Hierarchical Production Planning for General Jobs Shops : Part 1 : Modeling

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 1 : Modeling

Anshu Mehra - Ioannis Minis
Jean-Marie Proth

N° 2633
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)

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).

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

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

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).

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

Planification hiérarchisée de la production en ateliers

Première partie : Modélisation

Anshu Mehra* Ioannis Minis* Jean-Marie Proth†

Résumé

Cette étude comporte deux parties. Dans la première partie, nous donnons une bibliographie étendue sur le sujet et nous produisons un modèle hiérarchisé à deux niveaux pour la planification de la production. Le modèle en question est composé d'un modèle de niveau bas, d'un modèle de niveau haut, et des liens entre ces deux niveaux. Ces modèles sont supposés connus. Nous fournissons une formulation mathématique du problème de planification de la production et un algorithme itératif pour le résoudre.
Dans la seconde partie de l'étude, nous fournirons plusieurs exemples numériques, parmi lesquels une application industrielle. Nous fournirons également une comparaison entre l'approche monolithique et l'approche hiérarchique en termes d'efficacité, d'occupation mémoire et de complexité.

Mots clefs: Planification hiérarchisée, Complexité, Optimisation, Calcul parallèle en gestion de la production

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 1: Modeling

A. Mehra* I. Minis* J.M. Proth†

Abstract

This study is divided into two parts. In part 1, we provide an extensive bibliographical study on the subject and introduce a two-level hierarchical production planning model. The whole model is composed of the low level model, the high level model, and the links between both. These models are supposed to be known. We provide a mathematical formulation of the production planning problem and an iterative algorithm to solve it.
In the second part of the study, we will provide several numerical examples, including an industrial application. We will also provide a comparison of the hierarchical approach and the monolithic approach in terms of efficiency, memory requirement and computational complexity.

Keywords: Hierarchical planning, Complexity, Optimization, Parallel computation in production management

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

Hierarchical production planning (HPP) decomposes the large planning optimization problem, known as the monolithic problem, into subproblems. The higher levels of the hierarchy represent the planning problem in an aggregate, more global manner. In contrast, the lower levels of the hierarchy provide a more detailed description. The related optimization models are solved in sequence starting from the top level; the solution of the model at a given level provides values for the parameters to the model of the subsequent level. These parameters reflect the constraints which are transferred from a level to the next lower level. It is the mechanism used to transfer the information captured at a given level to the next lower level. The solution of the lowest level, i.e. the detailed production plan, is implemented on the shop floor. A bottom-up feedback mechanism is typically used in order to adjust and improve the values of the attributes of aggregate entities, including processing times. Information on the current system state is also provided from the shop floor to the higher decision levels, and is used to update the model parameters, such as buffer inventories.

Typically, a hierarchical approach (i) reduces complexity and, thus, renders the planning problem solvable within reasonable time, (ii) copes better with disturbances since only part of the problems have to be solved again in case of random events, (iii) fits with the management hierarchy, since it is possible to design the hierarchical production management system in such a way that each of its level corresponds to a level of the management hierarchy, (iv) reduces the need detailed information in long- and medium-term planning, and (v) allows the use of different criteria at each managerial level.

Hax and Meal [1] were the first to formalize the hierarchical production planning concept based on the ideas of Holt *et al.* [2] and Winters [3]. They proposed a set a coordinated heuristics for hierarchical planning of a multi-plant firm consisting of four decision levels. Each level relates to decisions over different horizons. The longer range decisions provide the constraints for shorter range ones, and, thus, the decisions made at higher levels have a more significant impact on the operation of the system.

Bitran and Hax [4] proposed a computational improvement to the Hax and Meal model by reformulating the family and item disaggregation problems as knapsack problems, for which they provided efficient solution algorithms [5]. Numerical results indicated that the solutions are near the optimal for the case of low setup costs.

Golovin [6] detected the occurrence of infeasibilities in the Hax and Meal model and provided a numerical example to illustrate it. Gabbay [7] introduced a set of constraints in the family disaggregation model to guarantee feasible disaggregation. However, his work is restricted to the static case, in which the disaggregation problems are solved only for the first period of the planning horizon, and the rolling horizon procedure is not used for dynamic replanning.

Bitran, Haas and Hax improved the knapsack-based method in [8]. At the family disaggregation level they: (i) Used a "Look-Ahead Feasibility Rule" that considers the demand of the next period to prevent the family disaggregation from being infeasible. (ii) Selected the production quantity for each family close to its Economic Order Quantity (EOQ). A set of simulations indicated that the enhanced work produces superior results in comparison to the previous model presented in [4].

Erschler, Fontan, and Merce [9] derived necessary and sufficient conditions for feasibility and consistency. They proposed that the "Look-Ahead" procedure be extended to all periods of the long-term planning horizon. However, this requires the detailed demand to be known over the entire horizon, thus, limiting some of the benefits of the hierarchical approach.

Bitran, Haas and Mutsuo [10] extended the family disaggregation problem for the case in which setup costs are very high, and changeover costs for items in the same family are significant. A mixed integer linear programming problem was formulated and the disaggregation procedure was applied to data from a consumer electronics company. Their model was shown to perform better for high setup costs in comparison to that of Bitran *et al.* [8].

Mohanty and Kulkarni [11] proposed a different heuristic method for family disaggregation, which performs better when there is a negative error in forecasted demand.

Graves [12] presented a hybrid approach for the Hax and Meal model in which the aggregate and family disaggregation problems are included in a global mixed-integer linear program (MILP).Aardal and Larsson [13] extended Graves method by using the Benders decomposition for the same hybrid mixed-integer linear program. They found their method to be computationally faster than Graves algorithm.

Bitran, Haas and Hax [14] proposed an extension of their previous work to a two-stage fabrication/assembly system. They used data supplied by a pencil manufacturer to compare the production plans obtained by their technique to those of an MRP-based system. Their approach was found to outperform MRP in more than 90% of the test cases.

Serious restrictions remain in the Hax and Meal model, mainly: (i) a single resource is considered, (ii) no randomness is taken into account, and (iii) the length of time period is constant and the same at each level.

Nagi [15] proposed a two-level hierarchical model that aggregates parts to families and machines to manufacturing cells. The criterion at both levels consists of minimizing of backlogging and holding costs. Setup costs are ignored, but machine capacities are considered including the loss of capacity due to random breakdown of machines. The main contributions of Nagi's model include: (i) Multiple resources are considered. (ii) Randomness in machine failure is taken into account. (iii) Consistency, feasibility and optimality of the model have been shown for the perfect case. The limitations of the model include: (i) It does not consider multistage production. (ii) It does not aggregate time periods. (iii) The feasibility conditions for non-perfect cases are based on conservative estimates of the aggregate processing times. (iv) The detailed problem for a typical industrial case can be of a very large dimension. Harhalakis, Mehra, Nagi and Proth [16] extended the Nagi model to include aggregation of time periods. However, they only considered the case of a single resource for which parts have identical attributes (manufacturing times and costs).

Several other models have been developed by Axsater [17], [18], [19], Tsubone et al. [20], Saad [21], Thompson et al. [22], [23], and Inman and Jones [24]. All these models are developed for specific type of problems.

In this paper, we consider medium- and short-term planning, which are part of the hierarchical DMP (Decision Making Process) and which belong to the tactical level. In the medium-term planning, the time horizon is expressed in months or weeks. The information is moderately aggregated, and its source is external as well as internal. Typically, medium-term planning determines the number of units of parts in part families, to be produced in groups of machines (e.g. manufacturing cells) during each period (e.g. a week) of the horizon (e.g. 3 months). Short-term planning disaggregates the plan of the first medium-term time period (a week) over the short-term time periods (days). It determines the number of units of parts to be produced on each machine within each short-term planning period. The tactical decisions of short-term planning form constraints for the operational level of the hierarchy.

The design of the decision hierarchy is supposed to be known. In other words, we assume that part families have been aggregated into families, resources into manufacturing cells, and time periods into aggregate time periods, in a way which: (i) minimizes the number of cells visited by families while keeping the size of the cell bounded, and (ii) guarantees that part belonging to the same part family spend a similar amount of time in each of the cells they visit. The notation related to the decision hierarchy are presented in Section 2. Based on the given decision hierarchy, the decision making problems of medium- and short-term planning are formulated in Section 3. The global problem is decomposed in such a way that the sub- problems related to short-term planning are independent of each other and, hence, can be solved in parallel. Section 4 presents a solution algorithm that adjusts the aggregate processing times in order to provide a feasible and near optimal solution. Section 5 is the conclusion.

2 Notations

We propose a two-level hierarchical framework for tactical production planning. The entities of interest at aggregate level are part families, machine cells, and aggregate time periods. The entities of interest at detailed level are part types, machines and elementary periods. The planning time horizon consists of Z consecutive *sub-periods* (aggregate time periods) 1,2,...,Z; let $H_A = \{1, 2, ..., Z\}$. Each sub-period is divided into z *elementary periods* (detailed time periods) of duration T each. Thus the time horizon consists of $Z \cdot z$ elementary periods numbered $1, 2, ..., Z \cdot z$; let $H = \{1, 2, ..., Z \cdot z\}$. For sub-period $\kappa \in H_A$, $h(\kappa)$ denotes the set of the corresponding elementary periods, i.e. $h(\kappa) = \{(\kappa - 1) \cdot z + 1, ..., \kappa \cdot z\}$. Note that $H = \bigcup_{\kappa \in H_A} h(\kappa)$.

