Hierarchical Production Planning for General Jobs Shops : Part 2 : Evaluation and Application

Anshu Mehra, Ioannis Minis, Jean-Marie Proth

INRIA

INSTITUT NATIONAL DE RECHERCHE EN INFORMATIQUE ET EN AUTOMATIQUE

Hierarchical Production Planning for General Jobs Shops : Part 2 : Evaluation and Application

Anshu Mehra - Ioannis Minis
Jean-Marie Proth

N° 2634
Août 1995

PROGRAMME 5

Rapport de recherche

Les rapports de recherche de l'INRIA
sont disponibles en format postscript sous
ftp.inria.fr (192.93.2.54)

INRIA research reports
are available in postscript format
ftp.inria.fr (192.93.2.54)

si vous n'avez pas d'accès ftp
la forme papier peut être commandée par mail :
e-mail : dif.gesdif@inria.fr
(n'oubliez pas de mentionner votre adresse postale).

if you haven't access by ftp
we recommend ordering them by e-mail :
e-mail : dif.gesdif@inria.fr
(don't forget to mention your postal address).

par courrier :
Centre de Diffusion
INRIA
BP 105 - 78153 Le Chesnay Cedex (FRANCE)

by mail :
Centre de Diffusion
INRIA
BP 105 - 78153 Le Chesnay Cedex (FRANCE)

Planification hiérarchisée de la production en ateliers

Seconde partie : Evaluation et applications

Anshu Mehra* Ioannis Minis* Jean-Marie Proth†

Résumé

Dans la première partie, nous avons fourni un modèle hiérarchisé pour la planification à moyen et court terme, ainsi que les algorithmes qui fournissent des solutions proches de l'optimum pour ces problèmes.
Dans la seconde partie, nous évaluons ces algorithmes de différents points de vue. Nous comparons d'abord l'approche monolithique et l'approche hiérarchisée en termes d'occupation mémoire et de complexité. Nous étudions ensuite la qualité des plans de production obtenus en utilisant ces algorithmes. Finalement, nous présentons une application industrielle et comparons les résultats obtenus en utilisant l'approche hiérarchisée à ceux que fournit le système MRP II actuellement utilisé dans l'entreprise.

Mots clefs: MRP II, Gestion hiérarchisée de la production, Systèmes de gestion de la production monolithiques, Complexité, Occupation mémoire

Cette étude a été développée dans le cadre d'un accord entre la NSF (National Science Foundation) et l'INRIA par le CIM lab de l'Université du Maryland et le projet SAGEP de l'INRIA-Lorraine.

* Department of Mechanical Engineering and Institute for Systems Research, University of Maryland, College Park, MD 20742, USA

† INRIA-Lorraine, Technopôle Metz 2000, 4 rue Marconi, 57070 Metz, FRANCE

Hierarchical Production Planning for General Job Shops. Part 2: Evaluation and Application

A. Mehra* I. Minis* J.M. Proth†

Abstract

In the first part of this study, we provided a hierarchical model for medium- and short-term planning, as well as solution algorithms which iteratively provide near optimal solutions. In part 2, we evaluate these algorithms from different points of view. We first compare the hierarchical approach and the monolithic approach in terms of memory requirement and computational complexity. We then study the quality of production plans obtained using these algorithms. Finally, we present an industrial application and compose the results obtained using the hierarchical approach with those obtained by the MRP II system currently in use in the company.

Keywords: MRP II, Hierarchical production management system, Monolithic production management systems, Complexity, Memory requirement

This study has been performed by the CIM lab of the University of Maryland and the SAGEP project of INRIA-Lorraine as part of an agreement between NSF (National Science Foundation) and INRIA.

*Department of Mechanical Engineering and Institute for Systems Research, University of Maryland, College Park, MD 20742.

†INRIA-Lorraine, 4 Rue Marconi, Metz 2000, 57070 Metz, France.

1 Introduction

A hierarchical production management system has been proposed in part 1 of this study. The approach uses a two-level hierarchy and aggregates part types, machines, and time periods. The aggregation is performed using a simulated annealing and clustering analysis-based algorithm. The aggregate level of the hierarchy determines the quantities of part families to be processed by each cell during each aggregate time period (sub-periods). The detailed level determines the quantities of part types to be processed at each machine during each elementary time period. The proposed hierarchical scheme permits the computation of aggregate and detailed production plans when detailed demand/forecast information is not available for the medium-term planning horizon. It also allows absorption of random events without frequent recomputation. The second part is devoted to the evaluation of the solution algorithms presented in part 1 of the paper. Section 2 evaluates the solution algorithm. The quality of the solution, as well as memory requirement and computational complexity are considered. In section 3, we present an industrial application and compose the results obtained using the proposed hierarchical approach with the ones currently obtained using MRP II. Section 4 is the conclusion.

2 Evaluation of the Hierarchical Production Planning System

This section evaluates the hierarchical production planning system comprising the formulation presented in part 1. Three basic evaluation criteria are used to compare the hierarchical approach to the monolithic production planning approach: (i) the computational time required to obtain the production plan, (ii) the required computer memory, and (iii) the quality of the production plans. Note that the monolithic approach provides the optimum plan, by definition. Section 2.1 presents the savings achieved by the hierarchical production planning system in computational time and memory requirements, with respect to the monolithic planning system. Section 2.2 examines the quality of the solutions obtained from the hierarchical approach using a large set of realistic production planning problems.

2.1 Memory Requirements and Computational Time

In general, the size of the problems solved by the hierarchical approach is smaller than the monolithic planning problem. Also, the problems at the detailed level of the planning hierarchy are independent of each other, and can be solved in parallel, resulting in low computational times. In order to compute the memory requirements and computational times for the hierarchical and monolithic approaches, we make the following simplifying assumptions:

1. Each cell has the same number of machines $a = \frac{|\mathcal{M}|}{|\mathcal{C}|}$, where $|\mathcal{M}|$ is the total number of machines and $|\mathcal{C}|$ is the total number of cells.
2. Each part family has the same number of part types $b = \frac{|\mathcal{P}|}{|\mathcal{F}|}$, where $|\mathcal{P}|$ is the total number of part types and $|\mathcal{F}|$ is the total number of part families.
3. Each part type requires $c \cdot |\mathcal{M}|$ operations, where c is a positive real number.
4. Each part family requires $c \cdot |\mathcal{C}|$ macro-operations, i.e. the number of operations of each part type during a visit to a cell is a.

It is expected that these simplifications do not affect the comparison of the hierarchical and monolithic approaches significantly.

The evaluation of the computational time and memory size is based on the fact that both approaches are formulated in terms of linear programming problems. The latter are solved using the Simplex algorithm, for which there exist well known relationships to provide the computational time and memory size. Our evaluation determines the values of these parameters for both approaches.

Consider the following linear programming problem $\mathcal{L}$:

$$\text{Minimize } \left\{ \boldsymbol{c}^T \cdot \boldsymbol{x} : \boldsymbol{A} \cdot \boldsymbol{x} \leq \boldsymbol{b}; \boldsymbol{x} \geq 0 \right\}$$

where $\boldsymbol{A}$ is an $m_{\mathcal{L}} \cdot n_{\mathcal{L}}$ matrix and $\boldsymbol{b}$ an $m_{\mathcal{L}} \cdot 1$ vector of non-negative real numbers. Let $\gamma_{\mathcal{L}}$ be the number of non-zero elements in the matrix $\boldsymbol{A}$.

Computational Times

The computational time $CT_{\mathcal{L}}$ required to solve the above linear programming problem by the Simplex algorithm is given by Bazaraa [1] to be:

$$CT_{\mathcal{L}} = K_1 \cdot (m_{\mathcal{L}})^{2.17} \cdot (n_{\mathcal{L}})^{0.67} \cdot (\gamma_{\mathcal{L}})^{0.33} \tag{1}$$

where K_1 is a positive real number. Thus, the computational time of the monolithic approach is given by:

$$CT_{\mathcal{MP}} = K_1 \cdot (m_{\mathcal{MP}})^{2.17} \cdot (n_{\mathcal{MP}})^{0.67} \cdot (\gamma_{\mathcal{MP}})^{0.33} \tag{2}$$

The hierarchical approach consists of solving at each iteration the aggregate problem $\mathcal{APP}$, the part family disaggregation (PFD) problems $\mathcal{PFD}(f_r)$ $\forall r$, and the temporal and spatial disaggregation (TSD) problems $\mathcal{TSD}(\kappa, c_v)$ $\forall \kappa, v$. The PFD problems are independent of each other and hence can be solved in parallel. This is also true for the TSD problems. Thus, the computational time required for solving the problems in the planning hierarchy using parallel computation is given by:

$$CT^p_{\mathcal{HPP}} = \left[CT_{\mathcal{APP}} + CT_{\mathcal{PFD}(f_r)} + CT_{\mathcal{TSD}(\kappa, c_v)} \right] \cdot U \tag{3}$$

where U is the total number of iterations required for the convergence of the solution algorithm. On the other hand, if the problems at the detailed level are solved sequentially there are $|\mathcal{F}|$ part family disaggregation problems and $|\mathcal{C}| \cdot Z$ temporal and spatial disaggregation problems. The computational time in this case becomes:

$$CT^s_{\mathcal{HPP}} = \left[CT_{\mathcal{APP}} + CT_{\mathcal{PFD}(f_r)} \cdot |\mathcal{F}| + CT_{\mathcal{TSD}(\kappa, c_v)} \cdot |\mathcal{C}| \cdot Z \right] \cdot U, \tag{4}$$

The values of $CT_{\mathcal{APP}}, CT_{\mathcal{PFD}(f_r)}$, and $CT_{\mathcal{TSD}(\kappa, c_v)}$ are given by equation (1).

Memory Requirements

The memory size (in terms of the number of bytes) required by the Simplex algorithm for the solution of a linear program $\mathcal{L}$ is given by Bazaraa [1] to be:

$$MR_{\mathcal{L}} = K_2 \cdot (m_{\mathcal{L}} + 2(n_{\mathcal{L}} + \gamma_{\mathcal{L}})) \tag{5}$$

where $MR_{\mathcal{L}}$ represents the memory required (in bytes) and K_2 is a positive integer. For the monolithic problem, the memory requirement is given by:

$$MR_{\mathcal{MP}} = K_2 \cdot (m_{\mathcal{MP}} + +2(n_{\mathcal{MP}} + \gamma_{\mathcal{MP}})) \tag{6}$$

The maximum memory required to solve the planning problem by the parallel or sequential hierarchical approaches is the maximum of the memory size corresponding to the three subproblems; i.e.

$$MR_{\mathcal{HPP}} = \max\left(MR_{\mathcal{APP}}, MR_{\mathcal{PFD}(f_r)}, MR_{\mathcal{TSD}(\kappa,c_e)}\right) \tag{7}$$

The values of $MR_{\mathcal{APP}}, MR_{\mathcal{PFD}(f_r)}, MR_{\mathcal{TSD}(\kappa,c_e)}$ are given by equation (5).

