TECHNICAL RESEARCH REPORT

Temporal Aggregation in Production Planning

by G. Harhalakis, A. Mehra, R. Nagi, and J.M. Proth

T.R. 93-54

The Institute for Systems Research is supported by the National Science Foundation Engineering Research Center Program (NSFD CD 8803012), the University of Maryland, Harvard University, and Industry

TEMPORAL AGGREGATION IN PRODUCTION PLANNING

G. Harhalakis[1], A. Mehra[1], R. Nagi[1], J.M. Proth[2]

[1]Department of Mechanical Engineering and Institute for Systems Research,
University of Maryland, College Park, MD 20742, U.S.A.
[2]INRIA, 4, rue Marconi, Technopole Metz 2000, 57070 METZ, France, and
Institute for Systems Research, University of Maryland, U.S.A.

ABSTRACT

In this paper, the problem of temporal aggregation in production planning is addressed. A single facility with multiple part types is considered. The planning horizon consists of a sequence of elementary time periods, and the demand for all part types is assumed to be known over these periods. The production planning problem consists of minimizing the holding and backlogging cost for all part types. Due to usual errors in demand forecasting, and due to the large size of the linear programming problem commonly encountered in such problems, there is a need for aggregating the production variables over the time horizon (typically, from weekly to monthly) to result in a *hierarchical structure*. We consider a two-level hierarchy composing a sub-problem at each level, and we propose an iterative technique which solves these sub-problems in sequence. *A posteriori* bounds are developed, which are useful in evaluating the performance of the iterative algorithm. Quick lower and upper bounds of the original problem are also developed. Finally, numerical results for numerous test cases are presented.

Key words: Hierarchical planning, Temporal aggregation, Early/Tardy problem, Modeling, Inventory.

1 Introduction

Multi-product production planning is a simple problem to formulate, but a difficult problem to solve optimally. It consists of deciding the quantities of products to manufacture in a given sequence of elementary periods in order to optimize given criteria. Constraints such as capacity limitations have to be taken into account. This problem is further complicated by the uncertainty and fluctuations of demand forecasts. Two distinct approaches to production planning have been adopted in the past: (i) the monolithic approach, wherein the entire problem is formulated as a large LP (linear programming) problem, and (ii) the hierarchical approach, which partitions the global problem into a series of sub-problems that are solved sequentially. The advantage of using the hierarchical approach is the reduction in the complexity of the problem, as well as the gradual absorption of random events; for other advantages of hierarchical approaches see Dempster, Fisher *et al.* 1981. Hierarchical approaches have been suggested in the literature by Hax and Meal (1975), Bitran and Hax (1977, 1981), Graves (1982), Aardal *et al.* (1990), Tsubone *et al.* (1991), that consider production planning at different levels of the management hierarchy; they consider different criteria and cost structures. For instance, Nagi (1991) and Meier (1989) consider minimizing the losses due to holding and backlogging inventory costs.

The issue in hierarchical decompositions is how can one develop hierarchical models that provide good approximations to the respective monolithic problems. In multi-layer structures related to hierarchical management systems, the controller is decomposed into algorithms operating at different time intervals. All the layers of the controller directly affect the process but the higher ones control its slower aspects only: they intervene less frequently, they have longer optimization horizons, and are based on more aggregate models. In this paper, we address the problem of temporal aggregation that results in the hierarchical design process while going from a lower level of the hierarchy to the next higher level; this is due to their need to be operating at different time scales. On the other hand, we also need to consider the temporal disaggregation that results in the computation of a production plan from top to bottom; there is a gradual refining of the production plan along time as we go from higher levels to the lowest level.

We consider the production planning problem of determining the production levels of parts to be produced in a single facility over a given horizon which is discretized into elementary periods, given the demand requirements. The objective is to minimize the inventory holding and backlogging costs. A two-level hierarchy is constructed, the high level, which plans production for aggregate product entities over sub-periods (aggregate elementary time periods) composing a long horizon, and the low level, which disaggregates the high level production plan to that of detailed product entities over detailed elementary time periods composing a shorter horizon. The difficulty in employing this approach is that there is loss of detailed inventory information (eg. inventory cost information) at the high level, hence, this could result in significant loss of optimality. Thus, the inventory information at the low level is incorporated at the high

level in the form of "weights"; this is a standard practice while considering aggregate problems. However, good values for these weights can only be obtained once the detailed level problem is solved. Hence, we develop iterative schemes that solve the two problems iteratively, to converge to a good solution. This two-level structure can be employed between any two levels of a general n-level hierarchy. While the aggregation of parts is performed in the manner suggested by Nagi (1991), the aggregation of elementary periods into sub-periods of equal duration, and the corresponding disaggregation of production levels, constitute the major contribution of this work. The performance evaluation of this scheme is presented, and a number of error bounds are developed to further assess the "goodness" of this scheme.

This paper is organized as follows. In section 2, the literature survey is presented. In section 3, we describe the problem and formulate it. In section 4, we detail the aggregation scheme used and the design of a two-level hierarchy to solve the problem presented in the section 3. In section 5, an iterative algorithm employed to solve the problem is presented. Section 6 is devoted to the proof of convergence of the aforementioned algorithm. Section 7 describes another iterative algorithm that provides near optimal solutions, and is based on modifications to the constraints of the LP problems of the previous algorithm. Section 8 consist of the formulation of *a posteriori* error bounds for the proposed aggregation/disaggregation scheme and quick bounds (both upper and lower). Numerical results are presented in section 9. Finally, conclusions of this temporal aggregation methodology are drawn in section 10.

2 Literature Survey

Many techniques are used for forming hierarchical planning models, and for assessing their effectiveness with respect to recovering the optimal solution of the original problem. Rogers *et al.* (1991) identified three major components of the aggregate model, which are: (i) aggregation analysis, (ii) disaggregation analysis and (iii) error analysis - methods of determining the resulting error that may be introduced by employing aggregate models and disaggregate solutions. These three main elements of aggregation/disaggregation and their components are discussed throughout this paper. In this section, we present literature surveyed in the areas of hierarchical production planning systems, temporal aggregation, iterative procedures to aggregate problems and error analysis.

2.1 Hierarchical Production Planning Systems

Hax and Meal (1975) performed some pioneering work in the area of hierarchical production planning. They considered a multi-plant firm with four decision layers. The highest layer model performs a static (one time) distribution of products to individual plants. The next three decision layers are intended for the management of a single plant. The concept of a rolling horizon is employed, in order to perform repetitive open-loop optimizations. Bitran and Hax (1977), and Bitran *et al.* (1981) further refined Hax and Meal

hierarchical production planning methodology for a single stage production system. Bitran *et al.* (1982) extended the methodology to a two-stage process. These models considered three layers of aggregation for products: (i) *items*, (ii) *families*, and (iii) *types*. Graves (1982), adopts a different approach to the problem and introduces feedback between decision layers. Based on the product aggregation scheme of Hax and Meal, the problem is formulated as a monolithic mixed integer program, which is decomposed by Lagrangean relaxation. The best values of the Lagrangean multipliers are determined by an iterative procedure which may be interpreted as a feedback mechanism in the Hax-Meal framework. Aardal and Larsson (1990) present another hybrid approach to the mixed integer problem suggested by Graves (1982): the Hax and Meal model and its extensions are based on a typical cost structure, and the proposed aggregation schemes are relevant to only a particular class of production systems. The model does not take into account randomness in demand, capacity or any other production variables.

Axsater (1981), Mizrach (1985), Hillion, Meier and Proth (1987), Meier (1989) and Nagi (1991) address a double aggregation scheme over products and machines, where at the top layer the entities of relevance are machine sub-systems and part families. This class of literature does not address temporal aggregation issues in their formulations. Other related work in the area of hierarchical production planning is as follows: Mohanty and Kulkarni (1987) present a heuristic approach to the hierarchical production planning problem which performs better for a certain category of objective functions. Tsubone *et al.* (1991) present a new design of a two-stage hierarchical model by analyzing the relationship between the production planning rules and the role of the buffer inventory level in terms of system performance. Boskma (1982) perform simulation studies, and Axsater *et al.* (1984) present empirical evidence in support of hierarchical production planning.

Consistency between decisions made at different levels is typically ensured by taking into account the optimal solution obtained at a given level when computing the optimal solution at the next lower level. However, sometimes the disaggregation is infeasible, i.e. top level constraints may yield an empty feasible set at a lower level. Gabby (1975), Saad (1990) and Erschler *et al.* (1986) derive necessary and sufficient conditions for the consistent disaggregation of the Hax and Meal model. The sub-optimality issues have been addressed by Dempster, Fisher *et al.* (1981). Krajewski and Ritzman (1977) provide a survey of the problems and research in the area of the infeasibilities of disaggregation.

2.2 *Temporal Aggregation*

To clarify the term, an example of temporal aggregation in a two level hierarchical production planning structure is shown in Figure 2.1. The high level plans monthly production for all product entities over the entire planning horizon (for example - fifteen months), and the low level disaggregates the monthly results to a weekly plan for the product entities for the first month.

There exists very little literature on the temporal aspect of aggregation in hierarchical production planning systems. Abraham *et al.* (1985) have suggested the need of temporal aggregation in production planning and control of manufacturing systems, and Nagi (1991) has indicated the need for a temporal aggregation scheme, where similar input variables (product demand) are aggregated on a time scale. So far, to our knowledge, a general temporal aggregation scheme for production planning systems has not been addressed in the literature.

In view of the lack of literature on the temporal aspect of aggregation in production planning systems, we sought out some information by reviewing the application of the temporal aggregation in other problem domains. These applications are detailed below.

Engles, Larson *et al.* (1976) consider the hydro and thermal generation scheduling problem using an iterative dynamic programming procedure. The output of the monolithic problem is the hourly resource allocation for a scheduling horizon of at least one year. A two-layer hierarchical structure is considered; the high layer which schedules all resources weekly over one year, and the lower layer which disaggregates the weekly results to schedule the resources hourly for one week. These low and high layer horizons are selected because the demand is characterised by weekly cycles and the maintainence of resources and the decisions of acquiring a new resource are made yearly. The technique of successive approximations is used, whereby each resource is rescheduled at the low layer, and then the high layer problem is solved. Since the objective function is convex, the iterative procedure converges to the optimum. However, this procedure is problem specific, and cannot be applied or extended easily to general hierarchical production planning problems.

Monts (1991) proposes a temporal aggregation scheme to evaluate the marginal energy cost ($/mega watt-hour). The horizon for analyzing the given hourly marginal cost was set to one year. Due to the temporal structure of the marginal costs, aggregation along a time scale was performed according to the type of season (Winter, Spring, Summer or Fall) and the day type (weekday or weekend). This aggregation of the hourly marginal cost is not static in nature, and depends on the variance of the hourly marginal cost over a day. Both, two-layer and three-layer aggregation schemes are proposed, based on the hourly marginal cost variation. Monts's scheme is also problem specific, and does not always lead to the optimal solution.

Wei and Mehta (1980) addressed the information loss due to temporal aggregation in a dynamic system where the influence of an input variable on an output variable is distributed over several time periods. These models are known as distributed lag models. A proper time scale for the temporal aggregation is proposed, based on the time horizon of the aggregated information required. A Monte Carlo study on some commonly used forecasting least square estimators was presented, and the information loss in terms of the prediction of the input variables was examined.

Cunningham *et al.* (1992) examine the effect of temporal aggregation on money and interest rates. Aggregation was performed based on time averaging of the input variables, which was observed to cause a significant magnitude change in the output variables. Rossana *et al.* (1992) consider the modeling of real wages of the manufacturing industry, and present the effect of temporal aggregation on the output variables.

2.3 Iterative Aggregation Procedures

Dudkin *et al.* (1987) study in detail several iterative procedures for solving systems of equations of both constrained and unconstrained optimization models. These models were developed for large-scale economic planning models. At each iteration, the low level executes the decisions that differ from the high level plans, if doing so benefits the low level, i.e., the high level decisions are not obligatory to follow at the low level. Numerous two level iterative algorithms are discussed, along with proofs and rates of convergence for some simple cases.

Zipkin (1977, 1980a, b) explores the effect of aggregating variables in large linear problems. He employed an iterative technique to improve solutions obtained by given aggregation/ disaggregation problems. An arbitrary partitioning of the original set of variables of a large linear problem is performed and the columns of each group are replaced by their weighted averages. The aggregate problem is solved, and the solution obtained is disaggregated to yield a feasible solution to the original problem. New weights are then computed from the *disaggregate* solution, and the aggregate problem is solved again. Weak convergence of this iterative procedure to the optimal solution is proven assuming no degeneracy and including a successive approximation technique at the last step of the algorithm. The problem of degeneracy is also discussed.

