMRP1 with a succession of optimal relax-&-fix solutions. Tests using the relax-&-fix method on larger models with 20 products and 10 workstations were not optimally solvable in practicable time, motivating the development of the model approximations of the next section.

5. Approximations to the model

As shown in Section 4, model CMRP could well be impractical to solve optimally in reasonable computing time. Applying the relax-&-fix solution approach permits the identification of suboptimal solutions for larger models. However, the size of the model can be substantially reduced, the computing time much accelerated and solution quality greatly improved by making two approximations, as follows:

The first approximation involves the elimination of the binary y-variables representing the setups for period 2 onwards, compensating by increasing the values of the unit production times u_{iw} in those periods. The resulting model is denoted CMRP1. Constraints (2)–(3), (7)–(9), and (11)–(12) now all

apply only to period $t = 1$. The following constraints (14)–(16) are also included:

$$I^+_{i,t-1} - I^-_{i,t-1} + \sum_{w|i\in P(w)} x_{iwt} - I^+_{it} + I^-_{it} = d_{it} \quad t = 2, \ldots, T; i = 1, \ldots, P, \tag{14}$$

$$I^+_{i,t-1} - I^-_{i,t-1} + \sum_{w|i\in Q(w)} x_{iwt} - I^+_{it} + I^-_{it} = \sum_{w,j\in Q(w)|i\in C(j)} r_{ij}x_{jwt} + d_{it} \quad t = 2, \ldots, T; i = P+1, \ldots, Q, \tag{15}$$

$$\sum_{j\in Q(w)} u^*_{jw}x_{jwt} \leqslant A_{wt} \quad \forall w; t = 2, \ldots, T, \tag{16}$$

where u^*_{iw} is an increased unit production time that factors in likely setup times s_{ijw} in the same spirit as RCCP resource profile times (Proud, 1999). x_{iwt} is the provisional quantity of product i produced between on work centre w in period t.

Model CMRP1 can be solved optimally for smaller instances. For larger instances, it may be sub-optimally solved using the relax-&-fix method within period 1. In instances of all sizes, after solving for the setups of period 1, model CMRP1 is reformulated so that the former period 2 is now period 1, and number of periods is one less. The model is solved afresh, and so on, with $T-2$ further models being solved, until the setups for periods $1, \ldots, T$ have all been decided.

The second approximation is a development of model CMRP1, recognizing that it is not just the setups for period 2 onwards that can be eliminated and compensated for using u^*_{iw}. When using the relax-&-fix method to solve model CMRP1, why not also eliminate the relaxed binary y-variables representing future setups in period 1? The resulting model is denoted CMRP2 and greatly reduces the number of continuous variables. For period $t = 1$, the capacity constraints (12) becomes

$$\sum_{n=1}^{n^*}\sum_{j\in Q(w)}\left[\sum_{i\in Q(w)} s_{ijw}y^n_{ijwt} + u_{jw}x^n_{jwt}\right] + \sum_{j\in Q(w)} u^*_{jw}x_{jwt} \leqslant A_{w1} \quad \forall w, \tag{17}$$

where n^* is the current relax-&-fix iteration. The values of y^n_{ijwt} will have been decided and fixed for $n = 1, \ldots, n^* - 1$, so the variables being optimized in model CMRP2 are, for $t = 1$, $y^{n^*}_{ijwt}$ as well as x^n_{jwt} for $n = 1, \ldots, n^*$ and x_{jwt} for those products j whose lot-size is not modelled by an x^n_{jwt} variable. In order to force $x_{jwt} = 0$ for those products j represented by a x^n_{jwt} variable, constraints (18)–(20) below must be imposed in model CMRP2:

$$x_{jwt} \leqslant M_{jwt}z_{jw} \quad \forall w; j\in Q(w), \tag{18}$$

$$z_{jw} \leqslant 1 - \sum_{i\in Q(w)} y^n_{ijwt} \quad \forall n \leqslant n^*; w; j\in Q(w), \tag{19}$$

$$\sum_{j\in Q(w)} z_{jw} \leqslant \max(0, |Q(w)| - n^*) \quad \forall w, \tag{20}$$

where $z_{jw}\ \forall w\ \&\ j\in Q(w)$ are additional binary decision variables obliged by constraints (19) to take value 0 for those products j on work centre w whose lot-size is modelled by an x^n_{jwt} variable via a y^n_{ijwt} setup variable. As the relax-&-fix iterations proceed, an increasing number of the z_{jw} are thus fixed at zero, with just one of the remainder being solved at zero and the rest taking value 1. The inclusion of these additional binary variables is not over-onerous as the computational tests of Section 6 will show. In order to schedule all T periods, model CMRP2 is solved $\max_w |Q(w)|$ times in succession for each of periods $t = 1, \ldots, T$ in a manner similar to model CMRP1.

