MANAGEMENT SCIENCE
Vol. 20, No. 5, January, 1974
Printed in U.S.A.

MULTICOMMODITY DISTRIBUTION SYSTEM DESIGN BY BENDERS DECOMPOSITION*†

A. M. GEOFFRION AND G. W. GRAVES§

University of California, Los Angeles

A commonly occurring problem in distribution system design is the optimal location of intermediate distribution facilities between plants and customers. A multicommodity capacitated single-period version of this problem is formulated as a mixed integer linear program. A solution technique based on Benders Decomposition is developed, implemented, and successfully applied to a real problem for a major food firm with 17 commodity classes, 14 plants, 45 possible distribution center sites, and 121 customer zones. An essentially optimal solution was found and proven with a surprisingly small number of Benders cuts. Some discussion is given concerning why this problem class appears to be so amenable to solution by Benders' method, and also concerning what we feel to be the proper professional use of the present computational technique.

1. Introduction

1.1 *The Model*

The simplest version of the problem to be modeled is this. There are several commodities produced at several plants with known production capacities. There is a known demand for each commodity at each of a number of customer zones. This demand is satisfied by shipping via regional distribution centers (abbreviated DC), with each customer zone being assigned exclusively to a single DC. There are lower as well as upper bounds on the allowable total annual throughput of each DC. The possible locations for the DC's are given, but the particular sites to be used are to be selected so as to result in the least total distribution cost. The DC costs are expressed as fixed charges (imposed for the sites actually used) plus a linear variable charge. Transportation costs are taken to be linear.

Thus the problem is to determine which DC sites to use, what size DC to have at each selected site, what customer zones should be served by each DC, and what the pattern of transportation flows should be for all commodities. This is to be done so as to meet the given demands at minimum total distribution cost subject to the plant capacity and DC throughput constraints. There may also be additional constraints on the logical configuration of the distribution system.

The mathematical formulation of the problem uses the following notation.

i	index for commodities,
j	index for plants,
k	index for possible distribution center (DC) sites,

* Received August 15, 1973.

† An earlier version of this paper was presented at the NATO Conference on Applications of Optimization Methods for Large-Scale Resource Allocation Problems, Elsinore, Denmark, July 5–9, 1971. This research was partially supported by the National Science Foundation under Grant GP-36090X and the Office of Naval Research under Contract N00014-69-A-0200-4042. Reproduction in whole or in part is permitted for any purpose of the United States Government.

§ We wish to express our gratitude to Mr. Steven M. Niino, Manager of Operations Research at Hunt-Wesson Foods, Inc., for his outstanding contribution to the success of the practical application reported in this paper. We also want to thank Mr. Shao-Ju Lee of California State University at Northridge for his invaluable assistance in carrying out a difficult computer implementation

l index for customer demand zones,

S_{ij} supply (production capacity) for commodity i at plant j,

D_{il} demand for commodity i in customer zone l,

$\underline{V}_k$, $\bar{V}_k$ minimum, maximum allowed total annual throughput for a DC at site k,

f_k fixed portion of the annual possession and operating costs for a DC at site k,

v_k variable unit cost of throughput for a DC at site k,

c_{ijkl} average unit cost of producing and shipping commodity i from plant j through DC k to customer zone l,

x_{ijkl} a variable denoting the amount of commodity i shipped from plant j through DC k to customer zone l,

y_{kl} a 0–1 variable that will be 1 if DC k serves customer zone l, and 0 otherwise

z_k a 0–1 variable that will be 1 if a DC is acquired at site k, and 0 otherwise.

The problem can be written as the following mixed integer linear program.

$$\text{(1)} \qquad \text{Minimize}_{x \geqq 0;\, y,z=0,1} \textstyle\sum_{ijkl} c_{ijkl} x_{ijkl} + \sum_k [f_k z_k + v_k \sum_{il} D_{il} y_{kl}]$$

subject to

$$\text{(2)} \qquad \textstyle\sum_{kl} x_{ijkl} \leq S_{ij}, \qquad \text{all } ij$$

$$\text{(3)} \qquad \textstyle\sum_j x_{ijkl} = D_{il} y_{kl}, \qquad \text{all } ikl$$

$$\text{(4)} \qquad \textstyle\sum_k y_{kl} = 1, \qquad \text{all } l$$

$$\text{(5)} \qquad \underline{V}_k z_k \leq \textstyle\sum_{il} D_{il} y_{kl} \leq \bar{V}_k z_k, \qquad \text{all } k$$

(6) Linear configuration constraints on y and/or z.

The notation $y, z = 0, 1$ means that every component y_{kl} and z_k must be zero or one. It is understood that all summations run over the allowable combinations of the indices, since many combinations are either physically impossible (such as an ij combination which signifies a commodity that cannot be made at plant j) or so obviously uneconomical as not to merit inclusion in the model (such as a kl combination that would serve customers in Miami from a DC in Seattle).

The correspondence between this model and the verbal problem statement should be apparent. The quantity $\sum_{il} D_{il} y_{kl}$ is interpreted as the total annual throughput of the kth DC. Constraints (2) are the supply constraints, and (3) stipulates both that legitimate demand must be met (when $y_{kl} = 1$) and that x_{ijkl} must be 0 for all ij when $y_{kl} = 0$. Constraints (4) specify that each customer zone must be served by a single DC. Besides keeping the total annual throughput between $\underline{V}_k$ and $\bar{V}_k$ or at 0 according to whether or not a DC is open, (5) also enforces the correct logical relationship between y and z (i.e., $z_k = 1 \Leftrightarrow y_{kl} = 1$ for some l). Constraints (6) are deliberately not spelled out in detail for the sake of notational simplicity. The only requirement is that they be linear and do not involve any x-variables.

1.2 *Discussion of the Model*

There are several features of the model which warrant some discussion either to point out the flexibility they afford or to indicate the manner in which they differ from related models to be found in the literature.

The reader may have noticed that the transportation variables are quadruply subscripted, whereas previous intermediate location models (Bartakke et al. [2];

Ellwein and Gray [8 p. 296]; Elson [9]; Marks, Liebman and Bellmore [19]) employ separate transportation variables for plant-to-DC and DC-to-customer shipments. That is, we might have used two sets of triply subscripted variables (x_{ijk} and x_{ikl}, say) linked by a flow conservation constraint for each commodity-DC combination. This alternative suffers from a lack of flexibility for some applications because it "forgets" the origin of a commodity once it arrives at a DC. In the real application which sired the work reported in this paper, for instance, the so-called "storage-in-transit" privilege was a very important determinant of rail transportation costs for several of the commodities. A transit rate is figured as the direct plant-customer rate plus a nominal charge for stopping over at the DC which serves the customer, so long as this DC is not too far off the direct line. The transit rate is usually smaller than the simple sum of the plant-DC rate and the DC-customer rate. Obviously the x_{ijk} and x_{ikl} formulation cannot cope with the transit feature. Another advantage of the x_{ijkl} formulation over the x_{ijk} & x_{ikl} formulation arises when some commodities are perishable; it may be necessary to disallow the possibility of shipping such commodities over jkl routes for which the total journey times are likely to be excessive.

The quadruply subscripted transportation variables also make it easy to accommodate direct plant-customer zone shipments so long as a customer zone does not try to receive a given commodity both from a DC and a plant. For instance, suppose that a certain subset of customer zones is to obtain all commodities directly from the plants instead of via DC's. Then one simply adds a fictitious DC site k_0, say, with the associated z_{k_0} and y_{k_0l}'s fixed at unity, and specifies the rates c_{ijk_0l} appropriately for each associated ijl (there is no need for (5) to include a constraint for k_0). One may also accommodate the situation in which a customer zone obtains some commodities directly from the plants and the others through its DC. Just make the c_{ijkl}'s corresponding to the direct commodities independent of the possible *DC*'s for such a customer zone, and omit the il combinations corresponding to the directly shipped commodities from both $\sum_{il} D_{il}y_{kl}$ terms in the model.