The following sections evaluate the number of constraints, m, number of variables, n, and number of non-zero elements, γ, of matrix $\boldsymbol{A}$ for the monolithic, aggregate, part family disaggregation, and temporal and spatial disaggregation problems. These values are then used to compute CT and MR for the two approaches using equations (3) – (7).

2.1.1 Computation of Problem Parameters: Monolithic Approach

The number of constraints in the monolithic planning problem $\mathcal{MP}$:

$$\begin{aligned} m_{\mathcal{MP}} &= 2|\mathcal{P}| \cdot Z \cdot z \ + \ c \cdot |\mathcal{M}| \cdot |\mathcal{P}| \cdot Z \cdot z \\ &\quad + |\mathcal{M}| \cdot Z \cdot z \ + \ c(|\mathcal{M}| - |\mathcal{C}|) \cdot |\mathcal{P}| \cdot Z \end{aligned} \tag{8}$$

where

- $2|\mathcal{P}| \cdot Z \cdot z$ is the number of constraints introduced to transform problem $\mathcal{MP}$ into a linear program.
- $c \cdot |\mathcal{M}| \cdot |\mathcal{P}| \cdot Z \cdot z$ is the number of constraints used to limit $u^k_{j,w}$ to the input buffer inventory level $s^{k-1}_{j,w-1}$ at the end of the previous elementary period.

- $|\mathcal{M}| \cdot Z \cdot z$ is the number of capacity constraints.

- $c(|\mathcal{M}| - |\mathcal{C}|) \cdot |\mathcal{P}| \cdot Z$ is the number of constraints used to maintain the inter-cell buffer levels at the end of each sub-period to the required values.

The number of variables in problem $\mathcal{MP}$ is:

$$n_{\mathcal{MP}} = (|\mathcal{P}| + c \cdot |\mathcal{M}| \cdot |\mathcal{P}|)Z \cdot z \tag{9}$$

where

- $Z \cdot z \cdot |\mathcal{P}|$ is the number of artificial variables introduced to transform problem $\mathcal{MP}$ into a linear program.

- $c \cdot |\mathcal{M}| \cdot |\mathcal{P}| \cdot Z \cdot z$ is the number of variables $u^k_{j,w}$.

The number of non-zero elements of matrix $\boldsymbol{A}$ of problem $\mathcal{MP}$ is:

$$\begin{aligned} \gamma_{\mathcal{MP}} &= 2 \cdot |\mathcal{P}| \cdot Z \cdot z + |\mathcal{P}| \cdot Z \cdot z(Z \cdot z + 1) \\ &+ c(|\mathcal{M}| - |\mathcal{C}|)|\mathcal{P}| \cdot Z^2 \cdot z^2 \\ &+ 0.5 \cdot c \cdot |\mathcal{C}| \cdot |\mathcal{P}| \left(Z \cdot z(Z \cdot z + 1) + Z(Z-1)z^2\right) \\ &+ c \cdot |\mathcal{M}| \cdot |\mathcal{P}| \cdot Z \cdot z \\ &+ c \cdot |\mathcal{P}|(|\mathcal{M}| - |\mathcal{C}|)Z \cdot z(Z+1) \end{aligned} \tag{10}$$

where

- $2 \cdot |\mathcal{P}| \cdot Z \cdot z$ is the number of non-zero artificial variables, and $|\mathcal{P}| \cdot Z \cdot z(Z \cdot z + 1)$ is the number of non-zero production variables $u^k_{j,w}$ in the constraints introduced to transform problem $\mathcal{MP}$ into a linear program.

- The third term in the right hand side of equation (10) is the number of non-zero variables in constraint:

$$\sum_{a=1}^{k} u^a_{j,w} - \sum_{a=1}^{k-1} u^a_{j,w-1} \leq s^0_{j,w-1} \;\forall j, w, k \text{ such that } h_{j,w} = 0$$

For each operation $o_{j,w}$ of part type p_j such that $o_{j,w}$ is not the first operation in a sub-routing of p_j, and for elementary period k, there are $k + (k-1)$ non-zero variables $u^k_{j,w}$. Recall that each part type requires $c \cdot |\mathcal{M}|$ operations and $c \cdot |\mathcal{C}|$ macro-operations, and that the number of operations which are not the first operation in a part sub-routing is $c(|\mathcal{M}| - |\mathcal{C}|)$. Hence, for all j, w, k such that $h_{j,w} = 0$, the total number of non-zero variables in the above constraint is:

$$\begin{aligned} & \sum_{k=1}^{Z \cdot z} [|\mathcal{P}| \cdot c(|\mathcal{M}| - |\mathcal{C}|)\,(k + (k-1))] \\ = \; & c(|\mathcal{M}| - |\mathcal{C}|)|\mathcal{P}| \cdot Z^2 \cdot z^2, \end{aligned}$$

since $\sum_{k=1}^{Z \cdot z} k = \frac{1}{2} Z \cdot z \cdot (Z \cdot z + 1)$.

- The fourth term in the right hand side of equation (10) is the number of non-zero variables in constraint:

$$\sum_{a=1}^{k} u^a_{j,w} - \sum_{a=1}^{(\kappa-1)z} u_{j,w-1} \leq s^0_{j,w-1}$$

$$\forall \kappa, k \in h(\kappa), j \text{ and for } w = y^1_j, y^2_j, ..., y^{\bar{n}_r}_j, \text{ where } p_j \in f_r.$$

  The number of non-zero variables in the above inequality is $k + (\kappa - 1)z$. Since this inequality is true for $c \cdot |\mathcal{C}|$ operations of each part type, the number of non-zero production variables $u^k_{j,w}$ in the above constraint is:

$$\begin{aligned} & \sum_{\kappa=1}^{Z} \sum_{a=(\kappa-1)z+1}^{\kappa \cdot z} \sum_{j=1}^{|\mathcal{P}|} (c \cdot |\mathcal{C}|(k + (\kappa - 1)z)) \\ = \; & \frac{1}{2} c \cdot |\mathcal{C}| \cdot |\mathcal{P}| \left(Z \cdot z(Z \cdot z + 1) + Z(Z-1)z^2 \right) \end{aligned}$$

- The fifth term in the right hand side of equation (10) is the total number of non-zero variables $u^k_{j,w}$ in the capacity constraint.

- The sixth term in the right hand side of equation (10) is the number of non-zero variables in the constraints on the intra-cell inventory levels at the end of each sub-period.

The values of $CT_{\mathcal{MP}}$ and $MR_{\mathcal{MP}}$ are determined by substituting $m_{\mathcal{MP}}, n_{\mathcal{MP}}$, and $\gamma_{\mathcal{MP}}$ from equations (8) ÷ (10) in (2) and (6).

2.1.2 Computation of Problem Parameters: Hierarchical Approach

The hierarchical approach consists of solving the aggregate, the part family disaggregation (PFD), and the temporal and spatial disaggregation (TSD) problems. Below we evaluate the computational time and memory requirements for all three problems in a manner similar to that of the monolithic approach.

Aggregate Problem

The number of constraints in the aggregate problem $\mathcal{APP}$ formulated in part 1 is:

$$m_{\mathcal{APP}} = 2Z \cdot |\mathcal{F}| + c \cdot |\mathcal{C}| \cdot |\mathcal{F}| \cdot Z + Z \cdot |\mathcal{C}| \tag{11}$$

The number of variables in problem $\mathcal{APP}$ is:

$$n_{\mathcal{APP}} = Z(c \cdot |\mathcal{C}| \cdot |\mathcal{F}| + |\mathcal{F}|) \tag{12}$$

The number of non-zero elements in matrix $\boldsymbol{A}$ of problem $\mathcal{APP}$ is:

$$\begin{aligned} \gamma_{\mathcal{APP}} &= 2|\mathcal{F}| \cdot Z + |\mathcal{F}| \cdot Z(Z+1) \\ &+ \tfrac{1}{2}c \cdot |\mathcal{C}| \cdot |\mathcal{F}| \cdot Z(Z+1) + \tfrac{1}{2}(c \cdot |\mathcal{C}| - 1)|\mathcal{F}| \cdot Z(Z-1) \\ &+ c \cdot |\mathcal{C}| \cdot |\mathcal{F}| \cdot Z \end{aligned} \tag{13}$$

Part Family Disaggregation Problem

The number of constraints in the part family disaggregation problem $\mathcal{PFD}(f_r)$ formulated in part 1 is:

$$m_{\mathcal{PFD}(f_r)} = 2b \cdot Z + b \cdot c \cdot |\mathcal{C}| \cdot Z + c \cdot |\mathcal{C}| \cdot Z \tag{14}$$

The number of variables in problem $\mathcal{PFD}(f_r)$ is:

$$n_{\mathcal{PFD}(f_r)} = b \cdot c \cdot |\mathcal{C}| \cdot Z + b \cdot Z \tag{15}$$

The number of non-zero elements in matrix $\boldsymbol{A}$ of problem $\mathcal{PFD}(f_r)$ is:

$$\begin{aligned}\gamma_{\mathcal{PFD}(f_r)} &= 2b \cdot Z + b \cdot Z(Z+1) \\ &+\tfrac{1}{2}b \cdot c \cdot |\mathcal{C}| \cdot Z(Z+1) + \tfrac{1}{2}b(c \cdot |\mathcal{C}| - 1)Z(Z-1) \\ &+b \cdot c \cdot |\mathcal{C}| \cdot Z\end{aligned} \tag{16}$$

Temporal and Spatial Disaggregation Problem

The number of constraints in the temporal and spatial disaggregation problem $\mathcal{TSDC}(\kappa, c_v)$, which was formulated in part 1 is:

$$m_{\mathcal{TSDC}(\kappa,c_v)} = 2|\mathcal{P}| \cdot z + a \cdot c \cdot |\mathcal{P}| \cdot z + a \cdot z + c \cdot |\mathcal{P}| + (a-1) \cdot c \cdot |\mathcal{P}| \tag{17}$$

The number of variables in problem $\mathcal{TSDC}(\kappa, c_v)$ is:

$$n_{\mathcal{TSDC}(\kappa,c_v)} = |\mathcal{P}| \cdot z(1 + a \cdot c) \tag{18}$$

The number of non-zero elements in matrix $\boldsymbol{A}$ of problem $\mathcal{TSDC}(\kappa, c_v)$ is:

$$\begin{aligned}\gamma_{\mathcal{TSDC}(\kappa,c_v)} &= 2|\mathcal{P}| \cdot z + |\mathcal{P}| \cdot z(z+1) \\ &+(a-1)c \cdot |\mathcal{P}| \cdot z^2 \\ &+\tfrac{1}{2}c \cdot |\mathcal{P}| \cdot z(z+1) \\ &+a \cdot c \cdot |\mathcal{P}| \cdot z + c \cdot |\mathcal{P}| \cdot z + 2(a-1) \cdot c \cdot |\mathcal{P}| \cdot z\end{aligned} \tag{19}$$

$CT_{\mathcal{TSD}}$ and $MR_{\mathcal{TSD}}$ are determined by substituting $m_{\mathcal{TSD}}, n_{\mathcal{TSD}}$, and $\gamma_{\mathcal{TSD}}$ from equations (17) – (19) into equations (1) and (5).