Representative values of u^*_{iw} for use in models CMRP1 and CMRP2 can be estimated by approximately solving the following MIP:

Model estimate increased unit production time (EIUPT):

$$\text{minimize} \sum_{i=1}^{Q} g_i I^-_i \tag{21}$$

such that

$$\sum_{w|i\in Q(w)} x_{iw} + I^-_i = \bar{d}_i \quad i = 1, \ldots, P, \tag{22}$$

$$\sum_{w|i\in Q(w)} x_{iw} + I^-_i = \sum_{w,j\in Q(w)|i\in C(j)} r_{ij}x_{jw} + \bar{d}_i \quad i = P+1, \ldots, Q, \tag{23}$$

$$x_{iw} \leqslant M_{iw}y_{iw} \quad \forall w; i\in Q(w), \tag{24}$$

$$\sum_{i\in Q(w)} (\bar{s}_{iw}y_{iw} + u_{iw}x_{iw}) \leqslant \bar{A}_w \quad \forall w, \tag{25}$$

$$y_{iw} = 0 \text{ or } 1 \quad \forall w; i\in Q(w), \tag{26}$$

$$x_{iw} \geqslant 0 \quad \forall w; i\in Q(w), \tag{27}$$

$$0 \leqslant I_i^- \leqslant \bar{d}_i \quad \forall i, \tag{28}$$

where the decision variables are:

$$y_{iw} = \begin{cases} 1 & \text{if product } i \text{ is likely to be produced} \\ & \quad \text{on work centre } w; \\ 0 & \text{otherwise.} \end{cases}$$

x_{iw} is the mean quantity of product i produced on work centre w (non-zero only if product i is produced on work centre w), I_i^- the resulting mean backorders of product i and where $\bar{s}_{iw}$ is the mean setup time to product i on work centre w, $\bar{d}_i$ is the mean demand for product i over periods $1, \ldots, T$ after discounting initial stocks I_{i0}^+ and backorders I_{i0}^-, and $\bar{A}_w$ is the mean available time on work centre w.

Model EIUPT assumes that a representative optimal policy would be to try to meet demand on a lot-for-lot basis, with no between-periods stocks, but possibly with some end-item backorders due to lack of workcentre capacity. In such a policy, a setup time $\bar{s}_{iw}$ for each product i is incurred on at least one work centre w in each period, giving:

$$\sum_{w|i\in Q(w)} (\bar{s}_{iw}y_{iw} + u_{iw}x_{iw}) = \sum_{w|i\in Q(w)} u_{iw}^* x_{iw} \quad i = 1, \ldots, Q. \tag{29}$$

Assume that for each product i there is a common factor f_i^* over all work centres $w|i\in Q(w)$ by which the unit production time u_{iw} must be increased to factor in the mean setup time, i.e., that:

$$u_{iw}^* = f_i^* u_{iw} \quad \forall w; i\in Q(w). \tag{30}$$

Then

$$\sum_{w|i\in Q(w)} u_{iw}^* x_{iw} = f_i^* \sum_{w|i\in Q(w)} u_{iw}x_{iw} \quad i = 1, \ldots, Q \tag{31}$$

and so from (29) and (31)

$$f_i^* = \frac{\sum_{w|i\in Q(w)} (\bar{s}_{iw}y_{iw} + u_{iw}x_{iw})}{\sum_{w|i\in Q(w)} u_{iw}x_{iw}} \quad i = 1, \ldots, Q \tag{32}$$

which is used with (30) to estimate u_{iw}^*.

However, if the optimal policy were to produce lots less frequently than lot-for-lot, then the use of expressions (30) and (32) would overestimate the optimal value of u_{iw}^* and possibly cause over-production in period 1, resulting in more inventory than necessary, but reducing the possibility of backlogs. If an optimal policy were to produce more frequently than lot-for-lot, then (30) and (32) would underestimate u_{iw}^* and possibly not cause a necessary anticipation of production in period 1 of a demand peak in periods 2 onwards. Since demand forecasts generally change as the planning horizon rolls forward, such production in period 1 might in fact not be needed. Moreover, greater importance is usually attached to avoiding backlogs than to reducing inventory. Accordingly, expressions (30) and (32) seem reasonable estimators of u_{iw}^*.

