

605 043  
605043

COPY	1	OF	202
HARD COPY		\$.	1.00
MICROFICHE		\$.	0.50

ON A DYNAMIC PROGRAMMING APPROACH  
TO THE CATERER PROBLEM—I

By

RICHARD BELLMAN

P-866

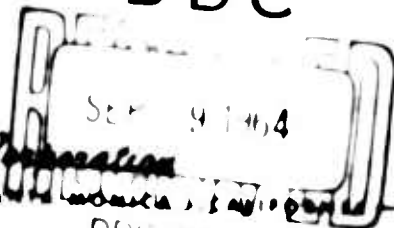
May 17, 1956

Approved for OTS release

DDC

The RAND Corporation

1700 MAIN ST



DDCIRA E

SUMMARY

In this paper, it is shown that the "caterer" problem, a problem in mathematical economics and logistics which has been discussed by Jacobs, Gaddum, Hoffman and Sokolowsky, and Prager, can be reduced to the problem of determining the maximum of <sup>a</sup>the linear form  $L_n = \sum_{i=1}^n v_i$ , subject to a series of constraints of the form  $v_1 \leq b_1$ ,  $v_1 + v_2 \leq b_2$ ,  $v_1 + v_2 + v_3 \leq b_3$ , ...,  $v_1 + v_2 + \dots + v_k \leq b_k$ ,  $v_2 + v_3 + \dots + v_{k+1} \leq b_{k+1}$ , ...,  $v_{n-k+1} + \dots + v_n \leq b_n$ ,  $0 \leq v_i \leq r_i$ ,  $i = 1, 2, \dots, n$ , under an assumption concerning the non-accumulation of dirty laundry.

This maximization problem is solved explicitly, using the functional equation technique of dynamic programming.

ON A DYNAMIC PROGRAMMING APPROACH  
TO THE CATERER PROBLEM—I

§1. Introduction

The purpose of this paper is to show how the functional equation method of dynamic programming may be used to obtain an explicit solution of the problem of determining the maximum of the linear form

$$(1) \quad L_n(v) = v_1 + v_2 + \dots + v_n,$$

over all  $x_1$  subject to the constraints

$$(2) \quad (a) \quad r_1 \geq v_1 \geq 0$$

$$(b) \quad v_1 \leq b_1$$

$$v_1 + v_2 \leq b_2$$

$$\vdots$$

$$v_1 + v_2 + \dots + v_k \leq b_k$$

$$v_2 + v_3 + \dots + v_{k+1} \leq b_{k+1}$$

$$\vdots$$

$$v_{n-k+1} + \dots + v_n \leq k_n.$$

The origin of this problem lies in the "caterer" problem, a problem of some interest in recent years in connection with economic, industrial and military scheduling problems.

## §2. Discussion

A large number of mathematical models of economic activities culminate in the problem of determining the maximum or minimum of a linear function subject to a set of linear constraints. The importance of having available computational algorithms for the numerical resolution of these problems can hardly be over-estimated, both as far as application of the results are concerned and as far as further theoretical study is concerned. Foremost of these algorithms is the "simplex" method of Dantzig, together with its modifications by Charnes, Lemke, Neale, and others.

In the study of universal methods, insufficient attention has been paid to the underlying structure of the processes generating the minimization and maximization problems. Ideally what is desired is a systematic fitting to each process of a computational algorithm specifically designed for the process. There has been barely a start made in the mathematical theory of computational algorithms; cf. the discussion in [1]. In particular, little effort has been devoted to the question of analytic solution of these minimization and maximization problems.

In this paper we wish to consider the interesting minimization problem posed above using the functional equation approach of dynamic programming, [2]. The problem from which it is derived, the "caterer" problem has been discussed by a number of mathematicians over the last few years, see Jacobs, [3], Gaddum, Hoffman and Sokolowsky, [4], and Prager, [5].

Our interest in the possibility of an explicit solution of

the type we present here was aroused by the solution obtained by O. Gross in the case where  $k = 2$ .

