INTERFACES
Vol. 6, No. 1, Pt. 2
November 1975

A MATHEMATICAL PROGRAMMING SYSTEM FOR FOOD MANAGEMENT APPLICATIONS *

Joseph L. Balintfy
School of Business Administration
University of Massachusetts
Amherst

ABSTRACT. A multistage multiple choice programming system was developed to reduce the food cost of institutional feeding programs by computer assisted menu planning and data control. Food management objectives were formulated as finding combinations of menu items for a sequence of days which satisfy desired nutritional, structural, compatibility and variety constraints at least cost, problems routinely faced by all volume feeding organizations and individuals. Solution was achieved by sequential solution of large integer programs with a specially developed truncated block enumeration algorithm. Programmed dialog made on-line postoptimal adjustments possible. A complete computerized food management information subsystem was also developed to support the matrix generator and provide a uniform file organization and data coding system applicable to any institution. System documentation, called S/360 CAMP, is available through the IBM Contributed Program Library.

After initial implementation in four hospitals proved that mathematical optimization had indeed reduced the food service cost by 10-15 per cent while maintaining prescribed nutritional and acceptability levels, two companies were formed to serve institutions in the application of CAMP and its derivatives. Published reports indicate successful applications in numerous hospitals, colleges, schools, and mental and penal institutions, realizing 5-34 per cent cost savings. The number of applications in the United States and abroad has been increasing over the years with beneficial impact on food management and nutritional health. The results motivated a new approach to food price index theory since the applications provide a practical mechanism to optimally readjust food consumption in the face of price fluctuation.

Introduction

Food management is concerned with the decision problems of feeding a given population by converting raw food into edible products – called menu items – and delivering meals which meet the preferences of the population, are nutritionally adequate and can be produced with the facilities and budget available. The essential interface between food management and customers is the menu. To a large extent, the menu from which the meals are selected defines the acceptability, nutritional adequacy and cost of the food service system. Furthermore, if the menu is nonselective, the food management's control over these attributes of the meals is complete. This is important to note because a very large and increasing number of the population such as students, patients, workers, servicemen, inmates, etc. are regularly on the diets served by such institutional feeding programs all over the world.

**Acknowledgements*
Parts of the work reported here have been supported by U.S.P.H.S. Grants HM-00216 and HS-00087, O.N.R. Contract N00014-67-0230-0006, and by the I.B.M. Corporation. The author gratefully acknowledges the assistance of Ms. J. Andrews, Dr. L. C. Dennis, Ms. M. J. Gelpi, Mr. M. E. McNabb and Dr. F. L. Shank for providing valuable information concerning their respective applications. Special recognition is given to the programming assistance of Mr. C. R. Blackburn in the development of the CAMP system.

In 1962, the author addressed himself to the questions: What is the methodology that food managers and dietitians use in planning menus? Are these menus optimal in some sense or can they be improved upon? If yes, could the approach of management science help in any way? What will be the expected benefit for food management? For the customers?

The answers to these questions gradually emerged through more than a decade of persistent research originated at Tulane University. Soon it was found that menu planning problems can be represented and, in most cases, solved as mathematical programming problems. The first pilot studies also established the central role of the computer in the future of food management as the logical tool for menu optimization and also as the necessary processing machinery for their large volume of data. It was further established that without mathematical programming models and computers neither the concept nor the existence of "best" menus could ever be defined in a scientifically acceptable way. The direct consequence of this thesis was observed through the very first application of a multiple choice programming "best buy" model of menu planning which indicated 19 - 30 per cent cost saving potential relative to conventionally planned menus while maintaining acceptability [5], [6].

Although these findings were based on realistic data, actual cost savings still were to be demonstrated through implementations. This took place under the support of a U.S.P.H.S. grant involving three hospitals in New Orleans and the University of Missouri Medical Center in Columbia from 1965 to 1970. The hospitals were linked to the Tulane University Computer Laboratory via remote terminals and each user received data processing and menu planning assistance from the research group. These applications established the need for a mathematical programming system which included a systematic approach to data collection and data base management as well as menu planning. Details of this work, described in [8], [9], [12], and [20], proved beyond doubts that the system in operation not only saved money for food management but also improved the nutritive and acceptability standards of food service.

In 1969, at the request of IBM Corporation, the Tulane system was converted into a self-contained program package: System/360 Computer Assisted Menu Planning (CAMP) [10]. The purpose of this conversion was to make the mathematical programming system available in a relatively machine-independent form for food management anywhere. Another conversion was made about the same time to the computer facilities of the Midwest Research Institute, leading to a most successful application in Kansas City [18] and to the incorporation of the first company formed for the explicit purpose of computerizing food management on a commercial basis.

The CAMP system and its use is essentially the mathematical programming system which is the subject of this report. Since it was released in the public domain, it has been successfully implemented in a large number of institutions. In many instances it has been modified to suit the user's special information processing needs but the basic structure of the data base and the menu planning model proved its applicability in a wide range of food service operations. The only difficulties which seemed to hinder wider acceptance were related to the lack of data and know-how needed for implementations. So a second company was soon formed to assist future users with data bank and instructions. This company later merged with another corporation but is still serving several client organizations in computer assisted food management.

The growing success of the "best buy" meal concept and food service control through the CAMP approach has not ended research for further improvements. The latest results in modeling food preferences [15], [16] will undoubtedly enhance and accelerate many more future applications.

The exposition of material is organized into the methodoligical pattern of a typical approach in management science: model building, solution algorithm, system design and applications. The conclusions spell out the revolutionary impact of the approach on food management technology.

The Model

A *menu* is an ordered list of menu items partitioned according to the structure of meals and courses within the meals for any given day. A *menu item* is a well-defined portion of *food ingredients* mixed and possibly processed according to the corresponding *recipe* (or formula) which specifies the list of food quantities that are required to produce the given item. The menu is called *selective* if more than one item is available for choosing within any one of the courses; otherwise, it is *nonselective*. The term *menu schedule* means the meal-by-meal, day-by-day listing of items over time. Scheduling or "planning" a menu is a decision process whereby it is determined which item should occur in which course, which meal and on which day. Such decisions are easiest to consider by allocating the items into meals sequentially, i.e., by a *multistage* decision rule. The complexity of scheduling for a fixed time horizon in a single stage is discussed in [14].