Nagi (1991) has proposed an iterative procedure for the production planning problem where aggregation is performed for parts with similar cost attributes. The principle of Nagi's algorithm is similar to that of Zipkin`s disaggregation procedure. The proof of convergence has been provided for this procedure. Along similar lines, we are adopting iterative aggregation procedures for the temporal aggregation case, where we assume values of weights at the high level, and then improve these values by iteratively solving the LP problems.

2.4 Error Analysis

While employing a hierarchical approach, the information loss due to aggregation/disaggregation procedure can lead to sub-optimality. Error analysis helps in assessing this error and verifying the applicability of a particular aggregation scheme. In general, two types of error bounds are often associated with the aggregation/disaggregation procedure. These bounds are (Rogers *et al.*, 1991):

(i) *a priori error bounds* - these error bounds are estimated after an aggregate model has been formulated but not solved; *a priori* analysis is comparable to the worst case analyses for heuristics.

(ii) *a posteriori error bounds* - these error bounds are estimated after an aggregate model has been formulated and solved; *a posteriori* analysis is comparable to linear programming relaxation or Lagrangean relaxation methods.

The goodness of the bounds can be studied by employing monolithic models of reasonable sizes, and comparing the actual error with these error bounds. Typically, *a priori* bounds are much looser than *a posteriori* bounds. Zipkin (1977, 1980b) proposes *a posteriori* bounds for large linear aggregation/disaggregation optimization problems. Zipkin provides a tighter bound, compared to one given by Whitt (1978, 1979). However, Zipkin's bounds need additional information about the upper bound of the weighted sum of the optimal variables in a particular subset of the partition. Zipkin's *a posteriori* bound also needs the dual solution of the aggregate problem. In this paper, Zipkin's *a posteriori* bound estimation method for LP problems is used to evaluate the *a posteriori* bound of the proposed temporal aggregation model.

3 Problem Description and Formulation

3.1 Problem Description

We consider a unique facility. The set $P = \{p_1,p_2,\ldots,p_n\}$ represents n part types to be manufactured. Let t_i represent the processing time of one unit of part type p_i; $i = 1,2,\ldots,n$. Set-up times are not considered. We consider a planning horizon H composed of Z sub-periods which are in turn divided into z elementary periods each; these elementary periods are of duration Δ each. It is assumed that the duration of Δ is much larger than the processing times of parts, i.e. $t_i << \Delta$; $i = 1,2,\ldots,n$. Let d_{ik}, x_{ik} and s_{ik} represent respectively the number of units of part type p_i required at the end of the k-th elementary period, the planned production for part type p_i during the k-th elementary period and the inventory of part type p_i at the end of the k-th elementary period after satisfying the demand, $i = 1,2,\ldots,n$; $k = 1,2,\ldots,Zz$.

We consider the minimization of the total inventory holding and backlogging costs as our criterion. Let w_i^+ represent the cost associated with holding one unit of part type p_i in stock for one elementary period. Similarly, let w_i^- represent the cost associated with backlogging one unit of part type p_i by one elementary period; $i = 1,2,\ldots,n$.

3.2 Problem Formulation

The production planning problem consists of determining the production x_{ik} in order to minimize the total inventory holding and backlogging costs.

The problem can be formally stated as the following linear programming problem:

(Problem P1) Minimize: $\sum_{k=1}^{Zz} \sum_{i=1}^{n} f_i(s_{ik})$ (3.2.1)

where:

$$f_i(s_{ik}) = w_i^+[s_{ik}]^+ + w_i^-[-s_{ik}]^+; \qquad i = 1,2,\ldots,n, \text{ and } k = 1,2,\ldots,Zz \tag{3.2.2}$$

$$[\bullet]^+ = \text{Max}\{0,\bullet\};$$

$$s_{ik} = s_{i0} + \sum_{a=1}^{k} x_{ia} - D_{ik}; \qquad i = 1,2,\ldots,n, \text{ and } k = 1,2,\ldots,Zz \tag{3.2.3}$$

$$D_{ik} = \sum_{a=1}^{k} d_{ia}; \qquad i = 1,2,\ldots,n, \text{ and } k = 1,2,\ldots,Zz \tag{3.2.4}$$

Subject to:

$$\sum_{i=1}^{n} t_i x_{ik} \le \Delta; \qquad k = 1,2,\ldots,Zz \tag{3.2.5}$$

$$x_{ik} \ge 0; \qquad i = 1,2,\ldots,n, \text{ and } k = 1,2,\ldots,Zz \tag{3.2.6}$$

$f_i(y)$ is the cost function for part type p_i and inventory level y. Here, we have considered holding and backlogging costs in the cost function (3.2.2). s_{i0} denotes the initial inventory of part p_i (3.2.3). D_{ik} represents the cumulative demand of the part type p_i till the end of k-th elementary period (3.2.4). We can equivalently formulate problem P1 with zero initial inventory and a corresponding effective demand (see Nagi, 1991). Thus, from here on, without loss of generality, we shall assume that the initial inventory is zero for all part types, with demands being transformed in order to take into account the initial inventories. The inventory relation can then be written as follows:

$$s_{ik} = \sum_{a=1}^{k} x_{ia} - D_{ik}; \tag{3.2.3}$$

Constraint (3.2.5) represents the capacity lilits over each elementary period. Constraint (3.2.6) implies the non-negativity of production.

We can transform the LP problem P1 to an equivalent problem where the parts have unity processing times and the other variables and constants are transformed appropriately as follows:

$$x'_{ik} = x_{ik} \times t_i ; \tag{3.2.7}$$

$$D'_{ik} = D_{ik} \times t_i ; \tag{3.2.8}$$

$$w'^{+(-)}_i = w^{+(-)}_i / t_i ; \tag{3.2.9}$$

The new equivalent LP problem, whose criterion value will be the same as in the original problem, can then be formulated as follows:

$$\text{Minimize: } \sum_{k=1}^{Zz} \sum_{i=1}^{n} w'^{+}_i \times [\sum_{a=1}^{k} x'_{ia} - D'_k]^+ + w'^{-}_i \times [D'_{ik} - \sum_{a=1}^{k} x'_{ia}]^+ \tag{3.2.10}$$

Subject to:

$$\sum_{i=1}^{n} x'_{ik} \le \Delta ; \qquad k = 1,2,\ldots,Zz \tag{3.2.11}$$

$$x'_{ik} \ge 0 ; \qquad i = 1,2,\ldots,n, \text{ and } k = 1,2,\ldots,Zz \tag{3.2.12}$$

This LP problem is solved for x'_{ik} and the x_{ik} variables can easily be obtained from equations (3.2.7-3.2.9) above. From here on we will use only the equivalent problem in our formulations.

4 Hierarchical Approach

The hierarchical approach is based on grouping parts into part families and elementary periods into sub-periods. Aggregation of parts into part families is performed in a way that parts are grouped in a family if they have the same holding costs, and the same backlogging costs. This aggregation scheme is defined as "perfect aggregation" in Nagi (1991). The set of N part families is represented by $F = \{f_1, f_2, \ldots, f_N\}$; note that, $N \le n$. Two parts p_i and $p_j \in f_r$, $r \in \{1,2,\ldots,N\}$, iff:

(i) $w^{+}_i = w^{+}_j$

(ii) $w^{-}_i = w^{-}_j$

Under this aggregation scheme, v^{+}_r represents the cost associated with holding one unit of family type f_r in stock for one elementary period and v^{-}_r represents the cost associated with backlogging one unit of family type f_r by one elementary period; r = 1,2,...,N. We can relate these new parameters as follows:

$$v^{+(-)}_r = w^{+(-)}_j, \qquad \text{if } j | p_j \in f_r$$

We also define a function a(i), where i = 1,2,...,n, which represents the family $f_{a(i)}$ to which part type p_i has been assigned; also $a(i) \in \{1,2,\ldots,N\}$. Similarly, for time periods, we define a function b(k), where k = 1,2,...,Zz, which represents the sub-period to which the k-th elementary period belongs; also $b(k) \in \{1,2,\ldots,Z\}$. Note that $b(k) = \lceil k/z \rceil$, where $\lceil \bullet \rceil$ denotes the smallest integer greater than or equal to $\bullet$.

The two levels of the hierarchy are constructed as follows: (i) the high level that plans production for part families in sub-periods, and (ii) the low level that computes the disaggregation of the high level

solution to determine the production plan for parts types in elementary periods. The form of the objective function are the same at the high and the low level. The disaggregation is performed over a short term horizon, z ($z < H$), in a manner that the high level constraints are respected and the value of the objective function does not increase over z. Production planning is usually performed on a rolling horizon basis. That is, although the high level solution is computed over H, only a part of it (over z) is implemented, followed by a recomputation of the plan after z. This is done in order to incorporate the future demands/forecasts progressively.

An iterative algorithm is developed which solves the high and the low level problems at each iteration. The convergence of this algorithm is proved in section 6.

4.1 High Level Problem

The high level problem consists of production planning for part families on sub-periods over the entire planning horizon (H). Let $X_{r\kappa}$ denote the production planned for part family f_r during the κ-th sub-period; $r = 1,2,\ldots,N$; $\kappa = 1,2,\ldots,Z$. The production planning problem is to determine the production $X_{r\kappa}$ such that the backlogging and holding costs for families, still computed at the end of each elementary period, are minimized.

The inventory of the part family f_r at the end of the k-th elementary period, after satisfying the demand of the part types corresponding to this family, is denoted by σ_{rk}.

For the first sub-period we define g_{ik} as the ratio of the cumulative production of part type p_i till the k-th elementary period, to the production of the part family $f_{a(i)}$ over the entire sub-period. These ratios or weights are defined in order to take into account low level information at the high level. Thus g_{ik} is defined for only elementary periods in the first sub-period as follows:

$$g_{ik} = \begin{cases} \dfrac{\sum_{s=1}^{k} x_{is}^{\sim}}{X_{a(i)1}^{\sim}}\ ; & \text{if } X_{a(i)1}^{\sim} > 0 \\ 0\ ; & \text{otherwise} \end{cases} \qquad (4.11)$$

Notation '~' stands for the previous iteration, i.e., $x_{is}^{\sim}$ means the low level production obtained in the previous iteration, and $X_{a(i)1}^{\sim}$ means the high level production of part family a(i) for the first sub-period, obtained in the previous iteration. For the first iteration, initial values of g_{ik} can be assumed or computed according to the procedure detailed in section 5.

For the subsequent sub-periods (i.e., $\kappa = 2,3,...,Z$) production of part families is divided equally among all elementary periods composing that sub-period (i.e., for $k=(\kappa-1)z+1,...,\kappa z$). This is referred to as linear production of part families within these sub-periods.

The high level problem can be formally stated as the following linear programming problem:

$$\textbf{(Problem P2)} \quad \text{Minimize: } \sum_{\kappa=1}^{Z} \sum_{k=z(\kappa-1)+1}^{z\kappa} \sum_{r=1}^{N} \Phi_r(\sigma_{rk}) \tag{4.1.2}$$

where:

$$\Phi_r(\sigma_{rk}) = v_r^+ \times [\sigma_{rk}]^+ + v_r^- \times [-\sigma_{rk}]^+ ; \qquad r = 1,2,...,N \tag{4.1.3}$$

$$\sigma_{rk} = X_{r1} \times \sum_{il p_i \in f_r} g_{ik} - \sum_{il p_i \in f_r} D_{ik} ; \qquad k = 1,2,...,z \tag{4.1.4}$$

$$\sigma_{rk} = \sum_{\kappa=1}^{b(k)-1} X_{r\kappa} + X_{r\kappa} \times \left[\frac{k-(b(k)-1)z}{z}\right] - \sum_{il p_i \in f_r} D_{ik} ; \qquad k = z+1,z+2,...,Zz \tag{4.1.5}$$

Subject to:

$$\sum_{r=1}^{N} X_{r\kappa} \le z\,\Delta ; \qquad \kappa = 2,3,...,Z \tag{4.1.6}$$

$$\sum_{r=1}^{N} X_{r1} \left(\sum_{il p_i \in f_r} g_{ik} - \sum_{il p_i \in f_r} g_{i(k-1)} \right) \le \Delta ; \qquad k = 1,2,...,z \tag{4.1.7}$$

$$X_{r\kappa} \ge 0 ; \qquad r = 1,2,...,N, \text{ and } \kappa = 1,2,...,Z \tag{4.1.8}$$

The cost function for part family f_r in the k-th elementary period is represented by $\Phi_r(y)$, where y is its inventory level. The objective function that represents the minimization of the inventory holding and backlogging costs for the part families over the entire horizon is represented in (4.1.2). The relation (4.1.3) defines the cost function for part families. Relation (4.1.4) provides the inventory level of part families for the first z elementary periods, i.e., elementary periods belonging to first sub-period. Relation (4.1.5) provides the inventory level of part families for the elementary periods with $k > z$.

Constraint (4.1.6) represents the capacity limits in each sub-period. Constraint (4.1.7) ensures that the solution of the high level guarantees that the set of feasible solutions at the low level is non-empty (see section 6 for the proof of convergence) and (4.1.8) implies the non-negativity of production for part families.

