*A criterion is formulated which will permit project selection corresponding to management's statement of objectives and their relative importance.*

*An algorithm is developed to implement the criterion. The accompanying programming problem is examined and experience gained in executing the algorithm is described.*

*Application of the algorithm is demonstrated by detailing the solution of a problem.*

# Project evaluation and selection

**by B. Dimsdale and H. P. Flatt**

We consider the problem of evaluating several proposals and selecting one or more which in some sense "best" meet certain objectives. For example, the executive may wish to select from among a set of available projects a combination which will meet certain objectives relative to sales, sales staff, profitability, and capital investment over some particular period of time. Generally, multiple objectives are in conflict. For example, sales can probably be increased by increasing sales staff, but this affects profitability and capital investment. Thus, the problem is one of "trade-offs" which is not difficult if there are only a few choices. However, if there are many choices available and numerous objectives, simple "enumerative" approaches to the problem cease to be of utility, since the number of possible combinations becomes astronomical in size.

Our purpose is to describe an algorithm applicable to the problem. First, further clarification of the problem is given by means of a simple example and the kind of input data required is detailed, followed by an explicit definition of the problem. The next sections contain the basic mathematical results underlying the algorithm and algorithm itself. Next, an extension of the method to a broader class of problems is included. Some programming experience in executing the algorithm on the IBM 7090 and 1620 computers is described, and a problem and its solution are exhibited.

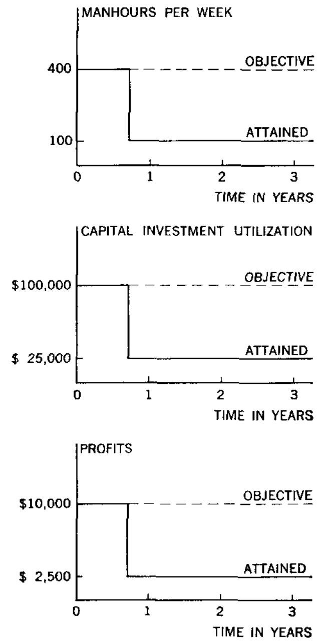
**illustration of the problem**

In order to clarify the problem, let us look at a simple illustration involving a small manufacturer. Suppose at present, products which require the total capacity of ten men and machine tools worth $100,000 are being manufactured. Annual profit is $10,000. However, one year from now there will be no longer any requirement for some of the products and, as shown in Figure 1, this will leave products which require 100 man-hours per week, utilize $25,000 of machine tools, and leave a profit of only $2,500.

Also suppose that the manufacturer is conservative, and wishes only to maintain stability for two additional years. Of course, profit increase would be welcome if the other objectives could be attained, but he will not consider investing more capital in order to increase profits.

Six new products have been proposed in addition to the "in house" products already committed. Each of the products has been described in terms of its requirement for man-hours, the amount of machine tools used, and its contribution to profit. This data is shown in Table 1 where "IH" denotes the "in house" products. The manufacturer knows that with six new products to consider, there are sixty-four possible combinations to examine. He tries to simplify the problem by looking for combinations which meet objectives, first by man-hours, then by machine utilization, and then by profit, arriving at the data given in Table 2.

Figure 1




TIME 0 IS PRESENT TIME

Table 1

| *Products* | *Man-hours required* | *Capital utilized* | *Profit* |
|---|---|---|---|
| IH | 100 | $25,000 | $2,500 |
| A | 200 | 75,000 | 6,000 |
| B | 100 | 50,000 | 3,000 |
| C | 300 | 75,000 | 2,500 |
| D | 100 | 25,000 | 5,000 |
| E | 200 | 25,000 | 1,500 |
| F | 100 | 10,000 | 2,500 |

Table 2

| *Products (IH +)* | *Man-hours required* | *Capital utilized* | *Profit* |
|---|---|---|---|
| A & B | 400 | $150,000 | $11,500 |
| A & D | 400 | 125,000 | 13,500 |
| A & F | 400 | 110,000 | 11,000 |
| B & D & F | 400 | 110,000 | 13,000 |
| C | 400 | 100,000 | 5,000 |
| E & F | 400 | 60,000 | 6,500 |
| D & E | 400 | 75,000 | 9,000 |
| B & E | 400 | 100,000 | 7,000 |
| A | 300 | 100,000 | 8,500 |
| C | 400 | 100,000 | 5,000 |
| B & D | 300 | 100,000 | 10,500 |
| B & E | 400 | 100,000 | 7,000 |
| D & F | 300 | 60,000 | 10,000 |
| A & E | 500 | 125,000 | 10,000 |
| C & D | 500 | 125,000 | 10,000 |

From the first part of Table 2, we see that of all the combinations which meet the manpower objective, only two (C and B & E) meet any of the other objectives. Even these combinations fall far short of meeting the profit objective. From the second part of the table, we observe by looking at those combinations which meet the capital investment requirement, that there are no new combinations meeting two or more objectives. Finally, from the latter part of the table, we note that any objectives which meet the profit objective fail to meet any of the other objectives.

Thus, the manufacturer discovers that there is no combination of products which will allow meeting all objectives exactly. The problem, which seemed simple, has become more complex, since it is necessary to look at every possible combination of products in order to determine that combination which in some sense "best" meets all objectives. In particular, since no combination meets all objectives, he must decide if all objectives are equally important or, if the objectives are not equally important, decide the relative importance of the different objectives.

This simple example illustrates several features of the more realistic and complex long-range planning problem:

- Objectives must be selected.
- Both old and new products may have to be manufactured whether or not it is optimal strategy to do so.
- The contribution of each proposed activity to each objective must be determined.
- It is in general unrealistic to expect to meet all objectives exactly, and hence it is necessary to give relative weights to the different objectives.
- It is necessary to formulate a criterion for judging which combination "best" meets all of the objectives.

**input data requirements**

Thus, there are three kinds of input data required: objectives, contributions of individual activities toward objectives, and weights which apply to the individual objectives. Furthermore all of this data in general varies with time. It is assumed in the following that the period of time under consideration remains fixed for any one calculation; that is, the goal is to meet all objectives over the same period of time. We will use the notation $t$ for time, and $T_0$, $T_1$ for the lower and upper bounds of $t$, respectively. Thus, $T_0 \leq t \leq T_1$ and $T_0$ and $T_1$ remain fixed throughout the problem.

Objectives are functions of $t$. We assume in general that there are $N$ objectives and write $C_j(t)$ to represent the $j$th objective, where $j = 1, 2, \cdots, N$.

Activities make contributions to fulfillment of objectives with the contributions also being functions of $t$. Thus, assuming that there are $M$ activities, we have $MN$ functions $\phi_{ij}(t)$ to describe the contribution of the $i$th activity to the fulfillment of the $j$th objective, so that $i = 1, 2, \cdots, M$, $j = 1, 2, \cdots, N$.

We note a basic assumption that if the requirement of activity

$i$ for resource $j$ is $\phi_{ij}(t)$ and of activity $k$ for resource $j$ is $\phi_{kj}(t)$, then the activities together require $\phi_{ij}(t) + \phi_{kj}(t)$ of resource $j$, this being true for all $i$, $j$, $k$. If this is not true for any pair, then a new activity must be added to the list consisting of the pair together and their resultant resource requirement. Naturally, this places a restraint on the freedom of choosing activities.

