MANAGEMENT SCIENCE
Vol. 33, No. 11, November 1987
Printed in U.S.A.

DESIGN AND OPERATION OF A MULTICOMMODITY PRODUCTION/DISTRIBUTION SYSTEM USING PRIMAL GOAL DECOMPOSITION*

GERALD G. BROWN, GLENN W. GRAVES AND MARIA D. HONCZARENKO
Department of Operations Research, Naval Postgraduate School, Monterey, California 93943
Graduate School of Management, University of California, Los Angeles, California 90024
Nabisco Brands, Inc., East Hanover, New Jersey 07936

An optimization-based decision support system has been developed and used by NABISCO to manage complex problems involving facility selection, equipment location and utilization, and manufacture and distribution of products such as the familiar Ritz Crackers, Oreo Cookies, Fig Newtons, etc. (all product names trademarks of NABISCO). A mixed-integer, multi-commodity model is presented for the problems at hand, and a new class of goal decompositions is introduced to yield pure network subproblems for each commodity; the associated master problems have several notable properties which contribute to the effectiveness of the algorithm. Excellent quality solutions for problems with more than 40,000 variables (including several hundred binary variables with fixed charges) and in excess of 20,000 constraints require only 0.6 megabytes region and less than one compute minute on a time-shared IBM 3033 computer; average problems (with fewer binary variables) require only a second or two. The solution method has more to recommend it than sheer efficiency: new insights are given for the fundamental convergence properties of formal decomposition techniques. Several applications of this powerful interactive tool are discussed.
(PROGRAMMING—LARGE-SCALE SYSTEMS; PROGRAMMING—INTEGER, APPLICATIONS; FACILITIES/EQUIPMENT PLANNING)

1. Introduction

The production/distribution system design problems considered here derive from current operations of the Biscuit Division of Nabisco Brands, Inc. Nabisco bakeries produce several hundred products for nationwide distribution (e.g., Ritz Crackers, Oreo Cookies, Fig Newtons, etc.). Production takes place in batches, each of which involves a relatively brief setup followed by a continuous run of product. The continuous production run involves two key operations: *baking*, in which raw ingredients are fed continuously into an oven, and *secondary operations*, such as sorting, packaging and labeling finished products. Thus, *primary facilities* (i.e., ovens) and *secondary facilities* (e.g., packing lines) must be operated synchronously.

Scheduling and operation of bakeries is a complex managerial task. Each oven is capable of producing many (but not all) products with varying efficiency. While baking a particular product, each oven uses a designated secondary facility for that product as determined by the physical layout of the bakery; the secondary facility may be shared with other ovens in operation at that time. Production must be assigned to bakeries so that total costs—manufacturing and transportation costs—are as low as possible.

Among the operational issues to be resolved are:

—Where should each product be produced?

—How much production of each product should be assigned to each primary facility?

—From where should product be shipped to each customer?

Over time, new products are introduced and customer demand shifts for each prod-

* Accepted by Donald Erlenkotter, former Departmental Editor; received May 17, 1983. This paper has been with the authors 19 months for 4 revisions.

0025-1909/87/3311/1469$01.25

uct and geographic region. Eventually, it becomes necessary to build new plants (or modify or close existing plants) and to introduce new facilities (or move or sell old ones). The strategic issues include:

—Where shall plants be located?

—What kind of facilities are needed, how many are required, and where should they be located?

We have endeavored to answer all these questions by developing a data base and decision support system that utilizes a large embedded optimization model. User-friendly interactive facilities provide rapid response to operational and strategic queries by simple editing and control of the model monolith. Reasonable model resolution entails approximately:

100–200 Products (product groups),
200–300 Facilities, including
80–100 primary facilities (ovens)
120–200 secondary facilities,
10–20 plants (bakeries), and
120–170 Customer zones.

Useful response time necessarily is very short. Consequently, a great deal of effort has been invested in the optimization methods.

In the sections that follow, the details of the mathematical model are presented, a new class of decompositions is introduced and applied, a mathematical justification for the decomposition is presented, and some general insights are given for the relative convergence properties of classical decomposition methods and the approach developed here. Finally, we give several examples of application of the decision support system.

2. Mathematical Model

Our formulation uses the following notation:

I is the index set of products;

J is the index set of facilities;

K is the index set of plants;

L is the index set of customers.

In addition to these primary index sets we require some additional derived index sets. The set of facilities J is partitioned into primary (baking) facilities, J_1, and secondary facilities, J_2.

The set of *activities* is defined by the combination of locations and facility types:

$$\underline{A} = J \times K = (J_1 \times K) \cup (J_2 \times K) = \underline{A}_1 \cup \underline{A}_2.$$

The activities are restricted as to location, and $\tilde{A} = \tilde{A}_1 \cup \tilde{A}_2 \subset \underline{A}$ are the subsets of allowable combinations.