2.1.3 Comparison of Memory Requirements and Computational Times of the Monolithic and Hierarchical Approaches

In order to compare the monolithic and hierarchical approaches we define the following ratios. Let $V = \frac{MR_{\mathcal{MP}}}{MR_{\mathcal{HPP}}}$ where $MR_{\mathcal{MP}}$ and $MR_{\mathcal{HPP}}$ are the memory requirements when solving a planning problem by the monolithic and hierarchical approaches, respectively. Furthermore, let $W = \frac{CT_{\mathcal{MP}}}{CT_{\mathcal{HPP}}}$, where and $CT_{\mathcal{HPP}}$ is the computational time required by the hierarchical approach for one iteration. W is equal to the number of iterations of the iterative algorithm, , for which the computational time is equal to that of the monolithic approach.

Table 1: Fixed values of system parameters

Parameter	Value
$\lvert\mathcal{M}\rvert$	50
$\lvert\mathcal{P}\rvert$	500
Z	10
z	5
a	5
b	20
c	0.3

In the following paragraphs we examine the dependence of the ratios V and W on the system parameters $a, b, c, |\mathcal{M}|, |\mathcal{P}|, Z$ and z. To facilitate the discussion we have grouped relevant system parameters and we examine the effects of each group on V and W separately. Group A contains two parameters, the number of sub-periods, Z, and the number of elementary periods in a sub-period, z. Group B contains two parameters, the number of part types, $|\mathcal{P}|$, and the number of part types in a family, b. Finally, group C contains three parameters, the number of machines, $|\mathcal{M}|$, the number of machines in a cell, a, and the ratio c of the number of operations in a part routing to the number of machines. When examining the effects of parameters Z and z of group A on the ratios V and W, the remaining parameters $|\mathcal{P}|$,b, $|\mathcal{M}|, a$, and c are fixed to the values given in Table 1. The effects of the parameters in the other groups are examined in an analogous manner.

Group A

Figure 1 shows the effects of the number of sub-periods Z and the number of elementary periods in each sub-period z on the memory requirement ratio V. The lowest curve corresponds to $z = 1$ and the others correspond to $z = 2, 3, 5$ and 7. The figure shows that the ratio V increases monotonically with Z and z. However, the function $V(Z, z)$ converges to a limit beyond a certain Z value.

The dependence of V on Z, and z can be explained as follows: For low values of Z

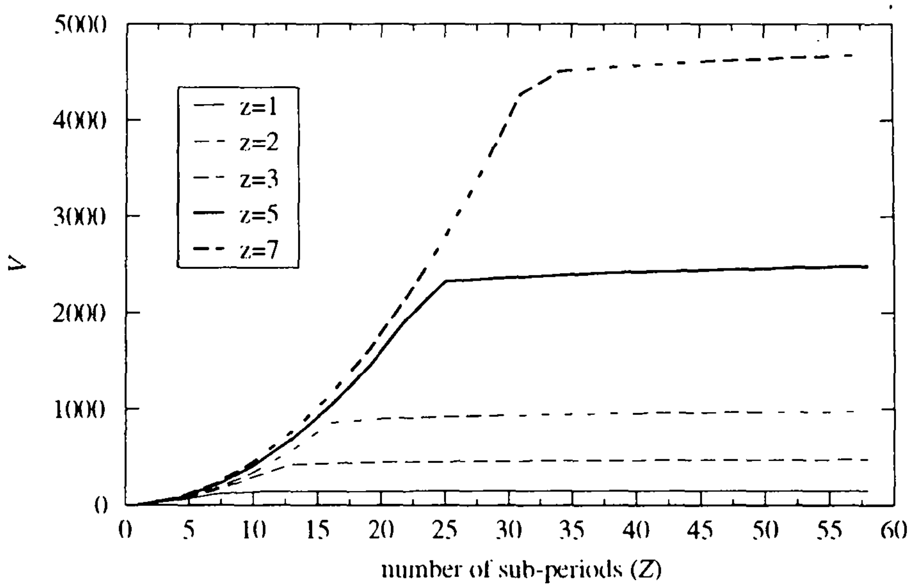
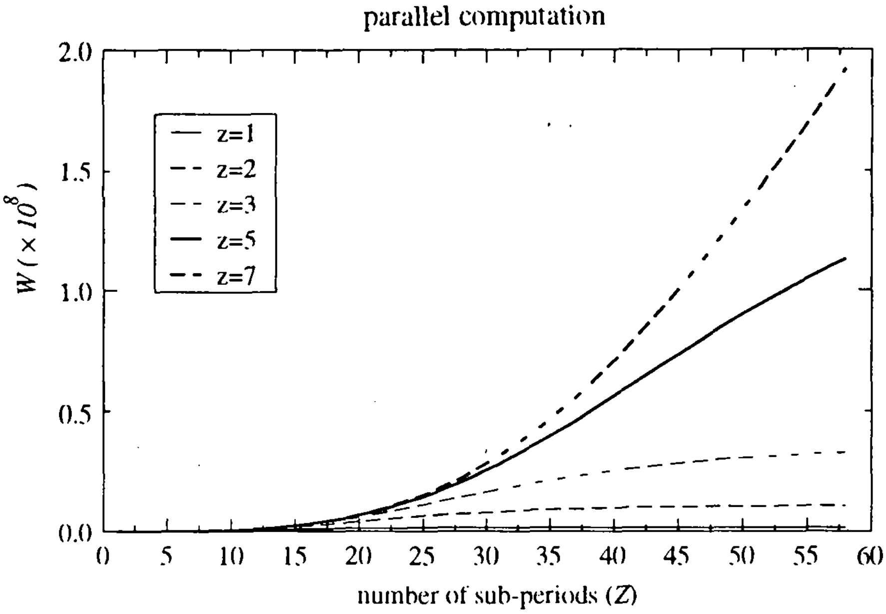


Figure 1: Dependence of the ratio of memory requirements V on parameters Z and z

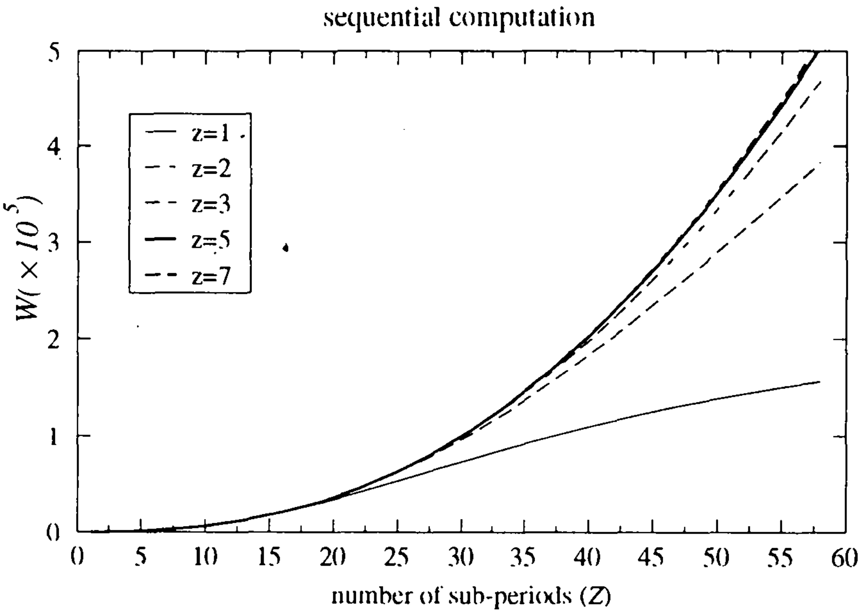
the function $V(Z, z)$ increases with Z for constant z, because $MR_{\mathcal{MP}}$ increases with Z while $MR_{\mathcal{TSD}(\kappa,c_v)}$ (which is the maximum among $MR_{\mathcal{APP}}$, $MR_{\mathcal{PFD}(f_r)}$, and $MR_{\mathcal{TSD}(\kappa,c_v)}$) remains constant. The function $V(Z, z)$ increases with z for constant Z because: (i) for low values of Z the percentage increase of $MR_{\mathcal{MP}}$ is greater than that of $MR_{\mathcal{TSD}(\kappa,c_v)}$ for the same increase in z, and (ii) for high values of Z, $MR_{\mathcal{MP}}$ increases with z while $MR_{\mathcal{APP}}$ remains constant. This also explains the larger difference in $V(z)$ (for increasing z and constant Z) for large Z values in comparison to low Z values.

$MR_{\mathcal{APP}}$ and $MR_{\mathcal{PFD}(f_r)}$ are invariant to z and $MR_{\mathcal{TSD}(\kappa,c_v)}$ is invariant to Z. For low values of Z and constant z, $MR_{\mathcal{TSD}(\kappa,c_v)}$ dominates $MR_{\mathcal{APP}}$ and $MR_{\mathcal{PFD}(f_r)}$ while for high values of Z, $MR_{\mathcal{APP}}$ dominates $MR_{\mathcal{PFD}(f_r)}$ and $MR_{\mathcal{TSD}(\kappa,c_v)}$. For high values of Z, $MR_{\mathcal{MP}}$ and $MR_{\mathcal{APP}}$ increase linearly with Z and therefore V converges to a constant value.

Figure 2 shows the effects of the number of sub-periods Z and the number of elementary periods in a sub-period z on the ratio W, for both parallel and sequential computation. W increases monotonically with Z and z for both types of computation. All $W(Z)$ curves approach a finite limit as $Z \to \infty$. This is because for high values



(i)



(ii)

Figure 2: Dependence of the ratio of computational times W on the parameters Z and z. (i) Parallel computation. (ii) Sequential computation

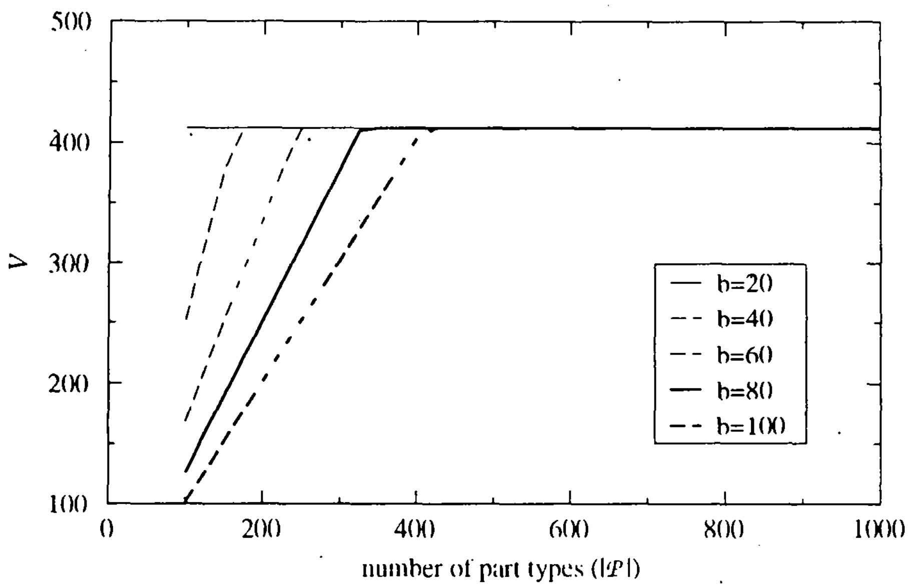


Figure 3: Dependence of the ratio of memory requirements V on parameters $|\mathcal{P}|$ and b

of Z, $CT_{\mathcal{APP}}$ is more than 99% of $CT_{\mathcal{HPP}}$ and the percentage increase in $CT_{\mathcal{APP}}$ and $CT_{\mathcal{MP}}$ for the same increase in Z is nearly the same. Note that the value of W for parallel computation is roughly 400 times greater than the value corresponding to sequential computation, since in the former case, problems $\mathcal{PFD}(f_r)$ $\forall r$ and problems $\mathcal{TSD}(\kappa, c_v)$ $\forall \kappa, c_v$ are solved in parallel.