Finally, we require a set of functions $a_j(t)$, $j = 1, 2, \cdots, N$, as weighting functions. These functions serve three purposes. First, they provide a means for establishing a common measure among the objectives, which may be given variously in terms of dollars or man-hours or floor space, etc., and therefore need to be converted to a common unit of measure. Second, they may be used to establish relative importance of objectives. Third, they permit one to modify weight with time. For instance, if a ten year plan is being considered, it would probably be desirable to weight the first year heavily as compared to the tenth year. Moreover, many considerations might indicate different time weighting for different objectives. If common time weighting is desired, we can replace $a_j(t)$ by $A_j w(t)$ in which case $A_j$ accounts for the first two purposes and $w(t)$ gives the uniform time weighting.

As observed earlier, it is not reasonable to expect that all objectives can be precisely met. The most we can hope to do is to minimize some measure of total deviation from objectives through time. Should the measure chosen have the property that its minimum value becomes zero in case the objectives are precisely met, it would also constitute a measure of the incompatibility of objectives. For this reason among others, the measure that suggests itself is the weighted total of the squared differences over the time interval in question, a measure which is somewhat similar to the total (weighted) area between objectives and summed activity resource requirements. The technique of least squares, aside from being amenable to mathematical treatment, has the further advantage that large discrepancies tend to be of short duration and, in many cases, difficulties of this kind can be overcome by appropriate management action.

**criterion and problem definition**

Explicitly, the criterion we select is stated as follows: Let $x = (x_1, \cdots, x_M)$ be a vector for which $x_i = 0$ or $1$; 0 if activity $i$ is to be excluded, 1 if activity $i$ is included, and let

$$G(x) = \int_{T_0}^{T_1} \left( \sum_{j=1}^{N} a_j(t) \left[ C_j(t) - \sum_{i=1}^{M} x_i \phi_{ij}(t) \right]^2 \right) dt \qquad (1)$$

The problem is to find vectors $x$ which minimize $G(x)$. We note that there may well be more than one such vector.

The average discrepancy, $\bar{D}$, over time and all resources is then, for any vector $x$, given by

$$\bar{D}^2 = G(x) \Big/ \int_{T_0}^{T_1} \left( \sum_{j=1}^{M} a_j(t) \right) dt.$$

$\bar{D}$, for minimizing $x$, is then the measure of incompatibility of objectives.

Expanding the integrand of Equation (1), we find that we may write

$$G(x) = (x, Ax) - 2(\Gamma, x) + K, \tag{2}$$

where $(x, Ax)$ and $(\Gamma, x)$ denote scalar products (of $M$-vectors, $x$ and $Ax$, and $\Gamma$ and $x$, respectively) and $A$, $\Gamma$ and $K$ are defined as follows:

$$A = \sum_{j=1}^{N} A_j,$$

$$A_j = (a_{nm}^j) \qquad \text{where}$$

$$a_{nm}^j = \int_{T_0}^{T_1} a_j(t)\phi_{mj}(t)\phi_{nj}(t)\, dt, \tag{3}$$

and we observe parenthetically the symmetry $a_{nm}^j = a_{mn}^j$.

$$\Gamma = \sum_{j=1}^{N} \Gamma_j,$$

$$\Gamma_j = (c_m^j) \qquad \text{where}$$

$$c_m^j = \int_{T_0}^{T_1} a_j(t)\phi_{mj}(t)C_j(t)\, dt. \tag{4}$$

Finally,

$$K = \sum_{j=1}^{N} K_j \qquad \text{where}$$

$$K_j = \int_{T_0}^{T_1} a_j(t)[C_j(t)]^2\, dt, \tag{5}$$

and we note that all elements are non-negative.

Let $B$ be identical with $A$ except that the diagonal elements of $B$ are zero, let $b_\nu$ represent row (or column) $\nu$ of $B$ and let $\alpha$ be a vector whose $\nu$th component is $\alpha_\nu$, where

$$\alpha_\nu = a_{\nu\nu} - 2c_\nu + 2(b_\nu, x). \tag{6}$$

Also, let

$$x' = x + \delta_\nu \qquad \text{where} \tag{7}$$

$$\delta_\nu = (0, 0, \cdots, \epsilon_\nu, 0, \cdots, 0) \tag{8}$$

and

$$\begin{aligned} \epsilon_\nu &= +1, \quad \text{if} \quad x_\nu = 0 \\ &= -1, \quad \text{if} \quad x_\nu = 1. \end{aligned} \tag{9}$$

It then follows by substitution that

$$G(x') = G(x) + \epsilon_\nu\alpha_\nu \tag{10}$$

and also that

$$\alpha(x') = \alpha(x) + 2\epsilon_\nu b_\nu. \tag{11}$$

From Equation (10), we see that a necessary condition for $x$ to minimize $G(x)$ is that:

$$\text{when} \qquad x_\nu = 1(\epsilon_\nu = -1), \qquad \alpha_\nu \leq 0; \tag{12}$$

$$\text{when} \qquad x_\nu = 0(\epsilon_\nu = +1), \qquad \alpha_\nu \geq 0. \tag{13}$$

Thus, given any $x$, it is either true that each of the components of $\alpha$ satisfy the above conditions, or there exists one $\nu$ such that either

$$x_\nu = +1, \qquad \alpha_\nu > 0 \tag{14}$$

or

$$x_\nu = 0, \qquad \alpha_\nu < 0. \tag{15}$$

We now consider three sets of vectors $x$, to be called $S_1$, $S_2$ and $S_3$. $S_1$ contains all those vectors which minimize $G(x)$. $S_2$ contains all those vectors $x$ such that if $x_\nu = 1$, then $\alpha_\nu \leq 0$; and if $x_\nu = 0$ then $\alpha_\nu \geq 0$. $S_3$ contains all vectors $x$ such that if $x_\nu = 1$, then $\alpha_\nu \leq 0$. Now $S_1 \subset S_2 \subset S_3$, for if there were a vector $x$ in $S_1$ not in $S_2$, then $G(x)$ could be reduced and would not be minimal and, clearly, $S_2 \subset S_3$ by definition. Thus, in scanning the $x$'s for a minimum of $G(x)$ it is sufficient to scan members of the set $S_3$.

Next, consider any $x \neq 0$ in $S_3$, and suppose $x'$ is obtained from $x$ by changing a component of $x$ which is 1 to 0, say $x_\nu$. By definition of $S_3$ it follows that $\alpha_\nu \leq 0$ and that $\epsilon_\nu = -1$. Hence $G(x') \geq G(x)$. Since every component of $b_\nu$ is non-negative it is also true that $\alpha_i(x') \leq \alpha_i(x)$ for every $i$; that is, $x'$ is also in $S_3$. By repeating the process, a sequence

$$x, x', x'', \cdots, 0$$

of vectors in $S_3$ is obtained for which

$$G(x) \leq G(x') \leq G(x'') \leq \cdots \leq g(0).$$

By reversing the sequence it will be observed that every vector in $S_3$ is a member of a sequence starting with $x = 0$, each vector is followed by a vector obtained by changing a *zero* component into a *one* in such a way that $G$ is not increased, and every member of the sequence is in $S_3$. These changes are made only at components $x_i$ for which $\alpha_i \leq 0$. Thus every member of $S_3$ is produced at least once by generating every possible sequence of this kind. Essentially, the algorithm consists of generating sequences of this type. It will be noted particularly that the corresponding sequence of $\alpha$ vectors are monotone non-decreasing, component by component.