The set of *production elements* is defined by the combination of products and primary activities:

$$P = I \times \underline{A}_1 = I \times J_1 \times K.$$

The production elements are restricted as to primary activity, and $\tilde{P} \subset P$ is the set of allowable combinations. Each allowable production element is associated with a unique secondary activity, as defined by the map M: $\tilde{P} \rightarrow \tilde{A}_2$. The map M induces a partition of $\tilde{P}$ according to the secondary activity used:

$$R_{a_2} = \{p \in \tilde{P} | M(p) = a_2 \in \tilde{A}_2\}.$$

The set of primary activities that produce a given product $i \in I$ is:

$$T_i = (a_1 \in \tilde{A}_1 | i \times \tilde{A}_1 \in \tilde{P}).$$

The given data for the model are:

D_{il} demand for product i in customer zone l,
S_{ijk} capacity for product i on facility j at plant k,
c_{ijk} average unit cost of producing product i on facility j at plant k,
f_{ikl} unit cost of shipping product i from plant k to customer l,
G_{jk} fixed portion of the annual possession and operating cost for facility j at plant k,
F_k fixed portion of the annual possession and operating cost for a plant at site k,
Y_{ijk} yield of product i on primary facility j at plant k,
U_{ijk} rate of utilization by product i of secondary facility j at plant k,
$\underline{C}_{jk}$, $\bar{C}_{jk}$ minimum and maximum utilization of facility j at plant k,
$\underline{N}_k$, $\bar{N}^k$ minimum and maximum number of facilities at plant k.

The variables of the model are:

X_{ijkl} the amount of product i produced on facility j at plant k shipped to customer l,
s_{ijk} the amount of product i produced on facility j at plant k,
W_{jk} a 0-1 assignment variable of facility j to plant k,
Z_k a 0-1 close-open variable for plant k.

The problem is formulated as the following mixed integer linear program:

$$\underset{X,s,W,Z}{\text{MINIMIZE}} \quad \sum_{\substack{(i,j,k)\in\tilde{P}\\ l\in L}} X_{ijkl}(c_{ijk} + f_{ikl}) + \sum_{(j,k)\in\tilde{A}} G_{jk}W_{jk} + \sum_{k\in K} F_k Z_k$$

SUBJECT TO

$$\sum_{(j,k)\in T_i} X_{ijkl} = D_{il}, \qquad i \in I, \qquad l \in L, \tag{2.1}$$

$$\sum_{l\in L} X_{ijkl} \leqslant s_{ijk}, \qquad (i, j, k) \in \tilde{P}, \tag{2.2}$$

$$\underline{C}_{jk}W_{jk} \leqslant \sum_{(i,j,k)\in\tilde{P}} s_{ijk}/Y_{ijk} \leqslant \bar{C}_{jk}W_{jk}, \qquad (j, k) = a_1 \in \tilde{A}_1, \tag{2.3}$$

$$\underline{C}_{jk}W_{jk} \leqslant \sum_{(i,j,k)\in R_{a_2}} U_{ijk}s_{ijk} \leqslant \bar{C}_{jk}W_{jk}, \qquad (j, k) = a_2 \in \tilde{A}_2, \tag{2.4}$$

$$\underline{N}_k Z_k \leqslant \sum_{j\in J} W_{jk} \leqslant \bar{N}_k Z_k, \qquad k \in K, \tag{2.5}$$

$$\sum_{k\in K} W_{jk} = 1, \qquad j \in J, \tag{2.6}$$

and the variable bounds

$$0 \leqslant X_{ijkl}, \tag{2.7}$$

$$0 \leqslant s_{ijk} \leqslant S_{ijk}, \tag{2.8}$$

$$W_{jk} = \{0, 1\}, \tag{2.9}$$

$$Z_k = \{0, 1\}. \tag{2.10}$$

The constraints (2.1) ensure that all demand is met. The constraints (2.2) ensure that products shipped are produced. The constraints (2.3) and (2.4) are multi-product capacity restrictions on the primary and secondary activities. The constraints (2.5) limit the assignment of facilities to plants. The constraints (2.6) ensure that a facility is assigned to only one plant.

A realistic prototypic problem has 150 products, 218 facilities, 10 plants and 127 customer zones. The number of production elements is 345 and thus there are

$345 \times 127 = 43{,}815$ X_{ijkl} variables. Adding the 345 s_{ijk} variables and the 10 Z_k variables, the problem has a total of 44,388 variables. There are $150 \times 127 = 19{,}050$ constraints of type (2.1), 345 of type (2.2), 218 of type (2.3) and (2.4), 10 of type (2.5), and 218 of type (2.6) yielding a total of 19,841 constraints.