At every stage of this decision process a version of the classical diet problem called the "menu problem" [5] emerges: find the optimum combination of menu items which satisfies not only specified nutritional but also certain structural and compatibility requirements. If these requirements can be formulated to safeguard a desired level of acceptability of the meals, the optimum is considered as the minimum cost combination of items, i.e., the "best buy" meal. The mathematical counterpart of this decision problem is shown to be a zero-one variable multiple choice programming problem with special structures.

From these preliminaries, the formal statement, i.e., the mathematical model for any one stage of a nonselective multistage menu planning problem can be presented as follows:

$$
\begin{array}{lllr}
P1 & \text{Minimize:} & c^T x_t; \quad t = 1, 2, \ldots . & \\
 & \text{Subject to:} & \text{(i)}\ A_1 x_t \geqslant b & \text{(nutritional constraints)} \\
 & & \text{(ii)}\ A_2 x_t = 1 & \text{(structural constraints)} \\
 & & \text{(iii)}\ A_3 x_t \leqslant 1 & \text{(attribute constraints)} \\
 & & \text{(iv)}\ a_t^T x_t = 0 & \text{(separation constraints)}
\end{array}
$$

where

$x_t^T = [x_{1t}^T, x_{2t}^T, \ldots, x_{Kt}^T]$ is a partitioned n-vector of (0, 1) components with partition k $(k = 1, 2, \ldots, K)$, representing n_k items belonging to course k of a K-course meal, with $n = \Sigma_{k=1}^{K} n_k$. Component $x_{jkt} = 1$ means that the $j$the item is in the kth course of the tth meal.

b = m-vector of nutritional allowances to be satisfied by the meal.

a_t = n-vector of availability indicators of items for meal t. It is updated from the previous availability of items as follows:

(1) $$a_t = \max[(a_{t-1} - 1), 0] + Sx_{t-1}$$

for $t > 1$ where S is a diagonal matrix of the minimum number of days of separations between possible rescheduling of the same item or same kind of items on the menu. The item is available for scheduling if the corresponding component of a_t is 0. The separation constraints in (iv) assure the desired level of variety in the sequence of meals.

$c^T = [c_1^T, c_2^T, \ldots, c_K^T]$ is an n-vector of the "as-purchased" portion costs of menu items. It is computed from the recipes of the menu items which are supposed to be known. Consider a conceptual $(l + n)$ recipe matrix R with elements r_{ij} representing the amount of the ith ingredient needed to produce one portion of the jth menu item. Consider further the l-vector p^T to represent the "as-purchased" price per unit of food ingredients. Then

(2) $$c^T = p^T R$$

is the computational rule to obtain the coefficients of the objective function.

$A_1 = [A_{11} A_{12} \cdots A_{1K}]$ is the $(m \times n)$ partitioned matrix of the nutrient contents per unit portion of menu items. The coefficients of A_1 must be nutrient amounts per edible portion of the items since the definition of the b vector presupposes ingested foods and food composition tables [26] also list the nutrients values per edible weights. Let the $(m \times l)$ matrix H represent these values for m nutrients and l foods. If r_{ij} is the as-purchased quantity of ingredient i in item j, that amount may be reduced in food production by preparation, cooking and edible portion losses represented by the $y_{ij}^p, y_{ij}^c,$ and y_{ij}^e yield factors, respectively. Values of these factors are measurable or known in quantity cookery. Consider now q_{ij} as the processed and edible part of ingredient i in menu item j as

(3) $$q_{ij} = y_{ij}^p y_{ij}^c y_{ij}^e r_{ij}$$

Then the $(l \times n)$ matrix Q is defined as $Q = \|q_{ij}\|$. Consequently

(4) $$A_1 = HQ$$

is the computational rule to obtain precisely calculated values of the nutrient composition of menu items. The assumption here is that the elements of A_1 are constants. This is, of course, not necessarily true and in reality A_1 is more likely a matrix of random elements. Properties of such formulations are discussed in [7] and [2].

$$A_2 = \begin{bmatrix} 111\ldots.1 & & & \\ & 111\ldots.1 & & \\ & & \ldots\ldots\ldots & \\ & & & 111\ldots.1 \end{bmatrix}$$

is the $(K \times n)$ matrix of (0,1) multiple choice structure coefficients. Any one row of (ii) forces one and only one of the menu items associated with the course into the menu.

$$A_3 = \begin{bmatrix} 1 & & & 1 & 1 & 1 & \dots & 1 & \\ 11 & & 1 & & & & \dots. & & 1 \\ 11 & & & 1 & 1 & & \dots. & & \\ . & & . & & . & & & & . \\ . & & . & & . & & & & . \\ 1 & 1 & 1 & & & 1 & \dots. & & 1 \end{bmatrix}$$

is an $(r \times n)$ matrix of (0,1) attribute codes. These codes are applied to control compatibility on the menu by preventing incompatible combinations. Any one row of (iii) is restricting the corresponding attribute from occurring on the menu more than once. So lamb twice a day or fruit cocktail twice in one meal will not come up in the solution, if so desired. Compatibility can be enforced by any two rows of (ii). This option is used to avoid serving additional items with "extenders", for example, tuna-noodle casserole which represents more than one course. For instance, two rows of (iii) with coefficients as follows:

$$(5) \qquad \left[\begin{array}{cccc|ccc} 1 & 1 & 0 & 0 & 1 & 0 & 0 \\ & & & & & & \\ 0 & 0 & 1 & 1 & 0 & 1 & 1 \end{array}\right] x_t \leqslant \begin{bmatrix} 1 \\ \\ 1 \end{bmatrix}$$

will force the first item from the second course into the solution whenever one of the last two items in the first set is nonzero. The assignment of the attribute codes is arbitrary and is usually learned at each institution by trial and error.

The solution of model P1 produces a sequence of nonselective "best buy" menus which are or can be made as attractive as the user wants them. Here the decision is subjective and it is assumed that dietary experts will apply their judgment through constraints (iii) and (iv), such that the resulting menus will meet their standard of acceptability at minimum cost.

