stichting

mathematisch

centrum

Σ
MC

M.A.H. DEMPSTER, M.L. FISHER, L. JANSEN,
B.J. LAGEWEG, J.K. LENSTRA, A.H.G. RINNOOY KAN

ANALYTICAL EVALUATION OF HIERARCHICAL PLANNING SYSTEMS

Preprint

2e boerhaavestraat 49 amsterdam

ANALYTICAL EVALUATION OF HIERARCHICAL PLANNING SYSTEMS

M.A.H. DEMPSTER (*Balliol College, Oxford/IIASA, Laxenburg*)

M.L. FISHER (*The Wharton School, University of Pennsylvania, Philadelphia*)

L. JANSEN (*Mathematisch Centrum, Amsterdam*)

B.J. LAGEWEG (*Mathematisch Centrum, Amsterdam*)

J.K. LENSTRA (*Mathematisch Centrum, Amsterdam*)

A.H.G. RINNOOY KAN (*Erasmus University, Rotterdam*)

ABSTRACT

Hierarchical planning systems have become popular for multi-level decision problems. After reviewing the concept of hierarchical planning and citing some examples, we describe a method for analytic evaluation of a hierarchical planning system. We show that multi-level decision problems can be nicely modeled as stochastic programs. Then any hierarchical planning system can be measured against the yardstick of optimality in this stochastic program. We demonstrate this approach on a hierarchical system that can be shown to be asymptotically optimal for a job shop design/scheduling problem.

1. INTRODUCTION

Many operations management planning and control problems require a series of decisions over time at an increasing level of detail. For example, there are at least two distinct decision making levels in most production operations. At the lowest level, detailed production scheduling decisions determine who will do a particular job on what machine and when. Considerations at this level include minimizing setups and meetings due dates. At a higher level, aggregate planning decisions are made concerning hiring and layoffs, over-time, production levels for product groups, ordering of raw materials, and due date setting. The time horizon for aggregate decisions ranges from a few months to one year. At the time aggregate decisions are made, much detailed information is not known with certainty. This can include future product demand, job processing times, machine breakdowns, worker availability, and raw material availability. In addition, other detail is intentionally ig-nored at the aggregate level. For example, the sequence dependent nature of setups is usually ignored, and product groups are used rather than individual stock-keeping units.

Hierarchical planning systems are becoming increasingly popular for multi-level decision problems like this. General discussions of hierarchical planning are given by Bitran and Hax [2] and Bradley, Hax and Magnanti [3,pp.212-213]. A hierarchical system uses separate mathematical programming models to make the decisions at each level. The solution of a higher level model creates a portion of the constraints for the model below it. Of course, there is nothing new about using interacting optimization models. What seems to be new in the hierarchical approach is the explicit emphasis on the link-ages between the models and on designing all models in the system simulta-neously so that they fit well together.

There are two fundamental reasons for using a hierarchical approach.
(1) *Reducing complexity*. Breaking a problem into subproblems is a standard method for simplifying the solution process. A tenet of hierarchical planning is that this partitioning can be done so that the interaction effects be-tween subproblems are acceptably weak.
(2) *Coping with uncertainty*. It is important to realize that the decisions at the various levels in the planning process need to be made at different

points in time. For example, aggregate planning decisions are made early enough to implement plans for hiring/layoff, raw materials acquisition, etc. On the other hand, a decision to assign a particular job to a specific machine can be postponed until the instant before the job begins processing. This is important in light of the fact that much data at the detailed level is uncertain at the time aggregate decisions are made. If detailed and aggregate decisions were combined in a single giant optimization model, as is sometimes proposed, the detailed decisions would be made earlier than necessary and hence would be based on less reliable forecasts of the uncertain data. The hierarchical approach postpones the detailed decisions as long as possible so that they can be based on more timely and hence more accurate data.