4.2 Low Level Problem

The low level problem consists of production planning for part types on elementary periods of the first sub-period. It determines the production volume x_{ik} for part type p_i from the aggregate high level solution $X_{a(i)1}$ in order to minimize the backlogging and holdings costs of all part types over the first sub-period.

The low level problem can be formally stated as the following linear programming problem:

(Problem P3) Minimize: $\sum_{k=1}^{z} \sum_{i=1}^{n} f_i(s_{ik})$ (4.2.1)

Subject to:

$$X_{r1} = \sum_{k=1}^{z} \sum_{i|p_i \in f_r} x_{ik} \; ; \qquad r = 1,2,\ldots,N \tag{4.2.2}$$

$$\sum_{i=1}^{n} x_{ik} \leq \Delta \; ; \qquad k = 1,2,\ldots,z \tag{4.2.3}$$

$$\sum_{k=1}^{z} x_{ik} = D_{iz} + \alpha_i \left(X_{a(i)1} - \sum_{j|p_j \in f_{a(i)}} D_{jz}\right) ; \qquad \forall\, i \,|\left(X_{a(i)1} - \sum_{j|p_j \in f_{a(i)}} D_{jz}\right) > 0 \tag{4.2.4}$$

$$x_{ik} \geq 0 \; ; \qquad i = 1,2,\ldots,n, \text{ and } k = 1,2,\ldots,z \tag{4.2.5}$$

Constraint (4.2.2) ensures that the high level production plan for part families is respected. Constraint (4.2.3) reflects the capacity limits over each elementary period. Constraint (4.2.4) ensures that the surplus production of part-families at the end of the first sub-period is assigned to individual part types according to the ratios α_i. The ratios α_i are determined according to either the expected long-term future demand or the demand in the subsequent elementary period(s) (if known), because, this surplus is intended to cater for the future demand. Note that

$$\sum_{i|p_i \in f_r} \alpha_i = 1; \qquad \forall r \,|\left(X_{r1} - \sum_{j|p_j \in f_{a(i)}} D_{jz}\right) > 0. \tag{4.2.6}$$

We do not specify such ratios for part families which are backlogged at the end of first sub-period because the decision to split this deficit among the part types does not affect the cost function for future periods. At this point, we also show that constraint (4.2.2) is redundant for $\forall r \,|\left(X_{r1} - \sum_{j|p_j \in f_{a(i)}} D_{jz}\right) > 0$.

Consider constraint (4.2.4) summed over parts belonging to the same part family:

$$\sum_{j|p_j \in f_{a(i)}} \sum_{k=1}^{z} x_{jk} = \sum_{j|p_j \in f_{a(i)}} D_{jz} + \sum_{j|p_j \in f_{a(i)}} \left(\alpha_j \left(X_{a(i)1} - \sum_{s|p_s \in f_{a(i)}} D_{sz}\right)\right)$$

After simplifying we obtain:

$$\sum_{k=1}^{z} \sum_{j|p_j \in f_{a(i)}} x_{ik} = X_{a(i)1}$$

The above constraint is the same as (4.2.2) which makes it redundant for $\forall r \,|(X_{r1} - \sum_{j|p_j \in f_{a(i)}} D_{jz}) > 0$.

5 Iterative Algorithm

In this section, we present an algorithm which solves problems P2 and P3 iteratively to obtain the production levels of part types for the elementary periods (k = 1,2,...,z). We initialize the weights g_{ik} by solving the low level problem (P3) without constraints (4.2.2) and (4.2.4). Production of part families in the first sub-period (X_{r1}) is estimated by adding the production of all part types in the part family for the first z elementary periods. The high level is then solved, assuming the weights obtained from relation (4.1.1). The production of part families in the first sub-period is then disaggregated at the low level. The new weights are then computed, and the two levels are solved again. The process is repeated until convergence is achieved. The algorithm is detailed as follows:

Algorithm 1

Step 1. Estimate the effective demand from the forecasted demand pattern.

Step 2. Transform the original problem to the equivalent LP problem with unity processing time.

Step 3. Aggregate part types into part families based on the aggregation scheme described in section 4, and determine cost attributes.

Step 4. Compute the cumulative demand D_{ik} (equation 3.2.4).

Step 5. Solve the low level problem (P3) for x_{ik} without constraints (4.2.2) and (4.2.4).

Step 6. Estimate the production of part families from the relation 4.2.2.

Step 7. Initialize the iteration number c; c = 0.

Step 8. c=c+1.

Step 9. Initialize the weights g_{ik}^c from the production of part types and part families using the relation (4.1.1); where g_{ik}^c represents the weights in iteration c.

Step 10. Solve the high level problem (P2) for the production of part families in the sub-periods, using the current value of the weights g_{ik}^c.

Step 11. Disaggregate part family production over the first sub-period, by solving the low level problem (P3) to determine x_{ik}; i = 1,...,n and k = 1,...,z.

Step 12. Check for the reduction of the high level criterion value, with respect to the high level criterion value of the previous iteration. If the reduction is greater than a user specified ε ($\varepsilon > 0$ and sufficiently small), then go back to step 8. Otherwise, STOP; the production values of the part types obtained at the last iteration is the solution to the problem.

6 Convergence

In this section the convergence of the iterative algorithm 1 presented in section 5 is proved.

Let $P2^c$ and $P3^c$ represent respectively the high level and low level problems at iteration c. Let $C(\phi, s)$ represent the criterion value corresponding to the feasible solution s of problem ϕ. Consider the division of the criterion value $C(P2^c, s)$ into two parts: (i) $C_1(P2^c, s)$, which represents the part of criterion value corresponding to the first sub-period (i.e. κ=1), and (ii) $C_2(P2^c, s)$, which represents the part of criterion value corresponding to sub-periods κ = 2,3,...,Z. Note that $C(P2^c, s) = C_1(P2^c, s) + C_2(P2^c, s)$. Let $X^c = (X_{r\kappa}{}^c;\ r = 1,2,...,N \text{ and } \kappa = 1,2,...,Z)$ in an $N \times Z$ solution vector of problem $P2^c$. Let $x^c = (x_{ik}{}^c;\ i = 1,2,...,n \text{ and } k = 1,2,...,z)$ in an $n \times z$ solution vector of problem $P3^c$.

We show that the solution of each problem $P2^c$ and $P3^c$ guarantees that the set of feasible solutions for the subsequent problem (i.e., $P3^c$ and $P2^{c+1}$), respectively is non-empty. We also determine, corresponding to the optimal solution: (i) X^{*c} of $P2^c$, a feasible solution x^c of $P3^c$ such that $C_1(P2^c, X^{*c}) = C(P3^c, x^c)$, and (ii) x^{*c} of $P3^c$, a feasible solution X^{c+1} of $P2^{c+1}$ such that $C(P3^c, x^{*c}) = C_1(P2^{c+1}, X^{c+1})$.

Property 1.

According to the result of (Nagi, 1991), in the perfect aggregation case, a property of the optimal solution of P1 is such that for any part type p_i and elementary period k:

- if $s^*_{ik} \geq 0 \Rightarrow s^*_{jk} \geq 0 \quad \forall j \mid p_j \in f_{a(i)}$
- if $s^*_{ik} \leq 0 \Rightarrow s^*_{jk} \leq 0 \quad \forall j \mid p_j \in f_{a(i)}$

Furthermore, it is proved that:

- if $\sigma^*_{rk} \geq 0 \Rightarrow s^*_{jk} \geq 0 \quad \forall j \mid p_j \in f_r$
- if $\sigma^*_{rk} \leq 0 \Rightarrow s^*_{jk} \leq 0 \quad \forall j \mid p_j \in f_r$

Lemma 1. *The optimal solution of problem* $P3^c$ *and the new weights computed thereof,* $g_{ik}{}^{c+1}$, *guarantee that the set of feasible solutions of problem* $P2^{c+1}$ *is non-empty.*

Let $x^*_{ik}{}^c$ (i = 1,2,...,n; k = 1,2,...,z) be the optimal solution of $P3^c$. Then, according to (4.1.1):

$$g_{ik}{}^{c+1} = \begin{cases} \dfrac{\sum_{s=1}^{k} x^*_{is}{}^c}{X^*_{a(i)1}{}^c}; & \text{if } X^*_{a(i)1}{}^c > 0 \\ 0; & \text{otherwise} \end{cases} \qquad (6.1)$$

We show that the set of feasible solutions of $P2^{c+1}$ is non-empty by verifying that $X^*_{r\kappa}{}^c$ ($r = 1,2,...,N$ and $\kappa = 1,2,...,Z$) is a feasible solution of problem $P2^{c+1}$ with the new weights $g_{ik}{}^{c+1}$. Constraints (4.1.6) and (4.1.8) are verified, as they are satisfied by the solution of problem $P2^c$. We now verify constraint (4.1.7) as follows:

$$\sum_{r=1}^{N} \sum_{il p_i \in f_r} X^*_{r1}{}^c \, (g_{ik}{}^{c+1} - g_{i(k-1)}{}^{c+1}) \le \Delta \; ; \quad k = 1,2,\ldots,z$$

By substituting for $g_{ik}{}^{c+1}$ and $g_{i(k-1)}{}^{c+1}$ from (6.1) we have:

$$\sum_{r=1}^{N} \sum_{il p_i \in f_r} X^*_{r1}{}^c \left(\frac{\sum_{s=1}^{k} x^*_{is}{}^c}{X^*_{r1}{}^c} - \frac{\sum_{s=1}^{k-1} x^*_{is}{}^c}{X^*_{r1}{}^c} \right) \le \Delta$$

After simplifying, we obtain:

$$\sum_{r=1}^{N} \sum_{il p_i \in f_r} x^*_{ik}{}^c \le \Delta \qquad (6.2)$$

The constraint (6.2) above is equivalent to the constraint (4.2.3) of problem $P3^c$ which was verified because x^*_{ik} ($i = 1,2,\ldots,n$ and $k = 1,2,\ldots,z$) is an optimal solutions of problem $P3^c$.

q.e.d.

Remark 1. *The criterion value obtained for the optimal solution of* $P3^c$ *equals the criterion value corresponding to the first sub-period of problem* $P2^{c+1}$ *with solution* X^{*c}; *i.e.* $C(P3^c, x^{*c}) = C_1(P2^{c+1}, X^{*c})$.

$C(P3^c, x^{*c})$ can be represented using (4.2.1) as:

$$\sum_{k=1}^{z} \sum_{i=1}^{n} w_i^+ [s^*_{ik}{}^c]^+ + w_i^- [-s^*_{ik}{}^c]^+ \text{ where } s^*_{ik}{}^c = \sum_{a=1}^{k} x^*_{ia}{}^c - D_{ik} \; ;$$

which can be rewritten as:

$$\sum_{k=1}^{z} \sum_{r=1}^{N} \sum_{il p_i \in f_r} w_i^+ [s^*_{ik}{}^c]^+ + w_i^- [-s^*_{ik}{}^c]^+$$

Due to the definition of attributes in the aggregation scheme of section 4:

$$\sum_{k=1}^{z} \sum_{r=1}^{N} v_r^+ \sum_{il p_i \in f_r} [s^*_{ik}{}^c]^+ + v_r^- \sum_{il p_i \in f_r} [-s^*_{ik}{}^c]^+ \qquad (6.3)$$

The inventory of part families is defined as follows (see also (3.2.3), (6.1) and (4.1.4)):

$$\sigma_{rk}{}^{c+1} = \sum_{il p_i \in f_r} s^*_{ik}{}^c \qquad r = 1,2,\ldots,N, \text{ and } k = 1,2,\ldots,z$$

According to property 1

$$[\sigma_{rk}^{c+1}]^+ = \sum_{il p_i \in f_r} [s^*_{ik}{}^c]^+ ; \qquad [-\sigma_{rk}^{c+1}]^+ = \sum_{il p_i \in f_r} [-s^*_{ik}]^+$$

which implies that (6.3) equals

$$\sum_{k=1}^{z} \sum_{r=1}^{N} v_r^+ [\sigma_{rk}^{c+1}]^+ + v_r^- [-\sigma_{rk}^{c+1}]^+$$

This shows that $C(P3^c, x^{*c}) = C_1(P2^{c+1}, X^{*c})$

Lemma 2. *The optimal solution of problem* $P2^c$ *guarantees that the set of feasible solutions of problem* $P3^c$ *is non-empty.*

Let $X^*_{r\kappa}{}^c$ (r = 1,2,...,N and κ = 1,2,...,Z) be the optimal solution of problem $P2^c$; $X^*_{r1}{}^c$ (r = 1,2,...,N) is of relevance to $P3^c$. We want to show that the set of feasible solutions of $P3^c$ is non-empty by constructing one such feasible solution. This is performed by a two step procedure.

Step 1.