Another unique feature of the model is that no customer zone is allowed to deal with more than one DC, since the y_{kl}'s must be 0 or 1 and not fractional. Thus each customer's demands must be satisfied by a single DC or directly from a producing plant (as described above). This assumption, which is required by the decomposition technique developed below, is frequently justified in practice. Our first-hand experience with three firms, each in a different industry, is that their accounting systems and marketing structures are geared to serving each customer zone from a single DC. Any change in this convention would be expensive both in terms of added administrative costs and in terms of less convenient service as perceived by customers. There would also be economic disadvantages due to reduced economies of scale in DC-to-customer shipments. Evidently a similar situation exists for other firms, as the desirability of this feature is frequently mentioned by other authors with practical experience [2], [6], [9], [10].

Notice that lower bounds as well as the customary upper bounds may be stipulated on warehouse throughput. This is useful for its own sake when there are reasons why each DC must be larger than a certain minimum size, and also to facilitate using a simple trick to permit a piecewise-linear representation of economies of scale and other nonlinearities (or even discontinuities) in DC costs as a function of throughput: simply introduce alternative DC's at a given site with different size ranges controlled by $\underline{V}_k$ and $\bar{V}_k$, with f_k and v_k specialized accordingly. For instance, a piecewise-linear DC cost function with three pieces would require three alternative DC's (small, medium and large) each with f_k and v_k dictated by the corresponding piece of the DC cost function. A simple configuration constraint can be included among (6) to ensure that at most one of the alternative DC's is opened at each site if this is not an automatic economic consequence of the model. The same trick also allows some economies-of-scale in transportation costs to be incorporated. This is especially useful for the in-bound (plant-to-DC) component of transportation costs for nontransit commodities. The larger the size range of an alternative DC, the lower should be the unit in-bound rates. The annual throughput of a DC has a much smaller influence on economies-of-scale for the out-bound rates, because the mode of transportation and delivery requirements are largely determined by the customers. This is especially true in view of the model assumption that each customer zone must be supplied by a single DC (the degree of consolidation of out-bound shipments is therefore relatively predictable for a given DC-customer zone pair).

The arbitrary configuration constraints (6) give the model quite a lot of flexibility to incorporate many of the complexities and idiosyncrasies found in most real applications. For instance, (6) permits:

- upper and/or lower bounds on the total number of open DC's allowed;
- specification of subsets of DC's among which at most one, at least one, exactly two, etc., are required to be open;
- precedence relations pertaining to the open DC's (not A unless B, etc.);
- mandatory service area constraints (if DC A is open, it must serve customer zone B);
- more detailed capacity constraints on the size of a DC than (5) permits, as by weighting the capacity consumption characteristics of each commodity differently or by writing separate constraints for individual or subsets of commodities;
- constraints on the joint capacity of several DC's if they share common resources or facilities;
- customer service constraints like

$$\left(\sum_{kl} t_{ikl} D_{il} y_{kl}\right) / \sum_l D_{il} \leq T_i ,$$

where t_{ikl} is the average time to make a delivery of commodity i to customer zone l after receiving an order at DC k, and T_i is a desired bound on the average delivery delay for commodity i.

A few additional remarks are in order concerning how the present model fits into the existing literature. Its chief ancestors are, of course, the well-known and much simpler "plant location" models (see Balinski and Spielberg [1, p. 268ff.]; Gray [16]; Ellwein [7] for surveys). These are basically single commodity transportation problems with fixed charges for the use of a source. Often the sources are assumed to have unlimited capacity. Recent work on capacitated problems of this type includes Davis and Ray [5], Ellwein and Gray [8], Fieldhouse [10], Geoffrion and McBride [13], Khumawala and Akinc [17], and Soland [21]. These authors all use branch-and-bound, which has emerged clearly as the most practical optimizing approach.

A natural extension of the capacitated plant location problem to the optimal location of intermediate facilities in multi-echelon systems has been studied by Marks, Liebman and Bellmore [19]. They report reasonably good computational experience with a conventional branch-and-bound algorithm in which the linear programs, which specialize to capacitated trans-shipment problems, are solved by an out-of-kilter routine. The same model is considered very briefly by Ellwein and Gray [8], who indicate that their capacitated plant location algorithm can be generalized to this case but give no computational experience.

If we now add the multicommodity feature, there appears to be no existing literature on special purpose optimizing algorithms. The only studies of multicommodity intermediate facilities location problems of which we are aware have used general purpose mixed integer linear programming systems. Bartakke et al. [2] describe an application of Bonner and Moore's Functional Mathematical Programming System for the Univac 1108 to an industrial problem with 4 plants, 4 commodities, 10 intermediate distribution sites with 3 possible sizes for each, and 39 customer points. It reportedly required 45 minutes of CPU time to optimize the resulting model with 210 rows, 30 binary variables and 1600 continuous variables. Elson [9] describes a specialized matrix generator and report writer for use in conjunction with the OPHELIE MIXED system for multicommodity intermediate location problems. Computational experience is given for one relatively small problem. The author refers to other computational experience with problems of similar size, from which he estimates that problems with 15 plants, 3 commodities, 45 DC sites, and 50 customer zones can be solved in about $8\frac{1}{2}$ system minutes on the CDC 6600 (assuming a 3:1 conversion ratio of billable system time to central processor time).

The reader who wishes to delve into the literature more deeply is encouraged to consult the excellent and massive (273 page) annotated bibliography on location-allocation systems prepared recently by Lea [18].

1.3 *Plan of the Paper*

§2 specializes Benders' well-known partitioning procedure to our problem in such a way that the multicommodity LP subproblem decomposes into as many independent classical transportation problems as there are commodities. This decomposition makes it possible to solve problems with virtually any number of commodities. Possible points of interest in this section include the technique used to recover the optimal multipliers for each LP subproblem from its analytically reduced and separated components, a variation of Benders' original procedure which has proven effective in this context, and some remarks on the reoptimization capability of this approach via the use of previously generated Benders constraints for revised problems.

§3 briefly describes a full-scale computational implementation which we have used to redesign the national distribution system of a major food firm. This application is discussed at some length in §4, with considerable stress placed on the importance of certain types of pre- and postoptimality runs to the professional success of this study. Actual computational experience is quoted in detail. The reader will be surprised, as we were, that in every run just a few iterations of Benders' procedure sufficed to find and verify a solution optimal to within a few tenths of one percent. Since this was also true for another large (unrelated) practical problem, it would seem that the class of problems studied herein is unusually amenable to solution by Benders' method.

§5 passes along a lesson learned from early computational experience concerning alternative logically equivalent model representations which are really not equivalent at all when solved by Benders Decomposition. We found that the representation used here is far superior to the natural more compact one we had tried earlier. This phenomenon is examined and implications emerge which may well be useful in other applications of Benders' method.

Some conclusions from our experience to date are offered in §6.

2. Application of Benders Decomposition

Most real-life applications of problem (1)–(6) are too large to be solved economically by existing general mixed integer linear programming codes [12]. The application addressed below had 11,854 rows, 727 binary variables and 23,513 continuous variables. The model does, however, have a conspicuous special property that enables it to be decomposed in such a way that the multicommodity aspect becomes much less burdensome: when the binary variables are temporarily held fixed so as to satisfy (4)–(6), the remaining optimization in x separates into as many independent classical transportation problems as there are commodities. This can be seen either from the physical interpretation of the problem or directly from (1)–(3). The transportation problem for the ith commodity is of the form

$$\text{Minimize } \textstyle\sum_{jl} c_{ij\bar{k}(l)l} x_{ij\bar{k}(l)l}$$

(7i) subject to

$$\begin{aligned} \textstyle\sum_{l} x_{ij\bar{k}(l)l} &\le S_{ij}, && \text{all } j \\ \textstyle\sum_{j} x_{ij\bar{k}(l)l} &= D_{il}, && \text{all } l \\ x_{ij\bar{k}(l)l} &\ge 0, && \text{all } jl, \end{aligned}$$

where $\bar{k}(l)$ is defined, for each l, as the k-index for which $y_{kl} = 1$ in the temporarily fixed y-array (by (4), $\bar{k}(l)$ is unique for each l).

The simplicity of the problem for fixed (y, z) suggests the application of Benders Decomposition [4]. A conventional specialization of this approach is given in §2.1, and the following section explains how the necessary multipliers of the full subproblem may be analytically synthesized from the multipliers of the reduced and separated subproblems (7i). §2.3 describes a variant of Benders' approach which we have found to be more suitable for computational purposes. Finally, the cost-saving reoptimization capability inherent in this approach is pointed out in §2.4.