In conclusion, the memory requirements and computational times are significantly less for the hierarchical approach. Also, the savings reflected by both ratios increase for longer planning horizons.

Group B

Figure 3 shows the effects of the number of part types $|\mathcal{P}|$ manufactured in the facility and the number of part types b in a part family on the memory requirement ratio V. The lowest curve corresponds to $b = 100$ and the others correspond to $b = 80, 60, 40$, and 20. The ratio V increases monotonically with $|\mathcal{P}|$ and converges to a limit. Note that all lines converge to the same limit.

This behavior can be explained as follows: For low values of $|\mathcal{P}|$ (< 400 for $b = 100$), $MR_{\mathcal{PFD}(f_r)}$ dominates $MR_{\mathcal{APP}}$ and $MR_{\mathcal{TSD}(\kappa,c_v)}$. $MR_{\mathcal{MP}}$ and $MR_{\mathcal{TSD}(\kappa,c_v)}$

increase linearly with $|\mathcal{P}|$. On the other hand, at the range of larger $|\mathcal{P}|$ values, $MR_{\mathcal{TSD}(\kappa,c_v)}$ dominates $MR_{\mathcal{APP}}$ and $MR_{\mathcal{PFD}(f_r)}$. In addition, both $MR_{\mathcal{MP}}$ and $MR_{\mathcal{TSD}(\kappa,c_v)}$ increase linearly with $|\mathcal{P}|$ and, therefore, the ratio V remains constant.

Note that V decreases monotonically with b for low values of $|\mathcal{P}|$. This is because $MR_{\mathcal{MP}}$ is almost constant with respect to the number of part types b in a family for a fixed $|\mathcal{P}|$; on the other hand $MR_{\mathcal{PFD}(f_r)}$ increases with b, and, therefore, V decreases monotonically with b for low values of $|\mathcal{P}|$.

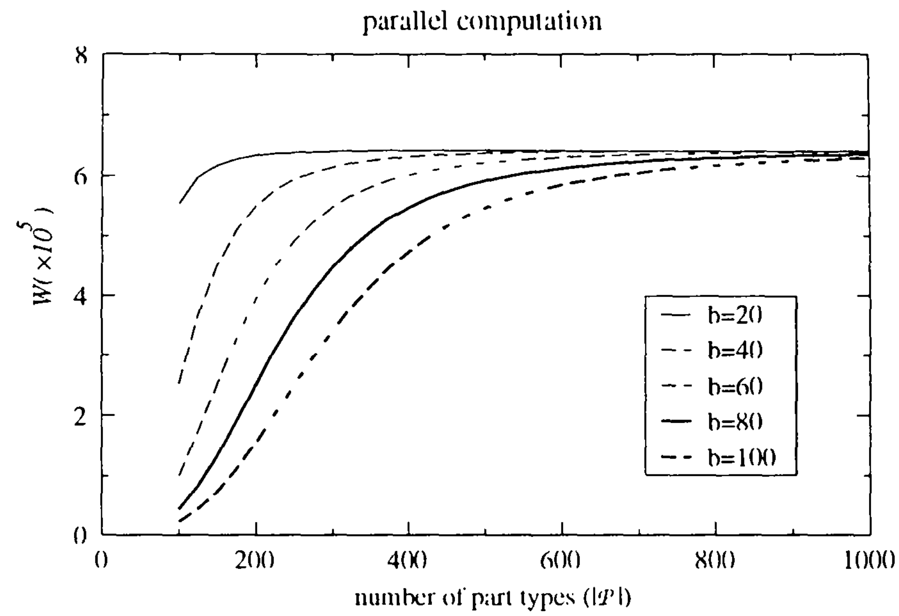
Figure 4 shows the effects of $|\mathcal{P}|$ and b on the computational times ratio W for both parallel and sequential computations. In general, W increases monotonically with $|\mathcal{P}|$ and decreases monotonically with b converging to the same finite limit for all values of b and $|\mathcal{P}|$. This is because for larger values of $|\mathcal{P}|$, the TSD problem contributes 99% of the total time required by the hierarchical approach, and the computational times for both the monolithic and the TSD problems increase linearly with $|\mathcal{P}|$. W for parallel computation is roughly 100 times the value corresponding to sequential computation, since problems $\mathcal{PFD}(f_r)$ $\forall r$ and $\mathcal{TSD}(\kappa, c_v)$ $\forall \kappa, c_v$ are solved concurrently.

In conclusion, the memory requirements and computational times are significantly less for the hierarchical approach in comparison to the monolithic approach. Also, the savings in both ratios V and W increase for large number of part types in the facility and for small part family sizes.

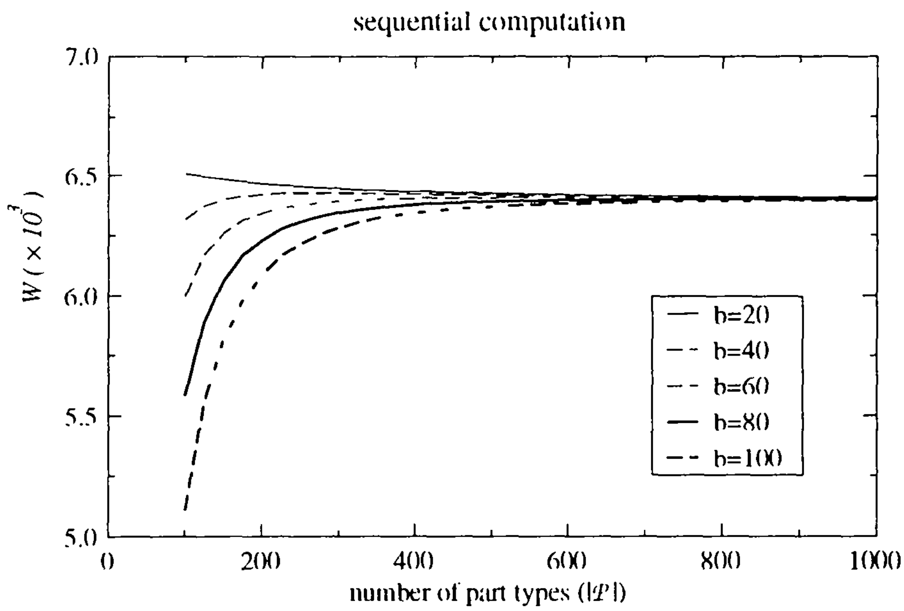
Group C

Figure 5 shows the dependence of the memory requirement ratio V on the number of machines $|\mathcal{M}|$ in the facility, the number of machines a in a cell, and the ratio c of number of operations per part type to number of machines. The curves in Figure 5(i) correspond to a = 4,8,12,16, 20, for $c = 0.30$, and the curves in Figure 5(ii) correspond to c = 0.05, 0.25, 0.5, 0.75, 1.00 for $a = 5$.

The ratio V increases monotonically with $|\mathcal{M}|$ and c, and decreases monotonically with a. V increases monotonically with $|\mathcal{M}|$ for fixed a and c because: (i) $MR_{\mathcal{TSD}(\kappa,c_v)}$ dominates $MR_{\mathcal{APP}}$ and $MR_{\mathcal{PFD}(f_r)}$, and does not depend on $|\mathcal{M}|$. Note also that $MR_{\mathcal{TSD}(\kappa,c_v)}$ depends on a which is constant. (ii) $MR_{\mathcal{MP}}$ increases linearly with $|\mathcal{M}|$.

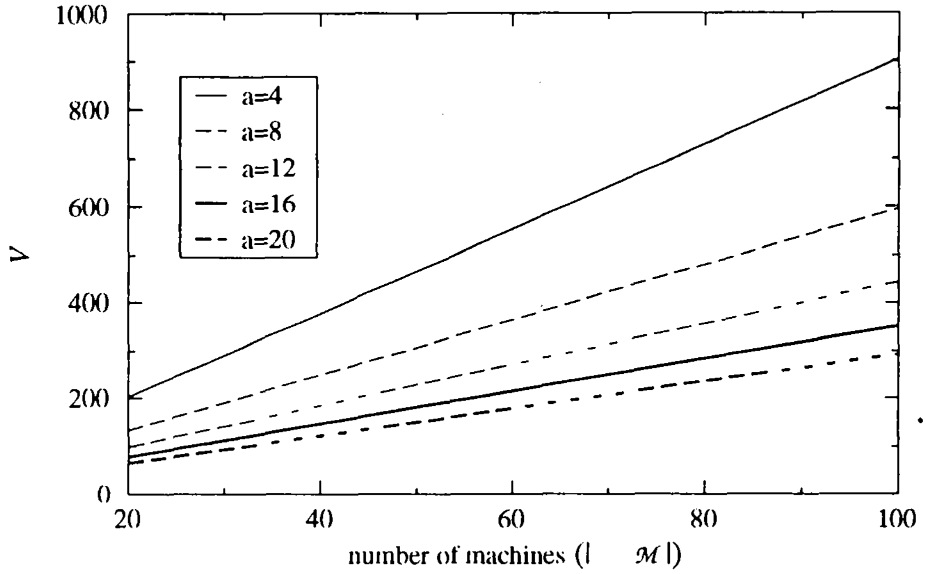


(i)