Model EIUPT has $\sum_w |Q(w)|$ binary variables. Table 2 shows the best, worst and typical computing times needed to solve each of the product-workcentre configurations shown in the

Table 2
Best, typical and worst instances of model EIUPT

Problem size		No. of binary variables	Optimal solution time (s)
Products	Workcentres		
10	5	14	0.02 s
		26	0.12 s
		44	40 s
20	5	29	0.05 s
		32	0.10 s
		37	0.34 s
50	5	52	0.03 s
		76	0.89 s
		85	15 s
100	10	127	0.34 s
		146	32 s
		198	> 1 h

table. Observe that model EIUPT can quickly be solved optimally for quite large problem instances, with up to 100 products on 10 work centres. Note that other instances of this size took over an hour to optimize. However, since the use of u^*_{iw} is an approximating device to factor in setup times, a practical response for large problem instances is to make do with the incumbent solution identified after a limited amount of branch-&-bound search time (such as the limit of 1 h used in the tests of Section 6).

6. Computational tests

How effective are models CMRP1 and CMRP2? In other words:

1. How well do the two models replicate the optimal outcome of model CMRP?

This question is not straightforward to answer as the provably optimal solution of model CMRP is obtainable only for very small instances. Thus a second question is:

2. What is the comparative quality of solutions of all three models that are obtained in equal computing time?

The model solutions will almost always be applied on a rolling horizon basis. In this situation only the production decisions relating to the first planning period are implemented after which the horizon is rolled forward and the model applied once more with updated demand, inventory and capacity information. For the purposes of comparison with model CMRP, demand after period T is not considered in the rolling horizon computational tests of models CMRP1 and CMRP2. Thus the tests compare the period 1 impact of T consecutive rolling horizon decisions, ignoring demand after period T. This also filters out, for the purposes of testing, the effect of "nervousness" (whereby the demand in new periods added as the horizon rolls forward can cause additional setups due to changes in planned but unimplemented lot-sizes) which a static solution applied in its entirety would avoid (Kropp et al., 1983; Maes and van Wassenhove, 1986; Blackburn et al., 1986; Baker, 1993; Kadipasaoglu and Sridharan, 1997).

Question 1 is considered first for instances with 5 products processed on 2 workcentres, showing that models CMRP1 and, to a lesser extent, CMRP2 result in solutions that sometimes replicate the optimal solution of model CMRP and are on average not far from it. Question 2 is then considered for larger instances with up to 100 products on 10 work centres, showing that, given equal computing time, models CMRP1 and CMRP2 obtain far superior solutions to model CMRP, and that model CMRP2 can solve large product structures.

Tables 3 and 4 shows the mean solution values and times for the following five models and solution methods:

1. Model CMRP solved optimally over all T periods.
2. Model CMRP solved by relax-&-fix over all T periods.
3. Model CMRP1 solved as a whole T times in order to schedule all T periods.
4. Model CMRP1 solved by relax-&-fix T times in order to schedule all T periods.
5. Model CMRP2 solved as a whole $T(\max_w |Q(w)|)$ times in order to schedule all T periods.

for 5 products on 2 machines over a planning horizon of $T = 4$ periods (Table 3) and $T = 5$ periods (Table 4), under both very tight and moderately tight capacity, applied to ten random problem instances for which an optimal solution for model CMRP could be found for $T = 5$ in under 10 h CPU time. The data was generated using the parameters of Section 4. In Table 4, the 10 h time limit was reached for about half of all randomly sampled instances, showing that the MIP solver CPLEX is at its limit of being able to optimally solve model CMRP, suggesting an optimal solution is inviable in reasonable time for larger problems. In comparison, the total CPU time spent solving the series of MIPs involved in the CMRP1 and CMRP2 models solved as a whole and by relax-&-fix approach was just a few seconds. Note that model CMRP took far less time to solve optimally over 4 periods than over 5