Now consider $x = 0$ and its corresponding $\alpha$, where $\alpha_i = a_{ii} - 2c_i$. If attention is fixed on some specific component $\alpha_\nu$ it may be that $\alpha_\nu > 0$, in which case $x_\nu$ must be 0 in all generated sequences. Or it may be that $\alpha_\nu$ is so large and negative that $x_\nu$ must be 1 in all generated sequences. Or, finally, it may be that

$\alpha_\nu < 0$ but not large enough negatively to exclude $x_\nu = 0$ as a possibility in the generated sequences. In this case two new vectors are to be considered, one of which is the null vector, the other the null vector with $x_\nu = 0$ replaced by $x_\nu = 1$. In each instance the $\nu$th component is considered as subject to no further change.

The set (of one or two vectors) generated above is processed again in similar manner, that is, by applying the same process to a component of each vector which has not yet been fixed. Continuing in this fashion, a set of vectors will ultimately be reached each of which has all its components fixed, and this set is the set $S_3$. It is to be noted that whenever a vector is generated by replacing $x_\nu = 0$ by an $x'_\nu = 1$ there is the prospect for any $x_i = 1$ (for which $\alpha_i \leq 0$) that $\alpha'_i > 0$. In this case the vector may be discarded.

algorithm

We note that two processes are required, one for calculation of the numbers $\alpha^i_{mn}$, $c^i_m$, $K^i$ in Equations (3), (4), and (5) and the other for the remaining processes. Of course, the first of these depends on the form in which the input data is given and hence cannot be described here, except to observe that a general program can be written for accumulating the integrals, which can be computed by appropriate subroutines.

In order to describe the computing algorithm for the second process in detail, two sets of indices, $E$ and $\bar{E}$ are assigned, associated with a vector $x$. The set $E$ contains those indices corresponding to components of $x$ whose values have been fixed; the set $\bar{E}$ contains all the remaining indices. Evidently, if $\nu \in \bar{E}$ then $x_\nu = 0$. Let $\beta$ be a vector whose $\nu$th component is $\beta_\nu$, where

$$\beta_\nu = 2 \sum_{i \in \bar{E}} b_{i\nu}, \tag{16}$$

$b_{i\nu}$ being the $i$th component of $b_\nu$. It will be seen that if $\alpha_\nu + \beta_\nu < 0$ then $x_\nu = 1$ in all vectors generated from $x$. The principal part of the algorithm can now be stated:

*Step* 1. Initially set

$$\alpha^0_\nu = a_{\nu\nu} - 2c_\nu, \qquad \beta^0_\nu = 2 \sum_{i=1}^{n} b_{i\nu}, \qquad x^0_\nu = 0$$

for $\nu = 1, \cdots, n$, $G(x^0) = K$, and $E = 0$.

*Step* 2. For any $x$ such that $\bar{E} \neq 0$ select some $\nu \in \bar{E}$ and proceed as follows:

- If $\alpha_\nu \geq -\beta_\nu$ generate $x' = x$, $\alpha' = \alpha$, $\beta' = \beta - 2b_\nu$, $G(x') = G(x)$, move $\nu$ from $\bar{E}$ to $E$.
- If $\alpha_\nu \leq 0$ generate $x'$ by changing $x_\nu = 0$ into $x'_\nu = 1$, $\alpha' = \alpha + 2b_\nu$. If $x' \in S_3$, that is, $x_i = 1$ implies $\alpha'_i \leq 0$ for all $i$, generate also $\beta' = \beta - 2b_\nu$, $G(x') = G(x) + \alpha_\nu$, move $\nu$ from $\bar{E}$ to $E$. If $x' \notin S_3$, discard $x'$.

*Step* 3. Iterate until no generated vector has $\bar{E} \neq 0$.

While it is clear that the above algorithm generates $S_3$ it is also true that interest is centered in $S_1$. For this purpose a further restriction can be made. For if, during the course of the iterations, a particular vector $x$ has been generated, it will be seen, for any vector $x''$ obtained from $x$ by the above process, that

$$G(x'') \geq G(x) + \sum{}' \alpha_\nu, \tag{17}$$

where the summation $\sum'$ is extended over those indices in the set $\bar{E}$ (associated with $x$) for which $\alpha_\nu < 0$. If $\bar{G}$ represents the smallest $G(x)$ obtained at any given point of the process, then any vector $x$ such that

$$G(x) + \sum{}' \alpha_\nu > \bar{G} \tag{18}$$

cannot possibly lead to a member of $S_1$ and thus may be dropped from consideration.

Finally, the problem of choosing $\nu \in \bar{E}$ exists. Since $G(x') = G(x) + \alpha_\nu$ it seems reasonable to choose that $\nu \in \bar{E}$ which minimizes $\alpha_\nu$. For if min $\alpha_\nu$ is negative this choice makes the maximum reduction in $G$, and if min $\alpha_\nu \geq 0$ no reduction of $G$ is possible.

One further function can be incorporated in the process. Specifically, it may be desired to force some components of $x$ to be 1 regardless of other considerations. That is, certain of the projects are to be forced into the final solution. There are two ways in which this can be done. One is to modify the functions $C_i(t)$. Another way is to make use of the fact that up to now only the pairs (0, 0), (1, 0), (1, 1) have been used for $(E_i, x_i)$. If the pair (0, 1) is used in this last case the algorithm is easily modified to accomodate this situation.

illustration of the algorithm

For illustrative purposes consider the problem formulated by the following values for $A$, $\Gamma$ and $K$:

$$A = \begin{pmatrix} 15 & 0 & 3 & 0 & 0 & 2 & 0 & 0 \\ 0 & 4 & 3 & 2 & 1 & 2 & 1 & 0 \\ 3 & 3 & 8 & 3 & 2 & 1 & 1 & 0 \\ 0 & 2 & 3 & 9 & 4 & 2 & 0 & 3 \\ 0 & 1 & 2 & 4 & 7 & 2 & 2 & 2 \\ 2 & 2 & 1 & 2 & 2 & 7 & 4 & 1 \\ 0 & 1 & 1 & 0 & 2 & 4 & 11 & 2 \\ 0 & 0 & 0 & 3 & 2 & 1 & 2 & 5 \end{pmatrix},$$

$\Gamma = (7, 12, 10, 13, 11, 23, 13, 11)$, and

$K = 60$.