Another way to enhance acceptability of menus is to offer selections. The most popular form provides two choices per course. However, even this simple approach may generate up to 2^K different possible meals which cannot all be controlled deterministically for cost and nutrients anymore. Probabilistic control on the other hand would require the knowledge of conditional choice probabilities, for which data are nowhere available yet. Considering these difficulties a practical model was built to bypass the need for choice probabilities. If a selective menu is needed, problem P2 is solved immediately following P1 as follows:

P2 Minimize: $c^T x_t^s; \quad t = 1, 2, \dots .$

Subject to: (i) $A_2 x_t^s \leqslant 1$

(ii) $A_3 x_t^s \leqslant 1$

(iii) $a_t^T x_t^s = 0$

(iv) $x_t^T x_t^s = 0$

where x_t^s is the notation for the items representing the second choices on the menu. Model P2 does not have nutritional constraints since it can be argued that the first choices are available for those who want balanced meals.

In fact, hospital statistics show that 60-80 per cent of the meals served consists of these first, "standard" items. Consequently, the solution of P2 is constrained only by compatibility considerations, especially by (iv) which says that the choices must be different items. Although the choices are limited to two in this program, there is nothing to prevent food management from adding more choices to the menu by conventional methods, if desired.

A notable omission from the model is the explicit formulation of labor and cooking capacity constraints. Data on these aspects of food production are not usually transferable and the difficulty of data collection was found to be unjustifiable. To the extent that these factors mattered, the attribute constraints of the model were sufficient to achieve the desired production smoothing effects.

The model described above is the basis of the mathematical programming system in all of the computer assisted food management applications covered in this report.

The Algorithm

The multiple choice programming model of one stage of the multistage menu planning problem may contain 600-800 zero-one variables (menu items) partioned into 10-24 possible courses per day as well as 10-20 nutritional constraints, 10-24 multiple choice constraints and over 100 attribute constraints. In the early sixties there was no efficient algorithm available to handle integer programming problems of this size. Consequently, a special purpose algorithm was developed and more or less evolved using the technique of *truncated block pivoting*.

For the purpose of exposition the t subscript is suppressed in P1 and only the objective function and constraints (1) (nutritional) and (iii) (multiple choice) are considered for the time being. This system is referred to hereafter as the *reduced model*.

Any basic integer solution to this reduced system will satisfy the following equation in matrix-vector form:

$$(6) \quad \begin{bmatrix} 1 & 0 & c_B^T \\ 0 & -I & B \\ 0 & 0 & I \end{bmatrix} \times \begin{bmatrix} z \\ s \\ x_B \end{bmatrix} = \begin{bmatrix} 0 \\ b \\ 1 \end{bmatrix}$$

where B is a matrix of K columns, one from each of the submatrices of A_1, and s is a slack vector. The subscripts of the vectors c_B and x_B refer to the index set I_B of the variables in the basic solution.

With structure (6) in mind, there are two conditions of interest. If x_B is such that $s \geqslant 0$ does not hold, the solution is not feasible and the problem is in Phase 1. Otherwise, the problem is considered to be in Phase 2. In either case, change in the basis is considered under a rule which preserves the structure of (6).

Suppose that the index set of basic variables I_B in any solution is partioned such that

$$I_B = IO \cup IP$$

where IP is the index set of variables which will remain in the basis, and IO is the index set of variables leaving the basis at the next basis change, being replaced by a set of entering variables denoted by II. The cardinality of *II,* which is equal to that of *IO,* is called the block size in block pivoting.

$$r = N(II) = N(IO)$$

Obviously, $1 \leqslant r \leqslant K$ and $r = 1$ is assumed in conventional methods of basis change. Changing the basis while maintaining structure with $r = 1$, corresponds to moving on a path toward the optimum involving only adjacent lattice points in the solution space. Such a path is proven to exist only for a special class of set covering problems [4]. For any other problems, therefore, a path which is not limited to adjacent points, i.e., pivoting with blocks of $r > 1$ improves the chance to reach the optimum. Although the chances tend to improve and the length of the path tends to shorten with increased block size r, the combinatorial task of evaluating all possible sets of variables for a block becomes increasingly prohibitive. The algorithm described below minimizes the need to evaluate increasing block sizes by recursing, bounding, and truncation.

Phase 2 of Block Pivoting

In Phase 2 a feasible integer solution x_B exists with nonnegative slacks $s \geqslant 0$ and a corresponding objective value z_U which is also the upper bound on any other feasible solutions of interest. A lower bound z_L for the problem can be found by solving the reduced system as a linear programming problem.

An improved feasible solution $x_{(r)}$ is one with the property of $z_{(r)} < z_U$ after replacing a set of variables *IO* by a set of variables *II* where $r = N(II)$. Such a solution exists if a set *II* can be found such that

$$\text{(7)}\quad \begin{aligned} &\text{(i)}\quad \sum_{II} c_{jk} - \sum_{IO} c_{jk} < 0 \\ &\text{(ii)}\quad \sum_{II} a_{jk} - \sum_{IO} a_{jk} + s \geqslant 0 \end{aligned}$$

where the a_{jk} terms are column vectors of A_1.

There are $\binom{K}{r}$ ways of defining a block of size r, and for each case there are $\Pi_r(n_k - 1)$ distinct set memberships of *II*. When all such possible sets are enumerated, a set *II* can be found for which the difference in (i) is minimum while satisfying (ii), and the corresponding objective value is referred to as "min $z_{(r)}$." Another rule can be constructed by truncating the enumeration at the first set *II* where conditions (i) and (ii) are satisfied. The corresponding objective value is denoted by "first $z_{(r)}$."

Figure 1 shows the conceptual flowchart of block pivoting in Phase 2 with incremented block sizes, which are eventually truncated at some predetermined value $r_{\max}$. The reader may note that "min $a_{(r)}$" is evaluated only for r = 1 where the computational task is insignificant. Computational experience indicated that truncating enumerations at higher block sizes tends to yield solutions which could be further improved by smaller blocks, thus the upper bound is reduced more rapidly and usually the whole enumeration process becomes more effective. A more complete description of the algorithm in [9] and [2] describes the implicit bounding techniques used to reduce the number of necessary enumerations.