2.1 *Specialization of Benders Decomposition*

Application of Benders Decomposition to (1)–(6) in the standard fashion leads to the following algorithm.

Step 0. Select a convergence tolerance parameter $\epsilon \ge 0$. Initialize $UB = \infty$, $LB = -\infty$, $H = 0$. If a binary array (y^1, z^1) satisfying (4), (5) and (6) is given, go to Step 2; otherwise, go to Step 1.

Step 1. Solve the current master problem

$$\text{(8)} \qquad \text{Minimize}_{y,z=0,1;\,y_0} \textstyle\sum_k [f_k z_k + v_k \sum_{il} D_{il} y_{kl}] + y_0$$

subject to (4), (5), (6) and

$$\text{(9)} \qquad y_0 + \textstyle\sum_{ikl} \pi^h_{ikl} D_{il} y_{kl} \ge -\sum_{ij} u^h_{ij} S_{ij}, \qquad h = 1, \cdots, H$$

by any applicable algorithm. Let $(y^{H+1}, z^{H+1}, y_0^{H+1})$ be any optimal solution. Put LB equal to the optimal value of (8), which is a *l*ower *b*ound on the optimal value of (1)–(6). Terminate if $UB \le LB + \epsilon$.

Step 2.

(a) Solve the linear programming subproblem

$$\text{(10)} \qquad \text{Minimize}_{x \ge 0} \textstyle\sum_{ijkl} c_{ijkl} x_{ijkl}$$

subject to (2) and (3)

with $y = y^{H+1}$ by any applicable algorithm. Denote the optimal value by $T(y^{H+1})$ and the optimal solution by x^{H+1}. Then the quantity

$$\text{(11)} \qquad \textstyle\sum_k [f_k z_k^{H+1} + v_k \sum_{il} D_{il} y_{kl}^{H+1}] + T(y^{H+1})$$

is an upper bound on the optimal value of (1)-(6). If (11) is less than UB, replace UB by this quantity, store $(y^{H+1}, z^{H+1}, x^{H+1})$ as the Incumbent, and terminate if $UB \leq LB + \epsilon$.

(b) Determine an optimal dual solution for (10) with $y = y^{H+1}$: denote it by u^{H+1} (corresponding to (2)) and π^{H+1} (corresponding to (3)). Increase H by 1 and return to Step 1.

A few remarks on this procedure are in order. First, note that an ϵ-optimal termination criterion has been used. The available upper and lower bounds on the optimal value of (1)-(6) coincide to within ϵ upon termination, at which time the Incumbent has been demonstrated to be ϵ-optimal in (1)-(6). Prior to termination it is known only that the Incumbent is within (UB-LB) of the optimal value. Finite convergence is assured for any $\epsilon \geq 0$.

Second, note that no provision is made at Step 2 for the possibility that (10) may be infeasible for some choices of y. This possibility can be handled easily within the standard framework of Benders Decomposition by slightly complicating the above algorithm, but we elect to preclude it here by assuming without loss of generality that $\sum_j S_{ij} \geq \sum_l D_{il}$ for all i (otherwise (1)-(6) is infeasible) and that all possible jk combinations are technically allowed (if j_0k_0 corresponds to an uneconomical route, take $c_{ij_0k_0l}$ equal to any comparatively large number). It is not difficult to verify that these innocuous assumptions imply that (10) is feasible and has a finite optimal solution for every binary y satisfying (4).

Third, as indicated previously, the LP subproblem (10) is most easily solved by solving an equivalent collection of independent classical transportation problems—one for each commodity. This can be demonstrated by observing that since y^{H+1} satisfies (4), (3) implies

$$x_{ijkl}^{H+1} = 0 \text{ for all } ijkl \text{ with } k \neq \bar{k}(l)$$

where $\bar{k}(l)$ is the k-index for which $y_{kl}^{H+1} = 1$. Thus (10) simplifies to

$$\text{Minimize } \sum_i \left(\sum_{jl} c_{ij\bar{k}(l)l} x_{ij\bar{k}(l)l}\right)$$

subject to

$$\begin{aligned} \sum_l x_{ij\bar{k}(l)l} &\leq S_{ij}, && \text{all } ij \\ \sum_j x_{ij\bar{k}(l)l} &= D_{il}, && \text{all } il \\ x_{ij\bar{k}(l)l} &\geq 0, && \text{all } ijl. \end{aligned}$$

This problem obviously separates on i into independent transportation problems of the form (7i). If the optimal value of (7i) is denoted by $T_i(y^{H+1})$, then $T(y^{H+1}) = \sum_i T_i(y^{H+1})$.

The reduction of (10) to independent problems of the form (7i) greatly simplifies Step 2a, but Step 2b then becomes less straightforward. The required optimal dual solution for (10) must be synthesized from the optimal dual solutions of (7i). The relationship between the optimal primal solutions of (10) and (7i) is obvious, but the relationship between the optimal dual solutions requires some analysis. This analysis is as follows.

2.2 *Details on Step 2b*

Step 2b requires an optimal dual solution (u^{H+1}, π^{H+1}) to (10) with y fixed at y^{H+1}. Since (10) is solved via (7i) rather than directly, the required dual solution must be synthesized from the available dual optimal solutions to (7i).

For notational simplicity, the superscript $H + 1$ will be replaced by an overbar (e.g., y^{H+1} becomes $\bar{y}$). Denote the available optimal dual variables of (7i) by $\bar{\mu}_{ij}$ (corresponding to the supply constraints) and $\bar{v}_{il}$ (corresponding to the demand constraints). It will be shown that the appropriate formulae to be used at Step 2b are:

$$\text{(12a)} \qquad \bar{u}_{ij} = \bar{\mu}_{ij}, \qquad \text{all } ij$$

$$\text{(12b)} \qquad \bar{\pi}_{ikl} = \text{Max}_j \{-\bar{\mu}_{ij} - c_{ijkl}\}, \quad \text{all } ikl.$$

To derive (12), one must compare the duals of (10) with those of (7i), where y is fixed at $\bar{y}$. The dual of (10) is

$$\text{Maximize}_{u \geqq 0;\, \pi} \sum\nolimits_{ikl} \pi_{ikl}(-D_{il}\bar{y}_{kl}) + \sum\nolimits_{ij} u_{ij}(-S_{ij})$$

(13) subject to

$$-u_{ij} - \pi_{ikl} \leq c_{ijkl}, \quad \text{all } ijkl.$$

Notice that for any fixed u, the optimal choice of π is obvious since there are no joint constraints on π and each π_{ikl} is constrained only from below by the bound

$$b_{ikl}(u) \triangleq \text{Max}_j \{-u_{ij} - c_{ijkl}\}.$$

If $(-D_{il}\bar{y}_{kl}) < 0$ then the best choice of π_{ikl} is $b_{ikl}(u)$, while if $(-D_{il}\bar{y}_{kl}) = 0$ then the optimal choice is any number greater than or equal to $b_{ikl}(u)$.

Notice also that when $(-D_{il}\bar{y}_{kl}) = 0$, as when $k \neq \bar{k}(l)$, the corresponding constraints may simply be dropped from (13) since they may always be satisfied without any effect on the value of the objective function. Thus (13) is equivalent to

$$\text{Maximize}_{u \geqq 0;\, \pi_{i\bar{k}(l)l}, \forall il} \sum\nolimits_{il} \pi_{i\bar{k}(l)l}(-D_{il}\bar{y}_{\bar{k}(l)l}) + \sum\nolimits_{ij} u_{ij}(-S_{ij})$$

(14) subject to

$$-u_{ij} - \pi_{i\bar{k}(l)l} \leq c_{ij\bar{k}(l)l}, \quad \text{all } ijl,$$

with the understanding that for ikl with $k \neq \bar{k}(l)$, $\bar{\pi}_{ikl}$ is any number greater than or equal to $b_{ikl}(\bar{u})$.