Then Table 3 follows where $E_i = 0$ implies that $i \in E$, $E_i = 1$

Table 3

| | | | | | | | | |
|---|---|---|---|---|---|---|---|---|
| $E$ | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| $x$ | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| $\alpha$ | 1 | −20 | −12 | −17 | −15 | −17 | −15 | −17 |
| $\beta$ | 10 | 18 | 26 | 28 | 26 | 28 | 20 | 16 |
| $G$ | 60 | | | | | | | |

Table 4

| | | | | | | | | |
|---|---|---|---|---|---|---|---|---|
| $E$ | 1 | 1 | 0 | 0 | 0 | 0 | 0 | 1 |
| $x$ | 0 | 1 | 0 | 0 | 0 | 0 | 0 | 1 |
| $\alpha$ | 1 | −12 | −6 | −7 | −9 | −11 | −9 | −17 |
| $\beta$ | 10 | 18 | 14 | 18 | 20 | 18 | 14 | 16 |
| $G$ | 23 | | | | | | | |

Table 5

| | | | | | | | | |
|---|---|---|---|---|---|---|---|---|
| $E$ | 1 | 1 | 0 | 0 | 0 | 1 | 0 | 1 |
| $x$ | 0 | 1 | 0 | 0 | 0 | 0 | 0 | 1 |
| $\alpha$ | 1 | −12 | −6 | −7 | −9 | −11 | −9 | −17 |
| $\beta$ | 6 | 14 | 12 | 14 | 16 | 18 | 6 | 14 |
| $G$ | 23 | | | | | | | |

Table 6

| | | | | | | | | |
|---|---|---|---|---|---|---|---|---|
| $E$ | 1 | 1 | 0 | 0 | 0 | 1 | 0 | 1 |
| $x$ | 0 | 1 | 0 | 0 | 0 | 1 | 0 | 1 |
| $\alpha$ | 5 | −8 | −4 | −3 | −5 | −11 | −1 | −15 |
| $\beta$ | 6 | 14 | 12 | 14 | 16 | 18 | 6 | 14 |
| $G$ | 12 | | | | | | | |

implies $i \in \bar{E}$. It is apparent that components 1, 2, 8 can be assigned immediately. This leads to Table 4.

In Table 4 the unassigned component with most negative $\alpha$ is the sixth. This leads to the two cases displayed in Table 5 and 6, respectively. Continuing according to the rules laid down, the following results are rapidly obtained:

min $G(x) = 7$, and

$x_{\min}$ values are:

(0 1 1 0 1 1 0 1),

(0 1 0 0 1 1 0 1) and

(0 1 0 1 0 0 1 1).

It will be noted that $\beta$ finally is null, which serves as a check on the process.

constraints

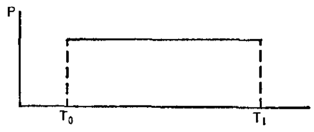
We have been looking at one type of problem in which we are forced to select any combination of activities which "best" meets our objectives. Other problems exist, however, in which we either do not have or do not desire this much freedom of choice.

For example, if we attempt to schedule a group of machines for preventive maintenance, we want to find an optimum schedule, but subject to the restriction that each machine receive preventive maintenance. Or, we may have decided to carry out six programs, and we are now concerned with the funding levels for each program. We wish to optimize this schedule, but subject to the restriction that we choose only one funding level per program.

As a final example, in considering advertising programs, it may be desired to choose some combination subject to the restriction that the combination include one or more choices among the media of television, radio, newspapers, or magazines.

Such problems may be handled as a simple extension to the basic method outlined above. For each subset from which one or more activities are required, an "artificial" objective is stated. This objective may be some non-negative constant $P$ over the entire time-interval of interest, with a weight function $\alpha_i(t) = 1$ (Figure 2). If any activity *does not* belong to the subset under consideration, then its contribution to the fulfillment of the objective is zero; if $M_0$ activities are to be selected from the subset, then the contribution of each member of the subset is $P/M_0$.

Figure 2




We begin by dealing with only one subset of activities which may, without loss of generality, be numbered 1, 2, $\cdots$, $M_1$, and suppose that our interest is in selecting exactly $M_0$ of these. That is, the constraint,

$$\sum_{i=1}^{M_1} x_i = M_0, \tag{19}$$

is established under which $G(x)$ is to be minimized. Let $\bar{x}$ represent a minimizing vector for $G(x)$ under this constraint, and consider

$$G_1(x) = G(x) + \int_{T_0}^{T_1} \left[P - \sum_{i=1}^{M_1} Px_i/M_0\right]^2 dx = G(x) + \lambda g(x), \tag{20}$$

where

$$\lambda = \frac{P^2(T_1 - T_0)}{M_0^2}, \qquad g(x) = \left(M_0 - \sum_{i=1}^{M_1} x_i\right)^2. \tag{21}$$

For any $\lambda$ we can find $x'$ which minimizes $G_1(x)$ (noting that $x'$ is a function of $\lambda$), by use of the preceding algorithm. Now

$$G(x') \leq G(\bar{x}), \tag{22}$$

for if the converse were true, that is, if $G(\bar{x}) < G(x')$, then $G(\bar{x}) + \lambda g(\bar{x}) < G(x') + \lambda g(x')$ since $g(\bar{x}) = 0$. That is to say, we should have $G_1(\bar{x}) < G_1(x')$ contrary to our hypothesis that $x'$ minimizes $G_1(x)$.

Now we show that there exists a value of $\lambda$ such that $G(\bar{x}) = G(x')$. We have

$$G(x') + \lambda g(x') = G_1(x') \leq G_1(\bar{x}) = G(\bar{x})$$

and either $g(x') = 0$ or $g(x') \geq 1$. If, for all $\lambda$, $g(x') \neq 0$ then $G(x') + \lambda g(x')$ can be made indefinitely large by choosing $\lambda$ sufficiently large. On the other hand, $G(\bar{x})$ does not change with $\lambda$. Hence the assumption that $g(x') \neq 0$ for all $\lambda$ leads to a contradiction. That is to say, for some $\lambda$ sufficiently large, $g(x') = 0$ and hence $x'$ minimizes $G(x)$ under the constraint $g(x) = 0$. Since this is a property of $\bar{x}$ it follows that $G(x') = G(\bar{x})$, that is, the vectors $x'$ solve the constrained problem.

There are two further points of interest. One, the above derivation is easily extended to more than one subset. Two, if activities are forced into the solution in such a manner that the constraint is violated, the derivation obviously collapses, since $\bar{x}$ does not exist. In practice there should be no difficulty on this latter point.

### application of the algorithm

To provide additional insight as to nature of the results that may be obtained with the algorithm we detail the solution of a more elaborate example. The problem to be considered involves three objectives: sales, profits, and engineering man-hours. Twelve time periods are to be considered, and the activity data is shown in Tables 7, 8, and 9.

Two of these projects, Numbers 5 and 8, are presently in house and required in the final solutions.

The objectives are as follows: sales, presently ($t = 0$) set at \$10,000,000 to increase uniformly to \$22,000,000 by the end of the twelfth period; profits, presently set at \$1,000,000 to increase to \$1,600,000; engineering man-hours to remain fixed at 100,000.

In Figures 3, 4, 5 the results of using four different sets of weights are shown. The results are labeled A, B, C, D and correspond to the weights given in Table 10 where $w_1(t) = 1$, and $w_2(0) = 1$, $w_2(12) = .40$ and is linear between these values. The mix of projects selected in each case is given in Table 11.

The effect on sales of increasing the weight associated with sales (solution B) is quite noticeable. Solution C is even better at the beginning of the range, but performs poorly toward the end. On the other hand solution $C$ gives the best answer for profits, which is to be expected. The results in man-hours seem relatively independent of weighting. Here the rapid "drop off" is the most noticeable feature. A glance at input data quickly reveals the reason for this, and may well indicate a need for quickly broadening the set of projects available.