2.1 The Low (Detailed Level) of the Hierarchy

We consider a single manufacturing facility consisting of $|\mathcal{M}|$ resources, $\mathcal{M} = \left\{m_1, m_2, ..., m_{|\mathcal{M}|}\right\}$, that manufacture a set of $|\mathcal{P}|$ part types, $\mathcal{P} = \left\{p_1, p_2, ..., \ p_{|\mathcal{P}|}\right\}$. Each part type follows a unique sequence of manufacturing operations. The operation sequence (or production routing) of part type p_j is denoted by $< o_{j,w} >_{w=1}^{n_j}$, and must be performed in a certain order; that is, operation $o_{j,w}$ can only start when operation $o_{j,w-1}$ has ended. Each operation is performed on a machine from the set $\mathcal{M}$ and has associated with it a deterministic processing time, which includes both the operation set-up and run times. Let $\alpha_{j,w} \in \mathcal{M}$ be the machine which performs operation $o_{j,w}$ required to manufacture part type p_j, and let $t_{j,w}$ be the time required to complete this operation. During an elementary period, a machine processes parts for a maximum time interval T. The duration of T is assumed to be much larger than the maximum processing time of an operation.

Let d_j^k be the demand of part type p_j to be fulfilled by the end of the kth elementary period. Also, let $u_{j,w}^k$ denote the quantity of part type p_j to be processed at operation $o_{j,w}$ during the kth elementary period. Let $s_{j,w}^k$ be the inventory level of part type p_j at the end of operation $o_{j,w}$ and at the end of kth elementary period. Negative values of inventory indicate backlog. For part type p_j, backlogging is allowed only at the end of the last operation (i.e. for finished parts), that is, $s_{j,w}^k \geq 0$ for $w = 1, 2, ..., n_j - 1$. For part type p_j, the variable $s_{j,w}^0$ indicates the initial inventory level at the output buffer of operation $o_{j,w}$ and $s_{j,0}^k$ indicates the raw-material inventory level at the end of the kth elementary period.

The objective function measures the total inventory and backlogging costs for both the finished goods and work-in-process. The costs are estimated at the end of a period, and are piece-wise constant functions of the inventory state. Let $I_{j,w}$ represent the cost associated with holding one unit of part type p_j in stock for one elementary period, after operation $o_{j,w}$ is completed. Similarly, let B_j represent the cost associated with one unit of part type p_j being delayed by one elementary period. Then, the objective function at the detailed level of the hierarchy for the kth elementary period is defined by:

$$\sum_{p_j \in \mathcal{P}} \left[\sum_{w=1}^{n_j - 1} \left[I_{j,w} \cdot s_{j,w}^k \right] + I_{j,n_j} \cdot \left[s_{j,n_j}^k \right]^+ + B_j \cdot \left[-s_{j,n_j}^k \right]^+ \right]$$

where, $[\bullet]^+$ equals $\max(0,\bullet)$.

2.2 The High (Aggregate Level) of the Hierarchy

The aggregation of machines to cells and part types to part families is discussed in Section 3. This section presents only the notation that is necessary for the representation of the aggregate level of the hierarchy.

Let $\mathcal{C} = \left\{c_1, c_2, ..., c_{|\mathcal{C}|}\right\}$ be a partition of the set of machines $\mathcal{M}$, that is

$$c_a \bigcap c_b = \emptyset \text{ for } c_a, c_b \in \mathcal{C} \text{ and } a \neq b, \text{ and } \mathcal{M} = \bigcup_{v=1}^{|\mathcal{C}|} c_v.$$

Let $\mathcal{F} = \left\{f_1, f_2, ..., f_{|\mathcal{F}|}\right\}$ be a partition of the set of part types $\mathcal{P}$, that is

$$f_a \bigcap f_b = \emptyset \text{ for } f_a, f_b \in \mathcal{F} \text{ and } a \neq b, \text{ and } \mathcal{P} = \bigcup_{r=1}^{|\mathcal{F}|} f_r.$$

Let W be the maximum cell size.

In the next section, the aggregation of part types to part families and machines to cells is done in such a way that parts types belonging to the same family follow a common sequence of cells during their manufacture. It is not necessary for the part types to follow the same machine sequence. Each part family f_r is manufactured following a unique macro-routing (or sequence of macro-operations), $< O_r^q >_{q=1}^{\bar{n}_r}$, such that macro-operation $O_{r,q}$ can only start when macro-operation $O_{r,q-1}$ has ended. Each macro-operation is performed within a cell from the set $\mathcal{C}$. Let $\pi_{j,q} \in \mathcal{C}$ be the manufacturing cell which can perform the qth macro-operation required to manufacture part type p_j, and let $\bar{t}_{j,q}$ be the time required to complete this macro- operation. Let $\mathcal{MO}_v$ be the set of all macro-operations $O_{r,q}$ which are performed in cell c_v. A cell can process part families for a maximum duration of $z \cdot T$ during a sub-period. $z \cdot T$ is assumed to be much larger than the maximum processing time of a macro-operation.

The routing of each part may be partitioned into a set of sub- routings, each sub-routing corresponding to a macro-operation of the part's parent family. Consider the routing of part $p_j \in f_r$, and define a set of $\bar{n}_r$ sub-routings $< o_{j,w} >_{w=y_j^q}^{v_j^q}$, $q = 1, 2, ..., \bar{n}_r$ (where y_j^q and v_j^q are the first and last operations in the sub-routing) such that $< o_{j,w} >_{w=1}^{n_j} = \circ_{q=1}^{\bar{n}_r} < o_{j,w} >_{w=y_j^q}^{v_j^q}$, where $\circ$ represents the concatenation of operations. Each sub-routing corresponds to one of the cells visited by the part type. Let $x_{j,w} = 0$ if operation $o_{j,w+1}$ is performed in the same cell as operation $o_{j,w}$ (i.e. if $o_{j,w}$ is not the last operation in a sub-routing); otherwise $x_{j,w} = 1$. Similarly, $h_{j,w} = 0$ if operation $o_{j,w-1}$ is performed in the same cell as operation $o_{j,w}$ (i.e. if $o_{j,w}$ is not the first operation in a sub-routing); otherwise $h_{j,w} = 1$.

$\tau_{r,q}^{\kappa}$ is the processing time required by one unit of a part in family f_r to complete macro-operation $O_{r,q}$ during the κth sub-period. This processing time depends on the bottleneck machine within the cell where the macro-operation is performed. Since the bottleneck machine depends on the part mix, which may vary over sub-periods, each processing time has a sub-period index, κ, associated with it. $\bar{u}_{r,q}^{\kappa}$ is the production volume of parts in family f_r processed at macro-operation $O_{r,q}$ during the κth sub-period. D_r^{κ} is the independent demand of parts in family f_r at the beginning of the

κth sub-period, $D_r^\kappa = \sum_{k \in h(\kappa)} \sum_{p_j \in f_r} d_j^k$. $\bar{s}_{r,q}^\kappa$ is the inventory of parts in family f_r at the end of macro-operation $O_{r,q}$ and at the end of the κth sub-period.

Let $\bar{I}_{r,q}$ be the cost associated with holding in stock one unit of a part in family f_r for one sub-period, after macro-operation $O_{r,q}$ is completed; also let $\bar{B}_r$ be the cost associated with one unit of a part in family f_r being delayed by one sub-period. To compute these aggregate costs which depends on the part mix, we use the weighted average of the costs with respect to the part type demands:

$$\bar{I}_{r,q} = z \cdot \sum_{k \in H} \sum_{p_j \in f_r} \left[d_j^k \cdot I_{j,v_j^q} \right] \cdot \left[\sum_{k \in H} \sum_{p_j \in f_r} d_j^k \right]^{-1} \tag{1}$$

$$\bar{B}_r = z \cdot \sum_{k \in H} \sum_{p_j \in f_r} \left[d_j^k \cdot B_j \right] \cdot \left[\sum_{k \in H} \sum_{p_j \in f_r} d_j^k \right]^{-1} \tag{2}$$

Note that we consider these costs to be same at each sub-period, even though these costs depend on the part mix which may vary over sub-periods. The computational tests shows that this approximation does not affect the results significantly.

Finally, we define $1\{\bullet\} = 1$ if argument $\bullet$ is true; $1\{\bullet\} = 0$ otherwise. The objective function at the aggregate level for the κth sub-period is defined by:

$$\sum_{f_r \in \mathcal{F}} \left[\sum_{q=1}^{\bar{n}_r - 1} \left[\bar{I}_{r,q} \cdot \bar{s}_{r,q}^\kappa \right] + \bar{I}_{r,\bar{n}_r} [\bar{s}_{r,\bar{n}_r}^\kappa]^+ + \bar{B}_r [-\bar{s}_{r,\bar{n}_r}^\kappa]^+ \right]$$

3 Mathematical Formulation of the Production Planning Problems

In this section, we formulate the global (monolithic) production planning problem and the problems related to the aggregate and detailed level of the planning hierarchy. These formulations are based on the aggregation of parts, machines, and time periods which are assumed to be given.

3.1 Monolithic Planning Problem Formulation

The monolithic production planning problem consists of determining the number of units processed at each operation in a part type's routing during each elementary period of the planning horizon. We assume that: (i) at most one operation in the part type's routing is performed during one elementary period, and (ii) a part visits at most one cell during each sub-period.