### §3. The Caterer Problem

Let us now state the caterer problem in the following form: (cf. Jacobs, [3], Prager, [5])

"A caterer knows that in connection with the meals he has arranged to serve during the next  $n$  days, he will need  $r_j$  fresh napkins on the  $j^{\text{th}}$  day,  $j = 1, 2, \dots, n$ . There are two types of laundry service available. One type requires  $p$  days and costs  $b$  cents per napkin; a faster service requires  $q$  days,  $q < p$ , but costs  $c$  cents per napkin,  $c > b$ . Beginning with no usable napkins on hand or in the laundry, the caterer meets the demands by purchasing napkins at  $a$  cents per napkin. How does the caterer purchase and launder napkins so as to minimize the total cost for  $n$  days?"

As is known from the above references, and also J. W. Caddum, A. J. Hoffman and D. Sololowsky, [4], this problem can be resolved by linear programming techniques in some cases.

In this paper we shall approach the problem using the approach of dynamic programming.

### §4. Dynamic Programming Approach - I

The first approach to the problem by means of dynamic programming proceeds as follows. The state of the process at any time may be specified by the stage, i.e. day, and by the number of napkins due back from the laundry in 1, 2, up to  $p$  days hence. On the basis of that information, we must make a decision as to

how many napkins to purchase, and how to launder the accumulated dirty napkins.

It is not difficult to formulate the problem in this way, using the functional equation approach. Unfortunately, if  $p$  is large, we founder on the shoals of dimensionality.

As we shall see, the proper dimensionality of the problem is  $p - q$ , when formulated in a different manner.

### §5. Formulation of Problem

In place of this approach, let us proceed with the equations defining the process in the usual way until an appropriate point at which we shall reintroduce the dynamic programming approach.

It is first of all clear from the above formulation of the problem that we may just as well purchase all the napkins at one time at the start of the process. Let us then begin by solving the simpler problem of determining the laundering process to employ given an initial stock of  $S$  napkins. Clearly

$$(1) \quad S \geq \max_k r_k.$$

Let us now make a simplifying assumption that all the dirty napkins returned at the end of each day are sent out to the laundry, either to the fast service or to the slow service. There are many justifications for this assumption as far as applications are concerned, which we shall not enter into at the moment.

The process then continues as follows. At the end of the  $k^{\text{th}}$  day, the caterer divides  $r_k$ , the quantity of dirty napkins

on hand, into two parts,  $r_k = u_k + v_k$ , with  $u_k$  sent to the  $q$ -day laundry and  $v_k$  sent to the  $p$ -day laundry.

Continuing in this way, we see that the quantity,  $x_k$ , of clean napkins available at the beginning of the  $k^{\text{th}}$  day is determined by the following recurrence relation,

$$(2) \quad x_1 = S,$$

$$x_k = (x_{k-1} - r_{k-1}) + u_{k-q} + v_{k-p}$$

where  $u_k = v_k = 0$  for  $k < 0$ .

The cost incurred on the  $k^{\text{th}}$  day is

$$(3) \quad bv_k + cu_k, \quad k = 1, 2, \dots, N-1$$

Hence the total cost is

$$(4) \quad C_N = b \sum_{k=1}^{N-1} v_k + c \sum_{k=1}^{N-1} u_k.$$

The problem is to minimize  $C_N$  subject to the constraints on the  $u_k$

$$(5) \quad \text{a.} \quad 0 \leq u_k \leq r_k$$

$$\text{b.} \quad x_k \geq r_k, \quad k = 1, 2, \dots, N.$$

In order to illustrate the method, we shall consider two particular cases.

$$(6) \quad \text{a.} \quad q = 1, p = 2$$

$$\text{b.} \quad q = 1, p = 3$$

The general case will be discussed following this.