Past work in hierarchical planning has consisted mainly of building clever systems. This paper considers the question of how one decides whether one system is better than another. To obtain an objective standard of performance, we introduce a stochastic programming model that encompasses the entire multi-level decision process. At each level we model lower levels accurately, but with stochastic parameters. The objective at each level is to minimize current costs plus the expected objective value of an optimal lower level solution. The performance of a hierarchical planning system can be studied analytically by determining how close it comes to optimality in this stochastic programming model.

In Section 2 we provide a rich set of examples of hierarchical planning systems. Section 3 describes the analytic evaluation approach in detail. In Section 4 we illustrate this approach by analyzing a hierarchical system for one of the examples given in Section 2. This hierarchical system is shown to have the desirable property of being asymptotically optimal as the number of jobs in the problem approaches infinity. This result provides concrete support for the concept of hierarchical planning.

2. EXAMPLES OF HIERARCHICAL PLANNING SYSTEMS

Examples are provided of hierarchical systems for four types of problems. Specific implementations of each type are also referenced.

2.1. Aggregate/detailed production scheduling

This example has already been discussed at some length in the introduction. The aggregate planning problem is usually modeled as a multiperiod linear program with a planning horizon of about one year. The basic decision variables are target production and inventory levels by period for aggregated groups of products. These target levels are fed to the detailed production scheduling model which is concerned with scheduling the actual production of each stock-keeping unit over a relatively short horizon. This detailed model is usually an integer program solved by a heuristic. It may be run much more often than the aggregate model.

Hax and Meal [14] and Hax and Golovin [12;13] have designed and successfully implemented a system of this type. They distinguish three increasingly aggregated product units: items, families and types. At the type level, they use a linear program to set long range target production levels that minimize production, overtime, and inventory carrying costs. At the family level, the production target for a type is allocated to the families within that type considering setup costs. At the item level, item production is planned over a short planning horizon using recent demand data.

Jaikumar [15] has developed a system for the Booth Fisheries Division of Consolidated Foods that has many novel features and has achieved impressive economic results. At the aggregate level, a linear program is used to determine levels of production, marketing promotions, and raw material acquisitions that maximize sales revenue minus the cost of inventory holding, overtime, hiring/layoff, production and raw materials. This model has a 26 week planning horizon with time periods varying from one to six weeks. The detailed model is an integer program with a planning horizon of one week and time periods of two hours. This model allocates the production of particular products to individual production lines.

A novel feature of Jaikumar's system is the use of a dual method to

4

link the aggregate and detailed models. Dual variables on the production ca-
pacity, raw material and manpower constraints in the aggregate model are fed
to the detailed model and used in the objective function to cost out usage
of those resources.

2.2. Job shop design/scheduling

This problem is concerned with specifying the number of machines of various
types to have in a job shop and the scheduling of work on those machines to
minimize the total of machine costs and some job based measure of perfor-
mance such as average job tardiness. The first algorithmic approach to this
problem was given by Fisher [6].

In a hierarchical approach, the higher level decision is obviously how
many machines to have, and the lower level decision is how to sequence jobs
in the shop. Armstrong and Hax [1] and Shwimer [16] have described hierarchi-
cal systems that use an integer programming model and a simulation with an
embedded heuristic sequencing rule to make the higher level decision. The
lower level decision is made with the heuristic sequencing rule.

2.3. Distribution system design/control

The higher level problem includes the distribution design questions of where
to locate plants and warehouses, whether to expand capacity at existing
plants and warehouses, whether to install automatic materials handling equip-
ment, etc. The lower level problem concerns such questions as the allocation
of customers to warehouses and the determination of commodity flows through
the system.

Hax [11] describes a system that he developed in the aluminum industry.
The higher level problem is a linear program that is run a number of times
for different cases. In addition to the standard distribution system design
questions, this model is concerned with whether or not the firm should enter
into long term product swapping contracts with competitors. The lower level
problem is concerned with which sources should supply a set of orders in
hand. It is modeled as a linear program with a heuristic post-adjustment of
the solution to eliminate order splitting.