Inventory state equations

For each part type $p_j \in \mathcal{P}$, the inventory state equation for operation $o_{j,w}, w = 1, 2, ..., n_j - 1$ and for elementary period $k \in H$ is:

$$s_{j,w}^{k} = s_{j,w}^{k-1} + u_{j,w}^{k} - u_{j,w+1}^{k},$$

and can be re-written as:

$$s_{j,w}^{k} = s_{j,w}^{0} + \sum_{a=1}^{k} \left[u_{j,w}^{a} - u_{j,w+1}^{a} \right]. \tag{3}$$

Equation (3) states that the inventory $s_{j,w}^{k}$ of part type p_j at the end of operation $o_{j,w}$ and the end of the kth elementary period is equal to the inventory $s_{j,w}^{k-1}$ at the end of the previous elementary period, plus the quantity of parts processed at operation $o_{j,w}$ during the kth elementary period, minus the quantity of parts processed at the next operation $o_{j,w+1}$ during the kth elementary period. Note that the inventory level at the end of an operation is updated only at the end of the elementary period. For the final operation o_{j,n_j} of part type $p_j \in \mathcal{P}$, and for elementary period $k \in H$, the inventory state equation is:

$$s_{j,w}^{k} = s_{j,w}^{k-1} + u_{j,w}^{k} - d_{j}^{k},$$

and can be re-written as:

$$s_{j,n_j}^{k} = s_{j,n_j}^{0} + \sum_{a=1}^{k} \left[u_{j,n_j}^{a} - d_{j}^{a} \right]. \tag{4}$$

The monolithic production planning problem can be described by the following optimization problem:

Problem $\mathcal{MP}$

$$\text{minimize} \quad \sum_{k \in H} \sum_{p_j \in \mathcal{P}} \left[\sum_{w=1}^{n_j - 1} \left[I_{j,w} \cdot s_{j,w}^{k} \right] + I_{j,n_j} \cdot \left[s_{j,n_j}^{k} \right]^{+} + B_j \cdot \left[-s_{j,n_j}^{k} \right]^{+} \right] \tag{5}$$

subject to:

$$u_{j,w}^{k} \leq s_{j,w-1}^{0} + \sum_{a=1}^{k-1} \left[u_{j,w-1}^{a} - u_{j,w}^{a} \right] \tag{6}$$

Constraint (6) applies to all operations of the routings of p_j except the first operation of its sub-routings.

$$u_{j,w}^{k} \leq s_{j,w-1}^{0} + \sum_{a=1}^{(\kappa-1)z} u_{j,w-1}^{a} - \sum_{a=1}^{k-1} u_{j,w}^{a} \tag{7}$$

Constraint (7) applies to the first operation of the sub- routings of p_j.

$$\sum_{p_j \in \mathcal{P}} \sum_{w=1}^{n_j} \left[1 \{ \alpha_{j,w} = m_i \} \cdot u_{j,w}^k \cdot t_{j,w} \right] \leq T \tag{8}$$

$$s_{j,w}^{\kappa \cdot z} \geq s_{j,w}^0 \tag{9}$$

$$u_{j,w}^k \geq 0 \tag{10}$$

$\forall k \in H$ in constraints (6) – (8), (10); $\forall \kappa \in H_A$ in constraint (9); $\forall p_j \in \mathcal{P}$ in constraints (6), (7), (9), (10); $\forall w = 1, 2, ..., n_j$ such that $h_{j,w} = 0$ in constraint (6); $\forall w = y_j^1, y_j^2, ..., y_j^{\bar{n}_r}$ such that $p_j \in f_r$ in constraint (7); $\forall w = 1, 2, ..., n_j$ in constraint (10); $\forall w = 1, 2, ..., n_j$ such that $x_{j,w.} = 0$ in constraint (9); $\forall m_i \in \mathcal{M}$ in constraint (8); for constraint (7) κ is chosen such that $k \in h(\kappa)$.

The objective function (5) consists of both (i) the inventory holding costs for work-in-process (for $w = 1, 2, ..., n_j - 1$) and finished parts ($w = n_j$) and (ii) the backlogging costs of finished parts over the entire planning horizon. Constraints (6), and (7) ensure that the production of part type p_j during an elementary period cannot exceed the number of part types p_j contained in the upstream buffer at the end of the previous elementary period. Constraint (6) is applicable to all operations other than the first operation in the sub-routings of a part type. Constraint (7) is applicable only to the first operation in the sub-routings of a part type. Note that in constraint (7), no inventory is transferred between cells during a sub-period, based on the assumption that a part type visits at most one cell during each sub-period. Constraint (8) guarantees that the resource capacity of each machine is not exceeded in a elementary period. Constraint (9) ensures that the intra-cell inventory at the end of each sub-period is always greater than or equal to the initial inventory. This is equivalent to restocking the intra-cell buffers at the end of each sub-period to at least their initial levels. Finally, constraint (10) ensures the non-negativity of production.

Note that although objective function in (5) is nonlinear, it can be transformed into a linear one by adding constraints as shown below:

$$\text{minimize} \quad \sum_{k \in H} \sum_{p_j \in \mathcal{P}} \left[\sum_{w=1}^{n_j - 1} \left[I_{j,w} \cdot s_{j,w}^k \right] + P_j^k \right]$$

subject to:

$$P_j^k \geq I_{j,n_j} \cdot s_{j,n_j}^k \quad \forall p_j \in \mathcal{P}, k \in H$$

$$P_j^k \geq -B_j \cdot s_{j,n_j}^k \quad \forall p_j \in \mathcal{P}, k \in H$$

where P_j^k is the inventory/backlogging cost related to operation o_{j,n_j} for the kth elementary period.

Although the monolithic optimization problem is straightforward to formulate, there are several reasons why it cannot be solved easily or implemented in practice, that is: (i) the linear program (LP) is of a very large dimension for a typical manufacturing system and planning horizon, (ii) detailed information about the demand of part types is not available for the entire horizon, (iii) the demand is subject to change due to order cancellations and acceptance of new orders, which, in turn, requires recomputation, (iv) the LP does not allow random events to be absorbed with a computational effort that is proportional to the impact of the random event, and (v) the LP does not allow different criteria to be used at different levels of the hierarchy.

The hierarchical formulation overcomes these problems. In addition, the hierarchical structure parallels the corporate management hierarchy and, thus, provides significant assistance to the overall management function.

3.2 Hierarchical Formulation: Aggregate Planning Level

The aggregate problem consists of determining, for each part family f_r, the production $\bar{u}^{\kappa}_{r,q}$ at each macro-operation $O_{r,q}$ and each sub-period κ of the planning horizon. The aggregate production is distributed at the detailed level among part types on machines during elementary periods. The solution of the detailed level is the number of units to be processed at each operation in a part type's routing during each elementary period. The criterion at the aggregate level is to minimize the inventory costs related to inter-cell work-in-process, and the inventory holding and backlogging costs related to the families of finished parts. At the detailed level, the criterion is to minimize the inventory costs related to work-in-process and the inventory holding and backlogging costs related to finished part types. Thus, we introduce a hierarchy among inventories; more importance is given to inter-cell and end-product inventories.

The aggregate processing time $\tau^{\kappa}_{r,q}$ is computed as follows:

$$\tau^{\kappa}_{r,q} = \frac{z}{z + 1 - \max_{p_j \in f_r} \left[v^q_j + 1 - y^q_j \right]} \cdot \max_{p_j \in f_r, w = y^q_j, \ldots, v^q_j} t_{j,w}. \tag{11}$$

The $\tau^{\kappa}_{r,q}$ in equation (11) represents the worst-case value of aggregate processing times for the part families and ensures consistency and feasibility during disaggregation. In order to ensure consistency and feasibility, the first factor in the above equation considers the case when the intra-cell buffers are empty; the second factor is the maximum value of the processing times $t_{j,w}$ for all operations in the qth sub-routing and for all part types belonging to family f_r.

The aggregate initial inventory is computed by summing up the initial inventories of all part types in a family:

$$\bar{s}^0_{r,q} = \sum_{p_j \in f_r} s^0_{j,v^q_j} \quad \forall O_{r,q}; \quad \bar{s}^{\kappa}_{r,0} = \sum_{p_j \in f_r} s^{\kappa \cdot z}_{j,0} \quad \forall \kappa \tag{12}$$

Inventory state equations

For each part family $f_r \in \mathcal{F}$, and for sub-period $\kappa \in H_A$, the inventory state equation for macro-operation $O_{r,q}, q = 1, 2, ..., \bar{n}_r - 1$ is:

$$\bar{s}^{\kappa}_{r,q} = \bar{s}^0_{r,q} + \sum_{a=1}^{\kappa} \left[\bar{u}^a_{r,q} - \bar{u}^a_{r,q+1}\right] \tag{13}$$

For the final operation $O_{r,\bar{n}_r}$ of each part family $f_r \in \mathcal{F}$, and for sub-period $\kappa \in H_A$, the inventory state equation is:

$$\bar{s}^{\kappa}_{r,\bar{n}_r} = \bar{s}^0_{r,\bar{n}_r} + \sum_{a=1}^{\kappa} \left[\bar{u}^a_{r,\bar{n}_r} - D^a_r\right] \tag{14}$$

Based on the above definitions, the aggregate level problem is formulated as follows:

Problem $\mathcal{APP}$

$$\text{minimize} \quad \sum_{\kappa \in H_A} \sum_{f_r \in \mathcal{F}} \left[\sum_{q=1}^{\bar{n}_r - 1} \left[\bar{I}_{r,q} \cdot \bar{s}^{\kappa}_{r,q}\right] + \bar{I}_{r,\bar{n}_r}[\bar{s}^{\kappa}_{r,\bar{n}_r}]^+ + \bar{B}_r[-\bar{s}^{\kappa}_{r,\bar{n}_r}]^+ \right] \tag{15}$$

subject to:

$$\bar{u}^{\kappa}_{r,q} \leq \bar{s}^0_{r,q-1} + \sum_{a=1}^{\kappa-1} \left[\bar{u}^a_{r,q-1} - \bar{u}^a_{r,q}\right] \tag{16}$$

$$\sum_{O_{r,q} \in \mathcal{MO}_v} \left[\bar{u}^{\kappa}_{r,q} \cdot \tau^{\kappa}_{r,q}\right] \leq T \cdot z \tag{17}$$

$$\bar{u}^{\kappa}_{r,q} \geq 0 \tag{18}$$

$\forall \kappa \in H_A$ in constraints (16) – (18); $\forall f_r \in \mathcal{F}$ in constraints (16), (18); $\forall q = 1, 2, ..., \bar{n}_r$ in constraints (16), (18); $\forall c_v \in \mathcal{C}$ in constraint (17).