As the flowchart shows, pivoting continues until all possible blocks of size $r_{\max}$ and less are fully evaluated. The solution corresponding to the last value of z_U is considered to be the optimum. Computer programs for

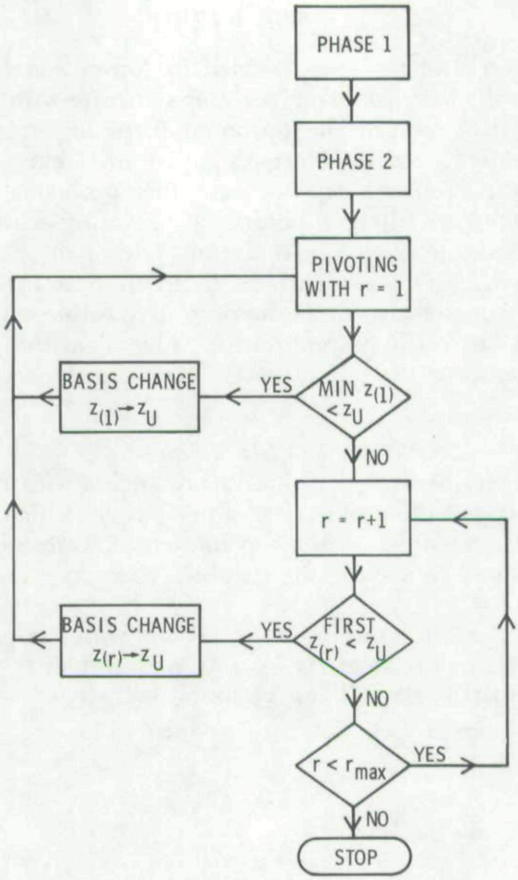

FIGURE 1. Conceptual Flowchart of Blockpivoting in Multiple Choice Programming

block pivoting with $r_{max} = 8$ were written, but computational experience indicates that optimum or near optimum solutions could be found with $r_{max} = 3$ in all cases. It seems that r_{max} should be considered a parameter to be set at the user's discretion. It is set to $r_{max} = 2$ in [10].

Phase 1 of Block Pivoting

Finding an initial feasible solution in mathematical programming is considered to be more an art than a technique, and a multiple choice programming block pivoting is no exception. Foolproof methods such as Balas' [3] are computationally demanding and are needed only when the set of feasible solutions is very small. Otherwise, linear programming solutions — available when z_L is computed — can be used with a rounding subroutine with favorable results. Finally, block pivoting itself is applicable with the modifications described in [2].

Let g_i be defined for each row i as

(8) $g_i = \sum_{II} a_{ijk} - \sum_{IO} a_{ijk} + s_i$

Another value, d_i, is defined for each row as follows:

$$(9)\quad d_i = \begin{cases} \sum_{II} a_{ijk} - \sum_{IO} a_{ijk} & | \quad s_i \leqslant 0; g_i \leqslant 0 \\ -s_1 & | \quad s_i \leqslant 0; g_i > 0 \\ 0 & | \quad s_i > 0; g_i > 0 \\ g_i & | \quad s_i > 0; g_i \leqslant 0 \end{cases}$$

Then the following sum, similar in notion to the "Balas value" [3] is computed:

$$(10)\quad D = \sum_{i=1}^{m} d_i$$

The value of D determines the possible reduction in total infeasibility due to the introduction of variables in set II in the basis. Consequently, if at the pth iteration the total infeasibility $T_s^{(p)}$ is defined as

$$(11)\quad T_s^{(p)} = \sum_{i=1}^{m} (s_i s_i < 0)$$

then at iteration $(p + 1)$

$$(12)\quad T_s^{(p+1)} = T_s^{(p)} + D$$

and if $D > 0$, a sequence of such steps will eventually eliminate infeasibility, given sufficient block size r.

In case there is more than one set II with $D > 0$, the pivoting rule is set to find

$$(13)\quad \min_{II,IO} \left\{ \left(\sum_{II} c_{jk} - \sum_{IO} c_{jk} \right) \Big/ D \, \Big| \, D > 0 \right\}$$

by evaluating all blocks of size r.

Gue et al [21] have found that the algorithm even with r = 1 was efficient compared to others and the only one feasible to apply in menu planning. Additional computational experience reported in [2] showed that the optimum was found with $r \leqslant 3$ in 40 of the 43 integer programming sample problems solved, with near optimums found within 99.8 per cent of the true value in the remaining 3 cases. Blackburn [17] and Warner [25] report successful applications to large capital budgeting and nurses' scheduling problems.

The multiple choice programming algorithm for use in menu planning includes a nonzero A_3 matrix in the structure as stated in P1. The extension of the algorithm to include the resulting large number of extra "packing" constraints was accomplished by starting with and maintaining feasible bases in terms of the columns of A_3. Feasibility requires that the basis columns of A_3 are mutually orthogonal. This condition is checked by representing the (0,1) coefficients of each column by binary bits and applying the logical AND function to the basis combination. At any basis change pivoting is restricted to those entries which are orthogonal, i.e., compatible with the others already in the basis.

The System

The mathematical programming model of menu planning assumes the availability of a large volume of accurate data. Expressions (2) through (4) show the use of these data in the model. Unfortunately, prior to computerization, food service systems usually do not have records in processable form. Therefore, implementation of mathematical programming is possible only in combination with an information system for securing the data base. There was (and still is) something else to be considered: Food management does not readily give up control to a computer as far as menu planning decisions are concerned. For these reasons the mathematical programming approach, including computerized information processing and planning, had to be introduced as a tool to assist food management. These were the premises on which the name and content of the System/360 Computer Assisted Menu Planning program documentation [10] were based. This documentation was released in 1969 through the IBM Contributed Program Library and is available in the public domain. The system was designed both as an operational and training device due to the novelty it represented in the computerization of food management. For this very reason, it contained many on-line options in order to realize the operant conditioning benefits of terminal access to the system.