Glover et al. [9] describe a comprehensive production and distribution planning model developed for the Agrico Chemical Company. This is a network flow model explicitly concerned with the lower level decision of scheduling the flow of shipments from plants through the distribution system to warehouses. The model has also been successfully used in a case-study mode to analyze higher level decisions on capacity expansion and equipment modernization at the plants and warehouses.

Federgruen and Lageweg [5] describe a distribution system developed for a producer of industrial gases in the Netherlands. At the higher level, the number and locations of national distribution centers and the associated territories are determined. At the lower level, the commodity flows from distribution centers through regional depots to final customers are routed. The system is modeled as a network flow problem and uses a vehicle routing algorithm to calculate the routing costs.

2.4. Vehicle routing/scheduling

Many organizations operate vehicle fleets to deliver their products to customers. Frequently, for administrative convenience and other reasons, the fleet is scheduled using a fixed route system. In a fixed route system, particular customers are assigned to a vehicle and this assignment is revised infrequently (e.g., every 6-12 months). Daily customer orders are random. A customer orders on a given day with some probability, and the amount of any order is random. On any given day, a delivery schedule must be developed for the customers that have ordered for each vehicle. In this daily problem one can consider the possibility of delaying delivery of a customer order, or of using an alternative mode of delivery such as airfreight.

This problem suggests an obvious hierarchical system in which one model is used to assign customers to vehicles to form the fixed routes, and another model is used to solve the daily scheduling problem ("cluster first, route second"). Fisher and Jaikumar [8] have developed a partitioning algorithm for vehicle routing which follows the hierarchical structure just outlined. Customers are assigned to vehicles using a generalized assignment model. Each vehicle is scheduled using a traveling salesman model with side constraints. Golden and Yee [10] have also suggested a general framework for this problem.

3. A STOCHASTIC PROGRAMMING FRAMEWORK FOR EVALUATION OF HIERARCHICAL PLANNING SYSTEMS

Past work on hierarchical systems has concentrated on building clever systems, many of which were chronicled in the previous section. We are interested in this paper in the question of how one evaluates the quality of the decisions produced by a hierarchical system. One is interested both in comparing different systems and in absolute evaluation of a single system.

Some methods already exist for partially answering these questions. Different systems can be compared empirically by Monte Carlo simulation. In this approach, higher level models are run with forecasts of the uncertain lower level data. Lower level models are run with actual data values generated randomly by the Monte Carlo method. Hax and Golovin [13] have used this approach in evaluating their system for different settings of various parameters. One can also evaluate either by analytic or empirical methods the degree of optimality in the solutions to the submodel at each level.

All of these evaluation methods fail to answer the nagging question of how good a particular hierarchical system is when compared with an optimal system. To answer this we need a measure of optimality for the overall system, not just for each subproblem. Put differently, hierarchical systems are often called suboptimizing systems. If so, what is the optimization problem being suboptimized?

A little thought should make it clear that the answer to this question cannot be a deterministic mathematical programming model. Such a model could not encorporate the uncertainty that exists at lower levels of the global decision problem. This phenomenon can only be accurately modeled by a multi-stage stochastic program. The stochastic program would model lower levels accurately, but with stochastic parameters. The objective at each level is to minimize known costs at that level plus the expected objective value of an optimal lower level solution. For example, consider the form such a model would take for the example given in Section 2.2. At the time the job shop is designed, only probabilistic information is available on the jobs to be processed by the shop. A two-stage stochastic programming model of this problem would select the machine configuration of the job shop so as to minimize the cost of machines plus the expected cost (e.g., job tardiness

or flow time) of operating the shop optimally with the given machine configuration. Similar stochastic programs can be created for all of the other examples given in Section 2. We note that all of these models would have integer variables at one or more of the levels.