Constraint (16) ensures that the production of a family within a sub-period cannot exceed the number of parts of that family in the upstream buffer at the end of the previous sub-period. Constraint (17) guarantees that the resource capacity of all cells is not exceeded for all sub-periods of the horizon.

3.3 Hierarchical Formulation: Detailed Planning Level

The detailed level of the production planning hierarchy consists of determining the quantities of parts types to be processed at each operation during each elementary period. This is accomplished by disaggregating the aggregate production volumes $\bar{u}^{\kappa}_{r,q}$ obtained from the solution of the aggregate problem $\mathcal{APP}$ for all macro-operations $O_{r,q}$ and all sub-periods κ.

The disaggregation consists of two steps: the *part family disaggregation* (PFD) and the *temporal and spatial disaggregation* (TSD). The PFD step accepts the aggregate plan as its input and computes the quantities of each part type to be processed at each macro-operation and each sub-period. The resulting plan from the PFD step is the input to the TSD step, which computes the quantities of each part type to be processed at each operation and each elementary period. It is noted that the disaggregation of aggregate production volumes in one step is not feasible, since at the detailed level, we cannot distribute the production corresponding to a part family among its part types for intermediate macro-operations. This is because the planning horizon at the detailed level is a sub- period, and part type production for intermediate macro-operations depends on the demand in the latter sub-periods. The two disaggregation steps are discussed in detail below.

Part Family Disaggregation

In order to disaggregate the aggregate production quantities, $\bar{u}^{\kappa}_{r,q}$, we define the parameter, $\beta^{\kappa}_{j,q} \in [0, 1]$, which is the ratio of the production of part type p_j at the qth macro-operation in the κ-th sub-period, over $\bar{u}^{\kappa}_{r,q}$. The ratios $\beta^{\kappa}_{j,q}$ distribute the production volume of each part family in each macro-operation and sub-period, among the part types belonging to that part family. They are computed by solving a linear program for each part family separately. The objective function is to minimize the holding costs for inter-cell WIP and finished part types, and the backlogging costs of finished part types.

Inventory state equations

For each part type $p_j \in f_r$, the inventory state equation for $q = 1, 2, ..., \bar{n}_r - 1$, and for sub-period $\kappa \in H_A$ is:

$$s^{\kappa \cdot z}_{j,v^q_j} = s^0_{j,v^q_j} + \sum_{a=1}^{\kappa} \left[\beta^a_{j,q} \cdot \bar{u}^a_{r,q} - \beta^a_{j,q+1} \cdot \bar{u}^a_{r,q+1}\right] \tag{19}$$

where $\beta^{\kappa}_{j,q} \cdot \bar{u}^{\kappa}_{r,q}$ is the number of units of part type $p_j \in f_r$ to be processed at macro-operation $O_{r,q}$ during the κth sub-period. For the final operation o_{j,n_j} of each part type $p_j \in f_r$, and for sub-period $\kappa \in H_A$, the inventory state equation is:

$$s_{j,n_j}^{\kappa \cdot z} = s_{j,n_j}^{0} + \sum_{a=1}^{\kappa} \left[\beta_{j,\bar{n}_r}^{a} \cdot \bar{u}_{r,\bar{n}_r}^{a}\right] - \sum_{k=1}^{\kappa \cdot z} d_j^k \tag{20}$$

The disaggregation problem for each part family f_r is formulated by the following linear program:

Problem $\mathcal{PFD}(f_r)$

minimize

$$\sum_{\kappa \in H_A} \sum_{p_j \in f_r} \left[\sum_{q=1}^{\bar{n}_r - 1} \left[I_{j,v_j^q} \cdot s_{j,v_j^q}^{\kappa \cdot z}\right] + I_{j,n_j} \left[s_{j,n_j}^{\kappa \cdot z}\right]^+ + B_j \left[-s_{j,n_j}^{\kappa \cdot z}\right]^+ \right] \tag{21}$$

subject to:

$$\beta_{j,q}^{\kappa} \cdot \bar{u}_{r,q}^{\kappa} \leq s_{j,v_j^{q-1}}^{0} + \sum_{a=1}^{\kappa - 1} \left[\beta_{j,q-1}^{a} \cdot \bar{u}_{r,q-1}^{a} - \beta_{j,q}^{a} \cdot \bar{u}_{r,q}^{a}\right] \tag{22}$$

$$\sum_{p_j \in f_r} \beta_{j,q}^{\kappa} = 1 \tag{23}$$

$$\beta_{j,q}^{\kappa} \geq 0 \tag{24}$$

$\forall \kappa \in H_A$ in constraints (22) - (24); $\forall p_j \in f_r$ in constraints (22, 24); $\forall q = 1, 2, ..., \bar{n}_r$ in constraints (22 - 24).

Constraint (22) ensures that the production of a part type within a sub-period cannot exceed the number of parts contained in the upstream buffer at the end of the previous sub-period. Constraint (23) ensures that the cumulative production of part types belonging to a part family on each macro-operation and each sub- period is equal to the one provided by the aggregate plan.

The problems $\mathcal{PFD}(f_r)$ $\forall r$ are independent of each other, and hence can be solved in parallel.

Temporal and Spatial Disaggregation

The second step of the disaggregation procedure determines the production of each part type for each operation during each elementary period. In this step, the production volume of a part family is distributed among the part types according to the ratios $\beta^{\kappa}_{j,q}$ obtained from the problems $\mathcal{PFD}(f_r)$ $\forall r$. It is noted that the production plan at this step satisfies the targeted plan obtained from the aggregate level.

Inventory state equations

For each part type $p_j \in \mathcal{P}$, the inventory state equations for operations $o_{j,w}$, $w = 1, 2, ..., n_j - 1$ and for elementary period $k \in h(\kappa)$ are:

$$s^k_{j,w} = s^{(\kappa-1)z}_{j,w} + \sum_{a=(\kappa-1)z+1}^{k} \left[u^a_{j,w} - u^a_{j,w+1}\right] \tag{25}$$

For the final operation o_{j,n_j} of each part type $p_j \in \mathcal{P}$, and for elementary period $k \in h(\kappa)$, the inventory state equation is:

$$s^k_{j,n_j} = s^{(\kappa-1)z}_{j,n_j} + \sum_{a=(\kappa-1)z+1}^{k} \left[u^a_{j,n_j} - d^a_j\right] \tag{26}$$

The optimization problem at the detailed level for the κ-th sub-period can be formulated as the following linear problem:

Problem $\mathcal{TSD}(\kappa)$

minimize

$$\sum_{k\in h(\kappa)} \sum_{p_j\in\mathcal{P}} \left[\sum_{w=1}^{n_j-1} \left[I_{j,w} \cdot s^k_{j,w}\right] + I_{j,n_j} \left[s^k_{j,n_j}\right]^+ + B_j \left[-s^k_{j,n_j}\right]^+\right] \tag{27}$$

subject to:

$$u_{j,w}^{k} \leq s_{j,w-1}^{(\kappa-1)z} + \sum_{a=(\kappa-1)z+1}^{k-1} \left[u_{j,w-1}^{a} - u_{j,w}^{a}\right] \tag{28}$$

Constraint (28) applies to all operations of the routings of p_j except the first operations of its sub-routings.

$$u_{j,w}^{k} \leq s_{j,w-1}^{(\kappa-1)z} - \sum_{a=(\kappa-1)z+1}^{k-1} u_{j,w}^{a} \tag{29}$$

Constraint (29) applies to the first operation of the sub- routings of p_j.

$$\sum_{j\in\mathcal{P}} \sum_{w=1}^{n_j} \left[1\left\{\alpha_{j,w} = m_i\right\} u_{j,w}^{k} \cdot t_{j,w}\right] \leq T \tag{30}$$

$$\sum_{k\in h(\kappa)} u_{j,v_j^q}^{k} = \beta_{j,q}^{\kappa} \cdot \bar{u}_{r,q}^{\kappa} \tag{31}$$

$$s_{j,w}^{\kappa \cdot z} \geq s_{j,w}^{0} \tag{32}$$

$$u_{j,w}^{k} \geq 0 \tag{33}$$

$\forall k \in h(\kappa)$ in constraints (28) – (30), (33); $\forall p_j \in \mathcal{P}$ in constraints (28), (29), (31) – (33); $\forall w = 1, 2, ..., n_j$ such that $h_{j,w} = 0$ in constraint (28); $\forall w = y_j^1, y_j^2, ..., y_j^{\bar{n}_r}$ where $p_j \in f_r$ in constraint (29); $\forall w = 1, 2, ..., n_j$ in constraint (33); $\forall w = 1, 2, ..., n_j$ such that $x_{j,w} = 0$ in constraint (32); $\forall q = 1, 2, ..., \bar{n}_r$ where $p_j \in f_r$ in constraint (31); $\forall m_i \in \mathcal{M}$ in constraint (30).

Constraints (28), (29) ensure that the production of a part within an elementary period cannot exceed the number of parts contained in the upstream buffer at the end of the previous elementary period. Constraint (30) guarantees that the capacity of each machine is not exceeded for each elementary period. Constraint (31) ensures that the production of part type p_j for operation o_{j,v_j^q} during the elementary periods of the κ-th sub-period is equal to the value determined from the solution of (i) the aggregate problem and (ii) the part family disaggregation problem. Constraint (32) ensures that the intra-cell inventory at the end of the κ-th sub-period is greater than or equal to the initial inventory.