Table 3
Solution values with 5 products on 2 work centres over 4 periods

Instance	CMRP	CMRP RF	CMRP1	CMRP1 RF	CMRP2	CMRP Time
Very tight capacity						
1	83,400	977,079	96,618	1,087,053	85,139	10 s
2	88,632	5,917,600	102,926	2,282,075	99,364	37 s
3	31,790	914,254	32,732	1,058,190	39,041	108 s
4	39,043	848,091	39,043	1,074,249	56,050	575 s
5	125,654	415,043	135,593	540,090	135,593	38 s
6	69,130	2,099,245	78,726	1,549,739	101,526	218 s
7	111,554	2,265,239	111,554	3,850,023	137,560	10 s
8	46,815	738,417	132,994	687,480	229,045	560 s
9	67,674	1,760,807	68,329	1,838,804	68,329	6 s
10	68,044	2,683,072	69,048	1,964,859	69,048	8 s
Mean solution	73,174	1,861,885	86,756	1,593,256	102,070	
Mean time	157 s	4 s	1 s	3 s	3 s	
Moderately tight capacity						
1	0	1,394,099	0	60,943	0	4 s
2	0	1,504,005	0	1,733,685	0	6 s
3	0	929,618	0	1,138,122	23,688	20 s
4	0	469,380	0	443,094	2120	43 s
5	0	412,576	0	300,181	14,662	13 s
6	0	107	21	20,548	21	19 s
7	0	1,979,192	50	2,478,998	52	1 s
8	0	489,015	69	586,080	69	11 s
9	0	0	0	0	0	4 s
10	0	0	0	1,323,062	0	3 s
Mean solution	0	717,799	14	808,471	4061	
Mean time	12 s	3 s	1 s	3 s	2 s	

(0 indicates an (optimal) solution value of zero inventory and zero backlogs)

periods and when capacity is just moderately tight instead of very tight.

Tables 3 and 4 show solution values. Thus 0 means that the solution had zero value in the objective function (5), i.e., the production schedule had no inventory and no backlogs. Note that in both tables, all 10 CMRP problem instances sampled have zero-valued optimal solutions under moderately tight capacity. It should be emphasized that all the CMRP solution values, zero and non-zero, are optimal, i.e., no better solution exists.

Observe that under *very* tight capacity over 4 periods (5 periods), the mean solutions of models CMRP1 as a whole and CMRP2 were 18.6% (23.1%) and 39.5% (36.9%) worse respectively than that for model CMRP. However, in two (one) instances, the solution of CMRP2 was actually better than that of CMRP1. Under *moderately* tight capacity, solution of CMRP was always zero, replicated in 7 (5) instances by model CMRP1 and also in 4 (5) instances by model CMRP2. The solution of CMRP2 was never better than that of CMRP1.

Note the very poor performance of the relax-&-fix approach applied to models CMRP and CMRP1. This was confirmed for all larger problems instances and so relax-&-fix results are not shown in the additional tests reported below.

Over all ten instances for both 4 and 5 periods, Tables 3 and 4 jointly show a mean solution for CMRP1 that is 21% worse than that for CMRP in just 0.1% of the computing time under very tight capacity, and gets very near the zero optimal solution of CMRP in about 1% of the computing time under moderately tight capacity. These are encouraging results for model CMRP1. The

Table 4
Solution values with 5 products on 2 work centres over 5 periods

Instance	CMRP	CMRP RF	CMRP1	CMRP1 RF	CMRP2	CMRP Time
Very tight capacity						
1	106,215	977,044	138,195	1,106,044	149,953	30 s
2	108,800	8,897,200	130,670	3,702,232	108,800	133 s
3	46,572	1,545,646	47,603	1,304,627	72,139	790 s
4	59,549	1,124,291	59,549	1,139,669	69,174	6070 s
5	133,600	837,571	206,425	889,740	225,081	507 s
6	102,171	2,735,175	119,104	1,565,028	151,457	5532 s
7	164,484	3,309,489	176,494	4,206,024	199,712	50 s
8	72,661	778,449	142,621	752,958	157,093	5496 s
9	98,673	2,617,809	98,673	2,765,753	98,673	35 s
10	104,785	3,323,577	108,536	1,758,369	133,976	20 s
Mean solution	99,751	2,614,625	122,787	1,919,044	136,606	
Mean time	1866 s	6 s	1 s	4 s	4 s	
Moderately tight capacity						
1	0	1,486,738	0	776,764	0	6 s
2	0	2,612,437	0	1,318,401	0	27 s
3	0	1,064,165	0	1,215,020	0	370 s
4	0	469,380	5	920,781	5	713 s
5	0	415,316	174	1,253,971	20,237	153 s
6	0	616,220	33	14,807	33	121 s
7	0	1,980,041	92	1,445,184	9930	4 s
8	0	381,454	82	707,145	927	28 s
9	0	1,207,421	0	1,725,580	0	8 s
10	0	0	0	132,365	0	6 s
Mean solution	0	1,023,317	39	951,002	3113	
Mean time	144 s	5 s	1 sec	3 s	3 s	