§6. The case  $q = 1, p = 2$

The equations in (4.2) assume the form

$$\begin{aligned}
 (1) \quad x_1 &= S \\
 x_2 &= (x_1 - r_1) + u_1 \\
 x_3 &= (x_2 - r_2) + u_2 + v_1 \\
 &\vdots \\
 x_{n-1} &= (x_{n-2} - r_{n-2}) + u_{n-2} + v_{n-3}, \\
 x_n &= (x_{n-1} - r_{n-1}) + u_{n-1} + v_{n-2}
 \end{aligned}$$

Let us now solve for the  $x_k$  in terms of the  $u_k$  and  $v_k$ . Namely

$$\begin{aligned}
 (2) \quad x_1 &= S \\
 x_2 &= (S - r_1) + u_1 \\
 x_3 &= (S - r_1 - r_2) + (u_1 + u_2) + v_1 \\
 x_4 &= (S - r_1 - r_2 - r_3) + (u_1 + u_2 + u_3) + v_1 + v_2, \\
 &\vdots \\
 x_{n-1} &= (S - r_1 - r_2 - \dots - r_{n-2}) + (u_1 + u_2 + u_3 + \dots + u_{n-2}) \\
 &\quad + (v_1 + v_2 + \dots + v_{n-3}) \\
 x_n &= (S - r_1 - r_2 - \dots - r_{n-1}) + (u_1 + u_2 + u_3 + \dots + u_{n-1}) \\
 &\quad + (v_1 + v_2 + \dots + v_{n-2}).
 \end{aligned}$$

Since  $r_k = u_k + v_k$ , this may be written

$$(3) \quad x_k = S - v_{k-1}, \quad (v_0 = 0), \quad k = 1, 2, \dots, n.$$

Turning to (4.4), we wish to minimize

$$(4) \quad C_N = c \sum_{k=1}^{N-1} r_k + (b - c) \sum_{k=1}^{N-1} v_k,$$

over all  $v_k$  subject to the constraints

$$(5) \quad (a) \quad 0 \leq v_k \leq r_k$$

$$(b) \quad S - v_{k-1} \geq r_k \quad \text{or} \quad S - r_k \geq v_{k-1}.$$

Since  $(c - b) > 0$ , we wish to choose  $v_k$  as large as possible.

Hence

$$(6) \quad v_k = \min(r_k, S - r_{k+1}), \quad k = 1, 2, \dots, N-1.$$

This determines the structure of the optimal policy. Using this explicit form of the solution it is not difficult to determine the minimizing value of  $S$ .

### §7. The Case $q = K, p = K + 1$

It is readily seen upon writing down the equations that the case  $q = K, p = K + 1$  leads to a system of equations of the same type as given above for  $q = 1, p = 2$ . This illustrates the fact that it is only the difference  $p - q$  which determines the level of difficulty of the problem.

### §8. The Case $q = 1, p = 3$

In order to illustrate the method which is applicable to the general case, let us consider the case  $q = 1, p = 3$ .

The equations in (4.2) assume the form

$$\begin{aligned}
 (1) \quad & x_1 = S, \\
 & x_2 = x_1 - r_1 + u_1, \\
 & x_3 = x_2 - r_2 + u_2, \\
 & x_4 = x_3 - r_3 + u_3 + v_1 \\
 & \vdots \\
 & x_n = x_{n-1} - r_{n-1} + u_{n-1} + v_{n-3}
 \end{aligned}$$

Thus

$$\begin{aligned}
 (2) \quad & x_1 = S, \\
 & x_2 = S - r_1 + u_1, \\
 & x_3 = (S - r_1 - r_2) + u_1 + u_2, \\
 & x_4 = (S - r_1 - r_2 - r_3) + u_1 + u_2 + u_3 + v_1 \\
 & \vdots \\
 & x_n = (S - r_1 - r_2 - r_3 - \dots - r_{n-1}) + u_1 + u_2 + u_3 + \dots \\
 & \quad + u_{n-1} + v_1 + v_2 + \dots + v_{n-3}.
 \end{aligned}$$

Hence

$$\begin{aligned}
 (3) \quad & x_1 = S \\
 & x_2 = S - v_1 \\
 & x_3 = S - v_1 - v_2 \\
 & x_4 = S - v_2 - v_3 \\
 & \vdots \\
 & x_n = S - v_{n-2} - v_{n-1}.
 \end{aligned}$$