It is emphasized that the production of a part type in a cell during a sub-period is transferred to the next cell only at the beginning of the next sub-period. Hence, the problem $\mathcal{TSD}(\kappa)$ can be decomposed into $|\mathcal{C}|$ individual optimization problems, each corresponding to one manufacturing cell. Since these problems are independent of each other, they can also be solved in parallel, greatly enhancing computational efficiency. The decomposition of problem $\mathcal{TSD}(\kappa)$ into $|\mathcal{C}|$ optimization problems is described below.

Let $\mathcal{O}_v$ be the set of all operations performed in cell c_v.

Consider the objective function (27) of problem $\mathcal{TSD}(\kappa)$:

$$J(\kappa) = \sum_{k \in h(\kappa)} \sum_{p_j \in \mathcal{P}} \left[\sum_{w=1}^{n_j - 1} \left[I_{j,w} \cdot s_{j,w}^{k} \right] + I_{j,n_j} \left[s_{j,n_j}^{k} \right]^{+} + B_j \left[-s_{j,n_j}^{k} \right]^{+} \right]$$

Substituting the values of $s_{j,w}^{k}$ from equation (25), and simplifying yields

$$J(\kappa) = \sum_{k \in h(\kappa)} \sum_{p_j \in \mathcal{P}} \left[\sum_{w=1}^{n_j - 1} \left[I_{j,w} \cdot \sum_{a=(\kappa-1)z+1}^{k} \left[u_{j,w}^{a} - u_{j,w+1}^{a} \right] \right] + I_{j,n_j} \left[s_{j,n_j}^{k} \right]^{+} + B_j \left[-s_{j,n_j}^{k} \right]^{+} \right] + \sum_{k \in h(\kappa)} \sum_{p_j \in \mathcal{P}} \sum_{w=1}^{n_j - 1} \left[I_{j,w} \cdot s_{j,w}^{(\kappa-1)z} \right]$$

The last term of the above expression does not depend on the decisions made during the κth sub-period, since $I_{j,w}$ is a static system parameter, and $s_{j,w}^{(\kappa-1)z}$ is the initial inventory at the beginning of the κth sub-period; $s_{j,w}^{(\kappa-1)z}$ is computed from problem

$\mathcal{TSD}(\kappa-1)$ for $\kappa = 2, 3, ..., Z$. After disregarding the last term of the above expression and simplifying further, we obtain:

$$J_*(\kappa) = \sum_{k\in h(\kappa)} \sum_{p_j\in\mathcal{P}} \left[\sum_{a=(\kappa-1)z+1}^{k} \left[u_{j,1}^a \cdot I_{j,1} + \sum_{w=2}^{n_j-1} \left[u_{j,w}^a (I_{j,w} - I_{j,w-1}) \right] - u_{j,n_j}^a \cdot I_{j,n_j-1} \right] + I_{j,n_j} \left[s_{j,n_j}^k \right]^+ + B_j \left[-s_{j,n_j}^k \right]^+ \right]$$

Let $J_*(\kappa) = \sum_{c_v\in\mathcal{C}} J_*(\kappa, c_v)$ where $J_*(\kappa, c_v)$ contains only the variable $u_{j,w}^k$ and $s_{j,w}^k$ such that operation $o_{j,w}$ is performed in cell c_v. $J_*(\kappa, c_v)$ can then be written as follows:

$$\begin{aligned} J_*(\kappa, c_v) &= \sum_{k\in h(\kappa)} \sum_{p_j\in\mathcal{P}} \Bigg[\sum_{a=(\kappa-1)z+1}^{k} \Big[u_{j,1}^a \cdot I_{j,1} \cdot 1\{\alpha_{j,1} \in c_v\} \\ &+ u_{j,w}^a \sum_{w=2}^{n_j-1} \left[(I_{j,w} - I_{j,w-1}) \cdot 1\{\alpha_{j,w} \in c_v\} \right] \\ &- u_{j,n_j}^a \cdot I_{j,n_j-1} \cdot 1\left\{\alpha_{j,n_j} \in c_v\right\} \Big] \\ &+ \left(I_{j,n_j} \left[s_{j,n_j}^k \right]^+ + B_j \left[-s_{j,n_j}^k \right]^+ \right) \cdot 1\left\{\alpha_{j,n_j} \in c_v\right\} \Bigg] \end{aligned}$$

Thus, the criterion of TSD problem can be decomposed into $|\mathcal{C}|$ criteria, one for each cell. Furthermore, the decision variables in constraints (28) – (33) of the TSD problem for a certain cell are independent of the decision variables corresponding to the other cells. Thus, the the temporal and spatial decomposition problem for cell c_v and for the κ-th sub-period can be formulated as the following linear problem:

Problem $\mathcal{TSDC}(\kappa, c_v)$

$$\text{minimize} \qquad J_*(\kappa, c_v)$$

subject to:

constraints (28) - (33)

$\forall k \in h(\kappa)$ in constraints (28) - (30), (33); $\forall o_{j,w} \in \mathcal{O}_v$ such that $h_{j,w} = 0$ in constraint (28); $\forall w = y_j^q$ where $p_j \in f_r$ and $O_{r,q} \in \mathcal{MO}_v$ in constraint (29); $\forall o_{j,w} \in \mathcal{O}_v$ in constraint (33); $\forall o_{j,w} \in \mathcal{O}_v$ such that $x_{j,w} = 0$ in constraint (32); $\forall p_j \in f_r, O_{r,q} \in \mathcal{MO}_v$ in constraint (31); $\forall m_i \in c_v$ in constraint (30).

The temporal and spatial disaggregation problems $\mathcal{TSDC}(\kappa, c_v)$ $\forall \kappa, v$ are independent of each other and can be solved in parallel.

3.4 Discussion

In hierarchical planning, the problem at the aggregate level is solved first. The aggregate solution is the number of units of each part family to processed at each macro-operation and each sub-period. The aggregate processing times $\tau_{r,q}^{\kappa}$ $\forall r, q, \kappa$ are obtained from equation (11) and are the worst-case values, which, nevertheless, ensure consistency and feasibility during disaggregation of the aggregate production volumes at the detailed level. Section 4 presents an iterative algorithm to improve the aggregate processing time value at each iteration, based on the detailed solution obtained in the previous iteration.

The disaggregation of the aggregate production volumes is performed in two steps. The first step (part family disaggregation) computes the quantity of each part type to be processed at each macro-operation and each sub-period. The second step (temporal and spatial disaggregation) computes the quantities of each part type to be processed at each operation and each elementary period. Table 1 summarizes the horizon and

Table 1: Summary of the entities and horizon for the monolithic, aggregate, part family disaggregation, and temporal and spatial disaggregation problems

problem	# of time periods periods considered in the horizon	time period	part	resource
aggregate problem $\mathcal{AP}$	Z	sub-periods	families	cells
part family disaggregation $\mathcal{PFD}(f_r)$	Z	sub-periods	types	cells
temporal and spatial disaggregation $\mathcal{TSD}(c_\eta, \kappa)$	z	elementary periods	types	machines
monolithic problem $\mathcal{MP}$	$Z \cdot z$	elementary periods	types	machines

the entities of interest for the monolithic, aggregate, part family disaggregation, and temporal and spatial disaggregation problems.

4 Solution Algorithms for Hierarchical Production Planning

We propose two algorithms for solving the hierarchical production planning problem formulated in Section 3.

4.1 Single-Step Algorithm

This algorithm solves the three subproblems of the hierarchical production planning formulation in sequence. It first solves the aggregate problem $\mathcal{APP}$ to compute the number of units of each part family to be processed by each macro-operation during each sub-period. The aggregate production plan is then disaggregated in two steps. In the first step, designated as *part family disaggregation*, the part family production is distributed among its part types, for each macro-operation and each sub-period. This is accomplished by solving the $\mathcal{PFD}(f_r)$ linear program for each part family. In the second step, called *temporal and spatial disaggregation*, the production quantity of a part type obtained from the part family disaggregation step is disaggregated further by solving the $\mathcal{TSDC}(c_v, \kappa)$ linear program for each cell and each sub-period. The resulting solution is the number of units of part types to be processed through each operation during each elementary period. The single-step algorithm, HPPS, is described in Figure 1, and detailed below.

Algorithm HPPS

Input:

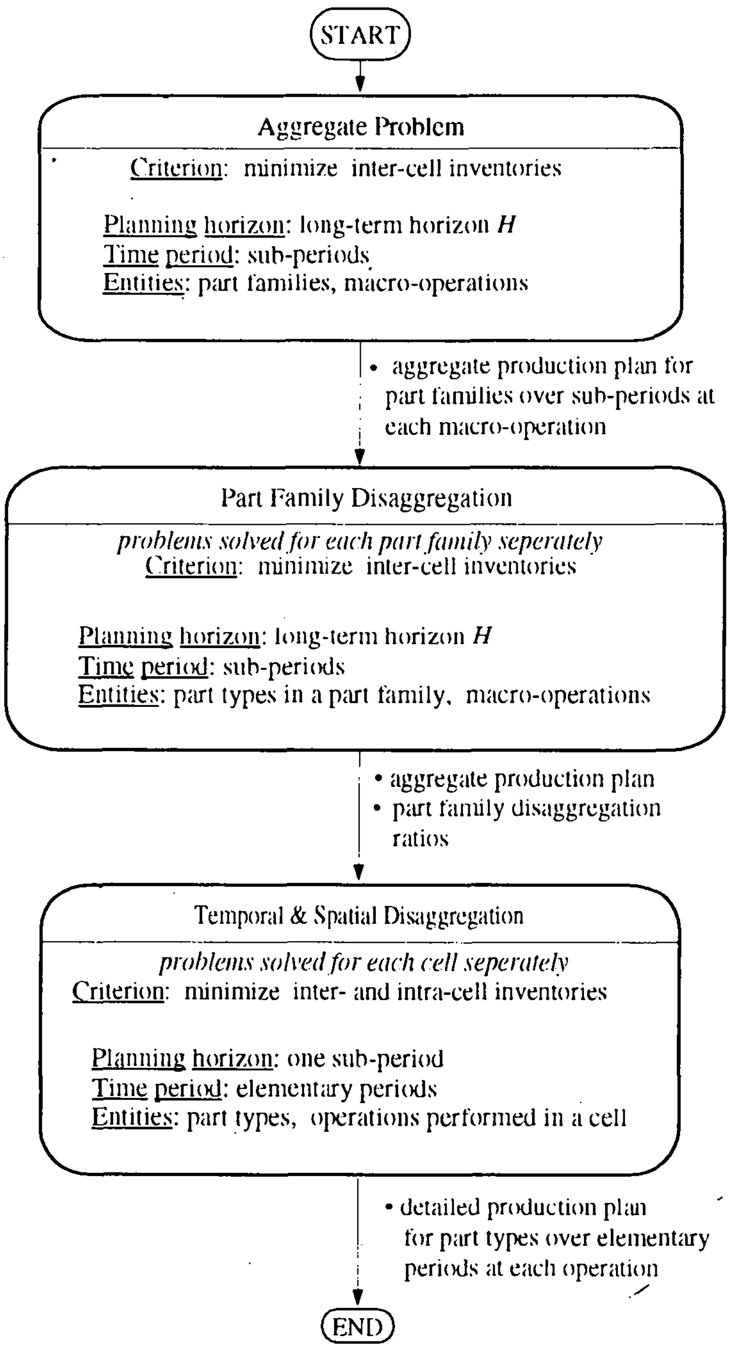

Figure 1: Single-step algorithm for hierarchical production planning

- For each machine: capacity per elementary period.
- For each part: routing, demand per elementary period, initial inventory, backlogging and holding costs.
- Number of sub-periods, and number of elementary periods. It is assumed that the lengths of the sub-periods and elementary periods are fixed.
- Cell and part family configuration.

Output: Detailed production plan, i.e. $u^k_{j,w}$ $\forall j, w, k$.

Initialization: Compute the aggregate costs $\bar{I}_{r,q}$ and $\bar{B}_r$ from equations (1) and (2), respectively. Compute the aggregate inventories $\bar{s}^0_{r,q}$ and $\bar{s}^\kappa_{r,0}$ from equations (12). Compute the aggregate processing times $\tau^\kappa_{r,q}$ from equation (11) to guarantee feasibility and consistency during disaggregation (shown later in this section).

Begin Algorithm.

1. Solve Aggregate Problem $\mathcal{APP}$.
2. Using the solution of the aggregate problem, solve part family disaggregation problem $\mathcal{PFD}(f_r)$ for every part family f_r.
3. Using the solution from problems $\mathcal{APP}$ and $\mathcal{PFD}(f_r)$, solve temporal and spatial disaggregation problem $\mathcal{TSDC}(\kappa, c_v)$ for every sub-period κ and every cell c_v.

End Algorithm.

This algorithm is applied on a rolling horizon basis. That is, although the production plan is computed over the entire planning horizon, it is implemented over the first sub-period only. At the end of the first sub-period, the plan is recomputed based on the actual state of the system. This is done in order to incorporate the future demands/forecasts progressively.

In Algorithm HPPS, the aggregate processing times for part families are computed from equation (11). Their values are very conservative and may lead to a high backlog of parts. To overcome this problem, an extension to the algorithm is provided later in this section.

Consistency and Feasibility During Disaggregation

In hierarchical decisions systems the term consistency implies that the solution of the detailed problems at the lower levels of the hierarchy satisfies the target production obtained from the aggregate problems at the upper levels of the hierarchy. It is shown below that the hierarchical production planning formulation presented in Section 3 ensures consistency by using the top-down constraints. The formal definition of consistency is as follows.

Definition 1 *For the formulations of Sections 3.2 and 3.3, a detailed plan* ($u^k_{j,w}$ $\forall j, w, k$) *is consistent with the aggregate plan* ($\bar{u}^\kappa_{r,q}$ $\forall r, q, \kappa$) *if the following equations are satisfied:*

$$\text{for problem } \mathcal{PFD}(f_r) \quad \sum_{p_j \in f_r} \left[\bar{u}^\kappa_{r,q} \cdot \beta^\kappa_{j,q}\right] = \bar{u}^\kappa_{r,q} \quad \forall r, q, \kappa \tag{34}$$

$$\text{for problem } \mathcal{TSDC}(\kappa) \quad \sum_{k \in h(\kappa)} u^k_{j,v^q_j} = \bar{u}^\kappa_{r,q} \cdot \beta^\kappa_{j,q} \quad \forall p_j \in f_r, r, q, \kappa \tag{35}$$

Consistency condition (34) implies that the cumulative production volume $\bar{u}^\kappa_{r,q} \cdot \beta^\kappa_{j,q}$ of part types p_j belonging to family f_r produced at macro-operation $O_{r,q}$ during sub-period κ is equal to the aggregate production volume $\bar{u}_{r,q}^{\kappa}$ determined from the aggregate problem. Consistency condition (35) implies that the cumulative production volume $\sum_{k\in h(\kappa)} u_{j,v_j^q}^{k}$ of part type p_j produced at operation o_{j,v_j^q} during elementary period k of the κth sub-period is equal to the value $\bar{u}_{r,q}^{\kappa} \cdot \beta_{j,q}^{\kappa}$ determined from the aggregate problem and the part family disaggregation problem.

Recall that a solution to an optimization problem is feasible if it satisfies all the constraints of that problem. Since the consistency equations (34) and (35) are also constraints to problems $\mathcal{PFD}(f_r)$ and $\mathcal{TSDC}(\kappa, c_v)$ respectively, *any feasible solution to these problems is also a consistent solution.* Hence, to prove consistency we verify that there exists a feasible solution to the PFD and TSD disaggregation problems. The existence of a feasible solution to these problems is guaranteed by Theorems 1 and 2. To prove Theorem 1 we first present Property 1 and Lemma 1. In order to reduce the notation, we assume that a part type visits a machine at most once during its visit to a cell.

Property 1 *If aggregate plan* $\bar{u}_{r,q}^{\kappa}$ $\forall r, q, \kappa$ *satisfies constraints (16) and (18) of aggregate problem* $\mathcal{APP}$, *then* $\bar{s}_{r,q}^{\kappa} \geq 0$ $\forall r,$ $q = 1, 2, ..., \bar{n}_r - 1,$ $\kappa = 1, 2, ..., Z - 1.$

Proof

Constraint (16) states that

$$\bar{u}_{r,q}^{\kappa} \leq \bar{s}_{r,q-1}^{0} + \sum_{a=1}^{\kappa-1} \left[\bar{u}_{r,q-1}^{a} - \bar{u}_{r,q}^{a}\right] \quad \forall r, q, \kappa$$

Using equation (13) the above constraint becomes:

$$\bar{u}_{r,q}^{\kappa} \leq \bar{s}_{r,q-1}^{\kappa-1} \quad \forall r, q, \kappa$$

From constraint (18), $\bar{u}^{\kappa}_{r,q} \geq 0$ and, therefore,

$$0 \leq \bar{s}^{\kappa-1}_{r,q-1} \quad \forall r, q, \kappa$$

or, $\bar{s}^{\kappa}_{r,q} \geq 0 \quad \forall r, q = 1, 2, ..., \bar{n}_r - 1, \kappa = 1, 2, ..., Z - 1.$

Lemma 1 $\sum_{p_j \in f_r} s^{(\kappa-1)z}_{j,v^{q-1}_j} = \bar{s}^{\kappa-1}_{r,q-1} \quad \forall r, q, \kappa$, *if the ratios* $\beta^{\kappa}_{j,q}$ *which define* $s^{(\kappa-1)z}_{j,v^{q-1}_j}$ *verify constraint (23).*

Proof

Summing the constraints (19) for all $p_j \in f_r$, we obtain:

$$\sum_{p_j \in f_r} s^{(\kappa-1)z}_{j,v^{q-1}_j} = \sum_{p_j \in f_r} s^{0}_{j,v^{q-1}_j} + \sum_{a=1}^{\kappa-1} \sum_{p_j \in f_r} \left[\beta^{a}_{j,q-1} \cdot \bar{u}^{a}_{r,q-1} - \beta^{a}_{j,q} \cdot \bar{u}^{a}_{r,q}\right]$$

for all $q = 2, 3, ..., \bar{n}_r$, and for all κ

Since $\sum_{p_j \in f_r} \beta^{\kappa}_{j,q} = 1$, we can rewrite the above equation as:

$$\sum_{p_j \in f_r} s^{(\kappa-1)z}_{j,v^{q-1}_j} = \sum_{p_j \in f_r} s^{0}_{j,v^{q-1}_j} + \sum_{a=1}^{\kappa-1} \left[\bar{u}^{a}_{r,q-1} - \bar{u}^{a}_{r,q}\right]$$

Using equation (13) for $q = 2, 3, ..., \bar{n}_r$ and equation (12) for $q = 1$, we can conclude that

$$\sum_{p_j \in f_r} s^{(\kappa-1)z}_{j,v^{q-1}_j} = \bar{s}^{\kappa-1}_{r,q-1} \quad \forall r, q, \kappa$$

Q.E.D.

Theorem 1 *For any feasible solution of the aggregate problem* $\mathcal{APP}$, *there exists at least one feasible solution to the part family disaggregation problem* $\mathcal{PFD}(f_r)$ *for every part family* f_r.

