commencement of the design, no less than a two-month course in the fundamentals of electronic computers and their programming. The groups from the administrations should be under dual subordination, one line to the administration and one line to the chief of the design group. Such an organization of effort will ensure the preparation of problems that are of actual worth to the sovnarkhoz.

The first part of the design should be to agree upon a concrete list of problems that can be solved on the computer with satisfactory efficiency. This will be a type of information model for the computer center.

The second part of work should be the formulation of a technical design. This will depend entirely upon the results of the applications prepared. In accordance with the characteristics of the "raw material" (e.g. the volume of data, the type of problems proposed for solution, the distribution of these throughout the course of the year), it will be necessary to choose the computer required, the type of arrangement of individual blocks, the means of communications and the physical plant.

Finally, it will be necessary to study the influence of the computation center on the work and structure of the apparatus of the sovnarkhoz. It will be necessary to insure, from the very beginning, a close and practical contact between these two.

It would be advisable to include all of this in a concrete organizational plan for carrying out the design work. This plan should be controlled directly by the top management of the sovnarkhoz. The creation of a computer center is not an ephemeral problem that can be delegated indiscriminately; it is one of the fundamental responsibilities of the top managers of the sovnarkhoz or enterprise.

At the present time, unfortunately, there is no unified plan for accomplishing the task of management automation; and such a plan is acutely needed. It is extremely important to determine, furthermore, the particular implementors of that plan who would direct this sort of work on a state-wide scale. In addition to the All-Union Institute of the Organization and Technology of Management in Minsk, it is necessary, in our opinion, to create similar institutes in a series of large economic regions. In these, design efforts should be carried out, and the institutes should have the responsibility for implementing their recommendations. It would be advisable to create (more accurately to recreate) the journal, *Technology of Management*, for the exchange of opinion. (Incidentally, such a thing would help to overcome the kind of sensational publicity on computers that occasionally appears in the popular press and which disorients readers.)

Electronics is a powerful tool, but it demands deep knowledge, great research effort, and state leadership and control.

TRANSLATED FROM THE RUSSIAN BY R. W. JUDY, 10 MAY 1963; TRANSLATION RECEIVED BY *Communications of the ACM* EDITOR, NOVEMBER, 1963.

Menu Planning by Computer

JOSEPH L. BALINTFY
Tulane University,* *New Orleans, Louisiana*

A computer code has been developed which plans menus by finding minimum cost combinations of menu items such that the daily dietary, gastronomic and production requirements can be satisfied for a sequence of days. A fast, special integer programming algorithm is described which approximates the theoretical solution to the problem. If necessary, any menu can be changed online and then post-optimized. Up to 30 percent saving on food cost is possible. A FORTRAN program for the IBM 1410 is available on request. A considerable amount of data processing must precede the implementation of the system.

Introduction

Decisions affecting the nutritional qualities and cost of food service in institutions are made in the form of "menu planning." The general objectives of menu planning are recognized as achieving palatable, nutritionally balanced and economical diets, that is, satisfying a certain set of simultaneous requirements for which the determination of optimum conditions has long been identified as a linear programming problem [1, 2]. Previous attempts in the well-known form of "diet problems" generally omitted the process of food preparation from the conceptions of the models, and therefore failed to become operational human-diet propositions.

The ongoing research of the author and his associates was based on the realization that decisions concerning the service and consumption of food are made in terms of menu items and not food items. The term *menu item* as used here, is the name of the combination of *food items* (ingredients) called for in the recipe. A menu item is either monoingredient (such as orange juice) or polyingredient (such as beef stew). On the menu the menu items are classified as *menu components* (such as appetizer, entree, etc.). This distinction seems to be a necessary and sufficient condition to establish the link between the operational objectives of menu planning and feasible mathematical programming techniques.

Since menu items are prepared from food items according to accepted recipes and hence can be regarded as palatable *per se*, the objectives of palatability in menu planning are carried through by imposing formal requirements on the structure of menu components and also on

This investigation was supported in whole by Public Health Service Research Grant HM 00216-02, from the National Institutes of Health.