We wish to maximize  $\sum_{k=1}^{N-1} v_k$  subject to the constraints



where  $R_k$  is the region defined by

$$(4) \quad (a) \quad x \geq v_k \geq 0,$$

$$b_{k+1} \geq v_k + v_{k+1}, \quad (b) \quad r_{k+1} \geq v_{k+1} \geq 0$$

$$\vdots \qquad \qquad \qquad \vdots$$

$$b_N \geq v_{N-1} + v_N, \qquad r_N \geq v_N \geq 0.$$

We have

$$(5) \quad f_{N-1}^*(x) = \text{Max} [v_{N-1} + v_N]$$

where

$$(6) \quad x \geq v_{N-1} \geq 0,$$

$$b_N \geq v_{N-1} + v_N, \quad r_N \geq v_N \geq 0.$$

Hence

$$(7) \quad f_{N-1}(x) = \text{Min} [b_N, x + r_N].$$

Employing the principle of optimality, [2], we see that

$$(8) \quad f_k(x) = \text{Max}_{0 \leq v_k \leq v_k^*} [v_k + f_{k+1}(\text{Min}(r_{k+1}, b_{k+1}^0 - v_k))]$$

where  $v_k^* = \text{Min} [x, b_{k+1}]$ , for  $k = 1, 2, \dots, N-1$ .

### §10. Explicit Solution

Let us assume that each  $f_k(x)$  has the form

$$(1) \quad f_k(x) = \text{Min} [P_k, x + Q_k],$$

for  $k = 1, 2, \dots, N-1$ . This is true for  $k = N - 1$ , upon referring to (9.7), and we shall establish it inductively for general  $k$ .

Assuming the relation true for  $k + 1$ , substitute in (9.8), obtaining

$$\begin{aligned}
 (2) \quad f_k(x) &= \text{Max}_{0 \leq v_k \leq v_k^0} \left[ v_k + \text{Min} \left[ P_{k+1}, \text{Min} (r_{k+1}, b_{k+1} - v_k) \right. \right. \\
 &\qquad \qquad \qquad \qquad \qquad \qquad \qquad \qquad \qquad \qquad \qquad \qquad \qquad \left. \left. + Q_{k+1} \right] \right] \\
 &= \text{Max}_{0 \leq v_k \leq v_k^0} \left[ \text{Min} \left[ P_{k+1} + v_k, \text{Min} (r_{k+1}, b_{k+1} - v_k) \right. \right. \\
 &\qquad \qquad \qquad \qquad \qquad \qquad \qquad \qquad \qquad \qquad \qquad \qquad \qquad \left. \left. + Q_{k+1} + v_k \right] \right] \\
 &= \text{Max}_{0 \leq v_k \leq v_k^0} \left[ \text{Min} \left[ P_{k+1} + v_k, \text{Min} (r_{k+1} + Q_{k+1} + v_k, \right. \right. \\
 &\qquad \qquad \qquad \qquad \qquad \qquad \qquad \qquad \qquad \qquad \qquad \qquad \qquad \left. \left. b_{k+1} + Q_{k+1}) \right] \right] \\
 &= \text{Min} \left[ P_{k+1} + v_k^0, r_{k+1} + Q_{k+1} + v_k^0, b_{k+1} + Q_{k+1} \right] \\
 &= \text{Min} \left[ P_{k+1} + \text{Min} (x, b_{k+1}), r_{k+1} - Q_{k+1} + \text{Min} (x, b_{k+1}), \right. \\
 &\qquad \qquad \qquad \qquad \qquad \qquad \qquad \qquad \qquad \qquad \qquad \qquad \qquad \left. b_{k+1} + Q_{k+1} \right] \\
 &= \text{Min} \left[ x + P_{k+1}, P_{k+1} + b_{k+1}, x + r_{k+1} + Q_{k+1}, b_{k+1} + Q_{k+1} \right] \\
 &= \text{Min} \left[ x + \text{Min} (P_{k+1}, r_{k+1} + Q_{k+1}), \text{Min} (P_{k+1} + b_{k+1}, \right. \\
 &\qquad \qquad \qquad \qquad \qquad \qquad \qquad \qquad \qquad \qquad \qquad \qquad \qquad \left. Q_{k+1} + b_{k+1}) \right].
 \end{aligned}$$