Now consider the duals of (7i) for each i, which may be combined into a single linear program since there are no variables in common. That is, $(\bar{\mu}, \bar{v})$ is an optimal solution of

$$\text{Maximize}_{\mu \geqq 0;\, v} \sum\nolimits_i \left[\sum\nolimits_j \mu_{ij}(-S_{ij}) + \sum\nolimits_l v_{il}(-D_{il})\right]$$

(15) subject to

$$-\mu_{ij} - v_{ij} \leq c_{ij\bar{k}(l)l}, \quad \text{all } ijl.$$

Comparison of (14) and (15) reveals that these are identical optimization problems (remember that $\bar{y}_{\bar{k}(l)l} = 1$), and hence the choice

$$\text{(16a)} \qquad \bar{u}_{ij} = \bar{\mu}_{ij}, \quad \text{all } ij$$

$$\text{(16b)} \qquad \bar{\pi}_{i\bar{k}(l)l} = \bar{v}_{il}, \quad \text{all } il$$

is optimal in (14). In view of the previous discussion, we also have the following necessary (given (16a)) and sufficient condition on the remaining $\bar{\pi}_{ikl}$'s:

$$\text{(16c)} \qquad \bar{\pi}_{ikl} \geq \text{Max}_j \{-\bar{\mu}_{ij} - c_{ijkl}\}, \quad \text{for all } ikl \text{ with } k \neq \bar{k}(l).$$

Relations (16a)–(16c) give the desired complete optimal solution to (13). Since (16a) is identical to (12a), it remains but to reduce (16b) and (16c) to the form (12b).

Relation (16c) is easily converted to the form of (12b) by selecting $\bar{\pi}_{ikl}$ in (16c) to be as small as possible, that is, so that equality holds—for, by the nonnegativity of $D_{il}y_{kl}$ in (9), this gives the best approximation to the optimal transportation cost function T. Second, by inspection of (15) we see that

$$\text{(17a)} \qquad \bar{v}_{il} = \text{Max}_j \{-\bar{\mu}_{ij} - c_{ij\bar{k}(l)l}\}, \quad \text{all } il\text{: } -D_{il} < 0$$

$$\text{(17b)} \qquad \bar{v}_{il} \geq \text{Max}_j \{-\bar{\mu}_{ij} - c_{ij\bar{k}(l)l}\}, \quad \text{all } il\text{: } -D_{il} = 0.$$

We may assume without loss of generality that equality holds in (17b), for if not then one may simply redefine $\bar{v}_{il}$ so that it does hold without upsetting the optimality of $(\bar{\mu}, \bar{v})$ in (15). Hence

$$\bar{v}_{il} = \text{Max}_j \{-\bar{\mu}_{ij} - c_{ij\bar{k}(l)l}\}, \quad \text{all } il,$$

which shows that (16b) reduces to (12b) and concludes the proof of (12).

2.3 *The Variant Actually Used*

There are numerous variants of the pure Benders Decomposition algorithm described in §2.1. One variant of particular interest is not to solve the current master problem at Step 1 to optimality, but rather to stop as soon as a feasible solution to it is produced which has value below $UB\text{-}\epsilon$. This implies, of course, that the master problem no longer produces a lower bound on the optimal value of (1)–(6) and so LB must be inactivated. The termination criterion of Step 2a must be deleted and that of Step 1 must be replaced by: "terminate if the current master problem has no feasible solution with value below $UB\text{-}\epsilon$; the current Incumbent is an ϵ-optimal solution of (1)–(6)."

It is not difficult to see that this variant must converge to an ϵ-optimal solution within a finite number of iterations. This follows from the finiteness of the number of dual solutions of (10) and from the easily verified fact that if any dual solution should be produced more than once at Step 2b, then the Incumbent must be improved by at least ϵ at each such repetition. There can be no more than a finite number of repetitions because the optimal value of (1)–(6) is bounded below.

The principal motivation behind this variant is that the early master problems have too little information about transportation costs to be worth optimizing very strictly. It takes several "Benders cuts" of the form (9) in order to give accurate information concerning these costs. This suggests that the master problems should be *sub*optimized, particularly when H is small. The degree of optimality achieved by this variant increases with H for two reasons: the minimal value of the master problem increases as H increases due to the accumulation of cuts, and the threshold $UB\text{-}\epsilon$ decreases each time an improved Incumbent is found.

A second motivation is that the variant's master problems are feasibility-seeking only:

$$\text{Find} \quad y, z = 0, 1 \quad \text{and} \quad y_0 \quad \text{to satisfy (4), (5), (6), (9) and}$$
$$\textstyle\sum_k [f_k z_k + v_k \sum_{il} D_{il} y_{kl}] + y_0 \leq UB\text{-}\epsilon$$

or, equivalently upon elimination of y_0,

$$\text{Find} \quad y, z = 0, 1 \quad \text{to satisfy (4), (5), (6) and}$$

$$\text{(9a)} \quad \textstyle\sum_k [f_k z_k + v_k \sum_{il} D_{il} y_{kl}] - \sum_{ij} u^h_{ij} S_{ij} - \sum_{ikl} \pi^h_{ikl} D_{il} y_{kl} \le UB\text{-}\epsilon, \qquad h = 1, \cdots, H.$$

Thus we may literally introduce any appealing (linear) objective function, say $\phi(y, z)$, and take the master problem to be:

$$\text{(8a)} \qquad \text{Minimize}_{y,z=0,1}\, \phi(y, z) \quad \text{subject to (4), (5), (6) and (9a).}$$

It is not necessary to optimize (8a), of course, but merely to produce a feasible solution if one exists. The choice of ϕ should be so as to encourage the production of useful feasible solutions. We have found the last (Hth) function appearing on the left-hand side of (9a) to be a good choice in practice.

We remark that (8a) is a *pure* 0–1 integer program, whereas (8) is a mixed integer program due to the appearance of y_0. This gives (8a) the added advantage of being somewhat more convenient to work with.

2.4 *Re-Optimization*

One of the advantages of the Benders Decomposition approach is that it offers the possibility of making sequences of related runs in considerably reduced computing times as compared with doing each run independently. The need for multiple runs is particularly acute in distribution system design studies because of the great economic consequences of the final solution, the difficulties of ascertaining future demands and costs with precision, and other reasons discussed at some length in §4.2.

The reoptimization capability of Benders' approach is due to the fact that the cuts (9a) generated to solve one problem can often be revised with little or no work so as to be valid in a modified version of the same problem. Assume for a moment that this is so. Then the modified problem can be started with these old (possibly revised) cuts included in the initial master problem and each master thereafter. If the optimal (y, z) solution of the modified problem is not too far from the optimal (y, z) solution of the original problem, then one would expect termination of the procedure in fewer major iterations than would be the case if it were begun from scratch.

The revision of cuts so as to be valid in a modified version of the problem is an easy matter so long as the c_{ijkl} coefficients do not decrease. This limitation is due to the requirement that (u^h, π^h) in (9) and (9a) must be feasible in the dual subproblem (13) corresponding to the modified version. Thus, increasing some c_{ijkl}'s and making arbitrary changes in the $\underline{V}_k$'s and $\bar{V}_k$'s and in the configuration constraints (6) require no revisions at all in (9a); except, of course, that appropriate values of UB and ϵ must be used. Changing an f_k or v_k is easily accomplished by a simple revision formula. Changing an S_{ij} or D_{il}, on the other hand, requires forethought in that the u_{ij}'s and π_{ikl}'s themselves enter into the revision formulae; normally these duals will not be saved since there is no need for them once a cut is calculated. Saving the u_{ij}'s poses no particular problem because the number of allowable ij combinations is relatively small in most applications. This would permit arbitrary changes in the S_{ij}'s. Saving all of the π_{ikl}'s would be burdensome storage-wise, so it is best to reconstruct them from the u_{ij}'s via (12). Thus arbitrary changes in the D_{il}'s are only slightly more difficult to accommodate than changes in the S_{ij}'s.

The usefulness of this reoptimization capability is indicated by the computational experience presented in §4.3.

3. Computer Implementation

An elaborate all-FORTRAN implementation has been carried out for the variant of Benders Decomposition described in §2. The objective of solving large problems in moderate computing times required the use of efficient algorithms for solving the master problems and subproblems, and careful data management techniques. These matters are discussed briefly in this section.

3.1 *Master Problem*

The master problems, of the form (8a), are pure 0–1 integer linear programs with a variable for every allowable DC-customer zone combination (y_{kl}) and for every possible DC site (z_k). Typically this leads to at least several hundred binary variables. Thus it was necessary to devise a specialized method which exploits the special structure of (8a). The method we employ is a hybrid branch-and-bound/cutting-plane approach with numerous special features.