(0 indicates an (optimal) solution value of zero inventory and zero backlogs)

equivalent results for model CMRP2 are less favourable, but still useful: 38% worse than CMRP in just 0.3% of the computing time under very tight capacity and "getting near" zero in 2% of the time under moderately tight capacity.

If both models took the same amount of time to solve, then model CMRP1 would not be useful, but the interesting point is that it took a small fraction of the time taken by model CMRP. A similar claim can be made for CMRP2, although it does do worse than CMRP1. The results for the small problems of Tables 3 and 4 for which model CMRP1 can be solved optimally prepare the way for tests with larger problems where optimal solutions for CMRP are impossible to find due to the combinatorial explosion of possible setup sequences.

How do the models scale up to a larger problems? Doubling the number of products to 10 confirmed that model CMRP is at its limits of optimality when planning 5 products on 2 machines over 5 periods. Instead of solving model CMRP optimally (which would take days or weeks), Table 5 shows tests results for 10 products on 2 machines over 5 periods where model CMRP's solution search time was limited to the mean amount needed by model CMRP1, thus permitting comparisons of solutions found in equal amounts of time. Observe that model CMRP's solutions were far worse than those of model CMRP1, both for very tight capacity (taking a mean 30 min) and moderately tight capacity (20 min), while model CMRP2 is fast, taking just 10 s. The trade-off is that although model CMRP2 occasionally gave a better solution than model CMRP1, it usually gave noticeably worse solutions and certainly so overall. However, note that under moderately tight capacity model

Table 5
Solution values with 10 products on 2 work centres over 5 periods

Instance	Very tight capacity			Moderately tight capacity		
	CMRP	CMRP1	CMRP2	CMRP	CMRP1	CMRP2
1	63,884	23,654	174,812	227,157	0	0
2	2,789,930	619	44,473	2,001,674	0	3325
3	2,197,547	488	92,814	1,623,433	44	44
4	3,288,966	188,875	267,718	10,190,519	0	0
5	868,310	69,218	261,233	2,000,155	85	409
6	1,799,483	32,912	136,933	911,945	38,233	40,953
7	2,861,868	50,642	437,736	2,236,964	506	506
8	1,978,556	211,489	119,984	1,738,724	188	6987
9	18,874,497	473,807	239,215	11,202,316	188	215
10	6,294,281	196,413	297,994	2,271,677	0	17,359
Mean solution	4,101,732	124,812	207,291	3,440,456	3924	6980
Mean time	1800 s	1800 s	10 s	1050 s	1050 s	10 s

(0 indicates an (optimal) solution value of zero inventory and zero backlogs)

Table 6
Solutions with 20 products on 5 work centres over 5 periods

Instance	Very tight capacity			Moderately tight capacity		
	CMRP	CMRP1	CMRP2	CMRP	CMRP1	CMRP2
1	—	—	887,656	267,698,238	131,717,765	15,842
2	—	—	642,472	—	244,087,600	419
3	—	—	889,215	—	31,823,265	23,882
4	—	—	550,101	—	—	33,421
5	—	—	381,632	—	153,422,888	56,333
6	—	—	299,210	—	—	1387
7	—	—	297,640	—	—	10,705
8	—	—	521,441	—	—	197
9	—	714,174,566	502,154	—	—	838
10	—	—	502,796	—	—	32,889
Mean solution	—	—	539,531	—	—	17,591
Mean time	25 s	25 s	22 s	25 s	25 s	23 s

The dash symbol indicates that no solution was identified within the time allowed

CMRP2 replicated the solution value of CMRP1 in 4 of the 10 instances.

A user does not know a priori which of models CMRP1 and CMRP2 will give a better solution for a given problem instance. How does one choose which model to use? There is a trade-off between solution time and schedule cost. If the user can afford to wait half an hour for a solution, then use model CMRP1. If not, then use CMRP2 which, for the test data, in a mean 10 s will provide a schedule that is on average about 1.7 times more costly than CMRP1.