3. Decomposition

Problems of dimension $19{,}841 \times 44{,}388$ can prove to be somewhat intractable when approached in a direct manner. Examination shows, however, that separating the X_{ijkl} variables by fixing s_{ijk}, W_{jk}, and Z_k in a primal decomposition breaks out disjoint transportation problems by product and leaves a relatively modest master problem. At the nth iteration, we have the following subproblems, one for each $i \in I$:

(ith subproblem at nth iteration)

$$\underset{X}{\text{MINIMIZE}} \quad \sum_{(j,k)\in \tilde{P}} X_{ijkl}(c_{ijk} + f_{ikl})$$

SUBJECT TO

$$\sum_{(j,k)\in T_i} X_{ijkl} = D_{il}, \qquad l \in L, \tag{3.1}$$

$$\sum_{l\in L} X_{ijkl} \leqslant s^n_{ijk}, \qquad (j,k) \in \tilde{P}, \tag{3.2}$$

and the variable bounds (2.7).

The supplies s^n_{ijk} in (3.2) for the nth iteration are obtained from a master problem.

The master problem at the nth iteration is given as follows, where I_c is the best known solution value for the entire problem through the nth iteration. Let v^n_{il} be the dual variables associated with the constraints (3.1) and u^n_{ijk} be the dual variables associated with the constraints (3.2).

(Master Problem)

$$\underset{s,W,Z}{\text{MINIMIZE}} \sum_{(j,k)\in \tilde{A}} G_{jk}W_{jk} + \sum_{k\in K} F_k Z_k$$

Subject to:

$$\sum_{(j,k)\in T_i} s_{ijk} \geqslant \sum_{l\in L} D_{il}, \qquad i \in I, \tag{3.3}$$

$$\underline{C}_{jk}W_{jk} \leqslant \sum_{(i,j,k)\in \tilde{P}} s_{ijk}/Y_{ijk} \leqslant \bar{C}_{jk}W_{jk}, \qquad (j,k) = a_1 \in \tilde{A}_1, \tag{2.3}$$

$$\underline{C}_{jk}W_{jk} \leqslant \sum_{(i,j,k)\in R_{a_2}} U_{ijk}s_{ijk} \leqslant \bar{C}_{jk}W_{jk}, \qquad (j,k) = a_2 \in \tilde{A}_2, \tag{2.4}$$

$$\underline{N}_k Z_k \leqslant \sum_{j\in J} W_{jk} \leqslant \bar{N}_k Z_k, \qquad k \in K, \tag{2.5}$$

$$\sum_{k\in K} W_{jk} = 1, \qquad j \in J, \tag{2.6}$$

$$s_{ijk} \overset{\circ}{=} s^g_{ijk}, \qquad (i,j,k) \in \tilde{P}, \tag{3.4}$$

$$\sum_{(i,j,k)\in \tilde{P}} u^t_{ijk}s_{ijk} + \sum_{j,k\in \tilde{A}} G_{jk}W_{jk} + \sum_{k\in K} F_k Z_k \leqslant I_c + \sum_{\substack{i\in I\\ l\in L}} v^t_{il}D_{il} - \epsilon, \qquad t < n, \tag{3.5}$$

and the variable bounds (2.8)–(2.10).

The constraints (3.3) are added to ensure sufficient supply for feasibility of the subproblem at each iteration. The *cuts* (3.5) are developed in the next section. The constraints (3.4) are the really novel feature of this formulation. They are a set of *production goals* used to provide additional information in the allocation of the activity capacity across the various products. The symbol $\overset{\circ}{=}$ indicates that each may be violated at a small linear penalty cost. The penalty cost must be small enough to ensure that these "goals" will be sacrificed as needed to satisfy the other relatively stiffer constraints.

In this model the goals are obtained initially by solving the subproblems with $s_{ijk}^0 = S_{ijk}$, the individual production capacities, while ignoring the multi-product capacity restrictions (2.3), (2.4). Since these goals are obtained as a relaxation of the original problem, this provides a lower bound on the value of the subproblems.

Typically the most desirable activities will be overutilized by the subproblems and exceed their joint product capacities as specified in the constraints (2.3), (2.4). Therefore the actual subproblem use ($\sum_{l \in L} X_{ijkl}$) is scaled down proportionally by product to equal exactly the joint product capacities $\bar{C}_{jk}$. This is done after a heuristic has been used to select a good initial configuration of assignments of facilities to plants. Each unassigned facility is assigned to the compatible plant from which the subproblems have drawn the highest total proportions of demands for products made there.

The use of decomposition goals profoundly influences the rate of convergence achieved with decomposition.

Dual decomposition models with goals have been suggested by Ruefli (1971) as a purely conceptual tool for interpretation of hierarchical organizational behavior. Subsequently, Freeland (1976) discussed primal decomposition goal models in the same vein.