Proof

We prove the above theorem by showing that the set of feasible solutions to problem $\mathcal{PFD}(f_r)$ is non-empty. We claim that the ratios $\beta^{\kappa}_{j,q}$ obtained from equation (36) below satisfy constraints (22) – (24) of problem $\mathcal{PFD}(f_r)$.

$$\beta^{\kappa}_{j,q} = \begin{cases} \dfrac{s^{(\kappa-1)z}_{j,v_j^{q-1}}}{\bar{s}^{\kappa-1}_{r,q-1}} & \text{if } \bar{s}^{\kappa-1}_{r,q-1} \neq 0 \\ \\ \dfrac{1}{|f_r|} & \text{otherwise} \end{cases} \tag{36}$$

Equation (36) states that the value of $\beta^{\kappa}_{j,q}$ is the ratio of the inventory level of part type p_j to the inventory level of part family f_r, both measured at the end of the previous sub-period $\kappa - 1$ and the previous macro-operation $O_{r,q-1}$ (for non-zero part family inventory level).

The ratios $\beta^{\kappa}_{j,q}$ verify constraint (22) as follows. From equation (19), constraint (22) becomes:

$$\beta^{\kappa}_{j,q} \cdot \bar{u}^{\kappa}_{r,q} \leq s^{(\kappa-1)z}_{j,v_j^{q-1}}$$

Substituting $\beta^{\kappa}_{j,q}$ from equation (36) in the above inequality, we obtain:

$$\bar{u}^{\kappa}_{r,q} \leq \bar{s}^{\kappa-1}_{r,q-1} \qquad \text{if } \bar{s}^{\kappa-1}_{r,q-1} \neq 0\text{, and} \tag{37}$$

$$\bar{u}^{\kappa}_{r,q} \leq s^{(\kappa-1)z}_{j,v_j^{q-1}} \cdot |f_r| \qquad \text{if } \bar{s}^{\kappa-1}_{r,q-1} = 0. \tag{38}$$

Inequality (37) is true due to constraint (16), since the aggregate solution $\bar{u}^{\kappa}_{f,q}$ satisfies all the constraints of the aggregate problem $\mathcal{APP}$. Inequality (38) is true since

$\bar{s}^{\kappa-1}_{r,q-1} = 0$ implies: (i) $\bar{u}^{\kappa}_{r,q} = 0$ from constraint (16), and (ii) $s^{(\kappa-1)z}_{j,v^{q-1}_j} = 0$ from Lemma 1. Hence, the ratios $\beta^{\kappa}_{j,q}$ obtained from equation (36) satisfy constraint (22) of problem $\mathcal{PFD}(f_r)$.

The ratios $\beta^{\kappa}_{j,q}$ also satisfy constraint (23) by Lemma 1. We prove that $\beta^{\kappa}_{j,q}$ satisfies constraint (24) by showing that $s^{(\kappa-1)\cdot z}_{j,v^{q-1}_j}$ and $\bar{s}^{\kappa-1}_{r,q-1}$ are non-negative for all j, r, q, κ. Note that the aggregate inventory $\bar{s}^{\kappa-1}_{r,q-1}$ is non-negative for the aggregate solution $\bar{u}^{\kappa}_{r,q}$ from Property 1. We prove that $s^{(\kappa-1)\cdot z}_{j,v^{q-1}_j}$ is non-negative using equations (19) and (36) to obtain (i) and (ii) below:

(i) For $\kappa = 1$, $s^{z}_{j,v^q_j} = s^0_{j,v^q_j} + \frac{s^0_{j,v^q_j}}{\bar{s}^0_{r,q}} \cdot \bar{u}^1_{r,q} - \frac{s^0_{j,v^{q+1}_j}}{\bar{s}^0_{r,q+1}} \cdot \bar{u}^1_{r,q+1}$. Variable s^{z}_{j,v^q_j} is non-negative since the initial inventory s^0_{j,v^q_j} $\forall j, q$ is always non-negative, and the aggregate production $\bar{u}^{\kappa}_{r,q}$ $\forall r, q, \kappa$ is non-negative due to constraint (18).

(ii) For $\kappa = 2$, $s^{2\cdot z}_{j,v^q_j} = s^{z}_{j,v^q_j} + \frac{s^1_{j,v^q_j}}{\bar{s}^1_{r,q}} \cdot \bar{u}^2_{r,q} - \frac{s^1_{j,v^{q+1}_j}}{\bar{s}^1_{r,q+1}} \cdot \bar{u}^2_{r,q+1}$. Variable $s^{2\cdot z}_{j,v^q_j}$ is non-negative since the inventory s^{z}_{j,v^q_j} $\forall j, q$ is non-negative from (i) and the aggregate production $\bar{u}^{\kappa}_{r,q}$ $\forall r, q, \kappa$ is non-negative due to constraint (18).

The results of (i) and (ii) hold true for all sub-periods. Hence, $s^{(\kappa-1)z}_{j,v^{q-1}_j} \geq 0$ $\forall r, q, \kappa$ which implies that constraint (23) is verified by the ratios $\beta^{\kappa}_{j,q}$ defined in equation (36). Since constraints (22) - (24) are satisfied for any feasible solution of the aggregate problem $\mathcal{APP}$, there exists an least one feasible solution to the part family disaggregation problem $\mathcal{PFD}(f_r)$ for every part family f_r.

Q.E.D.

Theorem 2 *For any feasible solution of the aggregate problem $\mathcal{APP}$, with aggregate processing times $\tau^{\kappa}_{r,q}$ defined by equation (11), there exists at least one feasible solution to the temporal and spatial disaggregation problem $\mathcal{TSDC}(\kappa, c_v)$ for every sub-period κ and every cell c_v.*

Proof

From Theorem 1, we know that there exists at least one feasible solution of problem $\mathcal{PFD}(f_r)$ for any feasible solution to aggregate problem $\mathcal{APP}$. We will now prove that there exists a solution $u^k_{j,w}$ that satisfies the constraints (28) - (33) of the temporal and spatial disaggregation problem $\mathcal{TSD}(\kappa)$. Note that a feasible solution to the TSD problem satisfies the target production obtained from the aggregate and PFD problems. Also, since the constraints of problems $\mathcal{TSD}(\kappa)$ are used in problems $\mathcal{TSDC}(\kappa, c_v)$, any feasible solution of the former is also a feasible solution of the latter.

Consider the $0-1$ variable $\varphi^k_{j,w}$ defined below:

$$\varphi^k_{j,w} = \begin{cases} 1 & \text{if } (\kappa-1)z + \psi_{j,w,q} \leq k \leq \\ & \kappa \cdot z + \psi_{j,w,q} - \max_{p_a \in f_r}[v^q_a - y^q_a + 1] \quad ; \quad \forall k \in h(\kappa) \\ 0 & \text{otherwise} \end{cases} \tag{39}$$

where the positive integer variables $\psi_{j,w,q}$ denote the rank of operation $o_{j,w}$ in the sub-routing $< o_{j,a} >^{v^q_j}_{a=y^q_j}$, i.e. $\psi_{j,w,q} = v^q_j - w + 1$.

For all operations $o_{j,w}$, except the first operation in each sub-routing (i.e. $h_{j,w} = 0$), and for $k > (\kappa - 1)z + 1$, equation (39) states that:

$$\varphi^k_{j,w} = \varphi^{k-1}_{j,w-1} \tag{40}$$

Let $\chi^{\kappa}_{j,q}$ be a positive real variable as defined below:

$$\chi^{\kappa}_{j,q} = \bar{u}^{\kappa}_{r,q} \cdot \beta^{\kappa}_{j,q} \cdot \frac{1}{z + 1 - \max_{p_a \in f_r} \left[v^q_a - y^q_a + 1 \right]}, \tag{41}$$

where $p_j \in f_r$ and $O_{r,q}$ is a macro-operation of family f_r.

We claim that $u^k_{j,w}$ obtained from equation (42) below satisfies constraints (28) – (33) of problem $\mathcal{TSD}(\kappa)$.

$$u^k_{j,w} = \chi^{\kappa}_{j,q} \cdot \varphi^k_{j,w} \tag{42}$$

where $o_{j,w}$ is in the sub-routing $< o_{j,a} >^{v^q_j}_{a=y^q_j}$ and $k \in h(\kappa)$. For the example sub-routing of

Constraint (28) can be re-written as:

$$\sum_{a=(\kappa-1)z+1}^{k} u^a_{j,w} \leq s^{(\kappa-1)z}_{j,w-1} + \sum_{a=(\kappa-1)z+1}^{k-1} u^a_{j,w-1} \tag{43}$$

Also, from equations (40) and (42),

$$\sum_{a=(\kappa-1)z+1}^{k} u^a_{j,w} = \sum_{a=(\kappa-1)z+1}^{k-1} u^a_{j,w-1}$$

Since $s^{(\kappa-1)z}_{j,w-1} \geq 0$, condition (43) is always satisfied and variable $u^k_{j,w}$ from equation (42) always satisfies constraint (28).