Temporal disaggregation of $X^*_{r1}{}^c$ (r = 1,2,...,N) to determine production levels of families using the weights $\sum_{il p_i \in f_r} g_{ik}{}^c$ (r = 1,2,...,N and k = 1,2,...,z).

Let $Y_{rk}{}^c$ represent the cumulative production of family f_r till the k-th elementary period, and is defined as follows:

$$Y_{rk}{}^c = \sum_{a=1}^{k} \sum_{il p_i \in f_r} x_{ik}{}^c = X^*_{r1}{}^c \sum_{il p_i \in f_r} g_{ik}{}^c \qquad r = 1,2,...,N \text{ and } k = 1,2,...,z \tag{6.4}$$

At this step, we want to show the $x_{ik}{}^c$ (i = 1,2,...,n and k = 1,2,...,z) to be obtained from (6.4) (see step 2) satisfy constraint (4.2.3).

$$\sum_{i=1}^{n} x_{ik} \le \Delta$$

According to (6.4) and after simplifying, we obtain

$$\sum_{r=1}^{N} X^*_{r1} \Big(\sum_{il p_i \in f_r} g_{ik} - \sum_{il p_i \in f_r} g_{i(k-1)} \Big) \le \Delta$$

which is equivalent to the constraint (4.1.7) of problem $P2^c$.

Step 2.

Disaggregation of the production levels of family $Y_{rk}{}^c$ to determine production levels of parts using (4.2.4) and property 1.

In order to satisfy (4.2.4) we add a fictitious demand, d'_{iz} to d_{iz}, $\forall p_i$, for which $X^*_{a(i)1} > \sum_{j|p_j \in f_{a(i)}} D_{jz}$.

$$d'_{iz} = \alpha_i \left(X^*_{a(i)1} - \sum_{j|p_j \in f_{a(i)}} D_{jz}\right) \tag{6.5}$$

We then apply the disaggregation algorithm of (Nagi,1991) to determine x_{ik} ($\forall i|p_i \in f_r$; k = 1,2,...,z and r = 1,2,...,N). The basic idea of algorithm is as follows (for details see algorithm 2 of appendix A):

We consider one family at a time. For a family f_r, we consider elementary periods k = 1,2,...,z in sequence and examine $\sigma^*_{rk}{}^c = Y_{rk}{}^c - \sum_{j|p_j \in f_r} D_{jk}$: if

(a) $\sigma^*_{rk}{}^c > 0$

we define:

$$h_{rk} = \underset{b}{\text{Min}} \left\{ \text{s.t. } \sigma^*_{rk}{}^c - \sum_{a=k+1}^{b} \sum_{i|p_i \in f_r} d_{ia} \leq 0 \right\} \tag{6.6}$$

That is, h_{rk} is the first elementary period after k, where the inventory of the family f_r can be entirely consumed if no further production is performed. We solve the following problem to determine $x_{ik}{}^c$; $\forall i|p_i \in f_r$.

$$\sum_{i|p_i \in f_r} x_{ik}{}^c = Y_{rk}{}^c - Y_{r(k-1)}{}^c ; \tag{6.7}$$

$$s_{ik}{}^c = s_{i(k-1)}{}^c + x_{ik}{}^c - d_{ik}{}^c \geq 0 ; \qquad i|p_i \in f_r \tag{6.8}$$

$$\sum_{b=k+1}^{h} d_{ib} - s_{ik}{}^c \geq 0 ; \qquad i|p_i \in f_r \tag{6.9}$$

$$\sum_{b=k+1}^{a} d_{ib} - s_{ik}{}^c \leq 0 ; \qquad i|p_i \in f_r \text{ and } a = k+1,...,h-1 \tag{6.10}$$

$$x_{ik}{}^c \geq 0 ; \qquad i|p_i \in f_r \tag{6.11}$$

(b) $\sigma^*_{rk}{}^c \leq 0$

we solve the following problem to determine $x_{ik}{}^c$; $i|p_i \in f_r$.

$$\sum_{i|p_i \in f_r} x_{ik}{}^c = Y_{rk}{}^c - Y_{r(k-1)}{}^c ; \tag{6.12}$$

$$s_{ik}{}^c = s_{i(k-1)}{}^c + x_{ik}{}^c - d_{ik}{}^c \leq 0 ; \qquad i|p_i \in f_r \tag{6.13}$$

$$x_{ik}{}^c \geq 0 ; \qquad i|p_i \in f_r \tag{6.14}$$

The disaggregation problems described in (a) and (b) are solved using the algorithm 2 described in appendix A. The solution $x_{ik}{}^c$ is a feasible solution of $P3^c$ because: (i) (6.4), or (6.7) and (6.12) verify (4.2.2), (ii) step 1 verifies (4.2.3), (iii) (6.5) causes (4.2.4) to be satisfied, and (iv) (6.11) and (6.14) verify (4.2.5).

q.e.d.

Remark 2. *The criterion value corresponding to the first sub-period obtained for the optimal solution of* $P2^c$ *equals the criterion value of problem* $P3^c$ *and solution* $\mathbf{x}^c$ *provided by the previous algorithm.*
That is, $C_1(P2^c, \mathbf{X}^{*c}) = C(P3^c, \mathbf{x}^c)$.

$C_1(P2^c, \mathbf{X}^{*c})$ can be represented using (4.1.2), (4.1.3) and (4.1.4) as:

$$\sum_{k=1}^{z} \sum_{r=1}^{N} v_r^+ [\sigma^*_{rk}{}^c]^+ + v_r^- [-\sigma^*_{rk}{}^c]^+ \qquad (6.15)$$

The inventory of part families is defined by:

$$\sigma^*_{rk}{}^c = \sum_{il p_i \in f_r} s_{ik}{}^c \qquad r = 1,2,\ldots,N, \text{ and } k = 1,2,\ldots,z$$

According to property 1, i.e. (6.8) and (6.13)

$$[\sigma^*_{rk}{}^c]^+ = \sum_{il p_i \in f_r} [s_{ik}{}^c]^+ ; \qquad [-\sigma^*_{rk}{}^{c+1}]^+ = \sum_{il p_i \in f_r} [-s_{ik}]^+$$

which implies that (6.15) equals

$$\sum_{k=1}^{z} \sum_{r=1}^{N} v_r^+ \sum_{il p_i \in f_r} [s_{ik}{}^c]^+ + v_r^- \sum_{il p_i \in f_r} [- s_{ik}{}^c]^+$$

Due to the definition of attributes in the aggregation scheme of section 4:

$$\sum_{k=1}^{z} \sum_{i=1}^{n} w_i^+ [s_{ik}{}^c]^+ + w_i^- [- s_{ik}{}^c]^+$$

which equals $C(P3^c, \mathbf{x}^c)$.

Theorem 1. *The iterative scheme proposed in algorithm 1 converges.*

Since the optimal solution of problem $P2^c$ guarantees a non-empty set of feasible solutions to problem $P3^c$ (lemma 2), and we determine one solution $\mathbf{x}^c$ such that $C_1(P2^c, \mathbf{X}^{*c}) = C(P3^c, \mathbf{x}^c)$, this implies that $C(P3^c, \mathbf{x}^{*c}) \leq C_1(P2^c, \mathbf{X}^{*c})$. Furthermore, the optimal solution of problem $P3^c$ guarantees a non-empty set of feasible solutions to problem $P2^{c+1}$ (lemma 1), and we determine one solution $\mathbf{X}^{c+1} = \mathbf{X}^{*c}$ such that $C(P3^c, \mathbf{x}^{*c}) = C_1(P2^{c+1}, \mathbf{X}^{c+1})$. We also note that $C_2(P2^c, \mathbf{X}^{*c}) = C_2(P2^{c+1}, \mathbf{X}^{c+1})$. This leads to $C(P2^{c+1}, \mathbf{X}^{*c+1}) \leq C(P2^c, \mathbf{X}^{*c})$. Since the high level criterion is bounded, the algorithm converges.
q.e.d.

7 New Iterative Algorithm

The algorithm presented in section 5, was applied to 60 test cases (for details on problem generation, refer to section 9). It was observed that the algorithm converged to a feasible solution in all cases. The results indicated that the percentage deviation of the solution obtained by the algorithm 1 from the optimal ranged between 25% to 0%, with a concentration at around 15%. The mean number of iterations to convergence for these test cases was 3.9, and the maximal number of iterations was found to be 11. The reason for this sub-optimality was attributed to the fact that the weights g_{ik} obtained from an iteration at the low level, impose a restriction at the high level, i.e., the high level solution has to strictly follow these weights (due to (4.1.4) and (4.1.7)). While in-turn, the low level is constrained in approaching the optimal value of weights due to the aggregate high level solution, i.e. constraint (4.2.2). Thus, the iterative algorithm presented in section 5 converges to a local optimum in a rather constrained manner.

With this motivation in mind, a new algorithm is proposed in this section. We impose a relaxation on the capacity constraint (4.1.7) at the high level. Let U_k (k = 1,2,...,z) represent the number of time units by which the capacity constraint is violated in the k-th elementary period. At the high level, we also define a penalizing factor W which represents the penalty for utilizing one unit of extra capacity. Therefore, the penalty for violating capacities of the elementary period belonging to the first sub-period is defined by $W\times \sum_{k=1}^{z} U_k$. The penalizing factor W tends to infinity as the number of iterations tends to infinity. Alongside, at the low level, we impose a relaxation on the requirement that production of a given family be met with strict equality at the low level (i.e. constraint (4.2.2)). Let L_r (r = 1,2,...,N) represent the penalty for the violation of the production level of family f_r at the low level. At the low level, W represents the penalty for violating the high level production level for a family by one unit. Therefore, the penalty for violating the high level solution over all families is defined by $W\times \sum_{r=1}^{N} L_r$. Once again the penalizing factor W tends to infinity as the number of iterations tends to infinity. Observe that a very high value of penalizing factor W forces the high level problem to strictly follow the capacity constraint and the low level problem to strictly obey the high level production. This implies that the relaxation on (4.1.4) for the high level and (4.2.2) for the low level decreases with the number of iterations, finally leading them to be strict constraints. The relaxed high and low level problems, and the new algorithm are presented in the following sections. The numerical results presented in section 9 demonstrate the performance of this algorithm.

7.1 New High Level Problem

The problem can be stated as the following linear programming problem:

$$\textbf{(Problem P4)} \quad \text{Minimize: } \sum_{\kappa=1}^{Z} \sum_{r=1}^{N} \sum_{k=z(\kappa-1)+1}^{z\kappa} \Phi_r(\sigma_{rk}) + W\times \sum_{k=1}^{z} U_k \tag{7.1.1}$$

Subject to:

(4.1.3) through (4.1.6) ; (4.1.8) and

$$\Big(\sum_{r=1}^{N} \sum_{i\in f_r} X_{i1}(g_{ik} - g_{i(k-1)}) - \Delta\Big) \le U_k; \qquad k = 1,2,\ldots,z \tag{7.1.2}$$

The objective function (7.1.1) minimizes: (i) the inventory holding and backlogging for part families over the entire horizon, and (ii) the penalty for violating capacities of the elementary periods belonging to the first sub-period. Constraint (7.1.2) provides the value U_k for capacity violation during the k-th elementary period (k = 1,2,...,z).

7.2 New Low Level Problem

The problem can be stated as the following linear programming problem:

$$\textbf{(Problem P5)} \quad \text{Minimize: } \sum_{k=1}^{z} \sum_{i=1}^{n} f_i(s_{ik}) + W\times \sum_{r=1}^{N} L_r \tag{7.2.1}$$

Subject to:

(4.2.3) ; (4.2.5) and

$$\Big(X_{r1} - \sum_{k=1}^{z} \sum_{i|p_i\in f_r} x_{ik}\Big) \le L_r; \qquad r = 1,2,\ldots,N \tag{7.2.2}$$

$$\Big(\sum_{k=1}^{z} \sum_{i|p_i\in f_r} x_{ik} - X_{r1}\Big) \le L_r; \qquad r = 1,2,\ldots,N \tag{7.2.3}$$

$$\sum_{k=1}^{z} x_{ik} = D_{iz} + \alpha_i \Big(\sum_{k=1}^{z} \sum_{i|p_i\in f_r} x_{ik} - \sum_{j|p_j\in f_{a(i)}} D_{jz}\Big); \forall\, i \,\Big|\Big(X_{a(i)1} - \sum_{j|p_j\in f_{a(i)}} D_{jz}\Big) > 0 \tag{7.2.4}$$

The objective function (7.2.1) minimizes: (i) the inventory holding and backlogging costs for the part types in the elementary periods of the first sub-period, and (ii) the penalty for violating the aggregate production levels of part families. Constraints (7.2.2) and (7.2.3) provide the value L_r for the violation of the high level production for part-family f_r at the low level (r = 1,2,...,N). Constraint (7.2.4) ensures that

the surplus production of part-families at the end of the first sub-period is assigned to individual part types according to the ratios α_i.