* School of Business Administration.

the variety of menu items within the components. Inasmuch as these requirements are combined with nutritional and economic objectives, a new version of the classical diet problem, the "menu problem" emerges. The menu problem is concerned with finding the optimum combination of menu items which satisfies specified nutritional, structural and variety requirements for a sequence of days. The optimality condition can be the minimization of cost or maximization of some measure of consumer's satisfaction, or both. The structural requirements are the customary array of the components in menus. These can be for instance, appetizer, entree, cereal, bread and beverage for breakfast, and appetizer, entree, starch, vegetable, salad, dessert, bread and beverage for dinner and similarly for supper. More or less of these components make up the structure of menus depending upon the habits and standards of households and institutions. Finally, the variety requirements are satisfied either through selectivity, i.e. offering more than one menu item per component or by restricting the repetition of items for a number of days or by the combination of both procedures (cafeteria system).

Though several linear and nonlinear programming models can be and have been constructed for different menu problems, discussion in this paper will be limited to the solution of the simplest type of institutional menu planning problem. This is the problem of planning nonselective menus for a specified number of days which satisfy at minimum cost the daily dietary requirements and a desired degree of variety. It has been found that even with these relatively simple conditions the menu problem poses excessive computational work on computers with standard mathematical programming codes. After some experimentation a short and fast computer algorithm was developed for the programming of menu planning which is described below.

Dietary Information Processing

The solution of any menu problem on a computer must be preceded with the acquisition of data on items being considered for the menus. A short description of the corresponding data processing work as it was done on the author's project is outlined here.

Several hundred menu items were numbered and classified into mutually exclusive classes of menu components. Deleting bread and beverage items, 15 components were used in the experimental menus. Menu items were deliberately classified as either dinner or supper components to control color and consistency requirements in their possible combinations in the menus. The recipes of the menu items were "translated" into lists of food numbers and quantities of food in a serving in 100-gm. units. This list supplied all the information to calculate the price (cost) and nutrient composition per portion of menu items, which were subsequently stored on a master tape [3]. The upper part of Figure 1 shows the scheme of data processing. A special feature of this program is a fast and independent subroutine which updates the master tape

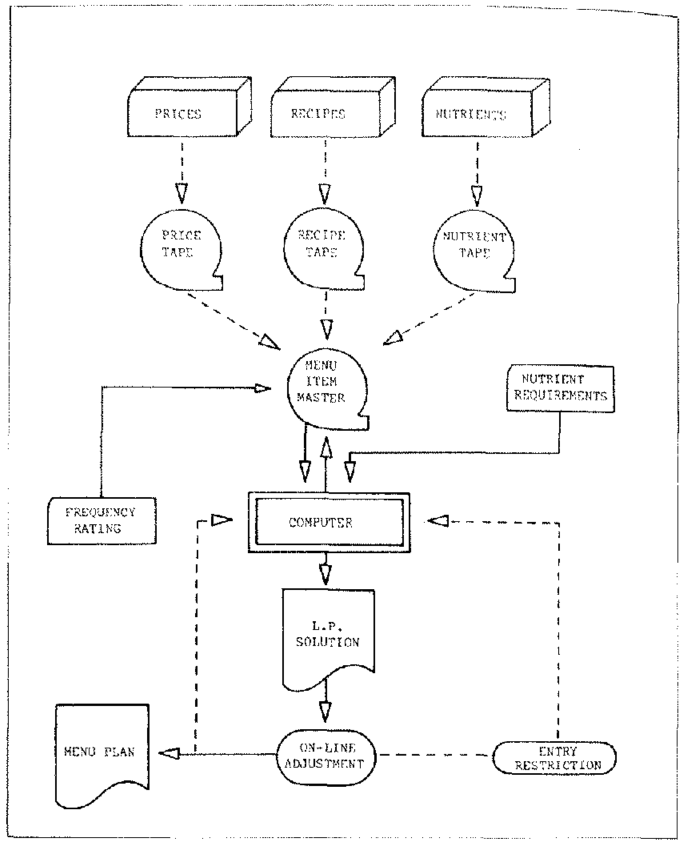


Fig. 1

in the event of food price changes. For this purpose disk storage might be preferred to tapes in actual operations.