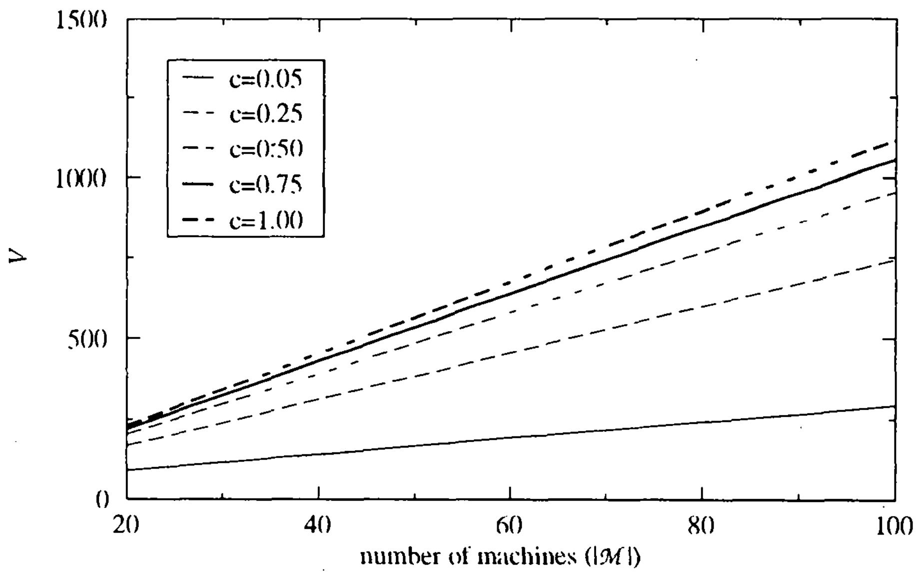


(ii)

Figure 4: Dependence of the ratio of computational times W on the parameters $|\mathcal{P}|$ and b. (i) Parallel computation. (ii) Sequential computation



(i)



(ii)

Figure 5: Dependence of the ratio of memory requirements V on the parameters (i) $|\mathcal{M}|$ and a and (ii) $|\mathcal{M}|$ and c

The ratio V decreases monotonically with a for fixed c and $|\mathcal{M}|$ because $MR_{\mathcal{MP}}$ is invariant to a and $MR_{\mathcal{TSD}(\kappa,c_v)}$ decreases linearly with a. The ratio V increases monotonically with c for fixed $|\mathcal{M}|$ and a because the percentage increase in $MR_{\mathcal{MP}}$ is higher than that of $MR_{\mathcal{TSD}(\kappa,c_v)}$ for the same increase in c. For $|\mathcal{M}| = 100$, and for an increase in c from 0.05 to 1.00, the percentage increase in $MR_{\mathcal{MP}}$ is about 1700% compared to an increase of about 450% for $MR_{\mathcal{TSD}(\kappa,c_v)}$.

Figures 6 and 7 show the effects of $|\mathcal{M}|$, a, and c on the computational time ratios W for both parallel and sequential computations. W increases monotonically with $|\mathcal{M}|$ and c, and decreases monotonically with a for both parallel and sequential computations. The dependence of W on $|\mathcal{M}|, a$, and c can be attributed to the fact that in general: (i) $CT_{\mathcal{TSD}(\kappa,c_v)}$ is equal to more than 99% of $CT_{\mathcal{HPP}}$, (ii) $CT_{\mathcal{TSD}(\kappa,c_v)}$ is invariant to $|\mathcal{M}|$ and decreases with increasing a, (iii) $CT_{\mathcal{MP}}$ is invariant to a, and (iv) the percentage increase in $CT_{\mathcal{MP}}$ is more than that of $CT_{\mathcal{TSD}(\kappa,c_v)}$ for the same increase in c. Note that the savings in computational times and memory requirements when using the hierarchical approach increase for increasing number of machines in the facility, for longer routings of part types and for smaller cell sizes.

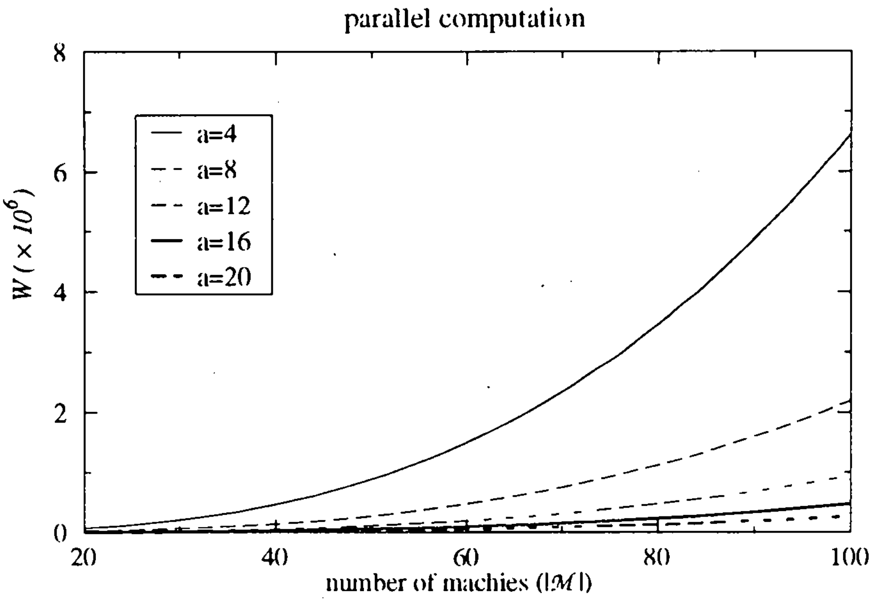
Summary of Comparisons

The following results can be deduced from the above analysis:

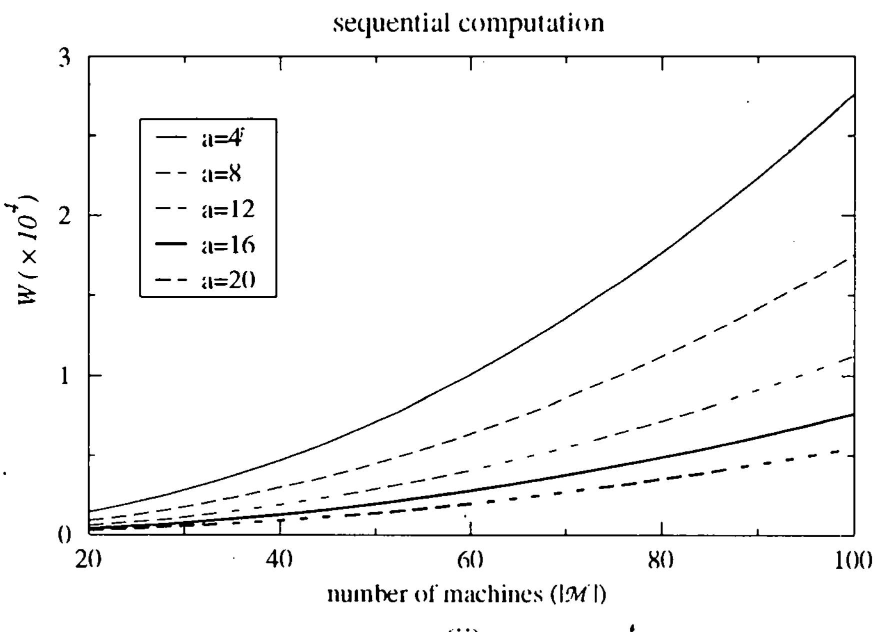
1. The memory size for the hierarchical approach is typically less than 1% of the memory required by the monolithic approach.

2. The number of iterations for which the computational time of the hierarchical approach (using sequential computation) is equal to the computational time of the monolithic approach is in the order of 10^4. Typically the hierarchical approach converges in less than 50 iterations; therefore, the ratio of computational times of the monolithic vs. the sequential hierarchical approach is about 200.

3. The ratio of computational times of the sequential hierarchical approach vs. the parallel hierarchical approach is, in general, greater than 100. Thus, the parallel

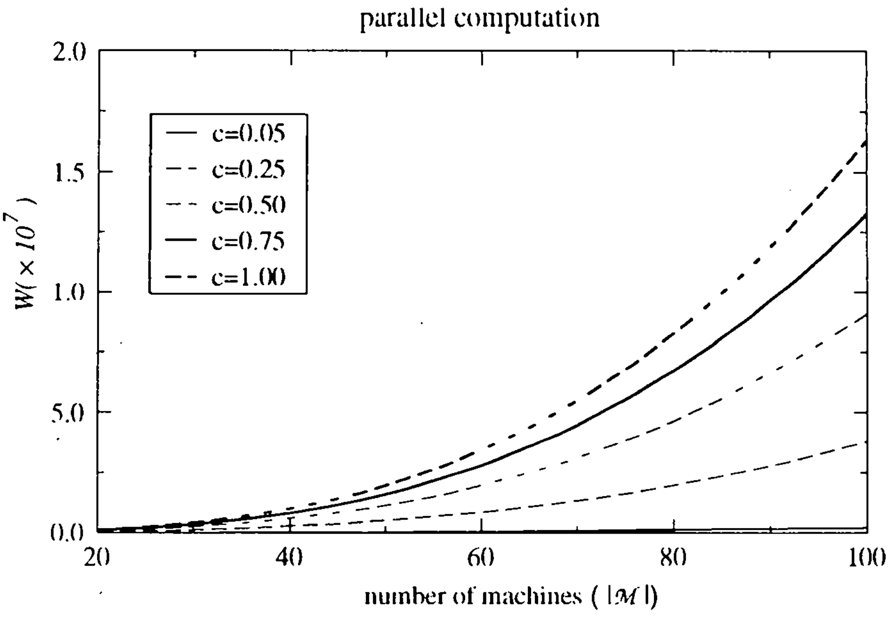


(i)

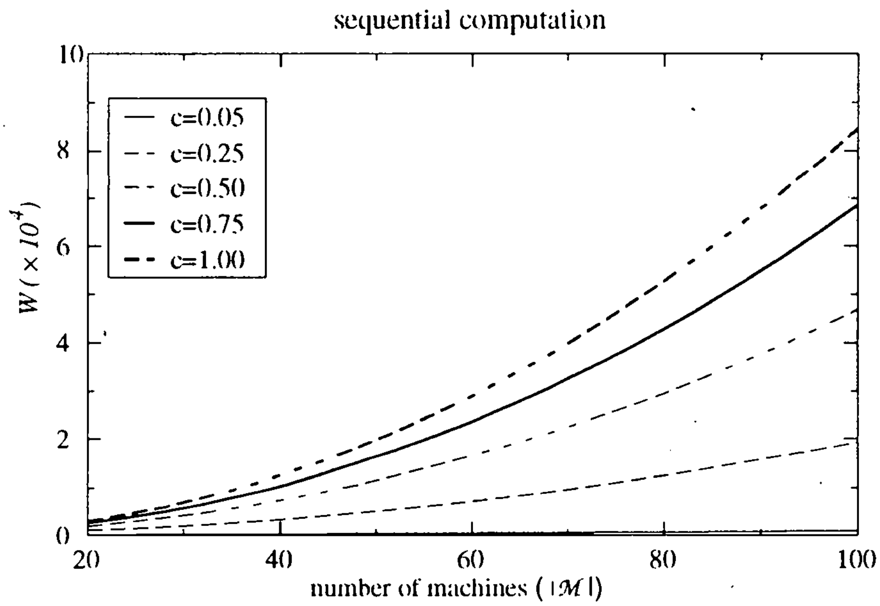


(ii)

Figure 6: Dependence of the ratio of computational times W on the parameters $|\mathcal{M}|$ and a. (i) Parallel computation. (ii) Sequential computation



(i)



(ii)

Figure 7: Dependence of the ratio of computational times W on the parameters $|\mathcal{M}|$ and c. (i) Parallel computation. (ii) Sequential computation

hierarchical approach is approximately 2×10^4 times faster than the monolithic approach.