7.3 New Iterative Algorithm

The new algorithm can be represented as follows:

Algorithm 3

Step 1-4. Same as *step* 1-4 in algorithm 1.

Step 5. Solve the low level problem (P5) for x_{ik} without constraints (4.2.5), (7.2.2), (7.2.3) and (7.2.4).

Step 6. Same as *step* 6 in algorithm 1.

Step 7. Initialize penalizing 'W' ; W = 0.
Initialize the iteration number 'c' ; c=0

Step 8. c=c+1.
W = W + O(Average Holding/Backlogging Cost); O(•) implies order of •.

Step 9. Initialize the weights g_{ik}^c from the production of part types and part families using the relation (4.1.1).

Step 10. Solve the high level problem (P4) for the production of part families in the sub-periods, using the current value of the weights g_{ik}^c.

Step 11. Disaggregate part family production over the first sub-period by solving the low level problem (P5).

Step 12. Same as *step* 12 in algorithm 1.

8 Error Bounds

In this section, we estimate the *a posteriori* error bounds while employing temporal aggregation. In addition, quick error bounds, both lower and upper are estimated for a given problem instance. All these error bounds take as a reference the optimal solution of a monolithic datum problem. (appendix B). The monolithic datum problem (PB) is one which is defined over entire horizon H: (i) the production variables corresponding to the elementary periods of the first sub-period are detailed, and (ii) production variables of the subsequent sub-periods are aggregate (along time); production over elementary periods corresponding to a sub-period is determined by dividing the aggregate production equally among these periods. The optimal solution of the monolithic datum problem is the best solution the iterative schemes (based on our hierarchical structure) can obtain.Such a demonstration is essential when a monolithic approach cannot be employed, since we can estimate the error introduced by the temporal aggregation *after* solving the problem by estimating the *a posteriori* error bounds. Similarly, when the optimal solution of the monolithic

problem is unavailable, quick estimation of the upper and the lower bounds of the monolithic solution can be utilized for evaluating the performance of the hierarchical solution.

8.1 A Posteriori Bound

Based on the work of Zipkin (1977), we summarize in this section, the *a posteriori* bound for the general column aggregation scheme of any LP problem. Consider a linear programming problem:

$$\underset{x_i}{\text{Max}} \sum_{i=1}^{n} c_i x_i ;$$

Subject to:

$$\sum_{i=1}^{n} a_{ij} x_i \le b_j ; \qquad j = 1,2,...,m$$

$$x_i \ge 0 ; \qquad i = 1,2,...,n$$

Let $\Omega = \{S_k: k = 1,2,...,K\}$ be a partition of $\{1,2,...,n\}$. The aggregate problem obtained can be formulated as follows:

$$\underset{X_k}{\text{Max}} \sum_{k=1}^{K} C_k X_k$$

Subject to:

$$\sum_{k=1}^{K} A_{kj} X_k \le b_j ; \qquad j = 1,2,...,m$$

$$X_k \ge 0 ; \qquad k = 1,2,...,K$$

Let, the dual solution of the aggregate problem be represented by U_j and the *a posteriori* bound be represented by ε_a. The *a posteriori* bound for the above mentioned aggregation can then be written as follows (Zipkin, 1977):

$$\varepsilon_a = \sum_{k=1}^{K} \left(\left[\underset{i \in S_k}{\text{Max}} \left(c_i - \sum_{j=1}^{m} U_j A_{kj} \right) \right]^{+} p_k \right); \qquad [\bullet]^{+} = \text{Max}\{0,\bullet\}$$

where the variable p_k is such that :(i) p_k is non-negative, $\forall k$, and (ii) p_k satisfy the relation (8.1.1) below, where x_i^* is the optimal solution to the original LP problem.

$$\sum_{i \in S_k} x_i^* \le p_k ; \qquad k = 1,2,...,K \qquad (8.1.1)$$

Using the above result, the *a posteriori* bound is estimated for the temporal aggregation scheme proposed in this paper. The numerical results presented in section 9 demonstrate the performance of this analysis.

8.2 Quick Bounds

In this section, we present the upper and the lower bounds of the monolithic datum problem (PB). The motivation of this analysis is to obtain the quick estimation of the upper and the lower bounds in the event that the optimal solution of the monolithic problem is unavailable. The hierarchical solution can then be compared with this range of values to assess performance.

8.2.1 Quick Lower Bound

In this section, we estimate the lower bound of the problem (PB). This lower bound is based on the boundary relaxation of production, similar to that developed by the Harhalakis *et al.* (1992). We consider each sub-period separately. For each sub-period κ, we perform the optimal allocation of the resource for the all part types due at the end of that sub-period. For the first sub-period (i.e., $\kappa = 1$), we solve for production of part types over elementary periods. For the consecutive sub-periods (i.e., $\kappa = 2, 3, ..., Z$) linear production of part types is considered over each sub-period. We plan the production of all part types on the sub-period while assuming that the sub-period is bordered on either side by periods of infinite duration other than the first sub-period. For the first sub-period, we assume only the (z+1)th period of infinite duration. The detailed algorithm of the quick lower bound procedure is presented in Algorithm 4 of appendix A.

8.2.2 Quick Upper Bound

In this section, we estimate the upper bound of the problem (PB). We consider each part type separately. The higher cost among the unit holding and backlogging cost is considered as the unique penalty weight for each part type, and parts are prioritized according to non-increasing values of these weights. For each part type i, we perform the optimal allocation of the resource for the entire horizon - the resource for the part type with the higher priority being allocated first. The optimal allocation is performed assuming the production of the part types to be linear for sub-periods greater than unity. It can be very easily proven that the solution (production for all part types) for which the quick upper bound is estimated is a feasible solution to the original problem. Hence the quick upper bound is always greater than or equal to the optimal criterion value of the original problem. The detailed algorithm of the quick upper bound procedure is presented in Algorithm 5 of appendix A.

9 Numerical Results

This section is devoted to the performance evaluation of algorithm 3 presented in section 7, the *a posteriori* bound, and the quick upper and lower bounds. The algorithm was applied to three problem sets with ninety-six test cases each. The number of part types considered were either eight or twelve; for the eight part case, four part families composed of three, two, two and one part types respectively were

considered, and for the twelve part case, six part families composed of three, three, two, two, one and one part types respectively were considered. The holding and backlogging costs of part types were generated at random, assuming a uniform distribution. The holding and backlogging costs for these three problem sets are presented in Table 9.1. The first set corresponds to the case where the holding and backlogging costs are of equal order. The second and third sets correspond to the cases where the holding costs are generally greater or lesser than the backlogging costs respectively.

PROBLEM SET	HOLDING COSTS	BACKLOGGING COSTS
[1]	U(10,20)	U(10,20)
[2]	U(5,50)	U(1,10)
[3]	U(1,10)	U(5,50)

Table 9.1 Holding and backlogging costs for three problem sets

Parts demand was assumed to be cyclic (sinusoidal) with varying amplitudes, and variation on the basic cyclic pattern was superimposed (see Figure 9.1). For each example, planning problems of varying degrees of difficulty were constructed as follows: (i) the number of sub-periods Z, (ii) the ratio of the capacity available over a horizon to the average capacity required (workload) over the horizon, denoted by CA, (iii) the ratio of the amplitude of cyclicity of a part's demand, to its average demand over the entire horizon, denoted by AMP, and (iii) the ratio of the variation in a part's demand to its average demand over the entire horizon, denoted by VAR. The details of these parameters are presented in Table 9.2. Finally, for each set of parameters, random demand streams were generated assuming uniform distributions on VAR.

N	[Z, z, Δ]	CA	AMP	VAR
8; 12	[12, 4, 60]; [15,4,60]; [8,5,32]; [12,5,32]	1.02; 1.06; 1.10	0.6; 0.8	0.2; 0.3

Table 9.2: Parameters selected to generate 96 test cases for each problem set

Thus, overall, 288 problems were attempted. The programs were coded in C, and the LP problems were solved by XMP (1981) subroutines on the SUN/SPARC station.

Figure 9.2 presents the performance evaluation of the iterative algorithm (presented in section 7) with respect to the monolithic datum solution. The results indicate that the iterative approach was less than 2% sub-optimal in 97% of the test cases, and the iterative solution was 0.4% sub-optimal on an average. The mean number of iterations until convergence was achieved for these 288 test cases was 5.4, and the maximal number of iterations was found to be 11.

Figure 9.3 presents the performance evaluation of the *a posteriori* error bound with respect to the monolithic datum solution. *A posteriori* bounds were about 20% lower than the monolithic datum solution in 52.3% of the test cases on an average. The performance of the bound deteriorated with higher CA (high unutilized capacity) and smaller values of AMP. Due to the high value of p_k variables (section 7.2), the *a posteriori* error bound was lower than the monolithic datum solution by 80-100% in 35.7% of the test cases. On an average, the *posteriori* bounds were 41.2% lower than the monolithic datum solution.

Figure 9.4 presents the performance evaluation of the quick upper bounds (presented in section 7) with respect to the monolithic datum solution. Quick upper bounds were around 25% higher than the monolithic solution in 4.2% of the test cases. The performance of these bounds deteriorated with a larger difference between holding and backlogging costs, larger Z, lower CA (tighter capacity requirement) and larger values of AMP. On an average, the quick upper bounds were 175.7% higher than the iterative solution.

Figure 9.5 presents the performance evaluation of the quick lower bounds (presented in section 7) with respect to the monolithic datum solution. Quick lower bounds were around 25% lower than the monolithic solution in 14.3% of the test cases. The performance of these bounds deteriorated with lower CA and larger values of AMP. The quick lower bounds perform better for the problem set [2] (when holding costs are generally greater than the backlogging costs) in comparison to problem sets [1] and [3]. On an average, the quick lower bounds were 61.4% lower than the iterative solution.

A summary of the performance of the bounds with respect to the problem parameters is presented in Table 9.3 and the numerical results for twenty-eight problems in problem set [1] are presented in Table 9.4.

Bounds	increase n	increase Z	increase CA	increase AMP	increase VAR
Quick upper bound	worsen	worsen	improve	worsen	no effect
Quick lower bound	worsen	worsen	improve	worsen	no effect
A posteriori bound	no effect	no effect	worsen	improve	no effect

Table 9.3: Performance evaluation of bounds with respect to problem parameters

10 Conclusion

This paper focuses on temporal aggregation which is an essential element in hierarchical production planning due to the different time-scales of decisions and the different horizons over which these are taken. We assume that the need to aggregate a certain number of elementary time periods into larger sub-periods

has been determined by the characteristics of the specific problem. Such characteristics include the time-scales of decisions and their cyclicity (see e.g., Engles , Larson *et al.*, 1976). Given this requirement, we develop the theory behind the development of a two-level hierarchical structure: (i) the aggregate high level that plans for production over sub-periods, and (ii) the detailed low level that disaggregates the high level plan. These problems are formulated as appropriate LP problems (as is the monolithic formulation of most production planning problems), and we develop iterative algorithms that provide a consistent and near-optimal solution. Error analysis is performed and lower and upper bounds of the problem instance are developed to assess the quality of solution obtained for the problem at hand. Numerical results confirm the applicability of this approach to solve large and complex problems with minimal loss of optimality. More importantly, the hierarchical scheme based on temporal and part aggregation permits the computation of aggregate as well as detailed production plans when detailed demand/forecast information is not known (or considered necessary) at high management levels with long planning horizons.

Employing the ideas proposed for a specific problem of minimizing holding and backlogging costs, we hope that other problems at different tactical levels of production planning can also be addressed. Future work should aim at addressing problems with multiple decisions and controls; capacity planning by hiring and overtime decisions are such areas which can be attempted directly. Finally, the accurate determination of the aggregation of elementary time periods into sub-periods for a general N-level hierarchy corresponding to a complex problem of multiple time-scales is another important focal area.

References

[1] Aardal, K., Larsson, T. (1990): "A Benders decomposition based heuristic for the hierarchical production planning problem," *European Journal of Operational Research*, 45 /4.

[2] Abraham, C., Dietrich, B., Graves, S., Maxwell, W. and Yano, C. (1985): "A Research Agenda for Models to Plan and Schedule Manufacturing Systems," in: an NSF workshop presented at as *Scheduling the Factory of the Future: A Research Planning Session*, Decision Sciences Department, University of Pensylvania, March 1985, and Boston TIMS/ORSA meeting.

[3] Axsater, S. (1981): "Aggregation of Product Data for Hierarchical Production Planning," *Operation Research*, 29 /4.

[4] Axsater, S. (1986): "On the Feasibility of Aggregate Production Plans," *Operation Research*, 34 / , 796- .

[5] Axsater, S., Jonsson, H., (1984): "Aggregation and Disaggregation in Hierarchical Production Planning," *European Journal of Operational Research*, 17/ , 338-.

[6] Bergstrom, R., Edin, P. A. (1992): "Time Aggregation and the Distributional Shape of Unemployment Duration," *Journal of Applied Econometrics*, 7 /1, 5.