The cuts employed are the original mixed integer cuts proposed by Gomory in 1960, and are applied to each node problem in order to strengthen the LP bounds and to drive variables toward integer values in preparation for the choice of a branching variable. Absolute priority is given to z-variables over y-variables in branching. Reversal bounds are calculated for variables which are branched upon using relaxed versions of (8a) which drop the integrality requirements on y (while keeping the integrality requirements on z) and transfer a linear combination of all constraints except (5) and individual variable bounds up into the objective function [11]. The multipliers which determine the linear combination are the appropriate dual variables of a node problem solved as a linear program (ignoring the integrality requirements on both y and z).

The linear programming subroutine takes full advantage of the generalized upper bounding constraints (4), and also exploits certain other aspects of the problem structure. It economizes on the use of core storage by generating columns as needed from compactified data arrays.

Finally, it should be mentioned that a number of logical relationships between the variables are built in at various points of the master problem algorithm so as to detect several kinds of infeasibility and "fix" the free variables when this is justified.

3.2 *Subproblem*

The transportation subproblems (7i) are solved using a new primal simplex-based algorithm with factorization (Graves and McBride [15]). Contrary to the conventional wisdom, such methods are superior to out-of-kilter type algorithms for most network flow applications [14], [15]. This is certainly true for the present application, where only the costs of the transportation subproblems change between successive solutions. An earlier implementation using an out-of-kilter algorithm was an order of magnitude slower on the average.

3.3 *Data Input and Storage*

Core storage requirements are economized by extensive use of overlay, cumulative indexing, and the creation of compact data sets from which model coefficients can be generated conveniently as needed. Most of the larger of these data sets are kept on disk. Raw problem data pertaining to permissible *ijkl* combinations, transportation costs, and customer demands are input from tape to a preprocessor program which creates the appropriate data sets on disk. These are then accessed directly by the main program, which receives the rest of the problem data (S_{ij}, $\underline{V}_k$, $\bar{V}_k$, f_k, v_k and configuration data for (6)) from direct keyboard input using the URSA conversational CRT-display remote job entry system at UCLA. The editing and scope display facilities of URSA make this an ideal means of entering and revising all but bulk data. Matrix generation and similar chores are accomplished entirely by the preprocessor and main programs.

The specific types of configuration constraints (6) accommodated in the current program include: fixing selected y_{kl} and z_k variables at specific values to set up regional or otherwise reduced versions of the full problem; mutual exclusivity constraints on DC sites; mandatory service area constraints for each DC; and a limit on the maximum number of DC's that may be open.

Newly generated cuts are stored on disk for use in the reoptimization mode described in §2.4. The last primal transportation solution is also stored on disk to serve as an advanced start in subsequent runs for which it is still feasible.

4. Solution of a Large Practical Problem

4.1 *Overview*

Hunt-Wesson Foods, Inc., produces several hundred distinguishable commodities at 14 locations (Wesson refineries, Hunt canneries, and co-packers) and distributes nationally through a dozen distribution centers. The firm decided in 1970 to undertake a thorough study of its distribution system design with particular emphasis on the question of distribution center locations. The study was prompted both by the need to resolve several expansion and relocation issues that had arisen, and by the recognition that a systematic global study of the entire distribution system would be likely to disclose opportunities for improvement that could not be identified by conventional analyses of individual cases and geographic regions.

The primary outcome of the study was that five changes were recommended in the firm's configuration of distribution centers (the movement of existing DC's to different cities and the opening of new DC's). The three most urgent of these changes have been carried out as of this writing and the other two are in process. The realizable annual cost savings produced by the study are estimated to be in the low seven figures.

§4.2 describes the various types of computer runs needed to carry out the study. Actual computational experience is summarized in §4.3.

4.2 *Eight Types of Computer Runs*

It is obvious that most distribution system design problems are of sufficiently major economic consequence to warrant the most careful computational treatment. Yet we were surprised by the large number of runs needed to deal properly with the various aspects of a real application. No less than 8 different types of runs can be distinguished, each of which may require several—sometimes many—distinct submissions:

- probationary exercises
- regional optimization
- global optimization
- "what if . . . ?"
- sensitivity analysis
- continuity analysis
- tradeoff analysis
- priority analysis.

For obvious reasons we cannot go into detail on all of these phases of the study, but we would like to make some general remarks on each in the light of our experience.

The purpose of the probationary exercises is to expose any possible shortcomings of the model, data, or computer code that may compromise their managerial usefulness. They must be regarded as "on probation" until proven otherwise, no matter how meticulous have been the data verification and program debugging efforts. A series of exercises is required in which the computer competes with management in carefully designed decision situations. Each situation must be limited in scope, as by restricting the number of free optimization variables, so that its complexity does not overwhelm the managers' ability to apply experience and familiar analytical techniques—yet it should be broad enough to exercise a significant portion of the model. The computer's solution and the managers' solution must be compared and any significant discrepancies must be reconciled, by hand calculation if necessary. For instance, it is useful to run the problem locked in to the current configuration of distribution centers so that the only optimization required is service area design and transportation flows. The series of exercises should involve each part of the model at least once. In this fashion the model, data and computer code truly earn their credibility.

Regional optimizations focusing on natural geographical regions are bridges between probationary exercises and global optimizations runs, to help tune internal algorithmic parameters and tactics while producing useful results. Four such regions were sufficient in our application.

Far from being the climax of a study, a global optimization run with all decision variables free requires considerable further study to confirm its validity and enhance its usefulness. It spawns additional runs to answer management's many inevitable "what if . . . ?" questions (what if a certain DC were kept open, or a certain customer zone were serviced by another DC, or a better rail rate negotiated here or DC lease there, etc.). It also raises questions concerning the sensitivity of the optimal solution to variation of the data. The need to address such questions is taken for granted in applications of linear programming but they are often slighted in large-scale integer programming applications—presumably on the grounds of their excessive computational cost. Our experience, however, is that such runs are indispensable as a source of useful insight into the behavior of the model and its tolerance for estimation errors. For instance, they revealed a serious error made during the initial formulation of the model concerning the specification of the lower limits $\underline{V}_k$ on distribution center throughput. Runs done using demand projected for several years beyond the primary target period of the study gave reassurance that dynamic factors were not unduly difficult to cope with via the static model used here.

Continuity analysis is similar to sensitivity analysis except that the purpose is to discover a possible pathology which cannot arise for ordinary linear programming models. We are referring to the possibility that a small change in the data may induce a sudden incommensurately large decrease in the optimal value of (1)-(6), a situation to which a modeler is likely to be quite averse since almost any datum can be changed by a small amount for a commensurately small cost (see Williams [23]). This situation can occur when the data changes lead to a discontinuous change in the feasible region. Changes in data appearing only in the objective function (1) [i.e., in the c_{ijkl}, v_k and f_k coefficients] cannot lead to such behavior. The other data should be checked by doing a run which relaxes each such coefficient somewhat; that is, each $\underline{V}_k$ should be decreased and each $\bar{V}_k$ and S_{ij} should be increased (it can be shown that relaxation of the $\underline{V}_k$'s and $\bar{V}_k$'s precludes the need to perturb the D_{il}'s). If the decrease in the

TABLE 1

Priority Analysis Results

DC Locations			Service Areas & Transport.	Total Cost	Differences	
OPT	Optimum (6 changes)		Optimum	100.00		
A	Current		Current	103.15		
B	Current		Optimum	101.43	1.72 save over A	
B.1		1	"	101.45	−0.02	
B.2		2	"	101.34	0.09	
B.3	Current & One	3	"	101.14	0.29	save
B.4	Change:	4	"	101.42	0.01	over
B.5		5	"	101.37	0.06	B
B.6		6	"	100.71	0.72	
B.7	Current & Best	1	"	100.01	0.01	loss
B.8	Subset of Changes	2	"	100.12	0.12	over
B.9	Omitting:	4	"	100.13	0.13	OPT
C	Current & Changes 3, 5, 6		Optimum	100.30	1.13 save over B	
C.1	Current & Changes	1	"	100.29	0.01	save
C.2	3, 5, 6 and Also:	2	"	100.17	0.13	over
C.3		4	"	100.13	0.17	C
D	Current & Changes 2, 3, 4, 5, 6		Optimum	100.01	0.29 save over C	

optimal value of (1)–(6) is excessively large by comparison with the estimated economic cost of changing these coefficients,[1] then additional more specific runs must be undertaken to localize the source of difficulty. A managerial decision would then have to be made concerning possible revisions of the problem data or even of the model itself. No serious discontinuities were detected for this application.