Another necessary set of attributes for menu items was obtained through opinion sampling studies, aimed at the determination of some measure to express preferences for menu items and variety. It was found that the preference and the desired frequency of serving of menu items are correlated closely enough to arrive at palatable menus based only on the latter measure [4]. Hence statistics of frequency ratings were calculated and stored on the master tape for each menu item. These numbers indicate the time interval in days for which the same item should be restricted from entering in the menus in order to preserve variety and an implied level of palatability on the menu plan. This concept of "entry restriction" defines an autogenerated sequence of varying subsets of menu items available for consideration each day once the planning process is started. Denoting the reciprocal of the frequency rating of menu item j in class k by f_{jk}, the condition of rotation is satisfied if $\sum_{j=1}^{n_k} f_{jk} = f_k > 1$ for all k. An attempt was made to keep $f_k \cong 3.5$ for each class of items included in the experimental runs.

Summing up, the dietary information processing should be based on standardized recipes, portion sizes and on the classification of menu items according to the desired number and structure of menu components for meals per day. After these prerequisites are met, the cost and the nutrient composition of menu items can be calculated and stored. Finally the frequency rating of items is determined and stored along with the other attributes on a master record. This record should be updated intermittently due to price or rating changes. In this paper the frequency rating is used as preference indicator but only for its simplicity and not from necessity. Additional direct measures of preference or acceptance as well as food processing time and capacity data may complete the records if it is so desired. Similarly, the nine nutrients adopted in the present program may be extended to a larger list including, for instance, the essential amino acids, sodium, fatty acids and so forth.

Mathematical Formulation of the Menu Problem

In conformity with the concept of daily dietary requirements [5] the formulation of the menu problem for one day only will be discussed. Let us denote the dietary requirements by the customary m-component b-vector, and the nutrient composition per serving of menu items in class k $(k = 1, 2, \cdots, L)$ by the A_k-matrix, with c_k' denoting the corresponding cost vector, and M_k indicating matrices with units in the kth row and zeros elsewhere. With this notation the simplest menu problem corresponds to the linear programming model (1) in revised simplex form where z is to be minimized and each

$$\left|\begin{array}{c|c|c|c|c|c} 1 & c_1' & c_2' & \cdots & c_L' & \mathbf{0} \\ \hline \mathbf{0} & A_1 & A_2 & \cdots & A_L & -I \\ \hline \mathbf{0} & M_1 & M_2 & \cdots & M_L & \mathbf{0} \end{array}\right| \cdot \left|\begin{array}{c} z \\ \hline \\ \mathbf{x} \end{array}\right| = \left|\begin{array}{c} 0 \\ \hline b \\ \hline \mathbf{1} \end{array}\right| \qquad (1)$$

variable, except the slacks, is bounded by the $0 \leqq x_j \leqq 1$ interval, such that $x_j = 0, 1$. On the right-hand side the requirements vector is partitioned into nutritional requirements b and "structural" requirements, represented by the sum vector $\mathbf{1}$, which contains L unit components. This specification states no more than the stipulation that the optimum menu under consideration in this paper should be a minimum cost combination of menu items such that each component is represented by one menu item and the total nutrient content of the menu for the day is greater than or equal to the nutritional requirements.