THE CAMP SYSTEM
(Computer Assisted Menu Planning)

A. DATA PROCESSING

(i) Off-Line Programs:

1. FOOD ITEM FILE
2. NUTRIENT FILE
3. MENU ITEM FILE
   (a) Subassembly
   (b) Recipe
   (c) Menu Item Master
4. CROSS REFERENCE

B. INFORMATION RETRIEVAL

(i) On-Line Programs:

1. RECIPE YIELD CHANGE
2. FOOD USAGE
3. NUTRIENT TALLY
4. RANKED NUTRIENT SEARCH

C. MENU PLANNING

(i) Off-Line Programs:

1. TRIPLE GENERATOR
2. MATRIX GENERATOR

(ii) On-Line Programs:

3. SELECTIVE MULTI-STAGE MENU PLANNING

FIGURE 2. The Program Organization of the System/360 Computer Assisted Menu Planning System

The family of programs incorporated in the system is shown in Figure 2, details of which are reviewed briefly as follows.

The Food Item File contains the as-purchased food prices, i.e., the p vector in (2). The Nutrient File contains the elements of the H matrix in (4). The Menu Item File consists of three parts. The first two contain the recipes, i.e., the elements of both the R and Q matrices as in (2), (3), and (4) for the items as well as for "subassemblies" of ingredients, such as gravies, salad dressings, etc. The last file is the Master File of condensed data containing the computed values of portion cost and nutrient composition of the menu items, i.e., the c vector and A_1 matrix elements in (2) and (4). Figure 3 shows the computer image of a CAMP recipe coded for the Naval Medical Center, Bethesda, Maryland, with minor format modifications. The yield factors and nutrient codes represent the ingredient-by-ingredient effects of food preparation. The portion cost and nutrients at the bottom are data computed from (2) – (4) and are stored on the master file.

The first two of the on-line information retrieval programs are vital to food management. The Recipe Yield Change program rescales a given

RECIPE NAME VEAL SCALLOPINI — RECIPE CODE L103B

NO. OF PORTIONS 100 — ESTIMATED PORTION SIZE 5.65 OZS.

COLOR MULTI-COLOR FLAVOR MILD TEXTURE SOFT TEMPERATURE HOT ATTRIBUTES 0 0

MEAL CODE 4 COURSE CODE 3

INGR. NO.	INGREDIENT NAME	QUANTITY LBS.	QUANTITY OZS.	UNIT AMT.	UNIT MEASURE	****** YIELDS******* PRE-PREP	PREP	COOK	E.P.	NUTR. CODE	S. FOOD GROUP	CD
L03-4055	FROZEN VEAL STEAKS	25.00	0.0	100.00	4 OZ STEAKS	1.00	1.00	0.66	1.00	23820	0	0
165-6867	HARD WHEAT FLOUR	1.00	2.0	4.50	CUPS	1.00	1.00	1.00	1.00	26500	0	0
262-8886	SALT	0.0	5.1	0.50	CUP	1.00	1.00	1.00	1.00	19630	0	0
L03-2730	PEPPER	0.0	0.2	1.00	TBSP	1.00	1.00	1.00	1.00	37410	0	0
L03-2670	LIQUID SHORTENING-FRYMAX	57.00	2.0	7.75	GALLONS	1.00	1.00	1.00	0.03	14017	0	0
L03-2670	LIQUID SHORTENING-FRYMAX	−55.00	−6.5	7.52	GALLONS	1.00	1.00	1.00	1.00	14017	0	0
823-7663	FRESH GARLIC, CHOPPED	0.0	1.0	7.00	CLOVES	0.88	1.00	1.00	1.00	10290	0	0
128-1179	DEHYDRATED CHOPPED ONIONS	0.0	4.0	1.00	CUP	1.00	1.00	1.00	1.00	14140	0	0
227-1387	DEHYDRATED GREEN PEPPERS	0.0	2.1	0.50	NO. 2½ CN	1.00	1.00	1.00	1.00	34560	0	0
616- 81	SALAD OIL	0.0	3.9	0.50	CUP	1.00	1.00	1.00	1.00	14018	0	0
582-4060	CANNED TOMATOES, CRUSHED	12.00	12.0	1.67	NO. 10 CANS	1.00	1.00	1.00	1.00	22840	0	0
234-6217	BEEF GRAVY BASE	0.0	2.0	5.00	TBSP	1.00	1.00	1.00	1.00	30800	0	0
N00-888	WATER	4.00	2.7	2.00	QUARTS	1.00	1.00	0.57	1.00	27010	0	0
262-8886	SALT	0.0	1.9	3.00	TBSP	1.00	1.00	1.00	1.00	19630	0	0
127-8922	FRESH PARSLEY, MINCED	0.0	4.0	2.00	CUPS	1.00	1.00	1.00	1.00	14720	0	0
L03-2728	OREGANO	0.0	0.0	1.00	TSP	1.00	1.00	1.00	1.00	37340	0	0

PORTION YIELD IN OZS. 5.64 — PERCENT DIFF. −0.12 — PORTION COST $0.438

RECIPE YIELD IN LBS. 35.27 — TOTAL RECIPE COST $43.82

PERCENT WASTE 0.02

NUTRIENT COMPOSITION/PORTION

CALORY	PROTEIN	FAT	SFA	OLEIC ACID	LINOLEIC ACID	CHOLESTEROL	CHO	CALC	IRON	SODIUM	POTASSIUM	VIT A	THIAMINE	RIBOFLAVIN	NIACIN	VIT C
	GM	GM	GM	GM	GM	MG	GM	MG	MG	MG	MG	IU	MG	MG	MG	MG
287.	20.9	18.7	7.	6.	4.	74.	7.8	23.	3.0	1027.	522.	610.	0.10	0.22	4.6	16.

Source: National Naval Medical Center, Bethesda, Maryland.

FIGURE 3. Computer Image of an Institutional Recipe

recipe – usually defined for 100 servings – to the actual number of servings required. The Food Usage program on the other hand computes the quantity of foods needed for a given menu and population census. The conceptually simple computation is

where s is the sales vector with s_j denoting the number of persons demanding item j from the menu, and f is the vector of food quantities consistent with the demand. Needless to say, these programs are considered the most useful tools in food production, issuing and purchasing decisions.