4. Justification of the Decomposition

Given the mathematical program

$$\begin{aligned} \text{MP} \quad & \min wy = w_1 y_1 + w_2 y_2 \\ \text{s.t.} \quad & Ay = A^1 y_1 + A^2 y_2 \le r, \\ & -y \le 0, \qquad Y_2 \in \Gamma, \end{aligned}$$

which becomes a linear programming problem given fixed values for y_2, the problem may be solved sequentially as a function of y_2:

$$\begin{aligned} \text{LP}(y_2) \quad & \min w_1 y_1 \qquad (+w_2 y_2) \\ \text{s.t.} \quad & A^1 y_1 \le r - A^2 y_2, \\ & -y_1 \le 0. \end{aligned}$$

Let F_p be the set of primal feasible solutions $y = (y_1, y_2)$ of MP. Let $R(y_2)$ be the set of $r(y_2) = r - A^2 y_2$ that yield a finite optimal solution to LP(y_2). Also let $F_p(y_2)$ be the set of primal feasible solutions $y_1(y_2)$ of LP(y_2) and $\bar{v}(y_2)$ be the optimal value of LP(y_2). Let $v(y_2) = \bar{v}(y_2) + w_2 y_2$. Now $(y_1, y_2) \in F_p$ yields a y such that $r(y_2) \in R(y_2)$ and $y_1 \in F_p(y_2)$ and conversely, and therefore every feasible solution of MP yields a feasible solution in the class of LP problems LP(y_2) and conversely. The optimal value of MP is $\min_{y_2 \in R(y_2)} v(y_2)$.

Let the dual of the relaxation of MP, deleting the restriction $y_2 \in \Gamma$, and the dual of LP(y_2) be respectively

$$\begin{aligned} \text{DP} \quad & \max \quad xr \\ & \text{s.t.} \quad xA \le w, \\ & \qquad\quad x \le 0, \end{aligned} \qquad\qquad \begin{aligned} \text{DP}(y_2) \quad & \max \quad x(r - A^2 y_2) \qquad (+w_2 y_2) \\ & \text{s.t.} \quad xA^1 \le w_1 \\ & \qquad\quad x \le 0, \end{aligned}$$

with corresponding feasible solution sets F_d and $F_d(y_2)$.

Let $x(y_2)$ be an optimal solution for $\mathrm{DP}(y_2)$ and then

$$\begin{aligned} v(y_2) &= \bar{v}(y_2) + w_2 y_2 = x(y_2) r(y_2) + w_2 y_2 \\ &= x(y_2)[r - A^2 y_2] + w_2 y_2 \\ &= x(y_2) r + [w_2 - x(y_2) A^2] y_2 . \end{aligned}$$

Now $\bar{v}(y_2)$ as a function of $r(y_2)$ can be strongly characterized. The function $\bar{v}(y_2)$ is a continuous, convex, piecewise linear function. Using the results that a continuous function of a continuous function is also continuous, and that the sum of convex functions is convex, $v(y_2)$ is also a continuous, convex, piecewise linear function.

If the convex function $v(y_2)$ were also continuously differentiable, we could apply the classical result about convex functions that $v(y_2) \geqslant v(y_2^0) + \nabla v(y_2^0)(y_2 - y_2^0)$ to obtain a linear lower bound function that would provide necessary conditions for $v(y_2) < v(y_2^0)$. Although $v(y_2)$ does have discontinuities in its first derivatives, a linear lower bound function is still easily obtainable. For any $x \in F_d$, $y \in F_p$, by the weak duality theorem of linear programming,

$$xr \leqslant xr + (w - xA)y = wy + x[r - Ay] \leqslant wy,$$

since

$$x \in F_d \Rightarrow x \leqslant 0, \qquad (w - xA) \geqslant 0,$$

$$y \in F_p \Rightarrow y \geqslant 0, \qquad (r - Ay) \geqslant 0.$$

Also,

$$(w - xA)y = [w_1 - xA^1]y_1 + [w_2 - xA^2]y_2 .$$

Restricting ourselves to $x(y_2) \in F_d(y_2)$, $y \in F_p$ for a fixed value $\bar{y}_2$, $[w_1 - x(\bar{y}_2)A^1]y_1 \geqslant 0$, and therefore,

$$\begin{aligned} l(y_2; \bar{y}_2) &\equiv x(\bar{y}_2) r + [w_2 - x(\bar{y}_2) A^2] y_2 \\ &= wy + x(\bar{y}_2)[r - Ay] - [w_1 - x(\bar{y}_2) A^1] y_1 \\ &\leqslant wy + x(\bar{y}_2)[r - Ay] \leqslant wy. \end{aligned}$$

The linear lower bound function $l(y_2; \bar{y}_2) = x(\bar{y}_2)r + [w_2 - x(\bar{y}_2)A^2]y_2 = v(y_2)$ at $y_2 = \bar{y}_2$ by direct comparison with the expression for $v(y_2)$. (This is an indirect consequence of the complementary slackness requirement that $[w_1 - x(\bar{y}_2)A^1]y_1(\bar{y}_2) = 0$.)