Even moderate size problems of this type lead to a matrix tableau with a minimum of 25 rows and 200 columns, requiring a new solution every day. The present state of art in mixed integer programming would not make the application of the exact model economically feasible [6]. Standard L.P. codes, especially with the customary two-phase technique and on moderate size computers, are also out of the question for the same reason, although the regular simplex algorithm itself offers a good approximation to the problem. It turns out that due to the upper bounds most of the nonslack variables in the solution are integers and on the average less than one fourth of them are fractions. This algorithm was used to calculate reference points for later comparisons (see Fig. 2). In order to arrive at an efficient computer code for the menu problem a new approach was needed and was derived from the special structure of the L.P. problem as represented by (1).

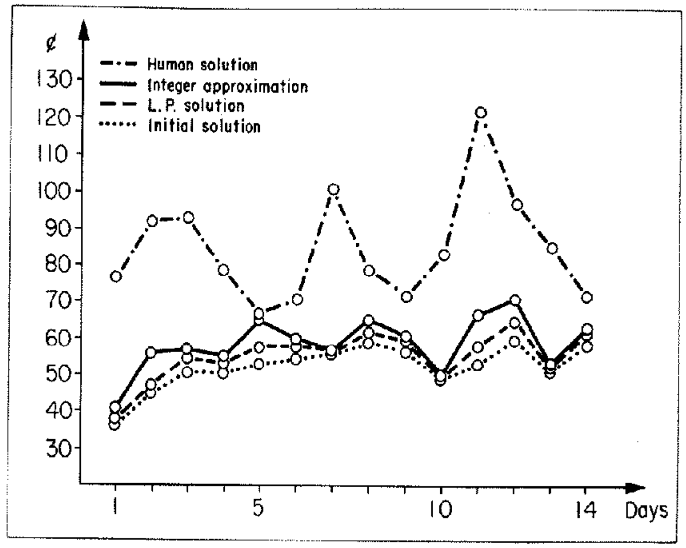


FIG. 2

Any feasible solution to the problem (1) would require that all the structural constrains are satisfied. If this solution happens to be in integers for all the L activity variables, the basis has the following structure:

$$B_* = \left|\begin{array}{c|c|c} 1 & 0 & c_B' \\ \hline \mathbf{0} & -I & B \\ \hline \mathbf{0} & \mathbf{0} & I \end{array}\right| \qquad (2)$$

and the solution is

$$\left|\begin{array}{c} z \\ \hline \\ x \end{array}\right| = B_*^{-1} \cdot \left|\begin{array}{c} 0 \\ \hline b \\ \hline \mathbf{1} \end{array}\right| = \left|\begin{array}{c} \sum_k c_{Bk} \\ \hline \sum_k a_{Bk} \\ \hline \mathbf{1} \end{array}\right| = \left|\begin{array}{c} z \\ \hline x_s \\ \hline \mathbf{1} \end{array}\right| \qquad (3)$$

since B_*^{-1} is the same as B^* with only the sign of c_B' changed. Here B and c_B' correspond to L columns, one from each of the k classes of (1) and x_s represents the vector of the nonnegative slack (surplus) variables.

It is easy to realize that at least one feasible configuration similar to (2) exists with $z = z_{\min}$ and corresponds to the optimum mixed integer solution. The approximation of this optimum can be achieved with great simplification if basis changes are performed with the preservation of the structure of B_* and feasibility, once this form is reached. In this case, the criterion for the jth vector entering the basis is $\min_j (z_j - c_j) = \min_{j,k} (c_{jk} - c_{Bk})$ for negative differences and the new variable replaces the old one in the same k-class at the unit level unless this change violates the feasibility $(x_s \geqq \mathbf{0})$. These iterations do not require matrix inversion and result in extremely fast programs. The end-result of the iterations is always a feasible solution in integers—except the slack variables—but it is not always the integer optimum, since the exact but complicated method of integer programming iterations is deliberately bypassed in favor of speed and simplicity. Experience has shown, however, that exchanging accuracy for efficiency pays off because the above described algorithm reaches the optimum integer solution in the majority of the cases and produces good approximations in the remaining ones (see Fig. 2).