The last two "information retrieval" programs serve mainly public relations purposes. Nutrient Tally computes for a known x_t vector (actual selection) the A_1x_t product and thus, it produces the calculated nutrient values of the meals. This is, of course, not necessary if the menu planning program is used. Finally, the last program prepares a list of items ranked according to their content of a specified nutrient. Again, the menu planning program obviates the need for such a search but it is a good device for teaching and for checking data accuracy.

The mathematical programming system of the CAMP documentation consists of one optional and two necessary parts. The optional part is the Triple Generator program which combines entree, starch and vegetable items into triples on the basis of the compatibility of their attributes because usually these three courses have the strongest interactions in compatibility. The program is basically a column generating device for the matrix generator with the potential of reducing the number of multiple choice constraints in A_2 and the number of rows in A_3. The second and third parts of the system operate both with and without triples.

The matrix generator program set up the coefficients of the system of equations in the format of models P1 and P2. There is provision for several independent matrix files in the system, each accommodating up to 600 menu items. Only those menu items can be "generated" which are already stored in the Master File, but by manipulating portion size and ingredient substitution options several versions of the same menu item can be generated for any of the matrix files. Different files may contain items for different seasons and, especially in hospitals, for different diets.

The program sets up the files according to the structure of the meals to be planned. The user answers the following questions:

1. How many meals per day are served?
2. How many courses (categories) are in each meal?
3. Which course in which meal is selective?
4. What are the menu components which do not change meal-to-meal (such as bread and beverage items)?
5. Which nutrients are to be controlled?
6. Is the triple generator going to be used?

After these questions are settled the user arranges input cards, one for each item to be included in the matrix. These cards have the separation ratings of the particular items and are followed by an input deck with the separation ratings of all the attributes. The manual suggests the collection of separation ratings through population surveys so that the menu will reflect the preference of the customers and not just that of food management.

HOSPITAL	1		2		3		4	
MENU PLANNING METHOD	CONV.	COMPUTER	CONV.	COMPUTER	CONV.	COMPUTER	CONV.	COMPUTER
FOOD COST ¢	77.46	69.60	96.76	83.78	125.40	106.22	118.49	102.62
Realized Cost Saving per patient per day	7.86¢ (10.15%)		12.98¢ (13.41%)		19.18¢ (15.30%)		15.87¢ (13.39%)	
Food Price Level	1 July 1966		24 March 1967		12 October 1967		12 March 1968	
COMMENTS	Selective Menu for 31 Days		Non-Selective Menu for 21 Days		Non-Selective Menu for 28 Days		Selective Menu for 14 Days	

Source: Computer System Research, Tulane University

FIGURE 4. Food Cost Evaluation of Conventional and Computerized Menu Planning in Four Hospitals

The Selective Multistage Menu Planning Program solves problems P1 and P2 in successive stages of meals and days and the optimum solution appears in English (or any other language) under the control of a dietary expert. In order to facilitate the exercise of this control, the program takes instructions on-line through a preprogrammed terminal dialog. Figure 4 shows some elements of this dialog. Some of the instructions are possibly of interest.

1. The SET command allows the user to force one or more items, such as fish on Friday, in the menu before planning begins. The algorithm will not change these items in the solution.
2. The ENTER command allows replacing an item by the one the user is ready to enter provided it is available. For this reason it is good to use the LIST command to find out what is available first. ENTER causes a postoptimal reoptimization process which may result in changing other items as well on the menu.
3. The FORCE command does the same as ENTER but without checking the availability of the item.
4. The MENU command freezes the items and makes changes possible without reoptimization.
5. The NUTRIENT and SNUTRIENTS command prints the nutrient composition of the first or second choices on the menus respectively.
6. GO is the command to start planning the next menu. It takes about 30 seconds of computer time to solve problem P1 (and P2).

Interactions between coding and operating procedures are permitted. For example, if these happen to be two items with gravy on the optimal menu, the user may correct this minor case of incompatibility on-line or the management may also elect to note the gravy attribute and apply a gravy code to these items in the matrix generator. In that case the source of any future incompatibilities of this sort is eliminated. Such adaptive operations soon will obviate the need for on-line adjustments and indeed most of the applications operated now without it. At the University of Tubingen, for instance, day-codes applied to the items do away with the SET command.

The system was designed to give maximum flexibility in operation and the users seem to be able to take full advantage of it, as the next section amply demonstrates.

Applications

The first large scale test of the modeling and system concepts took place at Tulane University with the cooperation of four hospitals where data processing and menu planning implementations were demonstrated for a period of about five years. Each hospital used the system and served the menus produced by the system for time periods ranging from two to three years. Most of this time was typically spent on data coding, especially recipe coding, learning and experimentation.

Figure 4 shows the concluding statistics of the implementations with respect to food cost saving and nutritional control. In each hospital the menus planned by conventional method were priced out and also analyzed for calculated nutrient values prior to serving the menus planned by computer assistance. The hospitals did not use uniform nutrient allowances but what was specified was always satisfied by the computer. The menus were served to the patients throughout a number of menu cycles without complaints either from management or patients. The 10-15 per cent relative cost saving is a conservative figure but it is in full agreement with the known hypothesis that the minimum of a mathematical programming problem cannot be found by nonmathematical means.

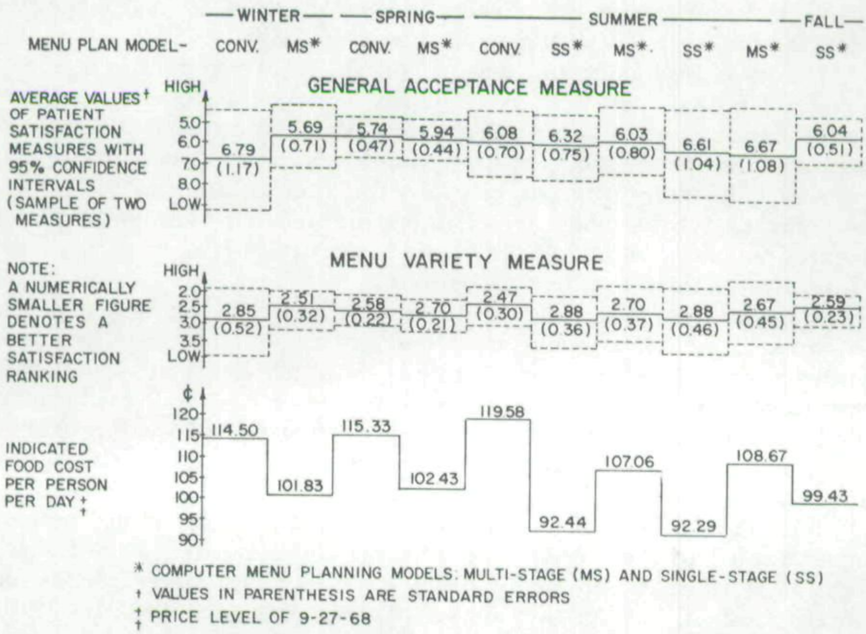

FIGURE 5. Comparison of Patient Satisfaction Measures and Raw Food Cost for the Regular Selective Menus