The class of linear lower bound functions $l(y_2; \bar{y}_2)$ parameterized by $\bar{y}_2$ can be employed to generate a sequence $\{y_2^n\}$ such that $v(y_2^n)$ converges to the minimum of $v(y_2)$. It is a necessary condition for an ϵ improvement over the best known incumbent value $v(y_2^c)$ that y_2^n satisfy the linear inequalities,

(Master Problem cuts)

$$l(y_2^n; y_2^t) = x(y_2^t) r + [w_2 - x(y_2^t) A^2] y_2^n \leqslant v(y_2^c) - \epsilon, \qquad t < n.$$

(Inequalities (3.5) are an instance of this.)

When the master problem is infeasible $v(y_2) > v(y_2^c) - \epsilon$ and $v(y_2^c)$ is an ϵ-optimal value, since any y_2 such that $v(y_2) \leqslant v(y_2^c) - \epsilon$ would satisfy $l(y_2; y_2^t) \leqslant v(y_2) \leqslant v(y_2^c) - \epsilon$ and be a feasible solution.

THEOREM. Assume $v(y_2)$ *is bounded below. Any sequence* $\{v(y_2^n)\}$ *generated from* y_2^n *that satisfies the master problem will terminate in a finite number of steps at an* ϵ*-optimal value.*

PROOF. Suppose the sequence does not terminate in a finite number of steps and yield an ϵ-optimal value. Each y_2^n generates an $x(y_2^n)$ which is a basic optimal solution of $\mathrm{DP}(y_2)$. There are only a finite number of bases $T(k)$ for $\mathrm{DP}(y_2)$ since $T(k)$ is not a function of y_2. Every $x(y_2^t)$ satisfies $T(k)x(y_2^t) = w_1$ for some k, and since w_1 is also independent of y_2 there are only a finite number of $x(y_2)$. Therefore let s be the first index such that $x(y_2^s) = x(y_2^t)$ for $t < s$. Then

$$v(y_2^s) = l(y_2^s; y_2^s) = l(y_2^s; y_2^t) \leq v(y_2^c) - \epsilon,$$

and $v(y_2^s)$ becomes the new incumbent. Since ϵ is strictly positive $v(y_2)$ will violate its lower bound in a finite number of steps. *Q.E.D.*

An analogous development is given for dual decomposition and hybrid approaches by Graves and Van Roy (1979).

Although any choice of y_2^n satisfying the master problem will ensure finite convergence, the rate of convergence is another matter. A two-dimensional representation of $v(y_2)$ as a piecewise linear convex function provides insight into the convergence process. (See Figure 1.)

The convexity of $v(y_2)$ implies that the slopes of the linear lower bound functions $l(y_2; \bar{y}_2)$ are small in the vicinity of the optimum and monotonically increase with y_2. The rate of change of $l(y_2; \bar{y}_2)$ is very sharp when remote from the optimum, and when $|y_2^n - y_2^{n-1}|$ is not strongly bounded an interminable oscillation can (and does) occur between the wings. Any strong minimization of the $l(y_2; y_2^t)$ such as the customary decomposition technique (Benders 1962):

(Customary Master Problem)

$$\begin{aligned} \min \quad & v \\ \text{s.t.} \quad & l(y_2; y_2^t) \leq v, \end{aligned}$$

seems to force this type of oscillation and exhibit very weak convergence; the cuts introduced here are much less prone to this unruly behavior, and admit a rich diversity of coercion via the objective function. Any heuristic that will yield a good starting $r(y^0)$ should be very beneficial. (Note $r(y^0) = r - A^2 y_2 = \bar{r}$ can be solved for y_2 as a goal program.) The piecewise linearity shows that $l(y_2; y_2) = v(y_2^n)$ in a sufficiently small