An important condition of success with this method is finding a proper initial basis in the form of (2). After some experimentation it was found that a version of the dual-simplex algorithm with the limitation of basis changes to the unit level provides the best results for this purpose with a minimum number of iterations. In many cases the first feasible solution obtained by this method proves to be the optimum without further iterations.

Computer Program

There are several reasons why running speed and simplicity should be highly preferred in programming menu plans on a computer. Maybe the most important of them is the human factor which necessarily remains in charge of ultimate judgment and responsibility in considering menus. Such an irrational and sensitive human trait as taste will defy computer logic for some time to come. This condition implies that some of the computer optimums will not necessarily pass as "optimum menus," in the opinion of some dietary authorities. Though effort can be made to minimize these occurrences, they never can be eliminated by programs alone. Consequently, menu planning on computers must be visualized as a joint man-machine decision-making activity. The prime importance is not that the computer can do a better job than a human being in planning menus; what is important is the fact that a human being, a dietary expert, can use a computer to help solve an extremely complex problem; that the computer results can be accepted, rejected or changed according to the objectives and standards of any individual or institution. These considerations call for a program with the option of online adjustments.[1] The online feature in turn limits the allowable time for calculations even on small computers to about a minute if it is to be a routine operation. The program described here works within this limitation.

The general scheme of the program follows Figure 1, and is based on the condition that menu plans are made for a number of days (usually two or more weeks) ahead of time. For menu planning runs, only the updated master tape is needed and a card-input containing the nutrient requirements. The sign of this data may indicate the type of inequality to be satisfied. The author's program was written for an IBM 1410 system (40K) which has sufficient capacity to place the upper part of (1) in the memory from the tape at the beginning of the run. Once input is provided, the program works in the following phases.

1. *Initial solution:* setting the least expensive, nonrestricted items in each component class, $k = 1, 2, \ldots, L$, to the upper bounds. This arrangement usually leaves some of the b-vector requirements unsatisfied ($b_i < 0$ for some i's).

2. *Initial Feasible Solution:* change of vectors on the upper bounds following rules similar to the dual simplex algorithm but observing the requirement that vectors are entering the basis at unit level and are replacing menu items only within their class. The criterion for vector a_{jk} in class k to enter is

$$\min_{j,k}[(c_{jk} - c_{Bk})/(a_{i_0jk} - a_{i_0Bk}) \mid a_{i_0jk} > a_{i_0Bk}; c_{jk} \geqq c_{Bk})] \tag{4}$$

where i_0 is determined from

$$b_{i_0} = \min_i(b_i \mid b_i < 0) \tag{5}$$

Change of vectors goes on until all b_i are positive and primal feasibility is obtained. The result of these iterations already is optimum in many cases.

3. *Optimum Feasible Solution:* further change of vectors on the upper bound as long as negative differences, $\min_{j,k}(c_{jk} - c_{Bk})$ can be found such that $(b + a_{jk} - a_{Bk}) \geqq 0$, i.e. the feasibility is not violated.

4. *Online Adjustment:* the menu is displayed on the printer as shown in Figure 3. Through a reactive typewriter, changes can be made, if needed, by forcing in desired menu items until the qualitative character of the menu is acceptable (like fish on Friday). Each change is followed by a post-optimization routine through phases 2 and 3 if the balance of the nutrients was disturbed.