Table 6 shows the results for 20 products on 5 work centres, limiting the time allowed for each model to the mean needed by CMRP2, i.e., at most 25 s. Note that the CPLEX solver could not even identify an initial solution for CMRP in this time, so huge are the problem instances. Model CMRP1 was allowed 5 s for its five MIPs, but CPLEX was only able to identify 5 solutions, out of a possible 20. It is clear that model CMRP2 provides far superior solutions, or the only solution, in the amount of CPU time it needs.

Table 7
Further mean results for model CMRP2 over 5 periods

	Number of instances	Very tight capacity		Moderately tight capacity	
		Value	Time	Value	Time
50 products 5 work centres	10	602,906	484 s	23,325	363 s
50 products 10 work centres	10	2,703,101	508 s	484,661	701 s
100 products 10 work centres	10	1,237,298	49,387 s	70,963	18,607 s

Table 8
CMRP2 solutions with 100 products and 10 work centres over 5 periods under moderately tight capacity (times in seconds)

Instance	First integer solution		1 min per MIP		5 min per MIP		60 min per MIP	
	Value	Time	Value	Time	Value	Time	Value	Time
1	3.0×10^9	1338	598,206	3480	114,555	12,020	31,233	19,063
2	1.5×10^9	442	42,281	1841	16,991	2418	33,534	8566
3	0.3×10^9	593	873,756	1779	105,773	2757	106,557	5896
4	1.0×10^9	655	149,997	1911	123,058	4124	72,679	25,930
5	0.9×10^9	912	66,474	2871	64,329	4287	91,528	31,474
6	1.7×10^9	830	40,830	1586	35,386	3192	35,841	22,828
7	2.0×10^9	1017	34,872	2426	75,524	4052	87,645	4330
8	3.5×10^9	700	6,725,573	2083	24,868	2324	24,676	5151
9	9.1×10^9	1096	187,006	3546	154,659	8237	140,823	37,830
10	5.6×10^9	822	84,927	2064	74,166	5134	85,113	25,004
Mean	2.9×10^9	841	880,392	2359	78,931	4885	70,963	18,607

Scaling up to larger structures, model CMRP is no longer solvable by CPLEX. Table 6 showed that CMRP1 is very inefficient compared to CMRP2. Both CMRP and CMRP1 are disconsidered in the tests of Tables 7 and 8 whose purpose is not to show the quality of the CMRP2 solutions compared to unknowable optimal solutions, but rather to show that model CMRP2 can provide solutions for fairly large problems and to provide insight into the trade-off between solution time and schedule cost.

Table 7 shows mean solution times and comparative solutions for model CMRP2 for 50 and 100 products on 5 and 10 workcentres. Observe that model CMRP2 was generally solvable within about 15 min for 50 products, a viable amount of computing time. However, for rolling horizon use, a period 1 schedule was actually identified and available for implementation after the application of model CMRP2 for period $t = 1$ only, i.e., after 4 min.

The mean CPU times for 100 products in Table 7 were not good—when capacity was moderately tight, model CMRP2 took a mean of 5.2 h when each MIP was allowed up to an hour's optimizing CPU time. However, when capacity was very tight, then a mean 13.7 h was needed to solve CMRP2. Both times are impracticable, but can be shortened by curtailing the branch-&-bound search at each MIP, as will now be shown.

The effect of limiting MIP search time under moderately tight capacity is shown in Table 8. When each MIP optimization was halted after identifying a first integer solution, the resulting CMRP2 solutions were awful. The solutions improved greatly when each MIP was allowed up

to 1 min. Most MIPs solved optimally in well less than this time and so the total time needed to solve CMRP2 was about 40 min (10 min for the period 1 schedule). The solutions improved still further, halving in mean value, when each MIP was allowed up to 5 min and the total time needed to solve CMRP2 doubled to about 80 min (20 min for the period 1 schedule). However, one of the ten instances had a (considerably) worse solution.