It may be of interest to investigate the case in which the policy is to accept as much sales and profits as possible, consistent with the above objective in engineering man-hours. For this problem the variation in solution would be expected with the variation in weight assigned to the manpower objective. The solutions of interest here are given in Table 12. The objective now will be \$50,000,000 in sales (constant), \$5,000,000 in profits (constant).

**Table 7** Sales (millions of dollars)

| *Project Number* | *Time* 0 | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | 10 | 11 | 12 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| 1 | | | | 0.8 | 6.8 | 7.0 | 9.2 | 7.3 | 4.5 | 2.1 | 1.0 | | |
| 2 | | 0.5 | 4.8 | 6.2 | 3.5 | 2.1 | 0.5 | | | | | | |
| 3 | 5.0 | 5.6 | 2.1 | 0.9 | | | | | | | | | |
| 4 | | | 0.5 | 3.5 | 4.0 | 8.5 | 8.4 | 6.3 | 5.1 | 0.6 | | | |
| 5 | 3.4 | 4.9 | 3.5 | 1.4 | | | | | | | | | |
| 6 | | | 2.3 | 8.6 | 4.4 | 6.9 | 2.3 | 0.9 | | | | | |
| 7 | | | | | | 1.3 | 3.2 | 5.5 | 7.6 | 3.8 | 6.0 | 4.5 | 2.1 |
| 8 | 4.2 | 3.2 | 1.1 | | | | | | | | | | |
| 9 | | | | | | | | | | | 0.3 | 6.6 | 12.4 |
| 10 | | | | | | | 2.1 | 6.8 | 7.5 | 8.0 | 7.4 | 5.0 | 3.2 |
| 11 | 1.2 | 2.0 | 3.1 | 1.6 | 0.3 | | | | | | | | |
| 12 | | | | | | | | 1.1 | 3.7 | 5.2 | 7.1 | 7.1 | 5.0 |
| 13 | | | | | | | | 1.3 | 2.4 | 4.7 | 5.5 | 2.0 | 1.0 |
| 14 | 1.6 | 2.2 | 1.4 | 0.5 | 0.3 | | | | | | | | |
| 15 | | | | | | | | | 0.6 | 5.4 | 3.8 | 4.9 | 2.7 |

**Table 8** Profits (millions of dollars)

| *Project Number* | *Time* 0 | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | 10 | 11 | 12 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| 1 | | | | | .08 | .51 | .72 | .75 | .54 | .31 | .04 | | |
| 2 | | | .01 | .32 | .48 | .23 | .17 | .06 | .03 | | | | |
| 3 | .28 | .47 | .66 | .49 | .43 | .13 | | | | | | | |
| 4 | | | | .02 | .08 | .11 | .56 | .54 | .50 | .14 | | | |
| 5 | .21 | .24 | .27 | .24 | .21 | | | | | | | | |
| 6 | | | | .03 | .30 | .44 | .39 | .31 | .23 | .05 | | | |
| 7 | | | | | | .06 | .15 | .38 | .52 | .48 | .37 | .16 | .03 |
| 8 | .12 | .16 | .33 | .24 | .05 | | | | | | | | |
| 9 | | | | | | | | | | | .01 | .20 | .60 |
| 10 | | | | | | | | .10 | .31 | .50 | .54 | .48 | .23 |
| 11 | .30 | .35 | .42 | .62 | .42 | .14 | .02 | | | | | | |
| 12 | | | | | | | | .04 | .12 | .34 | .40 | .20 | .14 |
| 13 | | | | | | | | | .10 | .39 | .58 | .73 | .60 |
| 14 | .41 | .32 | .11 | .02 | | | | | | | | | |
| 15 | | | | | | | | | | .12 | .44 | .60 | .51 |

**Table 9** Engineering man-hours (thousands)

| *Project Number* | *Time* 0 | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | 10 | 11 | 12 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| 1 | | | 5 | 37 | 42 | 52 | 41 | 17 | 12 | 6 | | | |
| 2 | 5 | 22 | 38 | 54 | 21 | 9 | | | | | | | |
| 3 | 32 | 38 | 35 | | | | | | | | | | |
| 4 | | | | 10 | 26 | 35 | 40 | 29 | 8 | | | | |
| 5 | 45 | 29 | 14 | | | | | | | | | | |
| 6 | | 3 | 18 | 20 | 33 | 32 | 15 | | | | | | |
| 7 | | | | | 14 | 23 | 33 | 32 | 25 | 14 | 5 | | |
| 8 | 32 | 21 | 7 | | | | | | | | | | |
| 9 | | | | | | | | | 11 | 17 | 29 | 21 | 5 |
| 10 | | | | | 2 | 8 | 16 | 41 | 19 | 14 | 8 | 4 | |
| 11 | 23 | 28 | 30 | 30 | 14 | | | | | | | | |
| 12 | | | | | | | 14 | 39 | 38 | 31 | 22 | 13 | 8 |
| 13 | | | | | | | | 5 | 13 | 23 | 18 | 17 | 16 |
| 14 | 8 | 16 | 20 | 3 | | | | | | | | | |
| 15 | | | | | | | | 2 | 12 | 14 | 16 | 15 | 7 |