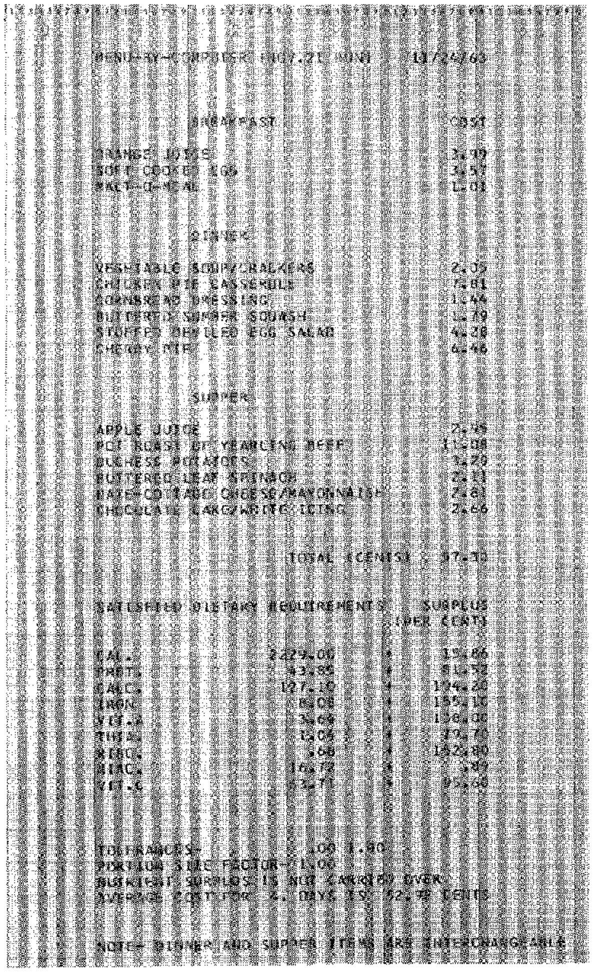
5. *Updating Entry Restrictions:* if the menu is accepted, the program marks entry restrictions on the items included in the menu and cancels restrictions expiring on the following day. The new menu is calculated for the next day, starting at phase 1 again, until planning is terminated.

Results and Conclusions

Experimental runs with the use of the above described menu program were conducted in such a way that an unbiased comparison between menu plans in general use and those calculated by the computer was possible. Figure 2 shows the results of the comparison of raw food cost per day of menus planned by different techniques. The average cost of food per day (less the cost of bread and beverages) was 82.86 cents for a two-week menu cycle in the hospital selected for the comparative studies. Applying the decision rules of the integer approximation in the menu program, decreased the food cost to 58.55 cents per day, equivalent to a 30 percent saving. It is also worth mentioning that the nutrients were satisfied each day by the computer program, while in the hospital menu plan deficiency was found in some nutrients (calories and B vitamins) 6 times out of 14 days. Figure 2 also shows the cost of the initial solution each day (end of phase 1) and the cost calculated by standard L.P. code. As it follows from the theory, the L.P. solution in fractions is always lower in cost than the corresponding optimum integer solution. The optimum integer solution could not be calculated by the exact method, but according to Figure 2 there is no doubt that optimum was reached at least 10 out of 14 times by the integer approximation. In any case the cost differential between the human and computer techniques is much higher than that of the exact and approximate solutions, and taking the running time differential into consideration no economically feasible substitute for the presented program could be found.

A FORTRAN program for the IBM 1410 system is now operational and available (at Tulane University Bio-Medical Computing System), and a similar program for

[1] The author is indebted to Dr. Andrew Vazsonyi for first recognizing and suggesting this option.



MENU-BY-COMPUTER [illegible] 11/24/63

BREAKFAST COST

ORANGE JUICE 3.99
SOFT COOKED EGG 3.57
MALT-O-MEAL 1.01

DINNER

VEGETABLE SOUP/CRACKERS 2.03
CHICKEN PIE CASSEROLE 7.81
CORNBREAD DRESSING 1.44
BUTTERED SUMMER SQUASH 1.79
STUFFED DEVILED EGG SALAD 4.28
CHERRY PIE 6.46

SUPPER

APPLE JUICE 2.95
POT ROAST OF YEARLING BEEF 11.08
DUCHESS POTATOES 3.29
BUTTERED LEAF SPINACH 2.11
DATE-COTTAGE CHEESE/MAYONNAISE 2.61
CHOCOLATE CAKE/WHITE ICING 2.66

TOTAL (CENTS) = 57.30

SATISFIED DIETARY REQUIREMENTS SURPLUS (PER CENT)