2.2 Quality of Production Plans Obtained from the HPP System

The quality of the production plans obtained from the HPP system is evaluated by comparing the resulting backlogging and holding costs to the ones obtained from the monolithic approach. Two different sets of computational experiments were performed, one focusing on a small shop and the other on a shop of medium size. For lack of space, we present only the results related to the small shop. For a large shop, the monolithic approach cannot be used due to prohibitive computational times and memory requirements.

Note that the monolithic approach yields the optimal cost value for the production planning problem. Thus the comparison tests provide the quality of the hierarchical production plans with respect to the optimum. Let C_h and C_m be the objective function values obtained from the hierarchical and monolithic approaches, respectively. Let $\theta = \frac{C_h - C_m}{C_m} \cdot 100$ i.e. the % distance between the hierarchical and monolithic solutions.

Numerical calculations were performed for different levels of two important problem parameters: machine capacity, and % similarity of the costs and processing times of parts in a family. The first parameter indicates the average load on the machines in the facility. A highly loaded facility typically results in frequent backlogging and holding of parts. Six levels were used for the work load parameter: Capacity level of 1 implies that all machines are loaded on an average between 5% – 20% of their capacity. Similarly, capacity level of 2, 3, 4, 5, and 6 implies that all machines are loaded on an average between 20% – 35%, 35% – 50%, 50% – 65%, 65% – 80%, and 80% – 95%, respectively. Three levels were used for the similarity parameter: 100% similarity implies that all parts in a family have identical attributes (costs and

Table 2: Parameters selected for the small shop

Parameter	Value
number of elementary periods	15
number of sub-periods	5
number of part types	35 – 45
number of machines	12 – 14
number of macro-operations for each family	2
number of operations in each sub-routing	1 – 3
number of cells	3
number of families	4
cell size	4 – 5
family size	9 – 11

times). A 20% (60%) similarity implies that the part attributes are 80% (40%) away from the family attributes. A high value of similarity typically leads to an inaccurate estimation of the family processing times for the first iteration of algorithm .

Table 2 provides the values of the problem parameters selected for the small shop. For each of the 18 (3×6) parameter combinations, 5 problems were solved for a total of 90 problems. For all parameters, for which Table 2 indicates range, the actual values for each problem instance were selected from a uniform distribution over the corresponding range.

The production planning problems were solved by both the hierarchical and the monolithic approaches. The numerical results were obtained by solving the problems on a SUN/SPARC station. The algorithms were coded in C and the linear programs were solved using the XMP solver [2]. For an instance of the planning problem of the small shop, Table 3 provides the number of constraints and the number of decision variables for the linear problems in the hierarchical and monolithic approaches.

Figure 8 presents the value of θ (the percentage distance between the hierarchical and the monolithic solution) at each iteration of the iterative algorithm for a sample

Table 3: Number of constraints and variables for linear problems of the small shop

Problem	# of constraints	# of variables
$\mathcal{MP}$	4350	2976
$\mathcal{APP}$	95	60
$\mathcal{PFD}(f_r)$	190 - 230	135 - 165
$\mathcal{TSD}(\kappa, c_\eta)$	465 - 578	436 - 505
total for HPP	750 - 903	631- 730

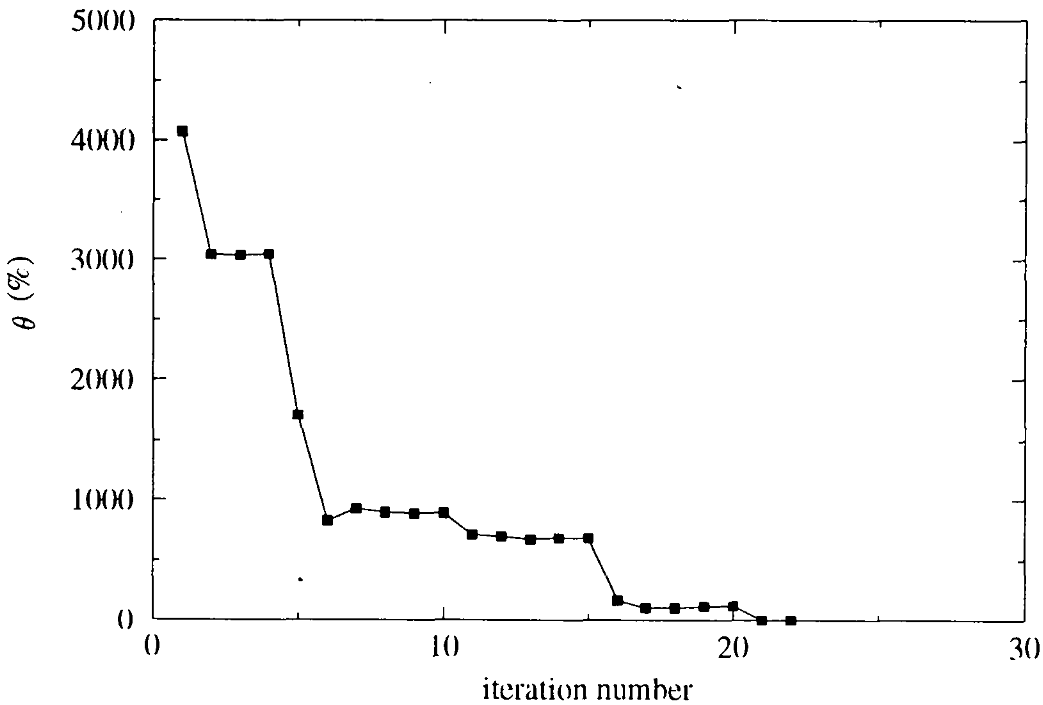


Figure 8: Small shop: Value of θ at each iteration of Algorithm

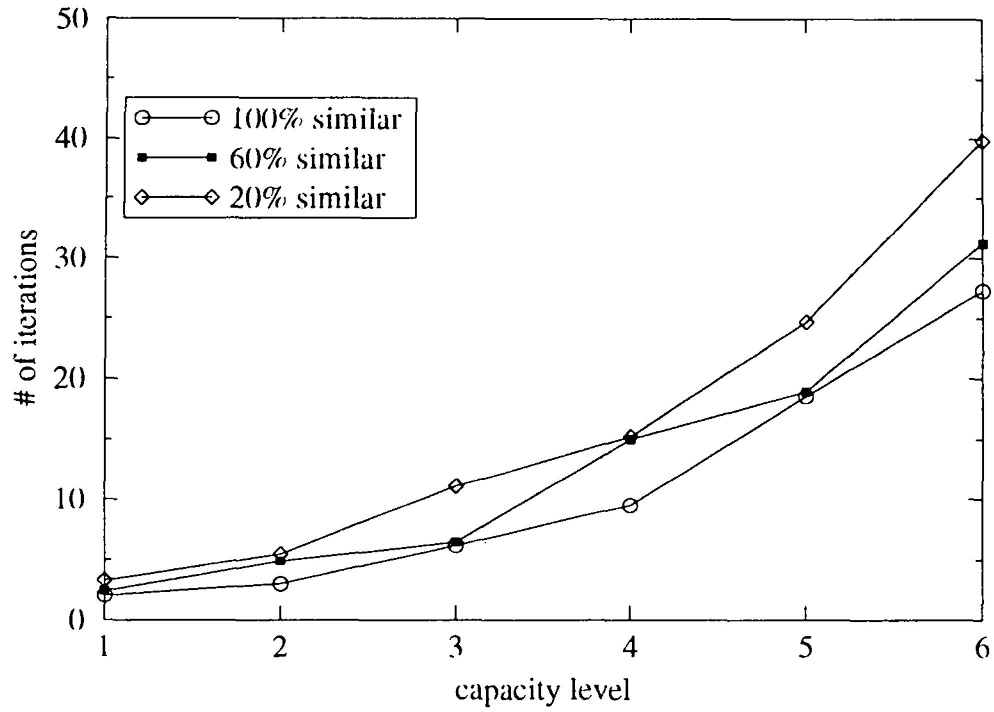


Figure 9: Small shop: Number of iterations required by the iterative algorithm for different capacity and part similarity levels

problem of the small shop. The figure shows that the hierarchical solution converges very fast to the near-optimal solution (in 22 iterations).

Figure 9 presents the number of iterations required for convergence of the iterative algorithm for different capacity levels and similarity of parts in a family. The number of iterations increases when the machines are more loaded. Also, more iterations are required when parts in a family are less similar. However, the number of iterations for convergence is not very sensitive to the similarity measure.

Figure 10 presents the average value of θ for 3 different values of the similarity of parts in the same family. The iterative algorithm always converges to within 1.5% of the optimum provided by the monolithic problem; thus the similarity of parts has little impact on the performance of the algorithm.

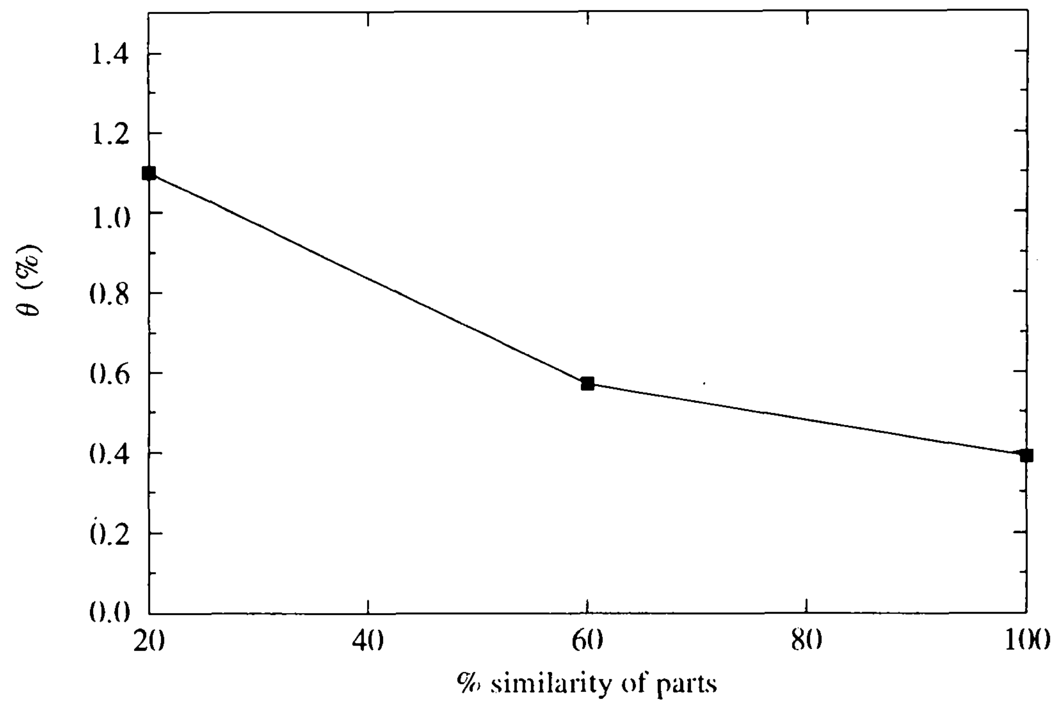


Figure 10: Small shop: Average value of θ for different parts similarity levels

3 Industrial Application

This section presents an application of the hierarchical production planning (HPP) system to an industrial case. Section 3.1 describes the production system under study and Section 3.2 presents the data collection and processing procedures. Section 3.3 provides the results of the iterative algorithm. Section 3.4 compares the HPP plan with the one obtained from the company's MRP II system. Section 3.5 summarizes the results of the case study.