Tradeoff analysis runs are appropriate when there are other major quantifiable criteria besides cost in evaluating the desirability of a given distribution system design. Perhaps the most important secondary criterion is the quality of customer service as it depends upon the distance between a DC and the customer zones it serves. One possibility is to adopt the average delivery delay criterion suggested in §1.2 and to solve the problem with successively tighter T_i's. In this manner one may generate the tradeoff curve between total distribution cost and the average delivery delay for any given product or weighted combination of products.

The last type of run on the list is priority analysis. When a study reaches the point where management is ready to consider practical implementation of the results, it is useful to distinguish the aspects of the solution yielding the largest savings from those of relatively marginal significance. Runs done to help refine this distinction suggest which aspects of the solution most urgently call for implementation and which should be postponed or even dropped as too marginal to be worth the organizational upset. In the present application this mainly involved trying to assess the relative economic value of each of the major changes recommended for the distribution center configuration then extant. The actual process is summarized in Table 1, which focuses on the distribution center locations because these are the decisions of primary managerial

[1] Only the coefficient changes actually required for feasibility of the new solution would, of course, enter into this estimation.

concern. As the first row indicates, the optimal DC configuration can be viewed as requiring 6 changes to the current (1970) configuration. Some of the changes require relocating an existing DC to a different city and the others require opening a new DC. The total distribution costs corresponding to the optimal configuration are normalized to 100. Row A gives the relative total cost corresponding to the current DC configuration and also the current service areas and transportation flows. Row B retains the current DC configuration but optimizes the service areas and transportation flows. Notice that slightly more than half of the total possible savings could be achieved by service area and transportation flow realignments alone.

Now comes the analysis of the relative value of each of the 6 changes, from which some subset is to be selected for implementation. Runs B.1–B.6 indicate the savings of each change if done individually. Changes 3 and 6 appear to be very attractive, changes 2 and 5 only moderately attractive, and changes 1 and 4 unattractive. Change 5, however, was quite appealing to management on the grounds that it would give additional warehousing space in a region of the country where space was in particularly short supply. Management therefore was inclined to give top implementation priority to changes 3, 5 and 6. This inclination was supported by the results of runs B.7–B.9, which examine the effect of omitting one of the *other* changes and selecting the best subset of the remainder. It turned out that changes 3, 5 and 6 were among those selected in every case. Top priority was therefore given to changes 3, 5 and 6 which, row C reveals, jointly save a little more than one would expect from simply adding their individual savings (1.13 versus 1.07). Changes 1, 2 and 4 were examined again individually given the acceptance of 3, 5 and 6. Changes 2 and 4 now look quite attractive, while 1 continues to be borderline. This conclusion is supported by runs B.7–B.9 because the same results would have been obtained if changes 3, 5 and 6 had been mandatory in these runs. In light of this analysis and of factors outside the scope of the model, management gave second priority to changes 2 and 4. Change 1 was considered too marginal for implementation. Row D shows that changes 2 through 6 are only 1/100 of 1% away from the system optimum.

4.3 *Computational Performance*

This section summarizes the code's computational performance on the Hunt-Wesson problem. All computing times refer to UCLA's IBM 360/91.

Table 2 presents ten representative runs without use of the reoptimization technique discussed in §2.4, and three with it (labeled R). None of these runs incorporated any type (6) configuration constraints beyond the locking open or closed of certain distribution center sites, so that the reader would be assured that the problem was not so severely constrained as to greatly facilitate optimization (our experience has been that while configuration constraints do tend to make the problem easier, the influence on computing times is rarely dramatic). Runs 6 and 7 are identical except for the specification of ϵ. Runs 8 and 9 are identical except that the $\underline{V}_k$'s were all 10% higher in run 8. Runs 2R, 3R and 6R are identical to runs 2, 3 and 6 respectively, except that each was initiated using all of the cuts generated by runs 1, 5 and 4, respectively. The largest number of free DC sites in any of these runs is 30 because the remaining sites were determined to be dominated as obviously uneconomical during the probationary exercises and regional optimizations.

The most striking conclusion to be drawn from Table 2, and indeed from our entire computational experience, is the surprisingly small number of iterations required for convergence even with very small values of the optimality tolerance ϵ. The num-

TABLE 2

Representative Runs

Run No.	DC's		Free 0-1 Variables[a]	Rows	ϵ (%)[b]	Major Iter.	Execution Time (Sec.)[c]
	Free	Locked Open					
1	0	16	249	4,403	0.06	3	16.7
2	0	16	254	4,488	0.03	4	23.8
2R	0	16	254	4,488	0.03	4	16.6
3	7	11	287	4,944	0.03	5	25.5
3R	7	11	287	4,944	0.03	4	17.5
4	15	4	336	5,657	0.06	4	23.2
5	20	1	349	5,783	0.15	4	24.9
6	20	5	411	6,857	0.06	7	50.5
6R	20	5	411	6,857	0.06	5	38.1
7	20	5	411	6,837	0.15	4	29.4
8	25	1	427	7,054	0.15	5	43.8
9	25	1	427	7,054	0.15	5	37.7
10	30	1	513	8,441	0.15	5	191.0

(*Notes*: (a) z_k's corresponding to the free DC's plus y_{kl}'s corresponding to DC's either free or locked open; (b) percentage of the optimal total cost; (c) in addition to execution time, each run required about one second of link editing time.)

ber of iterations increases only slowly with the size of the problem. Some partial explanations for this fortunate state of affairs are offered in the next section.

Table 3 gives further details on the runs listed in Table 2. For convenience, the optimal value of each run is normalized to 100. The difference between the "total" and "master" columns is the time at each major iteration spent extracting and solving the 17 transportation problems plus cut generation time. About half of this time, which runs quite consistently around 5 seconds, is spent performing the extraction from the data sets on disk.

From Table 3 it can be seen that suboptimizing the master problem as described in §2.3 is generally successful in helping to keep the time spent on it quite small. As one might expect, the final master problem tends to be relatively difficult for the larger problems. Notice also that the actual cost of the designs produced by the successive master problems usually (but not always) improves monotonely. Finally, we can observe that reoptimization saves computing time but not necessarily major iterations, and that it tends to yield a good first design.

It should be emphasized that the same standard internal and external parameter settings have been used in all of the runs. This was done in the interest of comparability. But, obviously, many useful alternatives exist which may lead to improved performance in specific cases. For instance, gradually reducing ϵ at each major iteration is a more effective way to achieve a desired low final ϵ at termination than keeping it constant. Initializing UB at a good known upper bound less than $+\infty$ is also possible and beneficial in most runs. And selectivity in choosing which prior cuts to use for reoptimization is helpful. All such ad hoc adjustments have been avoided here.

5. A Lesson on Model Representation

Anyone accustomed to working with linear programming applications is inclined to economize on the number of constraints he uses in a large-scale model. The model (1)–(6) presents an obvious opportunity to economize on the number of type (3)

TABLE 3

Detailed Results for the Runs of Table 2

Run No.	Major Iteration	Value of Design from Current Master (11)	Execution Time (Sec.)	
			Master	Total
1	1	103.51	5.8	11.5
	2	100.00	0.2	5.1
	3	Termination	0.1	0.1
2	1	102.78	7.5	13.1
	2	100.00	0.2	5.3
	3	100.01	0.6	5.1
	4	Termination	0.3	0.3
2R	1	100.02	0.9	6.9
	2	100.04	0.2	4.5
	3	100.00	0.3	5.0
	4	Termination	0.2	0.2
3	1	102.96	3.7	9.4
	2	100.04	0.2	5.3
	3	100.01	0.4	5.4
	4	100.00	0.3	5.3
	5	Termination	0.1	0.1
3R	1	100.04	0.9	6.9
	2	100.02	0.4	5.1
	3	100.00	0.5	5.3
	4	Termination	0.2	0.2
4	1	102.00	1.3	6.9
	2	100.01	0.5	5.3
	3	100.00	3.7	8.6
	4	Termination	2.4	2.4
5	1	101.94	1.3	6.8
	2	100.30	0.5	5.5
	3	100.00	5.4	10.4
	4	Termination	2.2	2.2
6	1	102.95	1.4	7.2
	2	100.40	0.6	5.7
	3	100.35	2.5	7.5
	4	100.29	2.6	7.5
	5	100.19	1.1	6.1
	6	100.00	0.3	5.3
	7	Termination	11.2	11.2
6R	1	100.39	0.8	7.3
	2	100.34	0.2	5.0
	3	100.30	5.9	10.8
	4	100.00	0.9	6.0
	5	Termination	9.0	9.0