CAL.	2229.06		19.86
PROT.	43.85		81.57
CALC.	127.10		194.20
IRON	8.08		155.10
VIT.A	3.69		130.00
THIA.	1.06		79.70
RIBO.	.60		152.80
NIAC.	16.72		.89
VIT.C	63.71		95.00

TOLERANCES- .00 1.90
PORTION SIZE FACTOR- 1.00
NUTRIENT SURPLUS IS NOT CARRIED OVER
AVERAGE COST FOR 4. DAYS IS 52.79 CENTS

NOTE- DINNER AND SUPPER ITEMS ARE INTERCHANGEABLE

FIG. 3

the 1440 system is under consideration. The program can be easily adjusted to solve a large variety of menu problems with different sets of objectives.

Acknowledgments. The author acknowledges the cooperation of Tulane Bio-Medical Computing System and Touro Infirmary, New Orleans, on the project with special appreciation for the support of Dr. James W. Sweeney, Co-Principal Investigator.

RECEIVED DECEMBER, 1963; REVISED JANUARY, 1964.

REFERENCES

1. STIGLER, G. J. The cost of subsistence. *J. Farm Economics 25* (1945), 303–314.
2. SMITH, V. E. Linear programming models for the determination of palatable human diets. *J. Farm Economics 41* (1961), 272–283.
3. WATT, B. K., MERRILL, A. L., ORR, M. L., ET AL. Composition of foods—raw, processed and prepared. U. S. Department of Agriculture Handbook No. 8, 1950.
4. PERYAM, D. R., POLEMIS, B. W., KAMEN, J. M., EINDHOVEN, J. AND PILGRIM, F. J. Food preferences of men in the U. S. Armed Forces. Dept. of the Army, Quartermaster Research and Engineering Command, Quartermaster Food and Container Institute for the Armed Forces, Jan. 1960.
5. Recommended dietary allowance. Natl. Research Council, Food and Nutrition Bd., Natl. Research Council Publ. 589, Rev. 1958.
6. DANTZIG, GEORGE B. *Linear Programming and Extensions.* Princeton U. Press, Princeton, 1963.

Problems in Automatic Abstracting

H. P. EDMUNDSON
The Bunker-Ramo Corporation, Canoga Park, California

A variety of problems concerning the design and operation of an automatic abstracting system are discussed. The purpose is to present a general view of several major problem areas. No attempt is made to discuss details or to indicate preferences among alternative solutions.

1. Introduction

Since automatic abstracting is in its infancy it is felt that a paper covering the subject as a whole is apt to be more helpful than one which pleads for a single course of research. In many ways the present situation in automatic abstracting in the United States is analogous to the early days of automatic translation. For example, only two or three research teams, totaling 12 people, are presently working on the problem of automatic abstracting, while a dozen teams with a total staff of some 100 researchers are now studying automatic translation. Moreover, various United States government agencies have invested several millions of dollars in automatic translation since 1953, while only several hundred thousand dollars have been made available for research on automatic abstracting since 1958.

In this exposition the problems of automatic abstracting are grouped into the following major classes: (1) conceptual problems, (2) input problems, (3) computer problems, (4) output problems, and (5) evaluation problems.

Present systems of automatic abstracting are capable of producing nothing more than extracts of documents, i.e. a selection of certain sentences of a document. This is not to say, however, that future automatic abstracting systems cannot be conceived in which the computer generates its own sentences by means of a suitable generative grammar program. Theoretically there is no linguistic or mechanical reason why such a system could not be designed and operated. The total system would then consist of a program which operates on the original document so as to produce an extract which in turn is fed into the generative grammar portion that then generates its own sentences using certain of the original sentences as grist This system is depicted in Figure 1. Such a system, however, is apt to be costly both in time and money.

Since the creation of a suitable generative grammar

This research was supported in part by the United States Air Force with funds from Contract No. AF 30(602)-2223, monitored by the Rome Air Development Ctr., Griffiss Air Force Base, N.Y.