3.1 Production System

Pangborn Corporation is a manufacturer of equipment for blast cleaning, descaling, deburring, shot peening and etching. Production planning is performed by an MRP II system, which issues the manufacturing orders for make parts and the purchase orders for raw materials.

This case study focuses on the machine shop of Pangborn Corporation which manufactures a total of 1100 different types of parts [457 make-to-stock (MTS) and 643 make-to-order (MTO)], which are fabricated by 31 different types of work-centers. Each work-center consists of one or more functionally identical machines, whereby each machine is defined as a unique piece of equipment. The total number of machines in the shop is 38. Table 4 describes the shop's work-centers.

3.2 Data Collection and Processing

The following data were collected for the production planning application:

1. The drawing of the machine shop layout.
2. The MTO and MTS part types and their description.
3. The total production of each MTS part type over a period of four months (01/01/94 - 04/01/94). Note that the typical lead time for the MTS parts is about 6 weeks.
4. The routing of each part type.
5. The number of machines that belong to each work-center.
6. The capacity of each work-center per shift, and number of shifts per day.
7. For each shop order of an MTS part type that was *completed* during this time period: (i) order quantity, (ii) order issue date, (iii) order due date, and (iv) order completion date.
8. The cost code X, A, B, or C of each MTS part; code X is assigned to the most expensive and code C to the least expensive parts.
9. The limit on the maximum number of machines in each manufacturing cell.

The above data were transformed to the inputs required by the iterative algorithm. The following paragraphs describe the data transformation procedure for each major input.

Table 4: List of work-centers in Pangborn's machine shop

Work-center	Description	Number of Functionally Identical Machines	Average Load
m_1	L & S engine lathe	1	0.2%
m_2	american engine lathe	1	2.0%
m_3	long bed engine lathe	1	0.2%
m_4	small engine lathes	4	0.4%
m_5	coya radial drill	1	1.9%
m_6	cinncinnati/american radial drills	2	9.1%
m_7	bolt threader	1	0.1%
m_8	pipe threader	1	0.1%
m_9	gray planer	1	0.2%
m_{10}	horizontal mills	2	1.8%
m_{11}	internal grinder	1	0.2%
m_{12}	universal grinder	1	5.3%
m_{13}	blanchard surface grinder	1	1.0%
m_{14}	thompson surface grinder	1	0.3%
m_{15}	vertical turret lathes	2	0.4%
m_{16}	horizontal boring mill	1	1.6%
m_{17}	makino n/c horizontal mill	2	44.7%
m_{18}	cinncinnati 10 v.c.	1	48.1%
m_{19}	n/c lathe - 12 universal	1	29.6%
m_{20}	n/c lathe - 12 chucker	1	93.4%
m_{21}	n/c lathe - 15 chucker	1	84.2%
m_{22}	bench or balance	1	47.0%
m_{23}	layout	1	7.5%
m_{24}	keyseat, broach & saw	1	7.9%
m_{25}	foundry	1	0.1%
m_{26}	band & abrasive saws	1	22.8%
m_{27}	receiving stores	1	–
m_{28}	raw material storage	1	–
m_{29}	outside operation	1	–

Elementary periods and sub-periods

Each week of the four-month time horizon was defined to be a planning sub-period; the total number of sub-periods is 13. Each working day in a week was defined as an elementary period; the number of elementary periods in a sub-period is 5 and the total number of elementary periods is 65.

Machine capacity and part processing times

The capacity of each work-center is 7.5 hrs per shift and the number of shifts per day is 2. Since the analysis considered only the MTS parts, the capacity of each work-center per day was reduced by the average daily workload imposed by the MTO parts. Thus, the capacity available per day for work-center m_i to manufacture MTS parts is:

$$\text{cap}(m_i) = 15 \cdot b_i - a_i \quad \text{(in hours)}, \tag{20}$$

where a_i is the average capacity required by the MTO parts per day on work-center m_i, and b_i is the number of functionally identical machines in work-center m_i.

Note that in formulating the planning problems of Section 3, we assumed that all machines have equal capacity. To respect this assumption, the processing times for the operations of each MTS part type were transformed such that the daily capacity available on each work-center for processing MTS parts equals 15 hrs, i.e.

$$t_{j,w} \leftarrow t_{j,w} \cdot \frac{15}{\text{cap}(m_i)}, \quad \text{where} \quad \alpha_{j,w} = m_i \tag{21}$$

where $t_{j,w}$ is the processing time for operation $o_{j,w}$ of part type p_j on work-center $\alpha_{j,w} = m_i$. Table 4 presents for each work-center the average daily available capacity consumed by MTS parts. In the calculations that yielded the values of Table 4, the capacity of each work-center was reduced using equation (20) and the manufacturing times were transformed using equation (21).

Part demand

The analysis considered work-centers m_i for $i = 1, 2, ..., 26$. The operations on work-centers m_{27} (receiving stores) and m_{28} (raw-material storage) do not require any manufacturing time. For work-centers m_{29} (outside operation) and m_{30} (fabrication shop), which are located outside the machine shop, the average manufacturing and queue time per operation was defined to be 5 days (after consultation with the Pangborn production staff). Although the production on work-centers m_{29} and m_{30} was not planned, the time spent by each part on these work-centers was accounted for by reducing the due date of the corresponding shop orders by 5 days. A similar procedure was followed for the operations performed on work-center m_{31} (inspection), for which no manufacturing time was reported in the part's routings. The average queue and manufacturing time at work-center m_{31} was reported to be equal to 1 day, which was subtracted from the due date of each work order that required inspection.

The order due dates were further modified as follows: Since we considered the time horizon consisting of elementary periods 1 through 65, an order due date greater than 65 was set to 65 (last elementary period of the planning horizon). Similarly, the due date of an order that was due prior to elementary period 1 was set to 1 (first elementary period of the planning horizon).

Part initial inventory

For those parts that were completed within the planning horizon but their manufacture had been initiated prior to elementary period 1, we determined the operations completed prior to elementary period 1 and only planned the remaining operations to completion. For the first of the remaining operations, say $o_{j,w}$, we determined the initial buffer inventory $s^0_{j,w}$ as described below.

First, the lead time of each part was determined; lead time = due date - issue date. For part type p_j that is completed in elementary period k such that k is less than the part lead time, the last operation w^* performed prior to elementary period 1 is determined from:

$$w^* = \left\lfloor \left(1 - \frac{\text{lead time - } k}{\text{lead time}}\right) * n_j \right\rfloor \qquad (22)$$

where $\lfloor \bullet \rfloor$ is the largest integer less than $\bullet$ and n_j is the number of operations required to manufacture part type p_j. The inventory s^0_{j,w^*} of part p_j for operation w^* was set equal to the order quantity.

Part holding and backlogging costs

Table 5: Holding and backlogging costs for MTS parts

part cost code	backlogging and holding costs
X	$I_{j,1}$ = U(0.75,0.90)
	B_j = U(0.75,0.90) · 50.0
A	$I_{j,1}$ = U(0.50,0.75)
	B_j = U(0.50,0.75) · 50.0
B	$I_{j,1}$ = U(0.25,0.50)
	B_j = U(0.25,0.50) · 50.0
C	$I_{j,1}$ = U(0.10,0.25)
	B_j = U(0.10,0.25) · 50.0
$I_{j,w} = I_{j,w-1} \cdot$ U(1.01,1.02) $\quad \forall j, w > 1$	

The MTS parts were classified by Pangborn personnel into four categories, X, A, B, and C; category X denotes the most expensive parts and category C denotes the least expensive parts. For part p_j of cost category X, the inventory holding cost $I_{j,1}$ for the first operation was selected to be $U(0.75,\ 0.90)$, where $U(a, b)$ is a random number between a and b selected from the corresponding uniform probability distribution. Note that these values are cost units and do not represent dollars. For all other operations of part type p_j of cost category X, the inventory holding cost $I_{j,w}$ was selected as $I_{j,w-1} \cdot$ U(1.01,1.02). Note that $I_{j,w}$ is 1 – 2 % higher that $I_{j,w-1}$ to account for the value added at the end of each operation. On the other hand, since high backlogging costs (in comparison to the inventory costs) is a normal industrial practice, the backlogging cost B_j for part types of cost category X was selected to be U(0.75, 0.90) · 50. Table 5 provides the holding and backlogging costs for all cost categories.

Table 6: Work-centers in each cell

cell	cell size	work-centers
c_1	6	$m_2, m_3, m_{10}, m_{12}, m_{19}, m_{25}$
c_2	6	$m_4, m_{13}, m_{17}, m_{20}, m_{21}.m_{26}$
c_3	6	$m_5, m_6, m_{18}, m_{22}, m_{23}, m_{24}$
c_4	2	m_{11}, m_{14}
c_5	4	m_7, m_8, m_{15}, m_{16}
c_6	2	m_1, m_9

Table 7: From-to matrix showing the flow of parts between the cells of Table 6.4

to	from cell					
cell	c_1	c_2	c_3	c_4	c_5	c_6
c_1	–	51	39	–	–	–
c_2	100	–	36	–	–	1
c_3	48	216	–	–	6	–
c_4	2	–	2	–		–
c_5	2	1	–	–	–	–
c_6	–	3	3	–	–	–

3.3 Results from the Iterative Algorithm

The first step in planning with the iterative algorithm was to determine the part families and manufacturing cells. The algorithms used are out of the scope of this paper. We obtained 6 cells and 28 part families. Table 6 provides the work-centers assigned to each cell. Table 7 presents the from-to matrix for the flow of part types between these cells. The entry $a_{i,j}$ of this matrix represents the number of different part types that flow from cell c_i to cell c_j for their manufacture. Note that the most significant flow of parts is between cells c_1, c_2, and c_3. Table 8 presents the number of part types in each part family and the cell sequences for these part families.