Figure 3 Sales

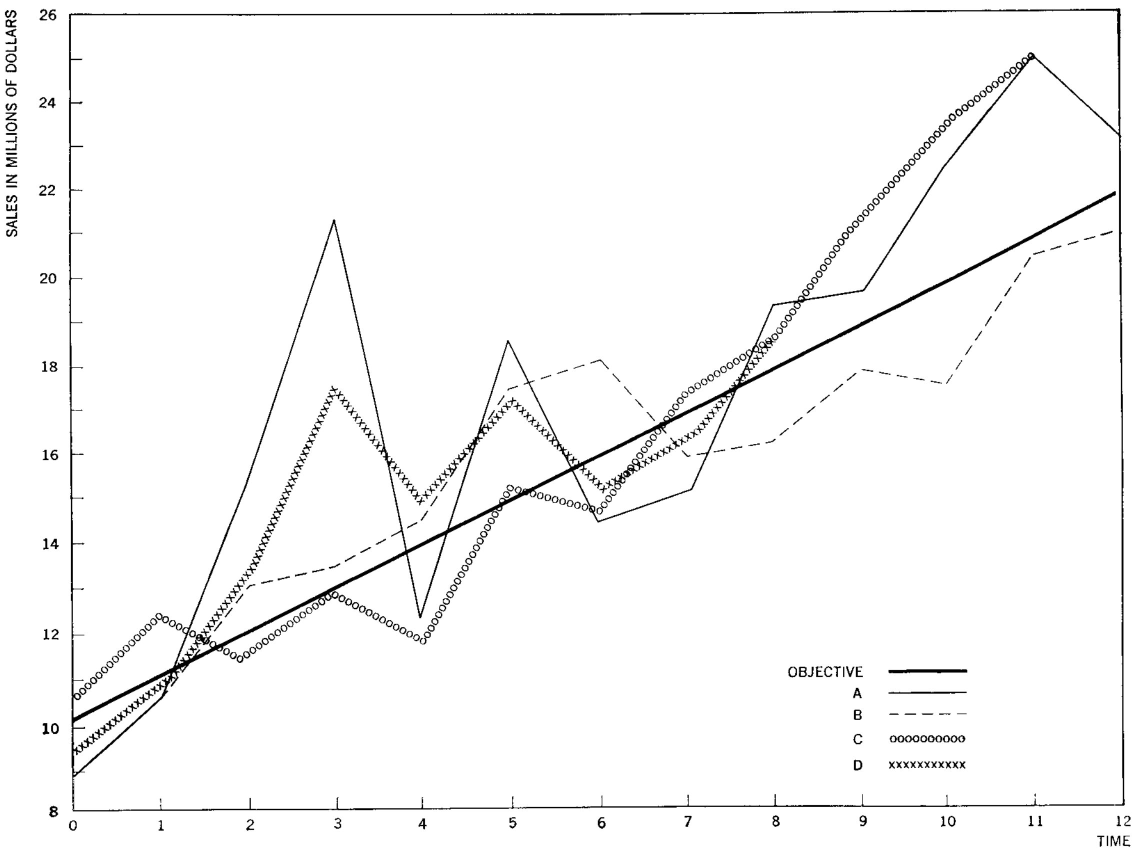



Figure 4 Man-hours

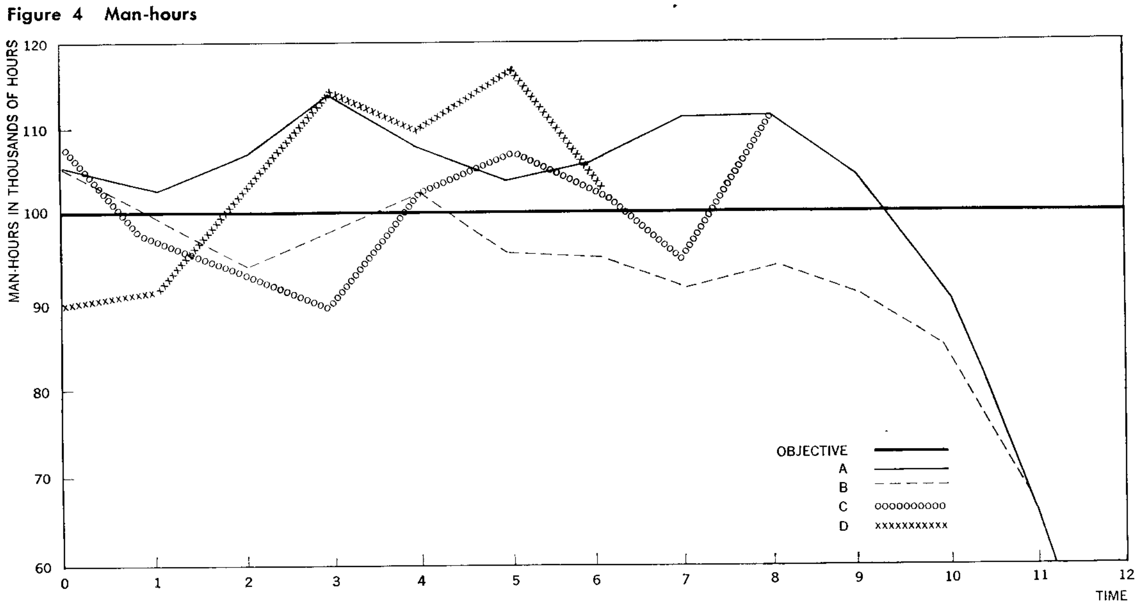


Figure 5 Profits

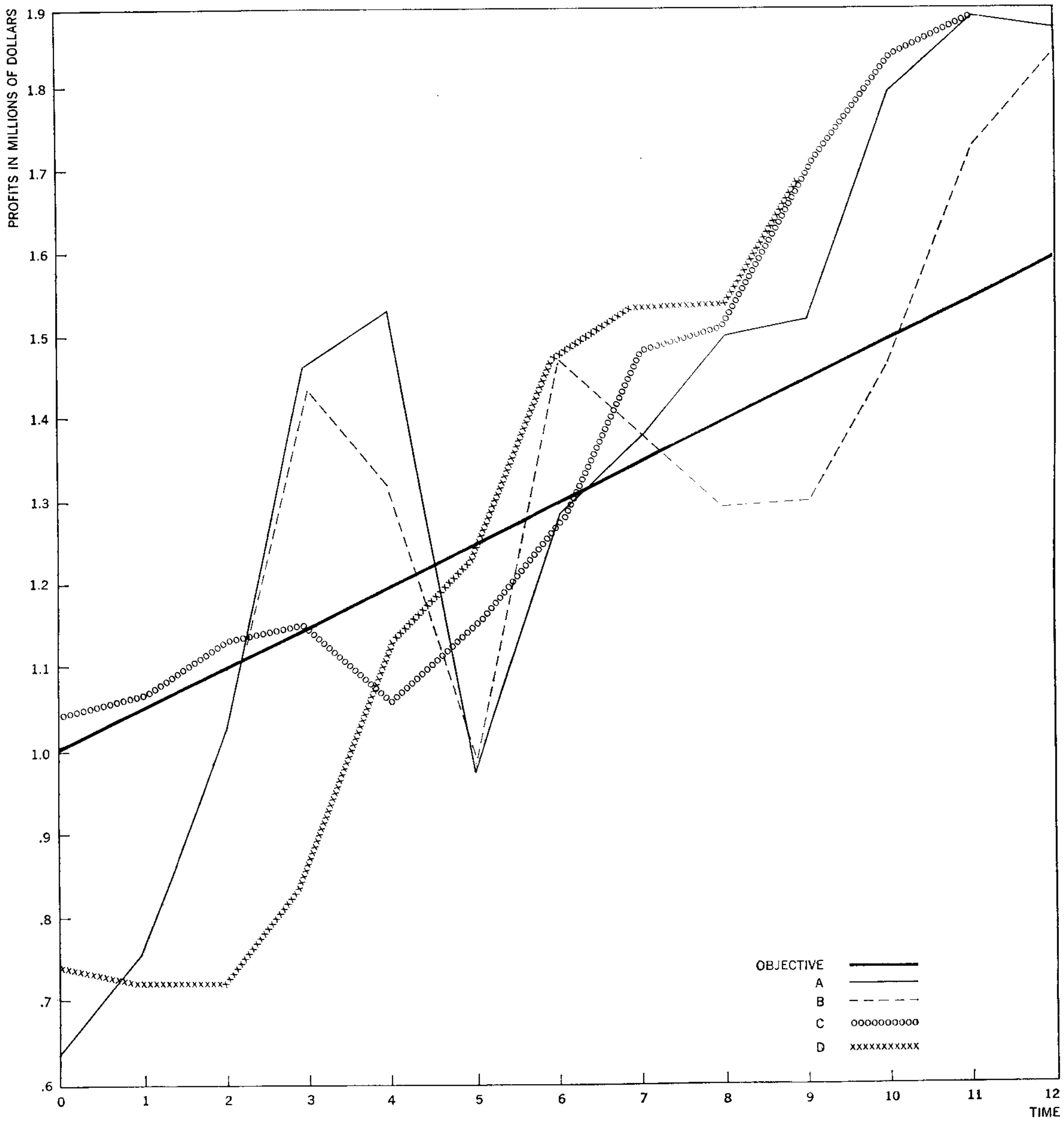



Table 10

| Solution | Weights $A_1$ | $A_2$ | $A_3$ | $w(t)$ |
|---|---|---|---|---|
| A | 1 | 10 | .1 | $w_1(t)$ |
| B | 10 | 10 | .1 | $w_1(t)$ |
| C | 1 | 100 | .1 | $w_1(t)$ |
| D | 1 | 10 | .1 | $w_2(t)$ |