[7] Bitran, G.R., Haas, E.A. and Hax, A.C. (1981): "Hierarchical Production Planning: A Single Stage System," *Operations Research*, 29 /4.

[8] Bitran, G.R., Haas, E.A. and Hax, A.C. (1982): "Hierarchical Production Planning: A Two Stage System," *Operations Research*, 30 /2.

[9] Bitran, G.R. and Hax, A.C. (1977): "On the Design of Hierarchical Planning Systems," *Decision Sciences*, 8/ .

[10] Bitran, G.R. and Hax, A.C. (1981): "Disaggregation and Resource Allocation using Convex Knapsack Problems with Bounded Variables," *Management Science*, 27 /4.

[12] Boskma, K. (1982): "Aggregation and Decision of Models for Medium Term Planning of Production," *European Journal of Operational Research*, 10/ ,244- .

[13] Cunningham, Thomas J. Hardouvelis, Gikas A. (1992): "Money and Interest Rates: The Effects of Temporal Aggregation and Data Revisions," *Journal of Economics and Business*, 44/1, 19.

[14] Dempster, M.A.H., Fisher, M.L., Lageweg, B., Jansen, L., Lenstra, J.K. and A.H.G. Rinnoy Kan (1981): "Analytical evaluation of Hierarchical Planning Systems," *Operations Research*, 29/4.

[15] Dudkin, L.M., Rabinovick, I, Vakhutinsky, I. (1987): "*Iterative Aggregation Theory*," Marcel Dekker, Inc., 270 Madison ave., NY 10016.

[16] Engles, L., Larson, R.E., Peschon, J., Stanton, K.N., (1976): "Dynamic programming applied to hydro and thermal generation scheduling," in: IEEE tutorial course on application of optimization methods in power system engineering, IEEE, NY.

[17] Erscheler, J., Fontan, G. and Merce, C. (1986): "Consistency of the Disaggregation Process in Hierarchical Planning," *Operations Research*, 34 / 3.

[18] Ewing, R.E. (1990): "A posteriori error estimation," *Computer Methods in Applied Mechanics and Engine*, 82 / 1 / 3, 59- .

[19] Gabby, H. (1975): "A Hierarchical Approach to Production Planning," TR-120, Operations Research Center, M.I.T, Cambridge, MA.

[20] Graves, S.C. (1982): "Using Lagrangean Techniques to solve Hierarchical Production Planning Problems," *Management Science*, 28 /3.

[21] Harhalakis, G., Nagi, R., Proth., J.M., (1992): "Single Machine Scheduling with Discrete Earliness and Tardiness," Technical Research Report, Systems Research Center, University of Maryland, College Park, USA, TR 92-79.

[22] Hax, A.C. and Meal, H.C. (1975): "Hierarchical Integration of Production Planning and Scheduling," in: *Studies in the Management Sciences*, M.A. Geisler, ed., Vol. 1, Logistics, North Holland - American Elsevier.

[23] Hillion, H., Meier, K. and Proth, J.M. (1987): "Production Subsystems and Part-families: The Top Level Model in Hierarchical Production Planning Systems," *Operational Research'87*, G.K. Rand, ed., Elsevier Science Publishers B.V. (North-Holland), © IFORS, 1988.

[24] Krajewski, L.J. and Ritzman, L.P. (1977): "Disaggregation in manufacturing and service organizations: survey of problems and research," *Decision Sciences*, 8 / 1.

[25] Lasserre, J.B. Merce, C. (1990): "Robust hierarchical production planning under uncertainty," *Annals of Operations Research*, 26/ 1 / 4, 73-.

[26] Libosvar, C. (1988): "Hierarchical Production Management: The Flow Control Layer," Ph.D. Thesis, University of Metz, France.

[27] Meier, K. (1989): "Commande Hierarchisee d'un Systeme de Production," Ph.D. Thesis, University of Metz, France.

[28] Mohanti, R.P. (1987): "Hierarchical Production Planning: Comparison of some heuristics," *Engineering Costs and Production Economics*, 11 /, 203-.

[29] Monts, K. (1991): "An Empirical Procedure for the Temporal Aggregation of Electric Utility Marginal Energy Costs," *IEEE Transactions on Power Systems: A Publicat*, 6 /2, 658-.

[30] Nagi, R. (1991): "Design and Operation of Hierarchical Production Management Systems," Ph.D. Thesis, University of Maryland, College Park.

[31] Oguchi, Noriyoshi Fukuchi, Takao (1990): "On Temporal Aggregation of Linear Dynamic Models," *International Economic Review*, 31 /1, 187-.

[32] Rogers, D.F., Plante, R.D., Wong, R.T., Evans, J.R. (1991): "Aggregation and Disaggregation Techniques and Methodology in Optimization," *Operations Research*, 39 /4, 553-.

[33] Rossana, R.J., Seater, J.J. (1992): "Aggregation, Unit Roots and the Time Series Structure of Manufacturing Real Wages," *International Economic Review*, 33 /1, 159-.

[34] Saad, Germaine H. (1990): "Hierarchical production-planning systems: extensions and modifications," *OR; The Journal of the Operational Research Society*, 41 /7, 609-.

[35] Tsubone, H., Matsuura, H., Tsutsu, T. (1991): "Hierarchical production planning system for two-stage process," *International Journal of Production Research*, 29 /4, 769-.

[36] Wei, W.W.S., Mehta, J. (1980): "Temporal aggregation and information loss in the distributed lag model," *Analyzing Time Series*, Proceeding of the international conference, Guerney, Channel Islands, Publisher: North-Holland, Amsterdam, Netherlands.

[37] XMP (1981) or Marsten, R. E.: "*The design of the XMPLP Library*," Transactions on Mathematical Software, 7/ 4.

[38] Whitt, W. (1978): "Approximations of dynamic programs, I," *Math. Opns. Res.*, 3 /, 231-.

[39] Whitt, W. (1979): "Approximations of dynamic programs, II," *Math. Opns. Res.*, 4 /, 179-.

[40] Zipkin, P.H. (1977): "Aggregation in Linear Programming," Ph.D. Thesis, Yale University, New Haven, Conn, 1977.

[41] Zipkin, P.H. (1980a): "Bounds for Aggregating Nodes in Network Problems," *Math. Prog.*, 19 /.

[42] Zipkin, P.H. (1980b): "Bounds on the effect of Aggregating Variables in Linear Programs," *Operations Research*, 28 /.

Appendix A

Algorithm 2

For r = 1 to N do
$\forall\ b|p_b \in f_r,\ d_{bz} = d_{bz} + \alpha_b\ \sigma_{rz}^c$

For r = 1 to N do
For k = 1 to z do
Case $\sigma^*_{rk}{}^c > 0$
$\forall\ b|p_b \in f_r,\ x_{bk}^c = d_{bk}$ /* Produce immediate demand */
remainder = $\sigma_{rk}^c - \sum_{b|p_b \in f_r} x_{bk}^c$
For j = k+1 to h_{rk} do
If $j < h_{rk}$
$\forall\ b|p_b \in f_r,\ e_{bj} = d_{bj}$
else
$\forall\ b|p_b \in f_r$, select e_{bj} such that $0 \le e_{bj} \le d_{bj}$, and $\sum_b e_{bj} \le$ remainder;
$\forall\ b|p_b \in f_r,\ x_{bk}^c = x_{bk}^c + e_{bj},\ d_{bj} = d_{bj} - e_{bj}$ /* Produce some later demand */
remainder = remainder - $\sum_b e_{bj}$

Case $\sigma^*_{rk}{}^c \le 0$
$\forall\ b|p_b \in f_r$, select x_{bk} such that $0 \le x_{bk} \le d_{bk}$, and $\sum_b x_{bk} = Y_{rk} - Y_{r(k-1)}$

If $k < z$
$\forall\ b|p_b \in f_r,\ d_{bk+1} = d_{bk+1} + d_{bk} - x_{bk}^c$
End.

Algorithm 4

Notation:

y_{if} = Number of units of part type p_i required at the end of a period, prior to the sub-period under consideration.

y_{il} = Number of units of part type p_i required at the end of a period after the sub-period under consideration.

$T_{i\kappa}$ = Number of units of part type p_i required at the end of sub-period κ.

Δ_∞ = Duration of periods prior to and after the sub-period under consideration; ($\Delta_\infty >> \Delta$).

QLB_κ = Quick lower bound for the κ-th sub-period.

For $\kappa = 1$ to Z do

Solve the LP problem $LB_\kappa( )$

Quick lower bound = $\sum_{\kappa=1}^{Z} QLB_\kappa$

END.

The LP problem $LB_1( )$ is formulated as follows (for $\kappa = 1$):

$$\text{Minimize: } \sum_{i=1}^{n} f_i(y_{if}) + QLB_1 + \sum_{i=1}^{n} f_i(y_{if} + \sum_{s=1}^{z} x_{is} + y_{il} - \sum_{s=1}^{z} d_{is})$$

Where:

$$QLB_1 = \sum_{k=1}^{z} \sum_{i=1}^{n} f_i(y_{if} + \sum_{s=1}^{k} x_{is} - \sum_{s=1}^{k} d_{is})$$

Subject to:

$y_{if} \le \Delta_\infty$; $\quad i = 1,2,\ldots,n$

$x_{ik} \le \Delta$; $\quad i = 1,2,\ldots,n$ and $k = 1,2,\ldots,z$

$y_{il} \le \Delta_\infty$; $\quad i = 1,2,\ldots,n$

$y_{if} \ge 0$; $\quad k = 1,2,\ldots,z$

$x_{ik} \ge 0$; $\quad k = 1,2,\ldots,z$ and $k = 1,2,\ldots,z$

$y_{il} \ge 0$; $\quad k = 1,2,\ldots,z$

The LP problem $LB_\kappa( )$ is formulated as follows (for $\kappa = 2,3,\ldots,Z$):

$$\text{Minimize: } \sum_{i=1}^{n} f_i(y_{if}) + QLB_\kappa + \sum_{i=1}^{n} f_i(y_{if} + T_{i\kappa} + y_{il} - \sum_{s=(\kappa-1)z+1}^{\kappa z} d_{is})$$

Where:

$$QLB_\kappa = \sum_{k=(\kappa-1)z+1}^{\kappa z} \sum_{i=1}^{n} f_i(y_{if} + \sum_{s=(\kappa-1)z+1}^{k} \frac{T_{i\kappa}}{z} - \sum_{s=(\kappa-1)z+1}^{k} d_{is})$$

Subject to:

$y_{if} \le \Delta_\infty$; $\quad i = 1,2,\ldots,n$

$T_{i\kappa} \le z \times \Delta$; $\quad i = 1,2,\ldots,n$

$y_{il} \leq \Delta_{\infty}$; $\quad i = 1,2,\ldots,n$

$y_{if} \geq 0$; $\quad k = 1,2,\ldots,z$

$T_{i\kappa} \geq 0$; $\quad i = 1,2,\ldots,n$

$y_{il} \geq 0$; $\quad k = 1,2,\ldots,z$

Algorithm 5

Notation:

C_{ik} = Capacity available when solving the LP problem for the part type p_i in the k-th elementary period.

$T_{i\kappa}$ = Number of units of part type p_i required at the end of sub-period κ.

QUB_i = Quick upper bound when solving the LP problem for the part type p_i.

Sort $\forall w_i = \max(w_i^+, w_i^-)$ in non-increasing order

Rearrange indices of w_i in increasing order, i.e. $w_1, w_2, \ldots, w_n$ are now in non-increasing order

For i = 1 to n do

if (i = 1)

For k = 1 to Zz do

$C_{ik} = \Delta$

else

For k = 1 to Zz do

$C_{ik} = C_{(i-1)k} - x_{(i-1)k}$

Solve the LP problem $UB_i(\,)$

Quick upper bound = $\sum_{i=1}^{n} QUB_i$

END.