It is clear that constraint (29) is satisfied by substituting the solution $u^k_{j,w}$ provided by equation (42) in it. We prove that constraint (30) is satisfied by considering any machine $m_i \in c_v$ and elementary period $k \in h(\kappa)$. The left hand side of the capacity constraint (30) is:

$$\sum_{p_j \in \mathcal{P}} \sum_{w=1}^{n_j} \left[1 \left\{ \alpha_{j,w} = m_i \right\} u^k_{j,w} \cdot t_{j,w} \right]$$

$$= \sum_{O_{r,q}\in\mathcal{MO}_v} \sum_{p_j\in f_r} \left[\sum_{w=y_j^q}^{v_j^q} 1\{\alpha_{j,w} = m_i\} \chi_{j,q}^{\kappa} \cdot \varphi_{j,w}^{k} \cdot t_{j,w} \right] \quad \text{from equation (42)}$$

$$\leq \sum_{O_{r,q}\in\mathcal{MO}_v} \sum_{p_j\in f_r} \left[\chi_{j,q}^{\kappa} \cdot \max_{p_a\in f_r, w=y_a^q,\ldots,v_a^q} t_{a,w} \right] \quad \text{since} \sum_{w=y_j^q}^{v_j^q} 1\{\alpha_{j,w} = m_i\} = 1$$

Using constraints (41) and (39), we obtain

$$\sum_{O_{r,q}\in\mathcal{MO}_v} \sum_{p_j\in f_r} \left[\chi_{j,q}^{\kappa} \cdot \max_{p_a\in f_r, w=y_a^q,\ldots,v_a^q} t_{a,w} \right]$$

$$\leq \sum_{O_{r,q}\in\mathcal{MO}_v} \sum_{p_j\in f_r} \left[\frac{\bar{u}_{r,q}^{\kappa} \cdot \beta_{j,q}^{\kappa}}{z + 1 - \max_{p_a\in f_r} [v_a^q - y_a^q + 1]} \cdot \max_{p_a\in f_r, w=y_a^q,\ldots,v_a^q} t_{a,w} \right]$$

$$= \frac{1}{z} \sum_{O_{r,q}\in\mathcal{MO}_v} \sum_{p_j\in f_r} \left[\bar{u}_{r,q}^{\kappa} \cdot \beta_{j,q}^{\kappa} \cdot \frac{z}{z + 1 - \max_{p_a\in f_r} [v_a^q - y_a^q + 1]} \cdot \max_{p_a\in f_r, w=y_a^q,\ldots,v_a^q} t_{a,w} \right]$$

$$= \frac{1}{z} \sum_{O_{r,q}\in\mathcal{MO}_v} \left[\bar{u}_{r,q}^{\kappa} \cdot \tau_{r,q}^{\kappa} \right] \quad \text{from (11) and (23)}$$

$$\leq T \quad \text{from constraint (17)}$$

Hence constraint (30) is also satisfied.

From equation (42), the left hand side of constraint (31) can be written as: $\left(z + 1 - \max_{p_j\in f_r} \left[v_j^q - y_j^q + 1\right]\right) \cdot \chi_{j,q}^{\kappa}$, which equals the right hand side of constraint (31) from equation (41). Hence, constraint (31) is satisfied.

Constraint (32) is satisfied by substituting $u_{j,w}^{k}$ from equation (42) in it. Constraint (33) is satisfied by the definition of variables $u_{j,w}^{k}$ in equation (42).

Since constraints (28) - (33) are satisfied for any feasible solution of the aggregate problem, there exists at least one feasible solution to the problem $\mathcal{TSDC}(\kappa, c_v)$ for every sub-period κ and every cell c_v.

Q.E.D.

4.2 Iterative Algorithm

The single-step algorithm HPPS provided in Section 4.1 uses equation (11) to estimate the aggregate processing times for each part family, macro-operation, and sub-period. In most cases, these values are highly conservative because they represent the worst-case values. These conservative values of the aggregate processing times may lead to high backlogs even when sufficient capacity is available at the detailed level. This is because the detailed production quantities are constrained by the target aggregate production plan, which, in turn, is constrained by the aggregate capacity constraint. Thus, the conservative values of the aggregate processing times lead to low aggregate and detailed production quantities. To overcome this problem, an extension to algorithm HPPS is provided in this section. The basic steps of this iterative algorithm, called HPPI, are presented below.

The first iteration of HPPI comprises the single-step algorithm of Section 4.1 and uses the aggregate processing times $\tau^{\kappa}_{r,q}$ computed from equation (11). The detailed plan obtained from the first iteration is analyzed to determine the parts which are either held or backlogged, as well as the load of each machine during each elementary period. For those part families which are either backlogged or held during a sub-period, the aggregate processing times $\tau^{\kappa}_{r,q}$ are reduced if capacity is available on machines employed in their production, during the relevant elementary periods. The new aggregate processing time for such a part family is given by:

$$\tau^{\kappa}_{r,q} \leftarrow \tau^{\kappa}_{r,q} - \mu \cdot \Delta\tau^{\kappa}_{r,q},$$

where μ is the step size at the second iteration. The value of $\Delta\tau^{\kappa}_{r,q}$ is determined by analyzing the state of the system at the end of iteration i. The aggregate problem is then resolved using the updated values of $\tau^{\kappa}_{r,q}$ and the aggregate production plan is disaggregated. This procedure is repeated until iteration i for which the updated values of the aggregate processing times lead to an empty feasible space at the detailed level. In this case the step size μ is reduced, and the values of the aggregate processing times are recomputed using the updated value of μ. The algorithm is repeated until either: (i) the solution of the aggregate problem does not change in two consecutive iterations (which implies that further improvement in the aggregate solution is not possible), or (ii) the value of the step size μ_i is less than a threshold ϵ specified by the user.

5 Conclusion

In this paper, we presented a hierarchical production planning model for general job shops. In this model, the high level sub-model is obtained by aggregating part types, machines and time periods, which corresponds to what production management implicitly do when making decisions; we just tried to rationalize a common approach in a sense which favors parallel computation. We also presented a solution algorithm that iteratively adjusts the aggregate processing times in order to provide a feasible and near optimal solution.

In the second part, we will present numerical examples, an industrial application, and an extensive comparison of the hierarchical and monolithic approaches in terms of memory requirements and computational complexity.

References

[1] A. C. Hax and H. C. Meal, "Hierarchical integration of production planning and scheduling," in *Studies in the Management Sciences* (M. A. Geisler, ed.), Logistics, North Holland - American Elsevier, 1975.

[2] C. C. Holt, F. Modigliani, J. F. Muth, and H. A. Simon, *Planning Production, Inventories and Work Force.* New Jersey: Prentice Hall, 1960.

[3] P. R. Winters, "Constrained inventory rule of production smoothing," *Management Sciences*, vol. 8, no. 4, pp. 470–481, 1962.

[4] G. R. Bitran and A. C. Hax, "On the design of hierarchical production planning systems," *Decision Sciences*, vol. 8, pp. 28–55, 1977.

[5] G. R. Bitran and A. C. Hax, "Disaggregation and resource allocation using convex knapsack problems with bounded variables," *Management Science*, vol. 27, no. 4, pp. 431–441, 1981.

[6] J. J. Golovin, "Hierarchical integration of planning and control," Tech. Rep. 116, Operations Research Center, Cambridge, M. I. T., 1975.

[7] H. Gabbay in *Optimal Aggregation and Disaggregation in Hierarchical Planning*, pp. 95–106, Boston: Martinus Nijhof, 1979.

[8] G. R. Bitran, E. A. Haas, and A. C. Hax, "Hierarchical production planning: A single stage system," *Operations Research*, vol. 29, no. 4, pp. 717–743, 1981.

[9] J. Erschler, G. Fontan, and C. Merce, "Consistency of the disaggregation process in hierarchical planning," *Operations Research*, vol. 34, no. 3, pp. 464–469, 1986.

[10] G. R. Bitran, Haas., E. A., and H. Matsuo, "Production planning of style goods with setup costs and forecast revisions," *Operations Research*, vol. 32, no. 2, pp. 226–236, 1986.

[11] R. P. Mohanty and R. V. Kulkarni, "Hierarchical production planning: Comparison and some heuristics," *Engineering Costs and Production Economics*, vol. 11, pp. 203–214, 1987.

[12] S. C. Graves, "Using lagrangian techniques to solve hierarchical production planning problems," *Management Sciences*, vol. 28, no. 3, 1982.

[13] K. Aardal and T. Larsson, "A benders decomposition based heuristic for the hierarchical production planning problem," *European Journal of Operational Research*, vol. 45, pp. 4–14, 1990.

[14] G. R. Bitran, E. A. Haas, and A. C. Hax, "Hierarchical production planning: A two stage system," *Operations Research*, vol. 30, no. 2, pp. 232–251, 1982.

[15] R. Nagi, *Design and Operation of Hierarchical Production Management Systems*. PhD thesis, University of Maryland, College Park, 1992.

[16] G. Harhalakis, A. Mehra, R. Nagi, and J.-M. Proth, "Temporal aggregation and disaggregation in production planning," Tech. Rep. 1957, INRIA-Lorraine, France, 1993.

[17] S. Axsater, "On the design of the aggregate model in a hierarchical production planning system," *Engineering and Process Economics*, vol. 4, pp. 89–97, 1979.

[18] S. Axsater, "Aggregation of product data for hierarchical production planning," *Operations Research*, vol. 29, no. 4, pp. 744–756, 1981.

[19] S. Axsater, "On the feasibility of aggregate production plans," *Operations Research*, vol. 24, no. 5, pp. 796–800, 1986.

[20] H. Tsubone, H. Matsuura, and T. Tsutsu, "Hierarchical production planning system for a two-stage process," *International Journal of Production Research*, vol. 29, no. 4, pp. 769–785, 1991.

[21] G. H. Saad, "Hierarchical production-planning systems: Extensions and modifications," *OR; The Journal of the Operational Research Society*, vol. 41, no. 7, pp. 609–624, 1990.

[22] S. D. Thompson and W. J. Davis, "An integrated approach for modeling uncertainty in aggregate production planning," *IEEE Transactions on Systems, Man and Cybernetics*, vol. 20, no. 5, pp. 1000–1012, 1990.

[23] S. D. Thompson, D. T. Watanabe, and W. J. Davis, "A comparative study of aggregate production planning strategies under conditions of uncertainty and cyclic product demands," *International Journal of Production Research*, vol. 31, no. 8, pp. 1957–1979, 1993.

[24] R. R. Inman and P. C. Jones, "Decomposition for scheduling flexible manufacturing systems," *Operations Research*, vol. 41, no. 3, pp. 608–617, 1993.

[25] T. S. Arthanari and Y. Dodge, *Mathematical Programming in Statistics*. New York: John Wiley and Sons, 1981.

[26] J. A. Hartigan, *Clustering Algorithms*. New York: John Wiley and Sons, 1975.

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 - 2633