One of the hospitals carried out further evaluations including explicit food acceptance measures as described in [20]. Figure 5 shows an interesting time series of data resulting from this experiment. The hospital switched back and forth in serving conventionally and computer planned menus while food cost and acceptance was measured in each period. The repeated drops in food cost again justify the benefits of mathematical programming and something else. The single stage (SS) model (described in [11]) was supposedly solving the same problem but without on-line access and hence adjustments. The cost differentials between these two models reflect among other things the postoptimal prerogatives of dietary experts in improving on the menus although improvement did not really show in the acceptance ratings.

By the time the Tulane system completed its task, the CAMP documentation became available and two of the hospitals converted to its use. However, the first documented application of CAMP took place at Central State Hospital in Georgia, where it is still in use and serving 7200 patients a day. The success story of this application is written up in [22] reporting about 5 per cent average food cost saving, which was more than enough to pay for the computerization.

Another application of the CAMP system with potentially far reaching impact was initiated by the U.S.D.A. Food and Nutrition Service for planning low cost school lunch menus. The system was implemented and evaluated in two school systems, one in Miami, Florida, and the other in Memphis, Tennessee, involving about 10 schools and 5,000 students in each case. The U.S.D.A. applications are not yet formally evaluated but it is already known that both school systems elected to keep the computer system in operation and plans for the extension of implementations are underway. Factors in this decision are related to better nutritional control, lower menu cost and last but certainly not least, the value of the computerized food management system in eliminating waste and errors in accountability.

The most ingenuous and original of the CAMP applications is at the 2100-bed University of Tubingen Hospital in West Germany. This system is unique for it contains a full conversion to the metric system and is the only one known to operate with the utilization of the triple option in the matrix generator. This use can be considered as the ultimate proof of the versatility and generality of the model for planning human diets. The menu planning program operates off-line but the menus are reviewed by a panel after which it is rerun with the use of the SET options.

Although the menus have been served in Tubingen for over a year, cost evaluation is not yet available but the major benefits of the system are recognized as providing better patient care in terms of menu variety and again in the information processing assistance of the computer to food management decisions and control. The application also revealed the usual problems in coding recipes and in enforcing the corresponding standardization process in food production.

Since CAMP is in the public domain its application is not necessarily communicated to the author or published. However, it is known that application was initiated in St. Luke's Hospital in Houston, Texas, and in St. Vincent Hospital and Medical Center, Toledo, Ohio, in 1969 and 1970, respectively. Also a number of teaching institutions such as the University of Missouri Medical Center, the University of Southern Mississippi, Southern University of Louisiana, and the University of Pretoria, South Africa, are using the system for educational purposes. One special conversion of the CAMP system was made under an O.N.R. contract according to the specifications of the National Naval Medical Center and is awaiting application there presently. Another conversion is underway involving the University of Massachusetts Food Service Department.

Further applications of the CAMP system are considered here to be those done by the Medicus System Corporation in the last two years without denying them credit for their own contributions to its viability. The commercial success of the Medicus conversions and applications is an implied proof of the overall benefits of the approach to food management. Further Medicus applications are included in the impressive list of Figure 6.

As was mentioned earlier, the Tulane system was also converted in 1969 and since then Trans Tech Inc. has been using a central computer facility to offer an on-line management information system to its clients. The initial success of this particular approach is documented in [18] claiming a record high 34 per cent saving in the Research Hospital, Kansas City. Figure 6 shows that at least seven more successful applications can be credited to Trans Tech to date. It is said that users seem to appreciate the food management information processing part of the system more than the menu planning part but this may be due to some of the shortcomings inherited from the earlier versions of the model used at Tulane University.

Figure 6 gives a good overview concerning the versatility and the spread

PROGRAMS	TYPES OF USE								
	Hospitals		Schools		Mental Institutions		Penal Institutions		Teaching
	Number	Persons Served	Number	Persons Served	Number	Persons Served	Number	Persons Served	Number
S/360 CAMP	4[1]	4,600	3[2]	25,000	1[3]	7,200	–	–	4[4]
Medicus	5	2,800	–	–	1	20,000	1	13,000	–
Trans Tech	8	4,600	–	–	–	–	–	–	–
TOTALS	17	12,000	3	25,000	2	27,200	1	13,000	4

1. Includes the metric conversion of CAMP at the University of Tübingen.
2. Includes two U.S.D.A. applications and conversions to the University of Massachusetts Food Service System.
3. Started with CAMP, now a Medicus client.
4. Includes three universities in the United States and one in South Africa.

FIGURE 6. Summary of Known Applications of Mathematical Programming Systems for Food Management as of March 1975

of the applications. It implies years of successful continuous use in most cases but it also includes applications just underway. It does not include a possibly large number of other applications not known to the author. It is fair to estimate that by the end of the year about 100,000 persons per day will be the recipients of scientifically balanced meals through the systems and more than a million have been enjoying its benefit for some time already.

Conclusions

The introduction and growing acceptance of the mathematical programming system described in the previous sections signify the advent of an important technological change about to be felt in food management and in human nutrition.

From food management's professional point of view [24], the approach is applauded and its multitude of benefits is readily recognized. The repeated proofs of significant cost saving through minimized "best buy" menus imply that without mathematical programming assistance, the food service industry is operating at about 10 - 15 per cent average opportunity loss. This loss can be directly attributed to the suboptimal decision rules applied presently in the conventional processes of menu planning. The repeated reference to the information process service of the system proves that computerization alone tends to serve managers' best interest and thus nicely overlaps the accessability of accurate data base for the models of management scientists. Improvements on these models are necessary, feasible, [1], [16] and inevitable.