where  $R_k$  is the region defined by

$$(4) \quad (a) \quad x \geq v_k \geq 0,$$

$$b_{k+1} \geq v_k + v_{k+1}, \quad (b) \quad r_{k+1} \geq v_{k+1} \geq 0$$

$$\vdots \qquad \qquad \qquad \vdots$$

$$b_N \geq v_{N-1} + v_N, \quad r_N \geq v_N \geq 0.$$

We have

$$(5) \quad f_{N-1}^*(x) = \text{Max} [v_{N-1} + v_N]$$

where

$$(6) \quad x \geq v_{N-1} \geq 0,$$

$$b_N \geq v_{N-1} + v_N, \quad r_N \geq v_N \geq 0.$$

Hence

$$(7) \quad f_{N-1}(x) = \text{Min} [b_N, x + r_N].$$

Employing the principle of optimality, [2], we see that

$$(8) \quad f_k(x) = \text{Max}_{0 \leq v_k \leq v_k^*} [v_k + f_{k+1}(\text{Min}(r_{k+1}, b_{k+1}^0 - v_k))]$$

where  $v_k^* = \text{Min} [x, b_{k+1}]$ , for  $k = 1, 2, \dots, N-1$ .

### §10. Explicit Solution

Let us assume that each  $f_k(x)$  has the form

$$(1) \quad f_k(x) = \text{Min} [P_k, x + Q_k],$$

We assume that  $n \geq K$ .

Let us first compute  $f_{n-K}(x_1, x_2, \dots, x_{K-1})$ . This is the maximum of

$$(3) \quad v_{n-K+1} + \dots + v_n$$

subject to the constraints

$$(4) \quad x_1 \geq v_{n-K+1}$$

$$(a) \quad x_2 \geq v_{n-K+1} + v_{n-K+2}$$

⋮

$$x_{K-1} \geq v_{n-K+1} + \dots + v_{n-1}$$

$$b_n \geq v_{n-K+1} + \dots + v_{r-1} + v_n$$

$$(b) \quad r_1 \geq v_1 \geq 0,$$

$$1 = n-K+2, \dots, n.$$

Thus

$$(5) \quad f_{n-K}(x_1, x_2, \dots, x_{K-1}) = \text{Min} (b_n, x_{K-1} + r_n, x_{K-2} + r_n + r_{n-1}, \dots, x_1 + r_{n-K+2} + \dots + r_n).$$

The recurrence relation for the sequence is

$$(6) \quad f_{k-1}(x_1, x_2, \dots, x_{K-1}) = \text{Max}_{0 \leq v_{k-1} \leq v_{k-1}^*} \left[ v_{k-1} + f_k(\text{Min}(x_2 - v_k, r_k), x_3 - v_k, \dots, x_{K-1} - v_k, b_{k+K-1} - v_{k-1}) \right],$$

where

$$(7) \quad v_{k-1}^* = \text{Min} [x_1, x_2, \dots, x_{K-1}, b_{k+K-2}]$$

Let us now assume that  $f_k$  has the form

$$(8) \quad f_k(x_1, x_2, \dots, x_{K-1}) = \text{Min} \left[ P_{0k}, x_1 + P_{2,k}, x_2 + P_{2,k}, \dots, x_{K-1} + P_{K-1,k} \right].$$

Then, substituting in (6),

$$(9) \quad f_{k-1}(x_1, x_2, \dots, x_{K-1}) \\ = \text{Max}_{0 \leq v_{k-1} \leq v_{k-1}^*} \left[ v_{k-1} + \text{Min} \left[ P_{0k}, \text{Min} (x_2 - v_{k-1}, r_{k+1}) + P_{1k}, x_3 - v_{k-1} + P_{2,k}, \dots, b_{k+K-2} - v_{k-1} + P_{K-1,k} \right] \right] \\ = \text{Max}_{0 \leq v_{k-1} \leq v_{k-1}^*} \left[ \text{Min} \left[ P_{0k} + v_{k-1}, x_2 + P_{1k}, v_{k-1} + r_{k+1} + P_{1k}, x_3 + P_{2,k}, \dots, b_{k+K-2} + P_{K-1,k} \right] \right].$$

The maximum is clearly assumed at  $v_{k-1} = v_{k-1}^*$ .