Table 11

| Solution | Projects selected |
|---|---|
| A | 2, 4, 5, 6, 7, 8, 9, 11, 12, 13, 15 |
| B | 1, 2, 4, 5, 8, 9, 11, 12, 13, 15 |
| C | 1, 5, 6, 7, 8, 9, 11, 12, 13, 14, 15 |
| D | 1, 2, 5, 6, 7, 8, 9, 12, 13, 14, 15 |

**Figure 6 Sales**

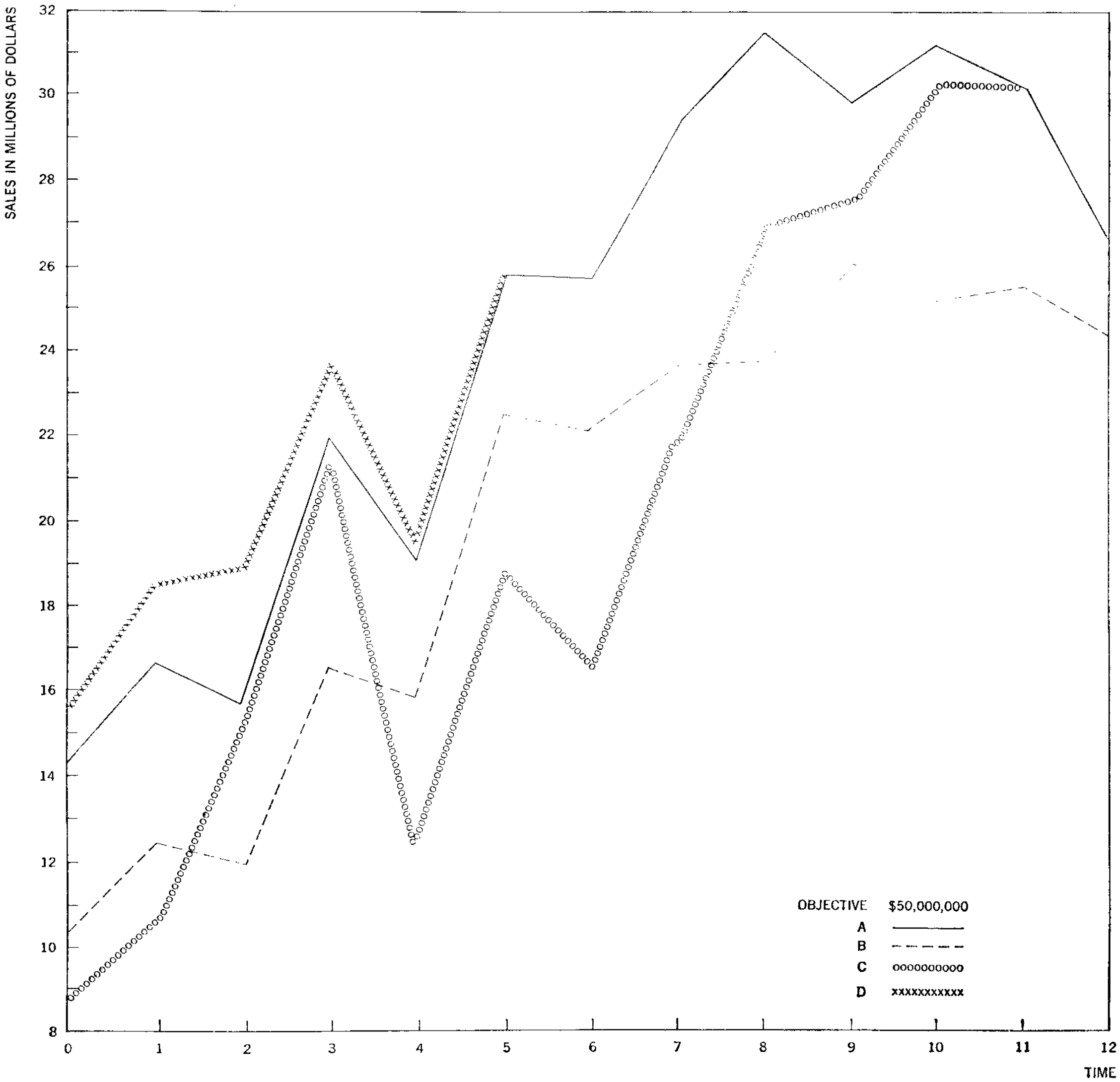



**Table 12**

| Solution | Weights $A_1$ | $A_2$ | $A_3$ | $w(t)$ |
|---|---|---|---|---|
| A | 1 | 10 | 0.1 | $w_1(t)$ |
| B | 1 | 10 | 1.0 | $w_1(t)$ |
| C | 1 | 100 | 1.0 | $w_2(t)$ |
| D | 10 | 10 | 0.1 | $w_1(t)$ |

**Table 13**

| *Solution* | *Projects selected* |
|---|---|
| A | 1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 12, 13, 14, 15 |
| B | 1, 4, 5, 6, 9, 10, 11, 12, 13, 14, 15 |
| C | 2, 4, 5, 6, 7, 8, 9, 10, 11, 12, 13, 15 |
| D | all fifteen |