We are not suggesting that any of these models should be solved optimally. Such an endeavor would be foolhardy given the reputation for intractability enjoyed by both integer programming and stochastic programming. Rather, we suggest that a hierarchical system for a multi-level decision problem can be usefully viewed as a heuristic applied to a stochastic programming model of the problem. Analysis of heuristics is now a well-established technique in the area of integer programming (e.g., see Fisher [7]). There is no reason why this same technique could not be applied to stochastic programming heuristics. This would allow objective statements about how closely a particular hierarchical system approaches the ideal of optimality in the appropriate stochastic programming model. In the following section we describe the results of such a study for a simplified version of the job shop design/scheduling example given in Section 2.2.

4. ANALYSIS OF A HIERARCHICAL SYSTEM FOR A JOB SHOP DESIGN/SCHEDULING PROBLEM

We are given n jobs to be processed on identical parallel machines. The problem is to decide how many machines to buy and how to sequence the jobs on the machines to minimize machine cost plus the maximum job completion time. Job processing times are independently distributed random variables whose values all become known with certainty at time t = 0 *after* a decision has been made on the number of machines to buy.

Define

c = cost of a single machine,

m = number of machines to be bought,

p_j = processing time of job j,

$p = (p_1, \ldots, p_n)$,

$C^*(m,p)$ = earliest time at which all jobs are completed when scheduled optimally on m machines, beginning at t = 0, and with known processing times $p = (p_1, \ldots, p_n)$.

A tilde (~) under a variable will indicate that it is a random variable, and E will denote expected value.

The problem is to choose m prior to t = 0 to solve

$$z^* = \min_m \{cm + EC^*(m,\underset{\sim}{p})\}. \tag{1}$$

Let m^* denote the optimal solution to (1). Then at time t = 0, sequence n jobs with known processing times on m^* machines to achieve the optimal completion time $C^*(m^*,p)$.

There is a natural two-stage hierarchical system for this problem in which the higher level problem is problem (1) and the lower level problem is sequencing the jobs once processing times are known. It is not obvious how to solve (1) optimally. Determining $C^*(m,p)$ for fixed m and p is itself an NP-hard problem so finding $EC^*(m,\underset{\sim}{p})$ as a function of m seems virtually impossible. We circumvent this difficulty with an idea that is fundamental to all of the hierarchical systems described in Section 2. In solving the higher level problem we suppress the combinatorial fine structure of the lower level problem by replacing $C^*(m,p)$ with P/m, where $P = \sum_{j=1}^{n} p_j$. The

quantity P/m is a lower bound on $C^*(m,p)$ for any m and p. This bound is quite good if n is large and $p_{max} = \max_j\{p_j\}$ is sufficiently small. This replacement leads to the higher level problem

$$\min_m \{cm + E\underset{\sim}{P}/m\}.$$

The derivative of the objective function is zeroed by $m = \sqrt{(E\underset{\sim}{P}/c)}$. Since m must be a positive integer, we choose $m^H = \lceil \sqrt{(E\underset{\sim}{P}/c)} \rceil$, where $\lceil a \rceil$ denotes the smallest integer not less than a.

The lower level problem is the problem of sequencing n jobs with known processing times on m^H machines. In our hierarchical system, we simply solve this problem by list scheduling (LS). This heuristic assigns jobs in arbitrary order, placing each job on the machine that has the least processing already assigned. Let

$C^{LS}(m,p)$ = earliest time at which all jobs are completed when
scheduled by list scheduling on m machines, beginning
at $t = 0$, and with known processing times
$p = (p_1,\dots,p_n)$.