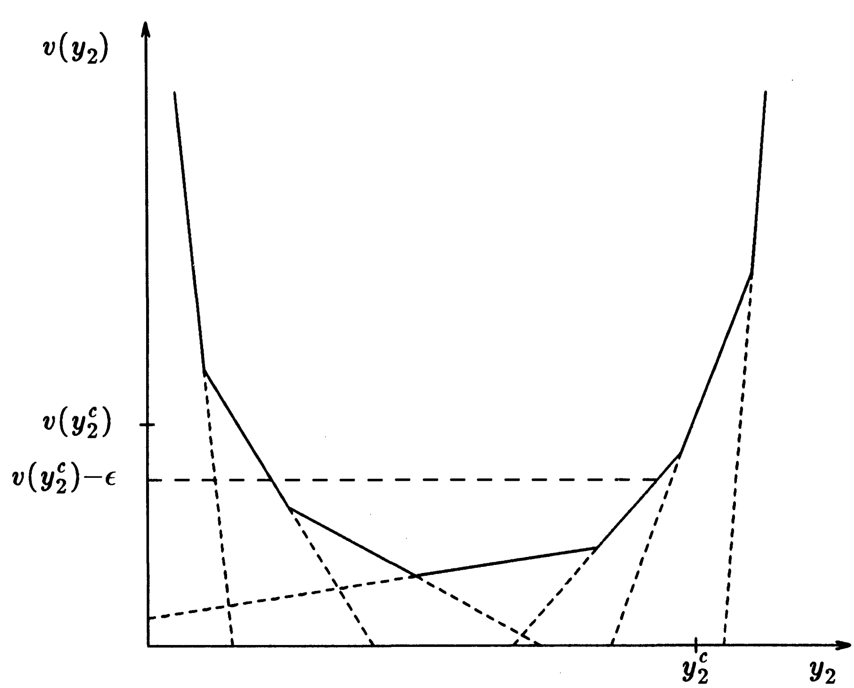

FIGURE 1. Graph of $v(y_2)$ and $l(y_2; \bar{y}_2)$ for $y_2 \in R^1$.

neighborhood of y_2^n (until a basis change). In general since $l(y_2; y_2^n)$ is determined by the dual variables $x(y_2^n)$, which are the derivatives of $v(y_2)$ with respect to $r(y_2)$, it is a good approximation to $v(y_2)$ only in a small neighborhood of y_2^n. To ensure stable and virtually monotonic convergence it is usually necessary to impose local bounds $|y_2^n - y_2^{n-1}| \leqslant b$ on each step of the process. Computational experimentation has strongly confirmed these theoretical insights.

The view of decomposition as the minimization of a nonlinear function presented here is closely related to what has become known as "Lagrangian Relaxation" (e.g., Geoffrion 1974), especially in the context of "subgradient optimization". However, in subgradient optimization only the last $l(y_2; y_2^n)$ is used. (To see this, note that the Lagrangian Relaxation of DP leads to a subgradient problem using $l(y_2, y_2^n)$ as the objective goal for DP (y_2^n).) Since it is a necessary condition that all the inequalities of the Master Problem be satisfied in order to achieve a gain at a given step, the neglect of all but the last inequality greatly complicates and weakens convergence. An observant reader will find precisely this unstable behavior in the numerical examples of Held, Wolfe, and Crowder (1974).

The decomposition theory advocated here also readily assimilates the use of *goals*. Using a 3-level hierarchy of linear penalties, a high penalty for all noncut and nongoal constraints, a medium penalty for the cuts (e.g., (3.5)), and a low penalty for the goals (e.g., (3.4)) retains all the essentials for convergence by ensuring that the constraints will violate in the same hierarchical order. The violation of the goal constraints at each step is ignored. When a violation of the cut constraints (3.5) occurs a new ϵ-optimal bound is achieved. The best procedure in goal incorporated decomposition is to use a decreasing sequence $\epsilon_1 > \epsilon_2 > \epsilon_3 \cdots$ of optimality tolerances and a decreasing sequence $b_1 > b_2 > b_3 \cdots$ of neighborhood bounds. The tolerance level ϵ_i is reduced when cut infeasibility occurs. The local bound b_i is reduced at each break in the monotonic decrease of

Step 0: Initialization
- ϵ, ϵ_f initial, minimal convergence tolerances
- δ, δ_f initial, minimal trust regions
- $n = 0$, n_f initial, final numbers of cuts
- Incumbent value = Large number
- Configuration Limitations (some fixed assignments y_2)

Step 1: Initial Configuration and Decomposition Goals
- Heuristic assignment of all of y_2
- Set initial decomposition goals

Step 2: Solve Master Problem
- If feasible, ignoring goals, go to Step 3
  - Else if $n = 0$ stop infeasible.
    - Else if $\epsilon \leqslant \epsilon_f$ go to Step 4
      - Else reduce convergence tolerance ϵ and repeat Step 2.

Step 3: $n = n + 1$
- Solve Subproblems
- Incumbent test for improved solution value
- If no improvement and $\delta > \delta_f$ reduce trust region δ

Step 4: If $n \geqslant n_f$ go to Step 5
- If feasible, ignoring goals, go to step 2
  - Else if initial heuristic assignment still in force
    - Relax heuristic assignments
    - Increase convergence tolerance ϵ
    - Increase trust region δ
    - Go to Step 2
  - Else go to Step 5

Step 5: Termination

FIGURE 2. A Skeletal Example of a Primal Goal Decomposition Algorithm.

$v(y_2)$. Moderation is a virtue in the selection of the tolerance levels ϵ_i. They are in effect aspiration levels, and greed forces large departure from the current neighborhood where the dual local derivatives $x(y_2^n)$ have some validity.

Figure 2 depicts the overall operation of an instance of the entire decomposition algorithm developed here. The local trust region for continuous variables in the master problem can be managed and a fashion reminiscent of nonlinear programming, with movement limited to a fraction of the current values (e.g., the neighborhood bounds b_n initially derive from (2.8) and the restriction fraction parameter δ).