Figure 7 Profits

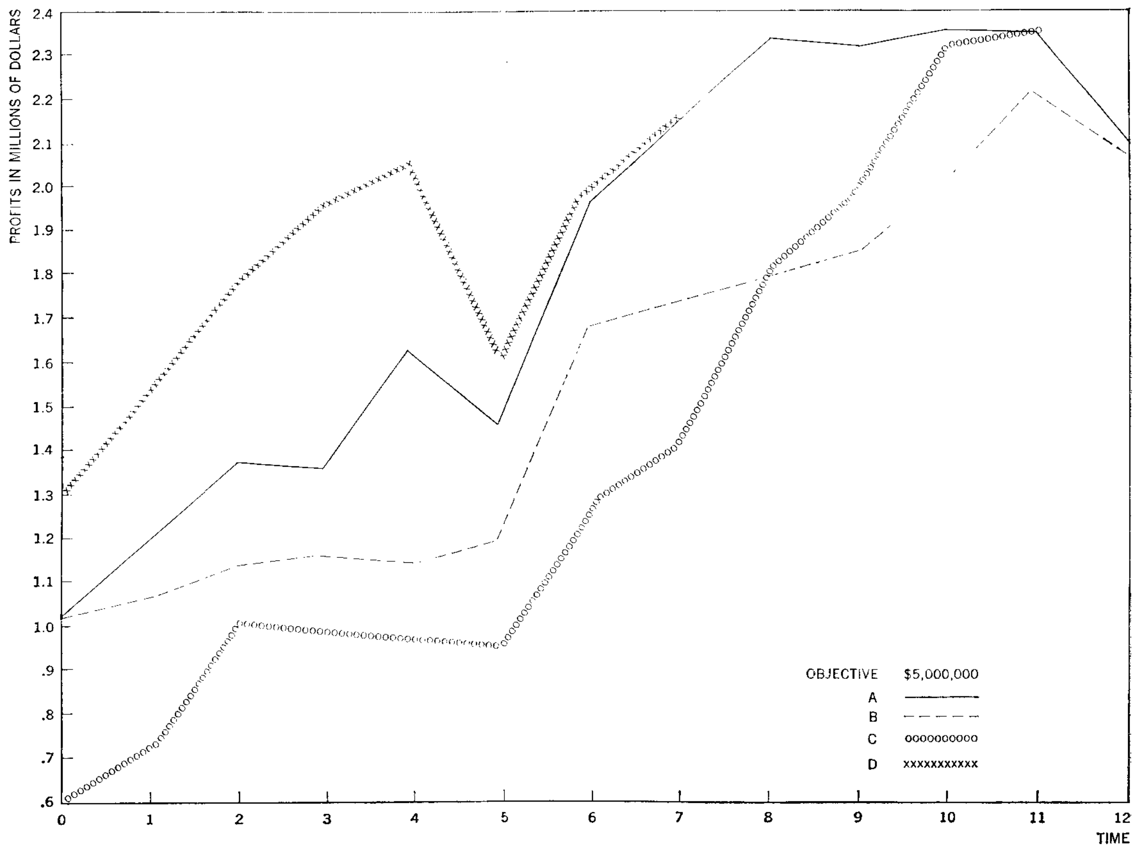
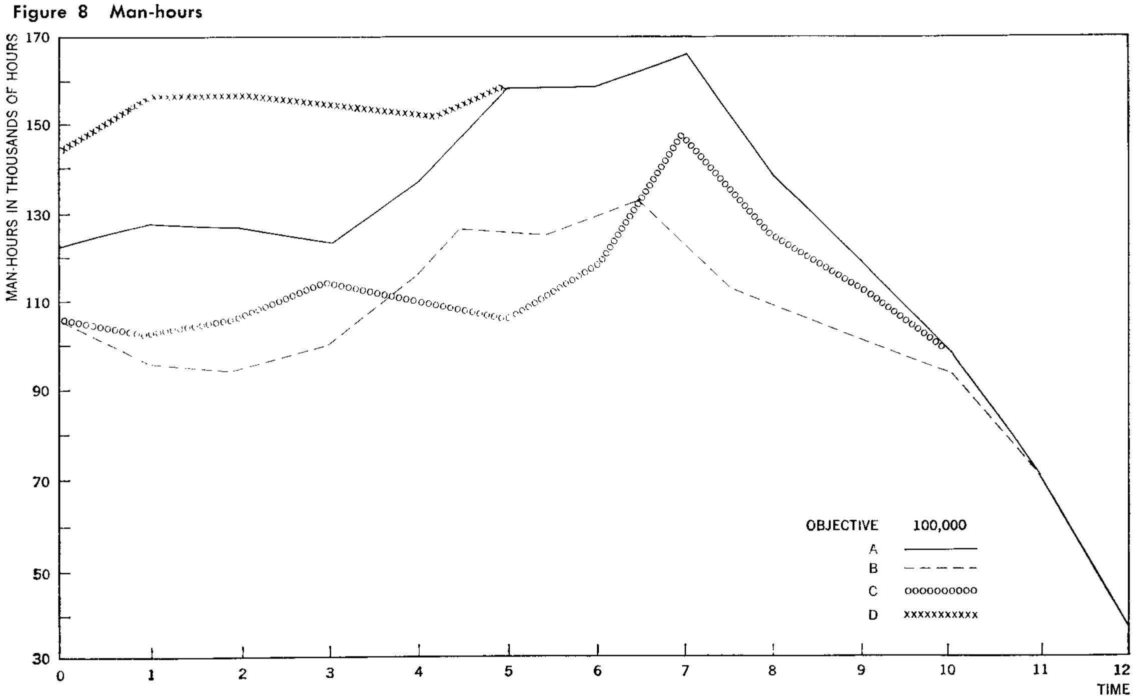



Figure 8 Man-hours



These numbers have been chosen to exceed the totals available at any given point in time. The product mixes are given in Table 13 and the results are shown in Figures 6, 7, 8. Solution D, which selects all fifteen projects, is the natural result of emphasizing sales (or profits, or both) in these circumstances. Finally, it is interesting to note that the set of weights 1, 10, 10, $w_1(t)$ gives a solution identical with solution $A$ of the preceding set. The general result seems to be that, the more weight attached to matching manpower objectives, the less sales and profits.

A 7090 was used for the above computations and running time was about fifteen seconds per case. The number of vectors examined varied from 12 to 41, the maximum number in storage varied from 1 to 6.

**computer programs**

Two important and related questions connected with the use of the above algorithm require an answer. The first question has to do with running time, the second with storage capacity required. The first question has two parts essentially, the length of time required to form the input needed for the above algorithm from original data, and the total number of vectors examined. The second question requires a knowledge of the maximum number of vectors in storage at any given moment.

Integration and summation processes can be estimated for running time, once the decision has been made as to the form of the input functions. The number of vectors examined and number in storage at any moment, can only be computed in general.

In passing, it may be noted that for an $M$ activity problem there are a maximum of $2^M$ combinations and that it is not difficult to construct an example in which every one of these is optimum in the above sense. Specifically, if $A$ is diagonal, that is, $a_{ij} = 0$ when $i \neq j$ and if moreover $a_{ii} = 2c_i$, then $(x, Ax) = 2(\Gamma, x)$ for every admissible $x$. That is to say $G(x) = K$ for all admissible $x$. To create such an $A$ and $\Gamma$, let $T_0 = 0$, $T_1 = M$, $\phi_{mj}(t) = 1$ for $m - 1 \leq t \leq m$, $\phi_{mj}(t) = 0$ elsewhere. Also let $C_j(t) = 1/2$, $a_j(t) = 1$ everywhere. Then $a_{ii} = N$, $c_i = N/2$. Thus, no upper bound less than $2^M$ for either number of vectors exists. It appears therefore that only experience can provide an indication as to the bounds that actually occur in practice.

For this reason, experimental programs have been written for the 7090 first, and later the 1620. In these programs all input functions are assumed to be continuous and piecewise linear. Furthermore, it is assumed that changes of slope occur only at integral values of time, $t = 0, 1, \cdots, T$, so that in fact the functions can be completely described by tables. The integrations in the programs are precise, under these assumptions. As for weights, the form $a_j(t) = A_j w(t)$ is used. The mechanism for forcing activities into the final solution is incorporated.

Another feature built into the program, which has not been mentioned before, is a device for locating solutions "close" to the optimum. This is accomplished by finding min $G(x)$ and then, on a second pass, retaining those vectors $x$ in $S_3$ such that

$G(x) < \min G + k[G(0) - \min G]$, that is, vectors $x$ such that $G(x)$ is within a certain percentage (described by $k$) of the minimum.

Although several small problems were run, the most significant single result to date has to do with a problem having 32 activities and 3 objectives over 20 time periods, with 3 activities being forced into the solution. In this problem, 21 different cases corresponding to differing choices of weights were run. Total running time on the 7090 was 20 minutes, the integrations being performed for each case, although a more sophisticated program would have made this unnecessary in this particular instance. The other operating statistics, which were printed out as part of the program, were as follows. The total number of combinations tested ranged from 100 to 1300. The maximum number of vectors in storage at any one time was 7.

It is not possible to make significant statements about so few statistics. However, the limited experience to date indicates that the results in the above case were fairly typical.