TABLE 3 (*Continued*)

Run No.	Major Iteration	Value of Design from Current Master (11)	Execution Time (Sec.)	
			Master	Total
7	1	102.90	1.5	7.1
	2	100.36	0.5	5.4
	3	100.00	3.3	8.8
	4	Termination	8.1	8.1
8	1	103.86	0.7	7.3
	2	100.37	0.7	5.7
	3	100.15	4.6	9.8
	4	100.00	0.5	5.4
	5	Termination	15.6	15.6
9	1	104.09	1.5	7.0
	2	100.37	0.6	5.6
	3	100.20	2.7	7.8
	4	100.00	0.5	5.5
	5	Termination	11.8	11.8
10	1	105.51	1.6	6.9
	2	100.38	0.5	5.3
	3	100.19	2.7	7.6
	4	100.00	0.3	5.1
	5	Termination	166.1	166.1

constraints without changing the logical content of the model in any way: replace (3) by

$$\text{(3a)} \qquad \textstyle\sum_{jk} x_{ijkl} = D_{il} \qquad \text{all } il$$

$$\text{(3b)} \qquad \textstyle\sum_{ij} x_{ijkl} = (\sum_i D_{il}) y_{kl}, \qquad \text{all } kl.$$

This formulation performs the two functions of (3) separately, namely ensuring that all demands are met and enforcing the appropriate logical relationship between the x's and the y's. The resulting representation of the problem is equivalent (has the same set of feasible solutions), and usually has fewer constraints. For the Hunt-Wesson application, the representation using (3a) and (3b) in place of (3) has 8,855 fewer constraints!

It turns out, however, that it would be a serious mistake to use this representation with any type of Benders Decomposition approach. The reason is that it leads to much weaker cuts. To see this, recall that all variants of Benders Decomposition work by accumulating linear supports to $T(y)$, which is defined in Sec. 2.1 as the optimal total transportation cost as a function of the configuration design y. For a given binary $\bar{y}$ satisfying (4),

$$\text{(18)} \qquad -\textstyle\sum_{ij} \bar{u}_{ij} S_{ij} + \sum_{kl} (-\sum_i D_{il} \bar{\pi}_{ikl}) y_{kl}$$

is such a support, where $\bar{u}$ and $\bar{\pi}$ are defined as in (12). This support is derived from the original formulation of the problem using (3) and is implicit in (9) and (9a). The corresponding support for the revised formulation using (3a) and (3b) in place of (3) can be written as:

$$\text{(18a)} \quad -\textstyle\sum_{ij}\bar{u}_{ij}S_{ij} + \sum_{kl}[-\sum_{i}D_{il}(\bar{\pi}_{i\bar{k}(l)l} + \text{Max}_{i'}\,\{\bar{\pi}_{i'kl} - \bar{\pi}_{i'\bar{k}(l)l}\})]y_{kl}\,.$$

It is evident by inspection (subtract $\bar{\pi}_{i\bar{k}(l)l}$ from both sides) that

$$\bar{\pi}_{ikl} \leq \bar{\pi}_{i\bar{k}(l)l} + \text{Max}_{i'}\,\{\bar{\pi}_{i'kl} - \bar{\pi}_{i'\bar{k}(l)l}\} \quad \text{for all} \quad ikl,$$

with the magnitude of the difference increasing with the number of commodity classes. Hence every y_{kl}-coefficient of (18) must be at least as large as the corresponding coefficient of (18a). That is, (18) uniformly dominates (18a) over the region of interest ($y \geqq 0$); it is a "tighter" support for the function $T(\cdot)$. The more commodity classes there are the greater will be the improvement of (18) over (18a).

The result that (18) dominates (18a) implies that the representation using (3) is to be preferred over the "equivalent" more compact representation using (3a) and (3b). Any variant of Benders Decomposition should converge in fewer major iterations for the first formulation than for the second. We have direct computational confirmation of this fact as a result of having turned to the first representation only after experiencing disappointing results with the second. Tables 4–6 show three approximately comparable disjoint regional optimizations using the original Benders Decomposition approach for both representations. We say "approximately" comparable because some internal parameters of the master problem algorithm were changed slightly during the time lapse between the runs, but we are confident that this does not alter the comparison significantly. The convergence parameter ϵ was set at 0.02 in all runs.

These comparative results indicate that the more compact representation consistently requires many more iterations for convergence, due principally to poorer

TABLE 4

First Comparison of Benders Decomposition for Two Alternative Model Representations

Major Iteration Number	Representation (3a) and (3b)		Representation (3)	
	LB	UB	LB	UB
1	—	5.410	—	5.053
2	4.150	5.023	5.000	5.028
3	4.349	"	≥5.008	(Convergence)
4	4.415	"		
5	4.534	"		
6	4.601	"		
7	4.631	"		
8	4.661	"		
9	4.714	"		
10	4.716	"		
11	4.750	"		
12	4.750	"		
13	4.774	"		
14	4.774	"		
15	4.808	"		
16	4.817	"		
17	4.817	"		
18	4.839	"		
	(No Convergence)			

TABLE 5

Second Comparison of Benders Decomposition for Two Alternative Model Representations

Major Iteration Number	Representation (3a) and (3b)		Representation (3)	
	LB	UB	LB	UB
1	—	5.134	—	5.083
2	3.892	“	4.937	4.960
3	4.245	“	≥4.940	(Convergence)
4	4.453	5.046		
5	4.534	“		
6	4.544	“		
7	4.574	5.043		
8	4.680	“		
9	4.680	“		
10	4.735	“		
11	4.735	“		
12	4.749	“		
13	4.749	5.027		
14	4.749	“		
15	4.759	“		
16	4.768	“		
17	4.768	“		
18	4.785	5.010		
19	4.785	“		
20	4.785	“		
	(No Convergence)			

TABLE 6

Third Comparison of Benders Decomposition for Two Alternative Model Representations

Major Iteration Number	Representation (3a) and (3b)		Representation (3)	
	LB	UB	LB	UB
1	—	5.158	—	5.158
2	4.425	5.036	4.925	4.957
3	4.431	“	≥4.937	(Convergence)
4	4.436	“		
5	4.438	4.967		
6	4.461	“		
7	4.494	“		
8	4.494	“		
9	4.496	“		
10	4.505	“		
11	4.508	“		
12	4.512	“		
	(No Convergence)			

lower bounds from the master problem. The time per iteration is approximately the same for both representations because the size and structure of the master problem and the individual transportation subproblems is exactly the same in both cases. Thus the representation using (3) is far superior. The other representation was all but unuseable in our application, considering the many validation and post-optimization runs required.

A closely analogous observation concerning the crucial importance of model representation has been reported recently by Beale and Tomlin [3]. They undertook to solve a practical problem concerning the optimal decentralization of office facilities using a direct branch-and-bound approach with a problem formulation which turns out to be very close to the one considered here. Their experience was that the problem proved to be much more tractable computationally when some of their constraints like (3a) and (3b) were replaced by constraints like (3).[2]

In this connection, we would like to point out an interesting relation between the two representations which becomes pertinent when problems of this sort are addressed by LP-based branch-and-bound. It can be shown that the convex hull of the feasible solutions to (3a), (3b), (4), $x \geqq 0$ and $y = 0, 1$ is given by the constraints (3), (4), $x \geqq 0$ and $y \geqq 0$. Thus the common practice of dropping integrality requirements in order to produce an LP relaxation at each node yields a tighter relaxation when (3) is used than when (3a) and (3b) are used. The price of this tighter bound and the reduction in branching which it affords is, of course, the additional time required to solve a larger LP at each node. It seems probable that some mixture of the two representations will be superior to either one alone in terms of total computing time (e.g., the separability of (3), (3a) and (3b) with respect to l suggests that (3) might be used just for the l's corresponding to the largest total demand). This appeared to be the case in Beale and Tomlin's study. It should be emphasized that the extra size of (3) by comparison with (3a) and (3b) does not offer any difficulty whatever when Benders' approach is used, thanks to the analytic reduction which takes placc prior to setting up the continuous subproblems to be solved at each major iteration. The ease with which Benders Decomposition can use such superior model representations is a comparative advantage over direct branch-and-bound which does not seem to be generally appreciated.