The overall value achieved by our hierarchical planning system is

$$Z^H = cm^H + EC^{LS}(m^H,\underset{\sim}{p}).$$

In [4] we show that

$$\frac{Z^H}{Z^*} \leq 1 + \frac{E\underset{\sim}{p}_{max}}{2\sqrt{(cE\underset{\sim}{P})}} = 1 + O(\frac{E\underset{\sim}{p}_{max}}{\sqrt{E\underset{\sim}{P}}}).$$

We also show that, if the $\underset{\sim}{p}_j$ have independent identical distributions with finite second moment, then the hierarchical planning system is asymptotically optimal in the sense that

$$\lim_{n\to\infty} \frac{Z^H}{Z^*} = 1.$$

Under the same assumption we have

$$\text{Plim}_{n\to\infty} \frac{cm^H + C^{LS}(m^H,\underset{\sim}{p})}{cm^* + C^*(m^*,\underset{\sim}{p})} = 1,$$

i.e., for any $\delta, \epsilon > 0$ there exists an n_0 such that for all $n > n_0$ the ratio between approximate and optimal solution value is no more than $1+\delta$ with probability at least $1-\epsilon$.

These results are extended in [4] to the case where the jobs are to be processed on uniform parallel machines (i.e., with different speeds) and the problem is to minimize machine cost plus the maximum job lateness with respect to a common constant due date.

ACKNOWLEDGMENTS

REFERENCES

1. R.J. ARMSTRONG, A.C. HAX (1977) A hierarchical approach for a naval-tender job-shop design. [3,Chapter 10].

2. G.R. BITRAN, A.C. HAX (1977) On the design of hierarchical production planning systems. *Decision Sci.* 8,28-55.

3. S.P. BRADLEY, A.C. HAX, T.L. MAGNANTI (1977) *Applied Mathematical Programming*, Addison-Wesley, Reading, Mass.

4. M.A.H. DEMPSTER, M.L. FISHER, L. JANSEN, B.J. LAGEWEG, J.K. LENSTRA, A.H.G. RINNOOY KAN (1980) Analysis of heuristics for stochastic programming: results for hierarchical scheduling problems. Report, Mathematisch Centrum, Amsterdam.

5. A. FEDERGRUEN, B.J. LAGEWEG (1980) Hierarchical distribution modelling with routing costs. Report BW 117, Mathematisch Centrum, Amsterdam.

6. M.L. FISHER (1969) A branch and bound algorithm for optimizing with a simulation model. M.S. Thesis, Sloan School of Management, Massachusetts Institute of Technology, Cambridge, Mass.

7. M.L. FISHER (1979) Worst-case analysis of integer programming heuristics. Decision Sciences Working Paper 79-10-01, University of Pennsylvania, Philadelphia; to appear in *Management Sci.*

8. M.L. FISHER, R. JAIKUMAR (1979) A generalized assignment heuristic for vehicle routing. Decision Sciences Working Paper 79-10-04, University of Pennsylvania, Philadelphia.

9. F. GLOVER, G. JONES, D. KARNEY, D. KLINGMAN, J. MOTE (1979) An integrated production, distribution, and inventory planning system. Research Report CCS 338, University of Texas, Austin.

10. B. GOLDEN, J. YEE (1979) A framework for probabilistic vehicle routing. *AIIE Trans.* 11,109-112.

11. A.C. HAX (1977) Integration of strategic and tactical planning in the aluminum industry. [3,Chapter 6].

12. A.C. HAX, J.J. GOLOVIN (1978) Hierarchical production planning systems. In: A.C. HAX (ed.) (1978) *Studies in Operations Management,* North-Holland, Amsterdam.

13. A.C. HAX, J.J. GOLOVIN (1978) Computer based operations management system (COMS). In: A.C. HAX (ed.) (1978) *Studies in Operations Management,* North-Holland, Amsterdam.

14. A.C. HAX, H.C. MEAL (1975) Hierarchical integration of production planning and scheduling. In: M.A. GEISLER (ed.) (1975) *Studies in Management Sciences, Vol. 1: Logistics*, North-Holland, Amsterdam.

15. R. JAIKUMAR (1974) An operational optimization procedure for production scheduling. *Comput. Oper. Res.* 1,191-200.

16. J. SHWIMER (1972) Interactions between aggregate and detailed scheduling in a job-shop. Ph.D. Thesis, Sloan School of Management, Massachusetts Institute of Technology, Cambridge, Mass.