The LP problem $UB_i(\,)$ is formulated as follows:

$$\text{Minimize: } QUB_i = \sum_{k=1}^{z} f_i\Big(\sum_{s=1}^{k} x_{is} - D_{ik}\Big) + \sum_{k=z+1}^{Zz} f_i\Big(\sum_{s=1}^{z} x_{is} + \sum_{s=z+1}^{k} \frac{T_{ib(s)}}{z} - D_{ik}\Big)$$

Subject to:

$x_{ik} \leq C_{ik}$; $\quad k = 1,2,\ldots,Zz$

$T_{ib(k)} \leq z \times C_{ik}$; $\quad k = 1,2,\ldots,Zz$

$T_{ib(k)} \geq 0$; $\quad k = z+1,z+2,\ldots,Zz$

$x_{ik} \geq 0$; $\quad k = 1,2,\ldots,z$

Appendix B

In this appendix, we present a linear programming problem. The criterion of the LP problem is the minimization of the inventory for all part types, at the end of each elementary period. The production of a part type in the sub-period, is assumed to be linear, for sub-periods greater than unity. Production of the part type p_i at the end of sub-period κ is represented by $T_{i\kappa}$. The problem can be formally stated as the following linear programming problem:

(Problem PB) Minimize: $$\sum_{k=1}^{z}\sum_{i=1}^{n} f_i(s_{ik}) + \sum_{\kappa=2}^{Z}\sum_{k=z(\kappa-1)+1}^{z\kappa}\sum_{i=1}^{n} f_i(s_{ik}) \qquad \text{(B.1)}$$

Where:

$$s_{ik} = \sum_{s=1}^{k} x_{is} - \sum_{s=1}^{k} d_{is}\,; \qquad \text{for } (k \le z) \qquad \text{(B.2)}$$

$$s_{ik} = \sum_{s=1}^{z} x_{is} + \sum_{s=z+1}^{k} \frac{T_{i\kappa}}{z} - \sum_{s=1}^{k} d_{is}\,; \qquad \text{for } (k > z) \qquad \text{(B.3)}$$

Subject to:

$$\sum_{i=1}^{n} T_{i\kappa} \le z\Delta\,; \qquad \kappa = 2,3,\ldots,Z \qquad \text{(B.4)}$$

$$\sum_{i=1}^{n} x_{ik} \le \Delta\,; \qquad k = 1,2,\ldots,z \qquad \text{(B.5)}$$

$$x_{ik} \ge 0\,; \qquad i = 1,2,\ldots,n, \text{ and } k = 1,2,\ldots,z \qquad \text{(B.6)}$$

$$T_{i\kappa} \ge 0\,; \qquad i = 1,2,\ldots,n, \text{ and } \kappa = 2,3,\ldots,Z \qquad \text{(B.7)}$$

Constraint (B.4) represents the capacity limits in each elementary period in the first sub-period. Constraint (B.5) represents the capacity limits in each sub-period. (B.6) implies the non-negativity of production of part types over each elementary periods and (B.7) implies the non-negativity of production of part types over each sub-period.