The theoretical result stated above also suggests a general methodology for discovering improved model representations: for various subsets of constraints involving some of the integer variables, try to explicitly derive the convex hull of the integer feasible points. Another related instance where this can be done is given in Geoffrion and McBride [13].

6. Conclusion

The major conclusion arising from this study is the remarkable effectiveness of Benders Decomposition as a computational strategy for static multicommodity intermediate location problems. The numerical experience quoted in §4.3 shows that only a few cuts are needed to find and verify a solution within one or two tenths of one percent of the global optimum. The same type of behavior was observed in another full-scale application carried out recently for a major manufacturer of hospital supplies with 5 commodity classes, 3 plants, 67 possible DC's and 127 customer zones. This behavior, together with the advantages of being able to decouple the multicommodity capacitated multiechelon transportation portion of the problem into a separate

[2] The authors are grateful to K. Spielberg for pointing out the following early reference containing related ideas: Guignard, M. and Spielberg, K. "Search Techniques with Adaptive Features for Certain Mixed Integer Programming Problems," Proceedings IFIPS Congress, Edinburgh, 1968.

classical transportation problem for each commodity, yields an extraordinarily powerful computational approach.

The reasons why Benders' approach requires so few cuts for this problem class are not yet clearly understood. The discussion of §5 shows that one essential ingredient is making an appropriate choice among alternative mathematical representations of the same physical problem. We were able to employ a representation which incorporates the many constraints describing the convex hull of a portion of tbe problem's integer feasible solutions. This was workable because of special opportunities for analytic simplification inherent in Benders' approach (it would not have been computationally feasible to use the same representation with a branch-and-bound approach to the problem). We hope that others will be motivated to study the questions raised by our observations with the objective of understanding more clearly the convergence behavior of Benders Decomposition and how to enhance it through appropriate choice of model representation.

Another conclusion we have reached on the basis of our experience is that every effort must be made to make it easy and economical to carry out the numerous pre- and postoptimality runs required to properly execute a practical application. This point, discussed in §4.2 and so well appreciated in the domain of linear programming, is rarely addressed in the existing integer programming literature. The burden of this requirement is exacerbated by the fact that many of the required runs must achieve very nearly optimal solutions if they are to be useful. This is certainly true of the probationary exercises, where significant suboptimality could shake management's confidence in the entire project, and is also true for "what if . . . ?," sensitivity, continuity, tradeoff and priority analysis runs as well because their very usefulness depends on the ability to measure *differences* between the solutions of different runs in a series. Obviously the tolerance on optimality must be quite tight if one is to avoid reaching spurious conclusions when making such comparisons. The results of §4.3 show that the approach developed here meets this requirement at reasonable computational cost.

The success with the present model suggests the desirability of expanding its scope. We shall mention here but two of the more appealing and easily accomplished possibilities. One is to include selection among alternative plant sites and plant capacity expansion projects via some additional 0–1 variables. Another is to take account of the service elasticity of demand, that is, of the fact that a customer zone's demand for various commodities tends to increase with the proximity of its assigned distribution center due to the advantages of decreased delivery delay [20], [22]. One way to incorporate this effect is to replace D_{il} in the model by D_{ikl}, the demand for product i by customer l if assigned to distribution center k. A (negative) net revenue term would also have to be appended to the objective function since total revenues to the firm would no longer be constant. Both of these extensions require but simple modifications to the algorithmic approach and do not upset the major factors controlling its efficiency (the use of a model representation yielding powerful Benders cuts and the separability of the multicommodity transshipment subproblem into an independent transportation problem for each commodity). We hope to be able to report on these and other extensions in a future paper.

References

1. Balinski, M. L. and Spielberg, K., "Methods for Integer Programming: Algebraic, Combinatorial and Enumerative," in J. S. Aronofsky (ed.), *Progress in Operations Research*, Vol. III, Wiley, New York, 1969.

2. Bartakke, M. N., Bloomquist, J. V., Korah, J. K., and Popino, J. P., "Optimization of a Multi-National Physical Distribution System," Sperry Rand Corporation, Blue Bell, Pa., presented at the 40th National ORSA Meeting, Anaheim, California, October 1971.

3. Beale, E. M. L. and Tomlin, J. A., "An Integer Programming Approach to a Class of Combinatorial Problems," *Mathematical Programming*, *3*, 3 (December 1972), 339–344.

4. Benders, J. F., "Partitioning Procedures for Solving Mixed-Variables Programming Problems," *Numerische Mathematik*, *4* (1962), 238–252.

5. Davis, P. S. and Ray, T. L., "A Branch-Bound Algorithm for the Capacitated Facilities Location Problem," *Naval Research Logistics Quarterly*, *16*, 3 (September 1969), 331–344.

6. De Maio, A. and Roveda, C., "An All Zero-One Algorithm for a Certain Class of Transportation Problems," *Operations Research*, *19*, 6 (October 1971), 1406–1418.

7. Ellwein, L. B., "Fixed Charge Location-Allocation Problems with Capacity and Configuration Constraints," Ph.D. Dissertation, Dept. of Industrial Engineering, Stanford University, August 1970.

8. —— and Gray, P., "Solving Fixed Charge Location-Allocation Problems with Capacity and Configuration Constraints," *AIIE Transactions*, *III*, 4 (December 1971), 290–298.

9. Elson, D. G., "Site Location via Mixed-Integer Programming," *Operational Research Quarterly*, *23*, 1 (March 1972), 31–43.

10. Fieldhouse, M., "The Depot Location Problem," University Computing Company, Ltd., London, presented at the 17th International Conference of TIMS, London, July 1970.

11. Geoffrion, A. M., "Lagrangean Relaxation and Its Uses in Integer Programming," Working Paper No. 195, Western Management Science Institute, UCLA, December 1972 (revised September 1973).

12. —— and Marsten, R. E., "Integer Programming Algorithms: A Framework and State-of-the-Art Survey," *Management Science*, *18*, 9 (May 1972), 465–491.

13. —— and McBride, R. D., "The Capacitated Facility Location Problem with Additional Constraints," Working Paper, Western Management Science Institute, UCLA, December 1973.

14. Glover, F., Karney, D., Klingman, D., Napier, A., "A Computational Study on Start Procedures, Basis Change Criteria, and Solution Algorithms for Transportation Problems," *Management Science*, this issue.

15. Graves, G. W. and McBride, R. D., "The Factorization Approach to Large-Scale Linear Programming," Working Paper No. 208, Western Management Science Institute, UCLA, August 1973.

16. Gray, P., "Mixed Integer Programming Algorithms for Site Selection and Other Fixed Charge Problems Having Capacity Constraints," Ph.D. Dissertation, Dept. of Operations Research, Stanford University, November 30, 1967.

17. Khumawala, B. and Akinc, V., "An Efficient Branch and Bound Algorithm for the Capacitated Warehouse Location Problem," presented at the 43rd National ORSA Meeting, Milwaukee, May 1973.

18. Lea, A. C., "Location-Allocation Systems: An Annotated Bibliography," Discussion Paper No. 13, Dept. of Geography, University of Toronto, May 1973.

19. Marks, D. H., Liebman, J. C., and Bellmore, M., "Optimal Location of Intermediate Facilities in a Trans-shipment Network," paper R-TP3.5 presented at the 37th National ORSA Meeting, Washington, D. C., April 1970.

20. Mossman, F. H. and Morton, N., *Logistics of Distribution Systems*, Allyn and Bacon, 1965, 245–256.

21. Soland, R., "Optimal Facility Location with Concave Costs," Research Report CS 126, Center for Cybernetic Studies, University of Texas at Austin, February 1973.

22. Willett, R. P. and Stephenson, P. R., "Determinants of Buyer Response to Physical Distribution Service," *Journal of Marketing Research*, *VI* (August 1969), 279–283.

23. Williams, A. C., "Sensitivity to Data in LP and MIP," presented at VIII International Symposium on Mathematical Programming, Stanford, California, August 1973.