Hence we have

$$(10) \quad f_{k-1}(x_1, x_2, \dots, x_{K-1}) = \text{Min} \left\{ P_{0,k} + v_{k-1}^*, x_2 + P_{1,k}, v_{k-1}^* + r_{k+1} + P_{1,k}, x_3 + P_{2,k}, \dots, b_{k+K-1} + P_{K-1,k} \right\} \\ = \text{Min} \left\{ \text{Min} \left[ P_{0,k}, P_{1,k} + r_{k+1} \right] + \text{Min} \left[ x_1, x_2, \dots, x_{K-1}, b_{k+K-2} \right], x_2 + P_{1,k}, x_3 + P_{2,k}, \dots, b_{k+K-2} + P_{K-1,k} \right\}$$

$$= \text{Min} \left\{ x_1 + w_k, x_2 + w_k, \dots, x_{K-1} + w_k, b_{k+K-2} + w_k, x_2 + P_{1,k}, \right. \\ \left. x_3 + P_{2,k}, \dots, b_{k+K-2} + P_{K-1,k} \right\},$$

where

$$(11) \quad w_k = \text{Min} [ P_{0,k}, P_{1k} + r_k ].$$

Hence

$$(12) \quad f_{k-1}(x_1, x_2, \dots, x_{K-1}) = \text{Min} \left\{ x_1 + w_k, x_2 + \text{Min} [ w_k, P_{1,k} ], \right. \\ \left. x_3 + \text{Min} [ w_k, P_{2,k} ], \dots, \text{Min} [ b_{k+K-1} + w_k, b_{k+K-2} + \right. \\ \left. + P_{K-1,k} ] \right\}.$$

From this equation we can read off the recurrence relations connecting the  $P_{1,k}$  and the  $P_{1,k-1}$ .

## §12. Discussion

The solution presented in the preceding section yields the optimal policy at each stage, as well as the value of the minimum cost.

There are a number of related problems which can be treated by similar methods. A particularly interesting one is the case where the demand is periodic. In this case, the problem reduces to maximizing

$$(1) \quad L_n = \sum_{i=1}^n v_i,$$

subject to a series of constraints

$$\begin{aligned}
 &v_1 + v_2 + \dots + v_K \leq b_1 \\
 (2) \quad &(a) \quad v_2 + v_3 + \dots + v_{K+1} \leq b_2 \\
 &\quad \quad \quad \vdots \\
 &\quad \quad \quad v_{n-K+1} + \dots + v_n \leq b_{n-K-1} \\
 &\quad \quad \quad v_{n-K+2} + \dots + v_1 \leq b_{n-K} \\
 &\quad \quad \quad \vdots \\
 &\quad \quad \quad v_n + v_1 + \dots + v_{K-1} \leq b_n.
 \end{aligned}
 \quad (b) \quad 0 \leq v_i \leq r_i.$$

Furthermore, there are the interesting problems in which there is a storage cost for each excess item, and in which there are more than two types of laundry service.

It is also easy to see that several more general classes of maximization problems subject to linear constraints may be treated by means of the same technique. We shall discuss these topics, together with the question of actual computational solution, in a further paper.

BIBLIOGRAPHY

1. R. Bellman, Mathematical Aspects of Scheduling Theory, J. Soc. Ind. Appl. Math., (to appear).
2. \_\_\_\_\_, The Theory of Dynamic Programming, Bull. Amer. Math. Soc., vol 60 (1954), pp. 503-516.
3. W. Jacobs, The Caterer Problem, Naval Research Logistics Quarterly, 1(1954), pp. 154-165.
4. J. W. Gaddum, A. J. Hoffman, and D. Sokolowsky, On the Solution to the Caterer Problem, *ibid.*
5. W. Prager, On the Caterer Problem, LEM-13, Division of Applied Mathematics, Brown University, 1956.