Table 8: Cell sequences for part families

part family	family size	cell sequences
f_1	2	c_3, c_2, c_3
f_2	1	c_2, c_6, c_2
f_3	1	c_1, c_5, c_3, c_1
f_4	1	c_2, c_1, c_2, c_3
f_5	2	c_2, c_3, c_4
f_6	1	c_1, c_3, c_1
f_7	1	c_4
f_8	3	c_2, c_1, c_3, c_1
f_9	33	c_2, c_1, c_3
f_{10}	8	c_2
f_{11}	22	c_2
f_{12}	2	c_2, c_3, c_1, c_4
f_{13}	1	c_2, c_5, c_3
f_{14}	3	c_6, c_3
f_{15}	14	c_1, c_3
f_{16}	1	c_2, c_1
f_{17}	13	c_2, c_1
f_{18}	2	c_1
f_{19}	5	c_1
f_{20}	5	c_2, c_3, c_1
f_{21}	4	c_5, c_3
f_{22}	58	c_3
f_{23}	7	c_2, c_3, c_2
f_{24}	5	c_1, c_2, c_3
f_{25}	94	c_1, c_2
f_{26}	27	c_2, c_3, c_2, c_3
f_{27}	127	c_2, c_3
f_{28}	11	c_2, c_3, c_1 c_3

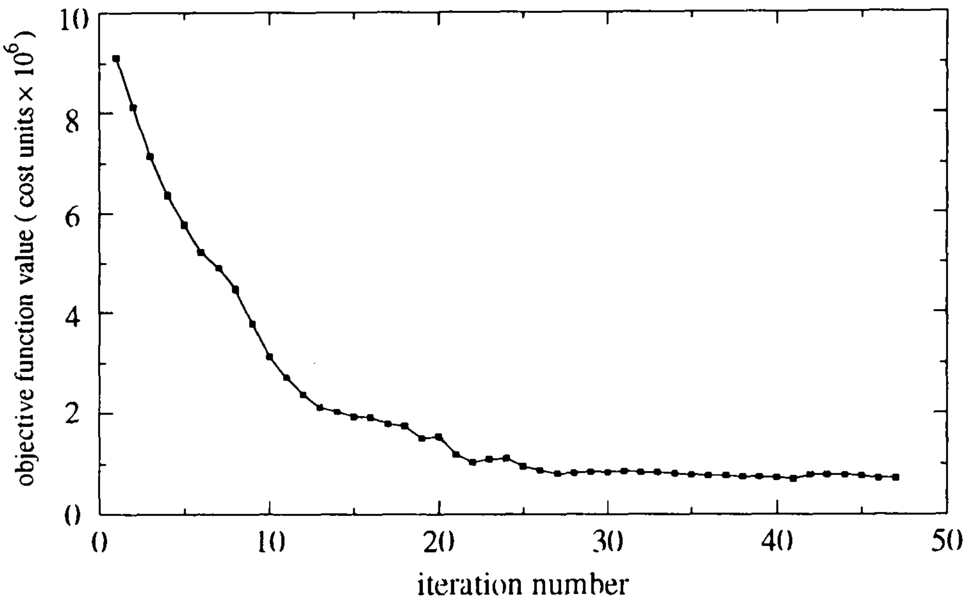


Figure 11: Objective function value at each iteration of

Based on the cell and part family configuration obtained from the aggregation algorithm, the production plan was computed using the iterative algorithm described in part 1. The algorithm converged in 47 iterations; the same aggregate solution was obtained in iteration 46 and 47. The total CPU time was 282 minutes. Figure 11 shows the evaluation of objective function (total holding and backlogging costs) vs. the iterations of the algorithm .

3.4 Comparison of the Production Plan Obtained from the iterative algorithm and the MRP II Systems

Figure 12 presents the percentage of orders produced early, on-time, and late using the plans of the iterative algorithm and MRP II systems (on-time delivery of an order implies that the order completion date is equal to the order due date). The figure clearly shows that the iterative algorihm provides a plan, which is significantly better than the MRP II plan. For example, on-time delivery was found to be 43% as

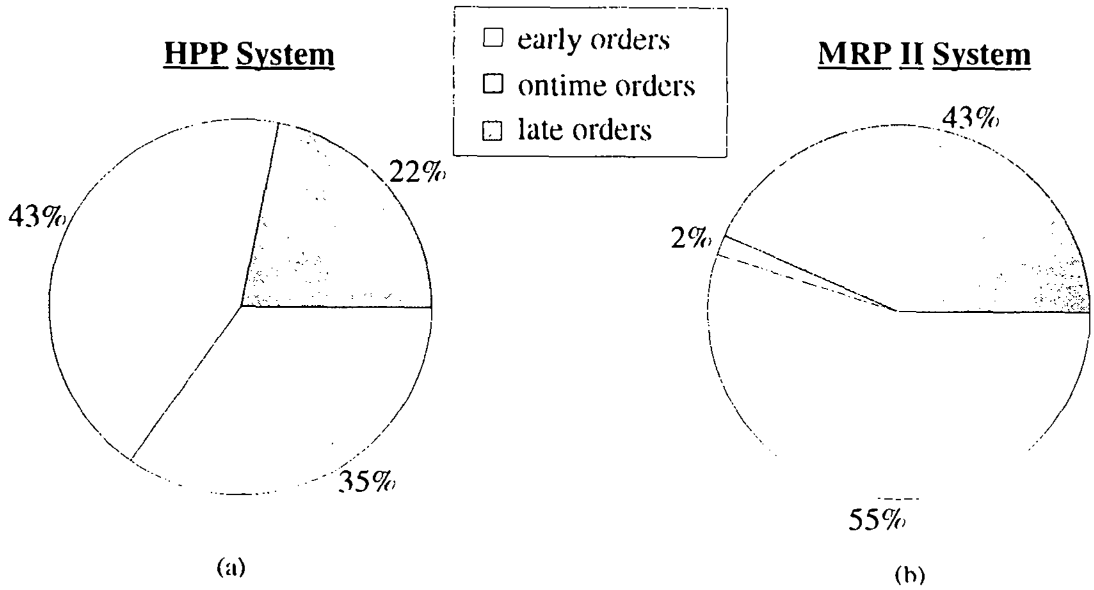


Figure 12: The percentage of orders produced early, on-time, and late. (a) HPP system. (b) MRP II system

compared to 2% for the MRP II plan. Furthermore, the iterative algorithm reduces late orders from 43% to 22% and early orders from 55% to 35%. Figure 13 presents the orders produced early, on-time, and late for parts of each cost category.

4 Conclusion

We developed an effective method that aggregates partsinto families and machines into cells to form the entities of the medium-term planning. We then proposed an iterative algorithm to reach a short-term planning which appeared to be close to the optimal solution provided by the monolithic problem. This short-term planning is obtained by disaggregating the medium-term planning.

We quantified in a rigorous manner the benefits of hierarchical production planning in terms of the savings in both computational complexity and memory requirements for non-parallel and parallel implementations. This is the first time that such an analysis has been performed for hierarchical production planning systems. The hierarchical production planning system can be readily used in an industrial setting. Although only minimization of the backlogging and holdings costs is presently con-

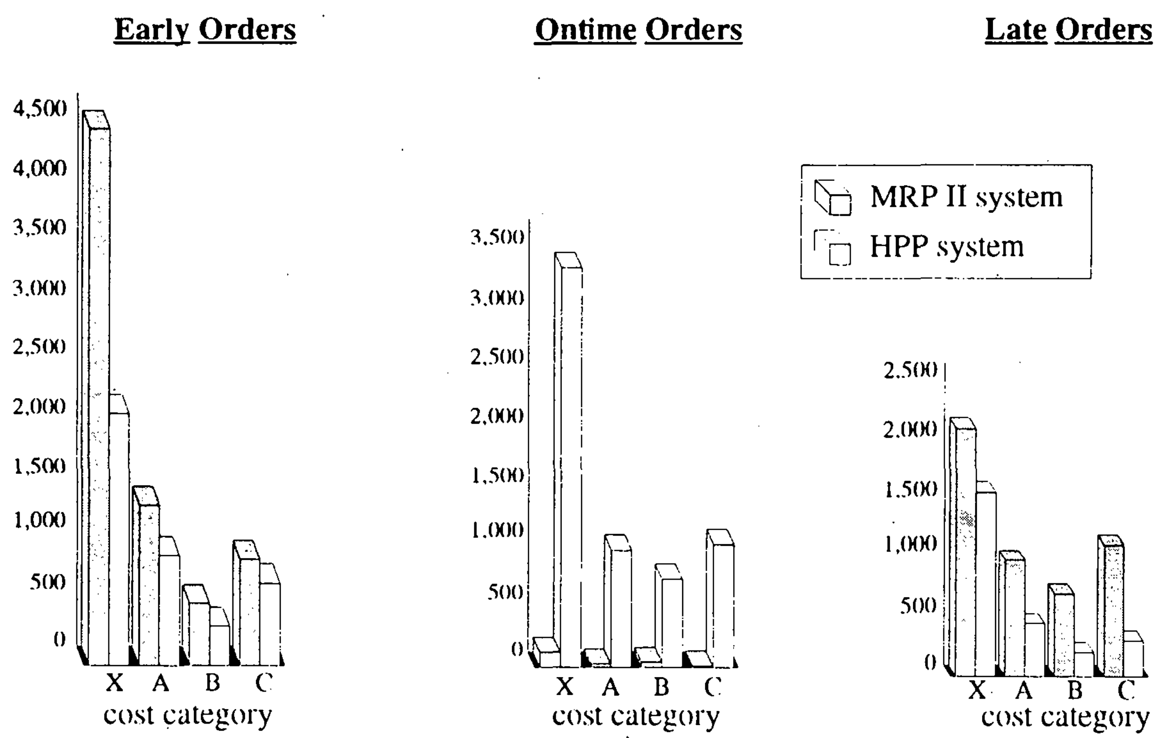


Figure 13: The number of orders produced early, on-time, and late for parts of each cost category

sidered, other costs such as setup, hiring and firing costs can be easily incorporated in the proposed system. The industrial application indicated that the proposed approach provides plans that are much superior to those of a typical MRP II, resulting in savings of 54.1% in total backlogging and holding costs. Also, the proposed approached led to an increase in on-time delivery of orders by 41%, a reduction in the number of late orders by 49%, and a reduction in the number of early orders by 37%.

References

[1] M.S. Bazaraa, J.J. Jarvis, and H. D. Sherali, Linear programming and networks flows. New-York: John Wiley and Sons, 1990.

[2] R. E. Marsten, "The design of the XMP library," ACM Transactions of the Mathematical Software, vol. 27, pp. 481 – 497, 1981.

Unité de recherche INRIA Lorraine
Technopôle de Nancy-Brabois - Campus scientifique
615, rue du Jardin Botanique - B.P. 101 - 54602 Villers lès Nancy Cedex (France)

Unité de recherche INRIA Rennes - IRISA, Campus universitaire de Beaulieu 35042 Rennes Cedex (France)
Unité de recherche INRIA Rhône-Alpes - 46, avenue Félix Viallet - 38031 Grenoble Cedex 1 (France)
Unité de recherche INRIA Rocquencourt - Domaine de Voluceau - Rocquencourt - B.P. 105 - 78153 Le Chesnay Cedex (France)
Unité de recherche INRIA Sophia Antipolis - 2004, route des Lucioles - B.P. 93 - 06902 Sophia Antipolis Cedex (France)

Éditeur
INRIA - Domaine de Voluceau - Rocquencourt - B.P. 105 - 78153 Le Chesnay Cedex (France)

ISSN 0249 - 6399

RR. 2634