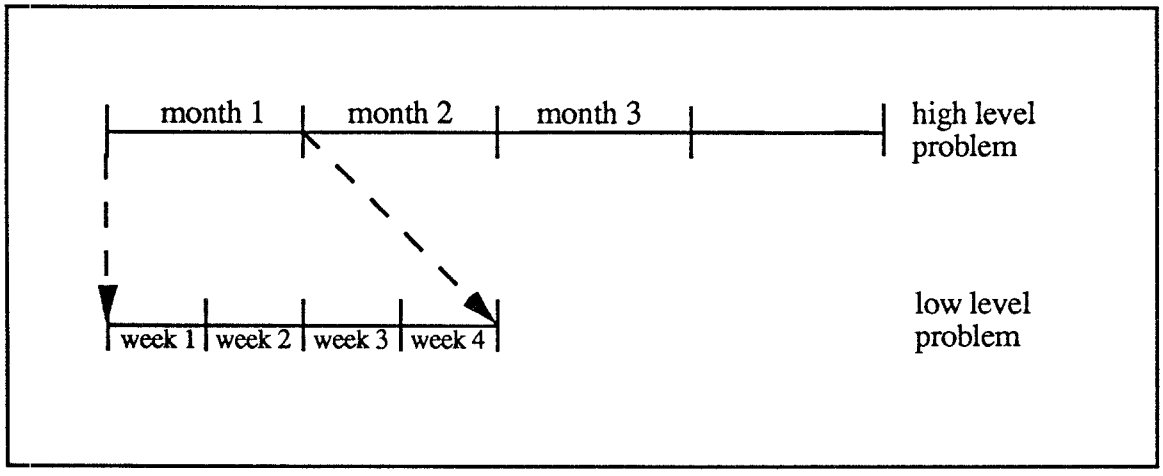

Fig. 2.1: Temporal aggregation in a two level hierarchical structure

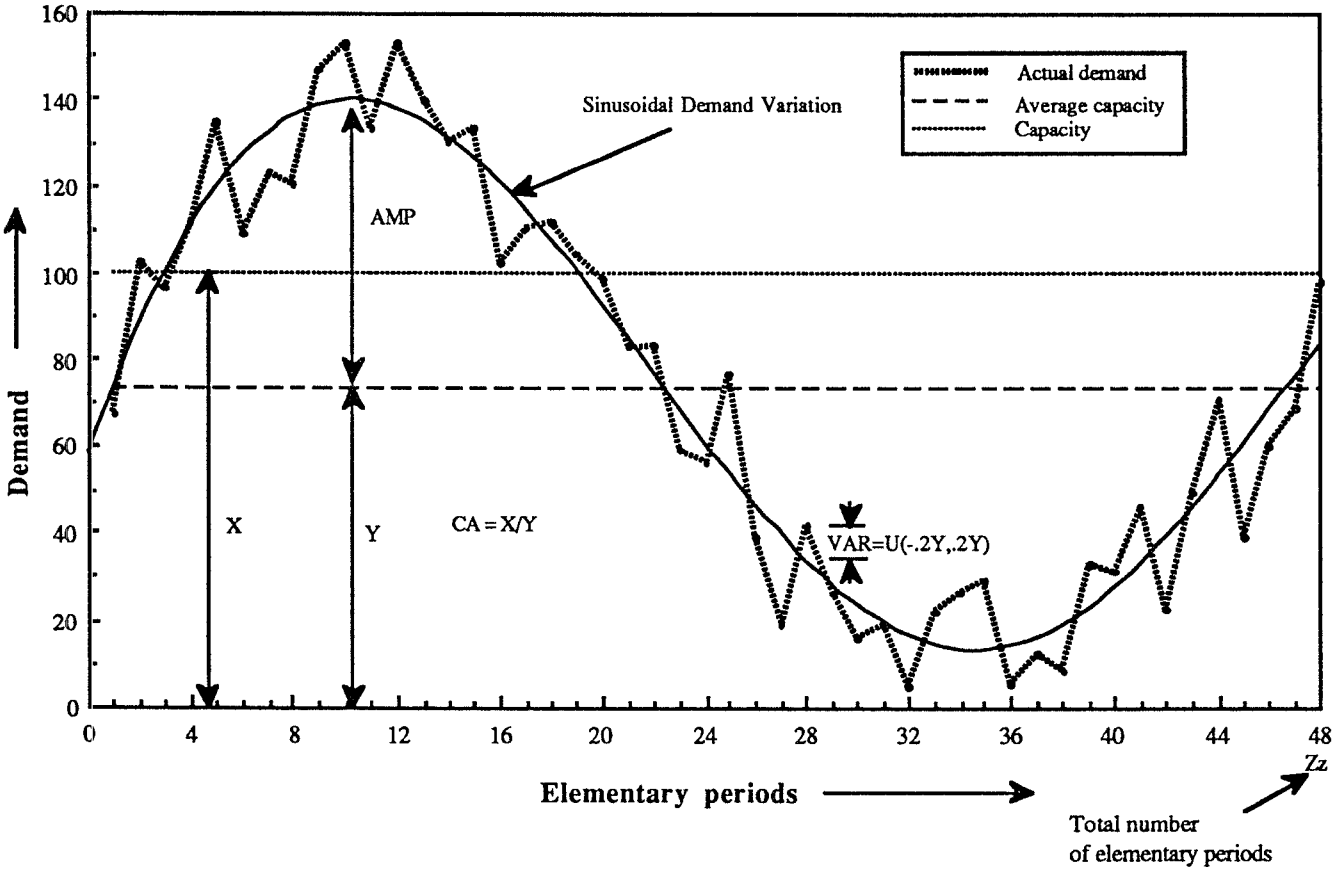

Fig. 9.1: Typical demand for a part type

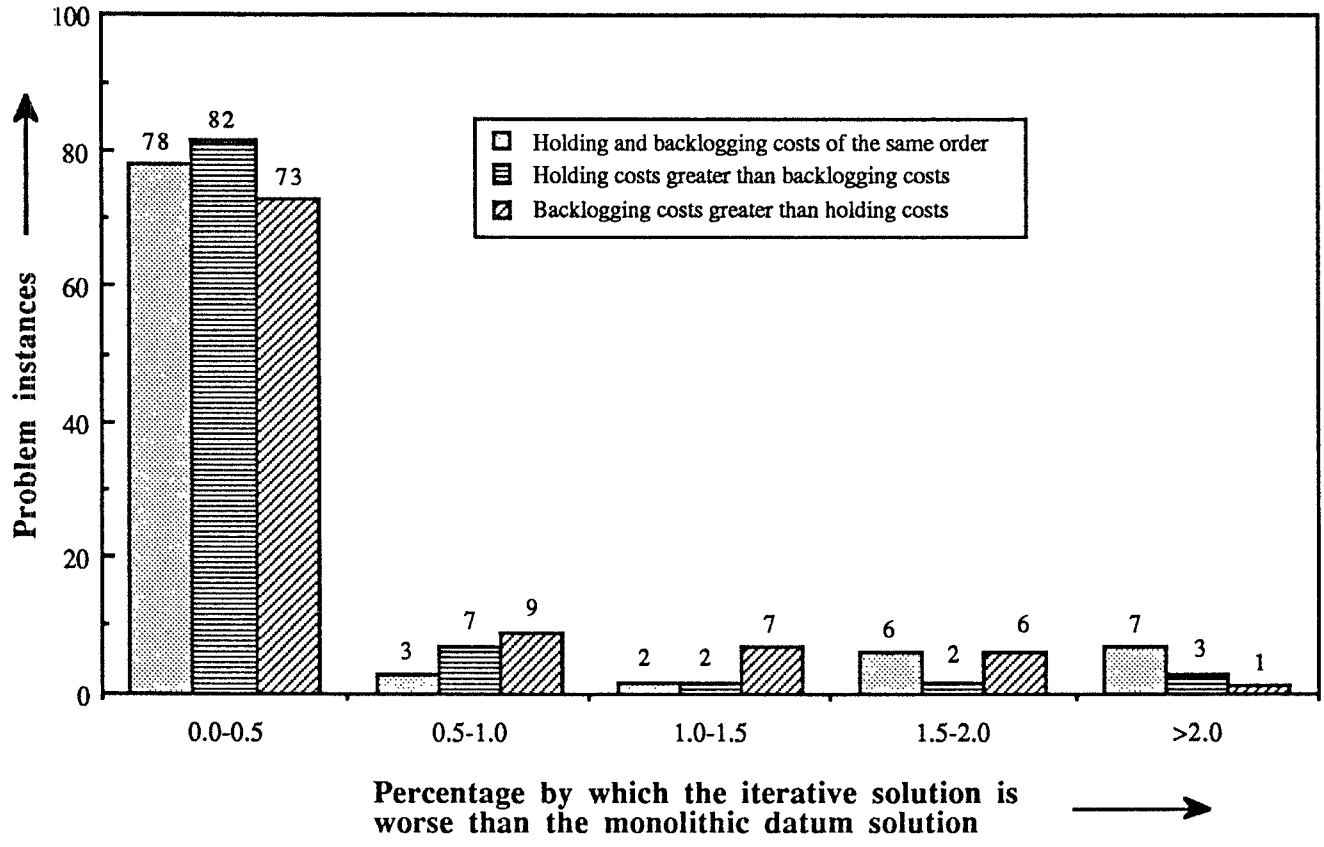

Fig. 9.2: Performance evaluation of algorithm 3

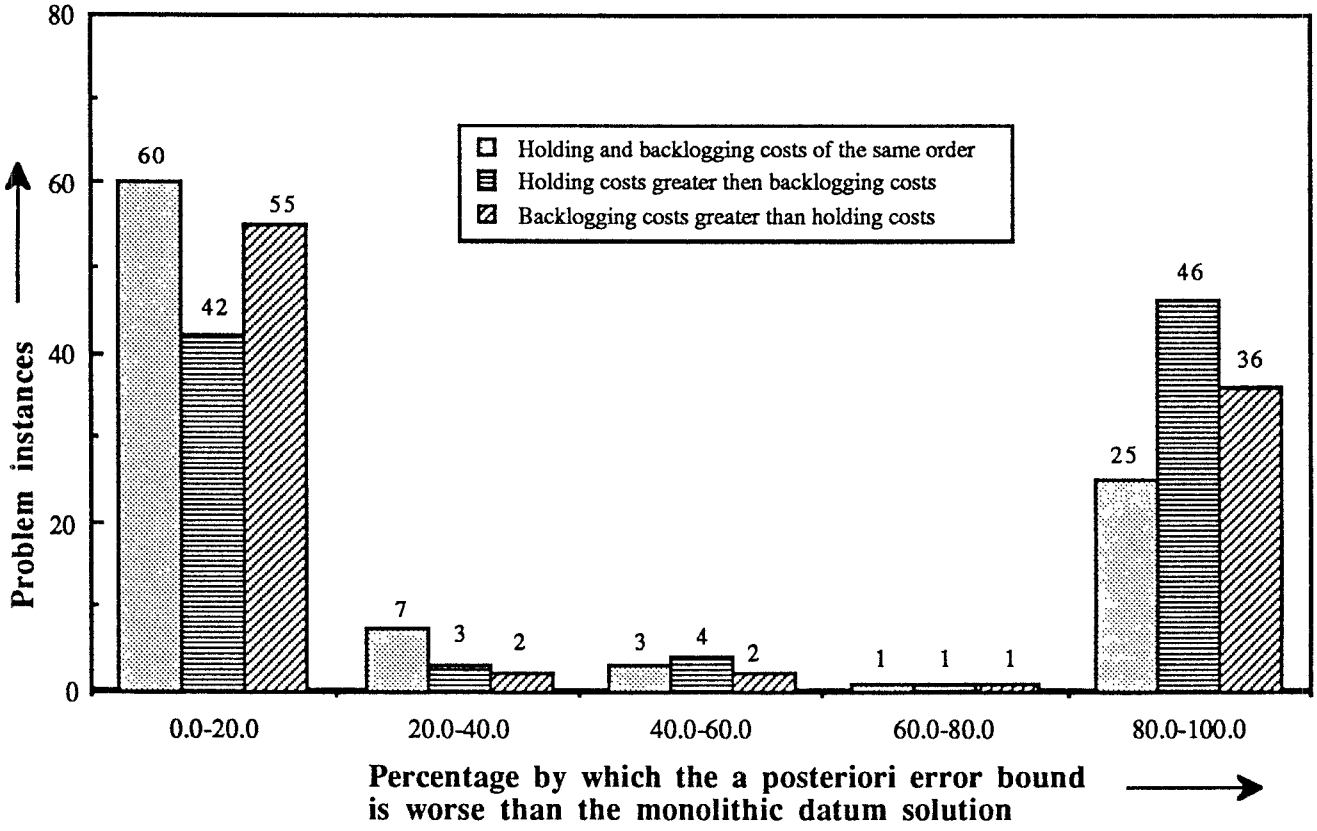

Fig. 9.3: Performance evaluation of the *a posteriori* error bound

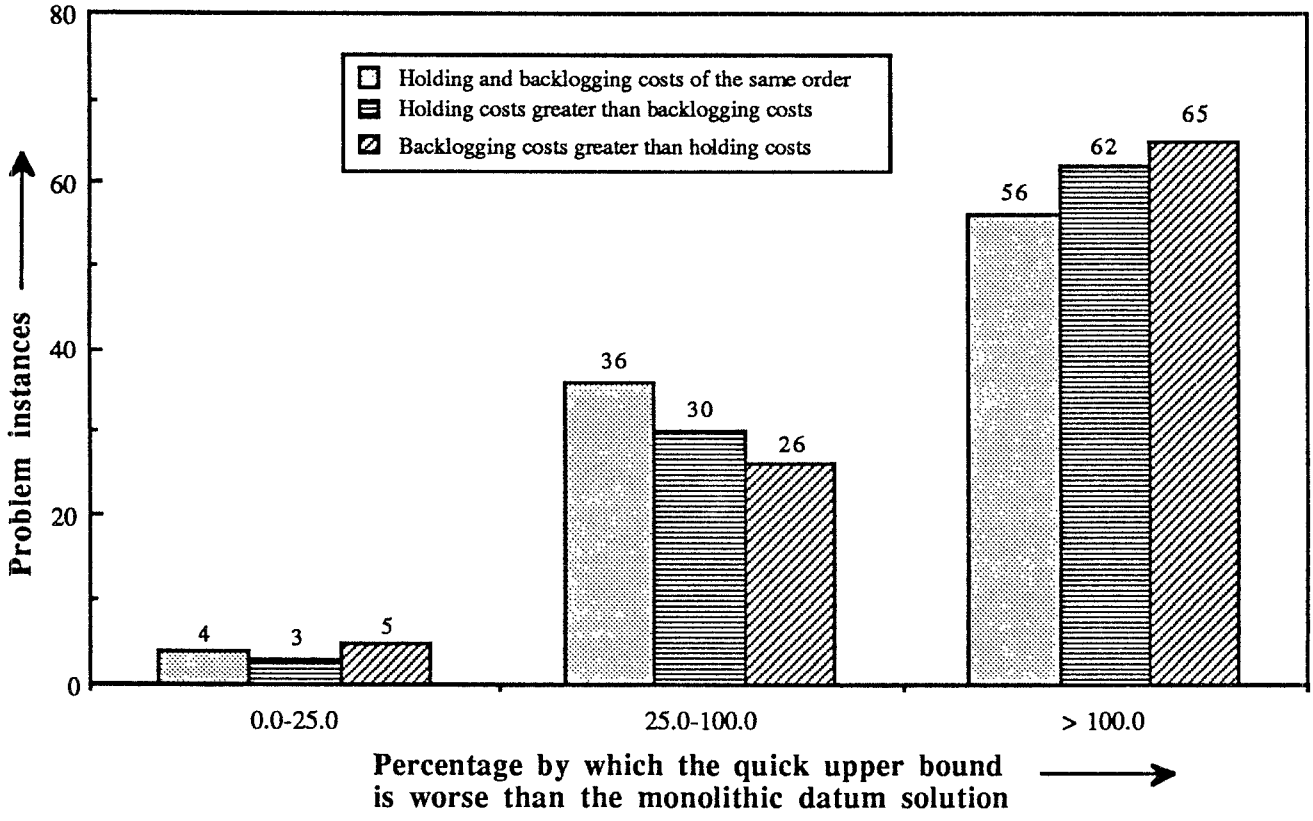

Fig. 9.4: Performance evaluation of the quick upper bound

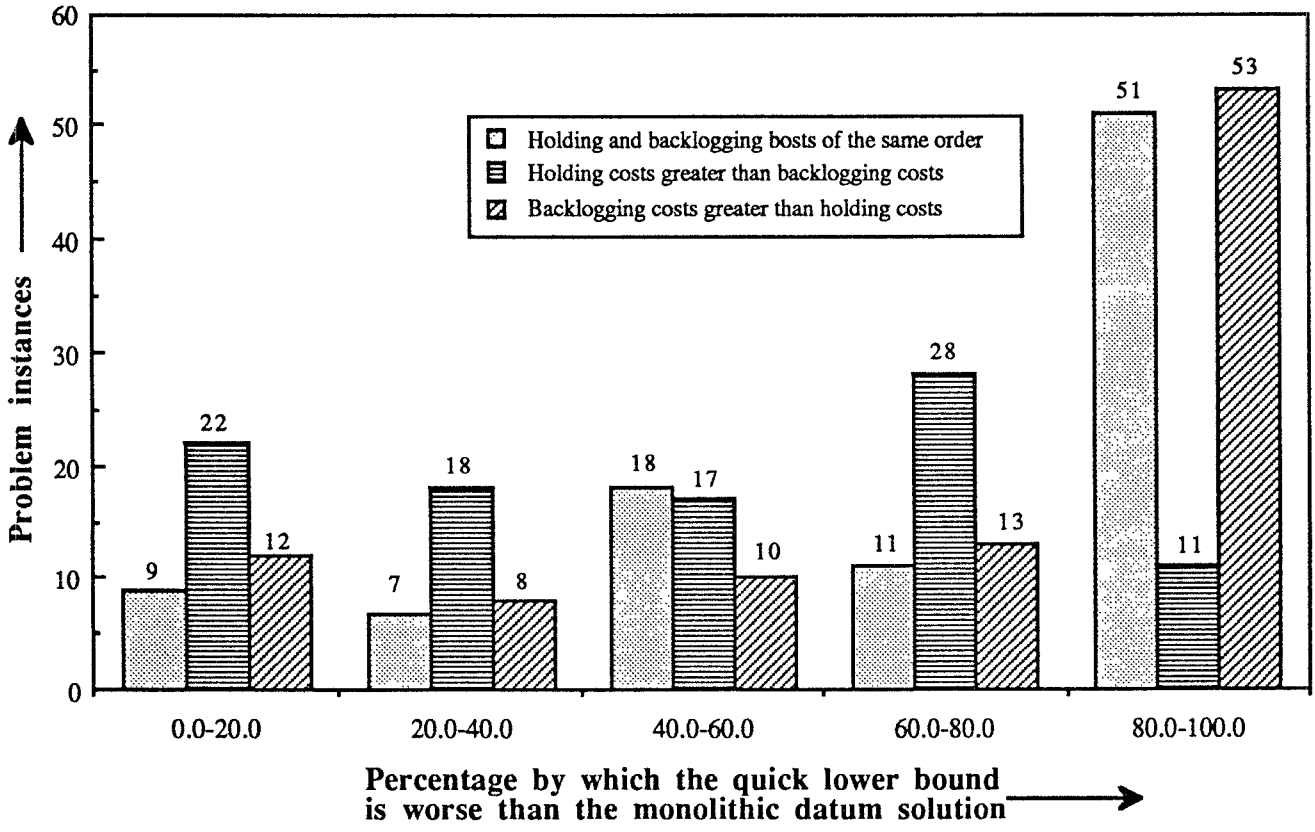

Fig. 9.5: Performance evaluation of the quick lower bound

Problem Instances	Datum Solution	Iterative Solution	A Posteriori Error Bound	Max(DS-APR , 0)	Quick Upper Bound	Quick Lower Bound	Z	z	N	Cap.	CA	AMP	VAR
1	57626	57632	2.44	57623.56	132229.505	5286.89773	12	4	8	60	1.02	0.85	0.3
2	60954	61004	2940	58014	125601.103	6553.02911	12	4	8	60	1.02	0.85	0.2
3	29692	29724	7040	22652	60087.132	3754.92604	12	4	8	60	1.02	0.7	0.3
4	20142	20146	0.0342	20141.9658	43392.807	4184.34227	12	4	8	60	1.02	0.7	0.2
5	9292.2	9368.6	0.1474	9292.0526	18435.7282	2472.1967	12	4	8	60	1.06	0.85	0.3
6	119412	119420	0.1358	119411.864	274226.816	7322.90528	12	4	8	60	1.06	0.85	0.2
7	1104.84	1139.26	9360	0	1455.29972	1027.2555	12	4	8	60	1.06	0.7	0.3
8	19357.4	19510.6	0.0576	19357.3424	40254.1469	5340.84687	12	4	8	60	1.06	0.7	0.2
9	51024	51068	0.532	51023.468	98746.7394	5583.58144	12	4	8	60	1.1	0.85	0.3
10	57438	57442	0.0788	57437.9212	113202.64	5154.13924	12	4	8	60	1.1	0.85	0.2
11	70028	70032	0.0438	70027.9562	143447.795	6336.80128	12	4	8	60	1.1	0.7	0.3
12	7431.8	7470.6	10480	0	14336.9507	2425.37737	12	4	8	60	1.1	0.7	0.2
13	48494	48526	0.0236	48493.9764	97033.1287	4705.42504	15	4	8	60	1.02	0.85	0.3
14	57592	57624	1.012	57590.988	123844.427	5087.38892	15	4	8	60	1.02	0.85	0.2
15	27704	27708	0.043	27703.957	57166.2826	4103.90883	15	4	8	60	1.02	0.7	0.3
16	61596	61612	0.492	61595.508	122427.802	7158.82681	15	4	8	60	1.02	0.7	0.2
17	85020	85052	3.44	85016.56	210950.982	8700.17177	15	4	8	60	1.06	0.85	0.3
18	45444	45474	1.246	45442.754	93640.9841	7185.7849	15	4	8	60	1.06	0.85	0.2
19	19136.8	19165.4	4820	14316.8	37464.062	2694.74354	15	4	8	60	1.06	0.7	0.3
20	3968.8	3985.2	9720	0	7311.95452	3002.16115	15	4	8	60	1.06	0.7	0.2
21	4623.2	4635.6	0.0952	4623.1048	10205.2923	2308.14515	15	4	8	60	1.1	0.85	0.3
22	9741.8	9771.4	1082	8659.8	20271.7082	3048.71366	15	4	8	60	1.1	0.85	0.2
23	939.4	943.52	5380	0	1088.56507	877.284452	15	4	8	60	1.1	0.7	0.3
24	1958.42	1994.02	0.063	1958.357	2619.81072	1693.84151	15	4	8	60	1.1	0.7	0.2
25	15063.2	15065	0.77	15062.43	35198.2872	3130.93521	8	5	8	32	1.02	0.85	0.3
26	17934.6	18010.2	12620	5314.6	36574.9443	2966.84557	8	5	8	32	1.02	0.85	0.2
27	14928.2	14965.6	5520	9408.2	30849.8367	3349.17728	8	5	8	32	1.02	0.7	0.3
28	18207.4	18304.8	10440	7767.4	37921.027	2712.35704	8	5	8	32	1.02	0.7	0.2

Notation: DS = Datum Solution; APR = A posteriori error bound

Table 9.4: Numerical results for twenty-eight problems in problem set [1]