5. Computational Experience

The decomposition has been implemented using the X-system (Brown and Graves 1975) for problem generation, coordination and master problem solution, and employing GNET (Bradley, Brown, and Graves 1977) for solving the pure network subproblems.

The X-system intrinsically incorporates and exploits goal constraints, so that the implementation of hierarchical penalties is very easy and the solution performance is good. GNET (ca. 1981) is even more efficient for the network problems than its progenitor (ca. 1974).

We were particularly interested in the actual behavior of the decomposition in light of the theoretical evidence. Accordingly, early experiments with real-life data were run with, and without the goal constraints.

Without the use of the goals (3.4), 30 iterations could not produce a solution of the prototypic problem within 4 million (annual) dollars of the optimum. Using the goals the initial solution was 3.5 million dollars better than the final solution without the

Step 0:	Initialization	(e.g.)
	Cost Conversion (master-to-subproblems)	(\$millions-to-mils, 10^9)
	Activity Conversion (master-to-subproblems)	(10^5)
	High Penalty (demand (3.3), facility assignment (2.5, 2.6))	(100.0)
	Medium Penalty (production (2.3, 2.4), cuts (3.5))	(20.0)
	Low Penalty (supply goals (3.4))	(0.5)
	Initial, minimal convergence tolerance (ϵ, ϵ_f)	(0.2, 0.05)
	Initial, minimal trust region (δ, δ_f)	(0.05, 0.0001)
	Configuration Limitations	(fix some W_{jk}, Z_k's)
Step 1:	Initial Configuration, initial supply goals, production goals	
	Solve subproblems with s^0_{ijk} = available S_{ijk}	(lower bound on $v(MP)$)
	Set production goals s^g_{ijk} to flows $\sum_l X_{ijkl}$	
	scaled equal to product capacities $\bar{C}_{jk}$	
	If all facilities assigned, go to Step 2	
	Unassigned facilities assigned to plants	(fix all W_{jk}, Z_k's)
	Find production capacities	
	Solve subproblems with S^0_{ijk} = available S_{ijk}	
	Set production goals s^g_{ijk} to flows $\sum_l X_{ijkl}$	
	scaled equal to product capacities $\bar{C}_{jk}$	
Step 2:	Solve Master Problem (with s^g_{ijk})	(for s_{ijk}, W_{jk}, Z_k)
	If ϵ can be reduced, $\epsilon = \max\{\epsilon/2, \epsilon_f\}$	(reduce ϵ)
Step 3:	Solve Subproblems with $s^n_{ijk} = S_{ijk}$	(for X_{ijkl})
	if δ can be reduced, $\delta = \max\{\delta/2, \delta_f\}$	(reduce δ)
Step 4:	If heuristic facility assignments relaxed	(free those W_{jk}, Z_k's)
	increase trust region $\epsilon = 4\epsilon_f$	(increase ϵ)
	increase convergence tolerance $\delta = 4\delta_f$	(increase δ)
Step 5:	Termination	

FIGURE 3. Example Details for Model at Hand.

goals. Further, the initial solution (with goals) was refined with 12 iterations to yield a further gain of 0.466 million dollars.

Although the solution without the goals was within about 1% of the optimum (which might seem an acceptable approximation), it is largely the ability of mathematical models to achieve these final refinements that justifies their use.

The solution of this relatively difficult initial prototypic problem on an IBM 3033 using FORTRAN IV H (Extended) with OPTIMIZE (2) required 64 seconds and 0.6 megabytes region.

The goals and hierarchical linear penalties can be generated in many ways, providing a rich experimental arena. We have tested static goals induced from initial capacity estimates, and dynamic goals derived from subproblem solutions with capacities fixed by preceding master problem solutions. Static penalties for the goals have been compared with dynamic asymmetric penalties determined by a heuristic which examines shortages and excess capacity in successive master problem solutions. The combination of static goals and dynamic penalties has performed best.

However, it is more significant that in our experience any goal-penalty combination exhibits profound improvement over classical decomposition. This robustness gives compelling evidence that goal decomposition is more effective in dealing with the complications of infeasibility than classical decompositions.

Figure 3 shows some of the details specific to the NABISCO problem in the context of the algorithm steps in Figure 2. The cost and activity conversion factors serve for translation of units between the mixed integer master problems and the pure integer network subproblems; the other penalties and tolerances inherit their units from these conversion factors.

6. Managerial Experience

The particular decision support system developed for Nabisco Brands has been named BPDOS—*B*iscuit *P*roduction-*D*istribution *O*ptimization *S*ystem. In actual decision-making situations, BPDOS has proven itself to be the management support system it was conceived and built to be. The thorough evaluation of the facility planning issues that Nabisco Brands must address on a frequent basis has been enhanced significantly. The top managers, who ultimately must make the decisions about closing old bakeries, introducing new technology or products, and moving production facilities from one bakery to another, now can do so with fully integrated and rapidly available information on the cost and capacity implications of those decisions.