When, as in Table 7, each MIP was allowed up to 1 h CPU time, the CMRP2 solution time quadrupled to about 5 h, but with only a 10% improvement in the mean solution value. In fact, six of the ten solutions worsened. Why is this so? It could be that the approximate nature both of the augmented coefficient u^*_{iw} used to factor in setup times in model CMRP2 and of its setup-by-setup solution method do put limits on any improvements attempted by running the MIPs for a long time. In other words, there is little, if any, added value in running each CMRP2 MIP for more than about 5 min. In this case, a reasonable solution for 100 products on 10 machines is likely to be reached after an hour or 90 min, with a period 1 schedule being identified after 15–20 min).

7. Conclusions

The contribution of the paper has been to derive and test models for capacitated materials requirements planning that take into account sequence-dependent setup times. The basic model was twice refined through linear approximations of the setup times, so that it can be used for production scheduling for medium-to-large sized product structures, with up to 100 end-items and components, processed over 10 workcentres. The models' simultaneous treatment of decisions about allocation of products and components to workstations, production sequencing and lot-sizing make them powerful tools.

For small problems, the tests showed the exact model CMRP to be superior, with model CMRP1 producing mildly suboptimal solutions and model CMRP2 producing definitely suboptimal ones, albeit far more quickly. However, for small to medium sized problems CMRP1 produced far superior solutions in equal computer time than model CMRP.

As problem size increases, CMRP2 produces solutions within a computing time in which CMRP1 usually cannot even find a feasible solution. When CMRP1 does find a solution it is much worse than that of CMRP2. The inferiority of the CMRP1 is due to the now very large number of 0/1 variables in each MIP. One way forward for future research is to develop heuristics for CMRP1 and desist from using CPLEX to solve model CMRP1 for larger problems, even for a limited amount of search time.

The ability of a model such as CMRP2 to tackle medium-to-large multi-stage production lot sequencing and sizing problems in practicable computing time is encouraging. It can be used operationally to make complex allocation and sequencing decisions of reasonable quality, particularly if applied on a rolling horizon basis to quickly schedule only the most immediate setups, a common practice given the prevalent reality of imperfect demand forecasts.

A further advantage of model CMRP2 is that no special algorithms needed to be developed. Rather, the accelerated solutions can be achieved by using powerful modelling approximations whose parameters are calculated from the specific demand and production data. As a result, model CMRP2 is easily implemented using the high-level mathematical programming tools (Fourer et al. 1993; Hentenryck, 1999) now found in ERP softwares such as SAP-APO (SAP-APO, 2002), making use of the good default search strategies within industrial strength MIP solvers such as CPLEX (ILOG, 1999). Again, research in the future can develop heuristics for CMRP2 to permit even larger problems to be solved.

References

Baker, KR., 1993. Requirements planning. In: Graves, S.C., (Ed.), Handbooks in Operations Research and Management Science, Vol. 4. Elsevier Science Publishers, New York, pp. 571–627.

Blackburn, J.D., Kropp, D.H., Millen, R.A., 1986. A comparison of strategies to dampen nervousness in MRP systems. Management Science 32 (4), 413–429.

Clark, A.R., 1998. Batch sequencing and sizing with regular varying demand. Production Planning and Control 9 (3), 260–266.

Clark, A.R., Armentano, V.A., 1993. Echelon stock formulations for multi-stage lot-sizing with lead times. International Journal of Systems Science 24 (9), 1759–1775.

Clark, A.R., Clark, S.J., 2000. Rolling-horizon lot-sizing when setup times are sequence-dependent. International Journal of Production Research 38 (10), 2287–2308.

Clark, A.J., Scarf, H., 1960. Optimal policies for a multi-echelon inventory problem. Management Science 6, 475–490.

de Bodt, M.A., van Wassenhove, L.N., 1983. Cost increases due to demand uncertainty in MRP lot sizing. Decision Sciences 13 (3), 345–362.

Dillenberger, C., Wollensak, A., Zhang, W., 1992. On practical resource allocation for production planning and scheduling with different setup products. Draft paper, German Manufacturing Technology Centre, IBM, Sindelfingen, Germany.

Drexl, A., Kimms, A., 1998. Beyond Manufacturing Resource Planning (MRP II). Advanced Models and Methods for Production Planning. Springer, Berlin.

Fourer, R., Gay, D.M., Kernighan, B.W., 1993. AMPL—a Modeling Language for Mathematical Programming. Boyd and Fraser, Danvers, MA http://www.ampl.com/.

Hentenryck, P.V., 1999. The OPL Optimization Programming Language. MIT Press, Cambridge, MA.