The menus provided by the system are not only low cost, but are also nutritionally balanced. It turns out that the system presently supplies the only scientifically acceptable methodology to plan nutritionally adequate meals. Reports such as in [20] and [23] assert that by the current pencil-and-paper methodology, dietitians cannot assure nutritional adequacy much less at least cost. In this respect the system heralds the need for a complete overhaul of dietary education as we know it today [13]. The impact of this prospect on the nutritional health of future generations is inestimable.

Some of the ramifications of the mathematical programming approach are realized in a new approach to food price index theory [11] and opens the way to think in terms of mechanisms which describe and prescribe optimum food intake changes in the function of food price variations. Advancements in the quantitative measurements of food preferences [15], [16] will certainly enhance progress in this direction.

Finally, contributions to the state of art in Management Science and Operations Research should also be mentioned. To date, at least four Doctoral dissertations and about the same number of masters' theses have been written on the subject, not to mention the large number of scholarly articles of which [1] is one of the latest and best examples.

References

1. Armstrong, R. D. and P. Sinha, "Application of Quasi-Integer Programming to the Solution of Menu Planning Problems with Variable Portion Size", *Management Science*, Vol. 21, No. 4 (December 1974) pp. 474-482.
2. Armstrong, R. D. and J. L. Balintfy, "A Chance Constrained Multiple Choice Programming Algorithm", *Operations Research*, Vol. 23, No. 3 (May-June 1975).
3. Balas, E., "An Additive Algorithm for Solving Linear Programs with Zero-One Variables", *Operations Research*, Vol. 13 (1965), pp. 517-549.

4. Balas, E. and M. Padberg, "On the Set Lowering Problem", *Operations Research,* Vol. 20 (1972), pp. 1152-1161.
5. Balintfy, J. L., "Menu Planning by Computer," *Communications of ACM,* Vol. 7, No. 4 (April 1964), pp. 255-259.
6. Balintfy, J. L. and E. C. Nebel, "Experiments with Computer Assisted Menu Planning", *Hospitals,* Vol. 40, No. 11 (June 1966), pp. 88-96.
7. Balintfy, J. L. and A. Prekopa, "Nature of Random Variations in the Nutrient Composition of Meals", *Health Services Research,* Vol. 1, No. 2 (Fall 1966) pp. 148-169.
8. Balintfy, J. L. and Research Staff, "Computerized Dietary Information System", Vols. I-III, Monograph, Computer System Research, Tulane University, New Orleans, La., (1967) University Microfilm No. DI-0052-54.
9. Balintfy, J. L. and C. R. Blackburn, "Planning and Scheduling Optimum Menus in Multi Stage", Computerized Dietary Information System, Vol. IV, Monograph, Computer System Research, Tulane University, New Orleans, La. (1968) 109 pages.
10. Balintfy, J. L., *"Systems/360 Computer Assisted Menu Planning."* Contributed Program Library No. 260D – 15.2.013, IBM Corporation, PID, Hawthorn, N. Y. (1969) 286 pages.
11. Balintfy, J. L., J. Neter and W. Wasserman, "An experimental Comparison Between Fixed Weight and Linear Programming Food Price Indexes", *J. of Amer. Stat.,* Vol. 65, No. 329 (March 1970), pp. 49-60.
12. Balintfy, J. L., "Computer Assisted Menu Planning and Food Service Management", Monograph, School of Business Administration, University of Massachusetts, Amherst, Ma. (1971) 85 pages.
13. Balintfy, J. L. "Mathematical Modeling and Human Nutrition", *Science,* Vol. 181, (10 August 1975) pp. 581-582.
14. Balintfy, J. L., "Large Scale Programming Properties of Menu Planning and Scheduling", in *Optimization Methods,* ed. R. W. Cottle and J. Krarup, The English University Press Ltd., London, 1974, pp. 81-98.
15. Balintfy, J. L., "Mathematical Models for Menus", *Naval Research Reviews,* Vol. 27, No. 7 (July 1974) pp. 1-10.
16. Balintfy, J. L., W. J. Duffy and P. Sinha, "Modeling Food Preferences Over Time", *Operations Research,* Vol. 22, No. 4 (July-August 1974) pp. 711-727.
17. Blackburn, C. R., D. R. Wood and J. L. Robertson, "A Mathematical Programming Approach to Capital Allocation with Risk", Abstract, *Operations Research Bulletin,* Vol. 19, Sup. 2 (Spring 1971).
18. Bowman, J. D. and E. M. Brennan, "Computer Assisted Menu Planning Provides Control of Food Service", *Hospitals,* Vol. 42, No. 103 (August 1969), pp. 57-79.
19. Cohen, E., "The Computer and St. Joseph's", *Food Management,* Vol. 8, No. 8 (August 1973) pp. 57-79.
20. Gelpi, M. J., "Integrated Nutrition and Food Cost Control by Computer", *J. Am. Diet. Assoc.,* Vol. 61, No. 6 (December 1972) pp. 637-646.
21. Gue, R. L., J. C. Liggett and K. C. Cain, "Analysis of Algorithms for the Zero-One Programming Problems", *Communications of the ACM,* Vol. 11, No. 12 (December 1968) pp. 837-844.
22. McNabb, M. E., "90-Day Nonselective Menus by Computer", *Hospitals,* Vol. 45, (May 16, 1971) pp. 88-91.
23. Head, M. K., R. E. Weeks and E. Gibbs, "Major Nutrients in the Type A Lunch", *J. Am. Diet. Assoc.,* Vol. 63, No. 6 (December 1973) pp. 620-625.
24. Patterson, W., "The Computer and You", *Food Management,* Vol. 8, No. 8 (August 1973) pp. 41-45.
25. Warner, D. M., "Scheduling Nursing Personnel According to Nursing Preferences: A Mathematical Programming Approach", Paper, The University of Michigan Program in Hospital Administration, May 1973.
26. Watt, B. K. and A. C. Merril, *Composition of Foods,* Agriculture Handbook N. 8, USDA, Washington, D. C. 1963.