For example, BPDOS has been used to help management analyze the operational and financial implications of closing an old bakery which was inefficient by today's standards in materials and product flows. Additionally, higher than average maintenance costs were eroding the profit margins of products made in that bakery. The question was,

> "Can the products made in the old bakery be produced elsewhere within existing capacity, and, if so, at what cost?"

Through a series of BPDOS runs, the production planning analysts were able to demonstrate to top management how the production capacity lost in closing one location could be "made-up" among those bakery locations that would remain. The realigned production-distribution system showed which products should be produced on each combination of facilities for each of the roughly 160 branches ("customers") to minimize total production and distribution costs.

Another application of BPDOS has been in the analysis of equipment requirements to convert all Ritz cracker production to "slug" packs vs. "dump" packs. In the traditional dump pack, the crackers are loose inside the box. With the slug pack, the crackers are stacked in three or more columns, and each column is wrapped separately in waxed paper. Through a series of BPDOS runs the production planning analysts were able to tell top management what additional equipment would be required, and in which locations, to convert fully to slug-type packaging.

Yet a third application of BPDOS has been in the roll-out planning for new products. Given market forecasts over five- to ten-year planning horizons, BPDOS can show how new production capacity should be introduced over time to minimize current production and distribution costs. When markets develop for new products, as reflected in increased demand, production facilities must be rebalanced to get the optimal facility utilization mix at any point in time.

The above kinds of facility planning issues have always existed, and they were not ignored prior to the implementation of BPDOS. However, prior to BPDOS, each planning scenario took about three days to work up manually; given the size of most scenarios, fully integrated and consistent evaluations could not be guaranteed, much less (nearly) optimal solutions. Using BPDOS, the scenario evaluation time has decreased to half a day, most of which is devoted to loading new data, such as demand forecasts. This 83% reduction in time to carry out a scenario analysis is significant since upwards of 50 such "What If" exercises may be done each year. That translates to saving more than half a man-year of a highly experienced production planner; and that, in turn, translates to a cost reduction of over $25,000.

In terms of actual computational time, the average BPDOS problem has taken 2–3 CPU seconds on an IBM 3033 with VM/CMS. The most difficult problems seldom require longer than 60 CPU seconds. For large problems, the actual turnaround time for the analyst sitting at a display terminal has been about three minutes per scenario. Including all the data file definitions and full-screen prompting menus that have been set up for BPDOS in the CMS operational environment at Nabisco Brands, the region requirements are just under 0.75 megabyte.

7. Conclusions

Although our model has a strategic perspective—fixed charges and binary decision variables contribute much of the computational optimization workload—it is surprising that significant savings are achieved just by resolution of production and transportation costs. Even for a fixed configuration of plants and facilities, the optimal assignment of production is a subtle affair, capable of producing remarkable cost reductions.

The new solution methodology has revolutionized our thinking about decomposition and relaxation methods. Many years of computational experience have convinced us that this new class of decompositions has much to recommend it.[1]

[1] We are indebted to Messrs. Les Brewster and Al Frank of Nabisco Brands, Inc., for their support of this research project.

Dr. Richard Powers of Insight, Inc., has been of invaluable assistance in bridging the significant gap separating sound theory and routine application.

Al Washburn has suggested several clarifications, and Rick Rosenthal has generously contributed improvements in style and presentation.

This paper was originally presented at ORSA/TIMS, Houston, October 14, 1981, under the title "Large-Scale Facility and Equipment Location: An Application of Goal Programming in Multicommodity Decomposition".

References

BENDERS, J., "Partitioning Procedures for Solving Mixed-Variables Programming Problems," *Numer. Math.*, 4 (1962), 238–252.

BRADLEY, G., G. BROWN AND G. GRAVES, "Design and Implementation of Large-Scale Primal Transshipment Algorithms," *Management Sci.*, 24 (1977), 1–34.

BROWN, G. AND G. GRAVES, "Design and Implementation of a Large-Scale (Mixed Integer) Optimization System," ORSA/TIMS, Las Vegas, November 1975.

FREELAND, J., "A Note on Goal Decomposition in a Decentralized Organization," *Management Sci.*, 23 (1976), 100–102.

GEOFFRION, A., "Lagrangian Relaxation and its Uses in Integer Programming," *Math. Programming Study*, 2 (1974), 82–114.

GRAVES, G. AND T. VAN ROY, "Decomposition for Large-Scale Linear and Mixed Integer Linear Programming," UCLA Technical Report, November 1979.

HELD, M., P. WOLFE AND H. CROWDER, "Validation of Subgradient Optimization," *Math. Programming*, 6 (1974), 62–88.

RUEFLI, T., "A Generalized Goal Decomposition Model," *Management Sci.*, 17 (1971), B 505–518.