ILOG, 1999. CPLEX 6.5 User's Manual. ILOG S.A, BP 85, 9 Rue de Verdun, 94253 Gentilly Cedex, France. http://www.cplex.com/.

ILOG, 2000. OPL Studio. ILOG S.A., BP 85, 9 Rue de Verdun, 94253 Gentilly Cedex, France. http://www.ilog.com/.

IOM, 2000a. Advanced Planning and Scheduling for Improved Business Performance. Cataloque: Short courses in Production and Operations Management for Autumn 2000–2001. Institute of Operations Management. http://www.iomnet.org.uk/.

IOM, 2000b. APS—From Myth to Reality. Institute of Operations Management seminar held on 28 September 2000. http://www.iomnet.org.uk/.

Kadipasaoglu, S.N., Sridharan, S.V., 1997. Measurement of instability in multi-level MRP systems. International Journal of Production Research 35, 713–737.

Kennerley, M., Neely, A.D., 2001. Enterprise resource planning: analysing the impact. Integrated Manufacturing Systems 12 (2), 103–113.

Kropp, D.H., Carlson, R.C., Jucker, J.V., 1983. Heuristic lot-sizing approaches for dealing with MRP systems nervousness. Decision Sciences 14, 156–165.

Kruse, G., 2000. Opportunities in APS—a broader view. Control 26 (7), 19–21.

Kuik, R., Salomon, M., van Wassenhove, L.N., Maes, J., 1993. Linear programming, simulated annealing and tabu search heuristics for lotsizing in bottleneck assembly systems. IIE Transactions 25 (1), 62–72.

Maes, J., van Wassenhove, L.N., 1986. Multi item single level capacitated dynamic lotsizing heuristics: a computational comparison (part ii: Rolling horizon). IIE Transactions 18, 124–129.

Maes, J., van Wassenhove, L.N., 1988. Multi-item single-level capacitated dynamic lot-sizing heuristics: a general review. Journal of the Operational Research Society 39 (11), 991–1004.

Meyr, H., 2000. Simultaneous lot-sizing and scheduling by combining local search with dual optimization. European Journal of Operational Research 120, 311–326.

Meyr, H., 2002. Simultaneous lotsizing and scheduling on parallel machines. European Journal of Operational Research 139, 277–292.

Ovacik, I.M., Uzsoy, R., 1995. Rolling horizon procedures for dynamic parallel machine scheduling with sequence dependent setup time. International Journal of Production Research 33, 3173–3192.

Potts, C.N., van Wassenhove, L.N., 1992. Integrating scheduling with batching and lot sizing: a review of algorithms and complexity. Journal of the Operational Research Society 43 (5), 395–406.

Proud, J.F., 1999. Master Scheduling: A Practical Guide to Competitive Manufacturing, 2nd Edition. Wiley, New York.

Robinson, A.G., Dilts, D.M., 1999. OR and ERP: can operations research play a role in fast-growing, enterprise-wide information systems? OR/MS Today 26 (3), 30–37.

Sait, S.M., Youssef, H., 1999. Iterative Computer Algorithms with Applications in Engineering. IEEE Computer Society, Los Alamitos, CA.

SAP-APO, 2002. Advanced Production Optimiser. http://www.sap.com/apo/.

Shapiro, J.F., 1993. Mathematical programming models and methods for production planning and scheduling. In: Graves, S.C. (Ed.), Handbooks in Operations Research and Management Science, Vol. 4. Elsevier Science Publishers, New York, pp. 371–443.

Silver, E.A., Pyke, D.F., Peterson, R., 1998. Inventory Management and Production Planning and Scheduling, 3rd Edition. Wiley, New York.

Thomas, L.J., McClain, J.O., 1993. An overview of production planning. In: Graves, S.C. (Ed.), Handbooks in Operations Research and Management Science, Vol. 4. Elsevier Science Publishers, New York, pp. 333–370.

Trade Matrix, 2000. http://www.i2.com/.

Wolsey, L.A., 1997. MIP modelling of changeovers in production planning and scheduling problems. European Journal of Operational Research 99, 154–165.

Wolsey, L.A., 1998. Integer Programming. Wiley, New York.

Wortmann, J.C., 1998. Evolution of ERP systems. In: Bitici, U.S., Carrie, A.S. (Eds.), Strategic Management of the Manufacturing Value Chain. Kluwer Academic Publishers, Boston.