DISCUSSION PAPER NO. 99

AN OPTIMUM SEEKING ALGORITHM FOR LOCATING LOCK BOXES

by

Avinoam Perry

August 8, 1974

AN OPTIMUM SEEKING ALGORITHM FOR LOCATING LOCK BOXES

ABSTRACT

The paper describes an algorithm which is based on the branch and bound procedure for solving the lock-box location problem. Because of the special structure associated with the mathematical formulation of this program, the solution of each subproblem takes negligible amount of time and it is achieved directly rather than iteratively. Nevertheless, some approximations are required for the estimation of the cost parameters associated with the objective function, a fact which is true in almost any real life application of mathematical programming.

AN OPTIMUM SEEKING ALGORITHM FOR LOCATING LOCK BOXES

by

Avinoam Perry

I. INTRODUCTION

The problem associated with the efficient location of lock boxes is described in detail by Ferdinand K. Levy [2] who also outlined a heuristic program for reaching a satisfactory solution. In this paper we developed a branch and bound algorithm which guarantees an optimum solution to the problem. The algorithm described in this paper is similar to the one suggested for solving the quantity discount transportation type problem [1], but fortunately, the problem may be formulated in such a way that the solution of every subproblem requires a negligible computational effort due to the fact that the solution of each subproblem is direct rather than iterative. For the purpose of our discussion we adopt the following notations:

(1) A_i = the total average face amount of checks originated at location $i \in I$ in a given period.

(2) M_{ij} = the number of mail days of checks originated at location $i \in I$ and sent to lock box $j \in J$.

(3) B_{ij} = the number of days that lock box $j \in J$ holds checks originated at $i \in I$ before the funds are available for use.

(4) C_j = the variable charge per check collected in lock-box $j \in J$.

(7) D_i = average number of checks originated at location $i \in I$ within a given period of time.

(8) $x_{ij} = \begin{cases} 1 & \text{if checks originated at location } i \in I \text{ are sent to lock-box } j \in J \\ 0 & \text{otherwise} \end{cases}$

(9) $y_j = \begin{cases} 1 & \text{if lock-box } j \in J \text{ is activated} \\ 0 & \text{otherwise} \end{cases}$

(10) d_{ij} = the variable charge associated with processing checks originated at location $i \in I$ and sent to lock-box $j \in J$, $d_{ij} = D_i C_j$.

(11) c_{ij} = the total variable charge (including opportunity cost of idle funds) associated with checks originated at location $i \in I$ and sent to lock-box $j \in J$, $c_{ij} = d_{ij} + rA_i (M_{ij} + B_{ij})$.

We can now formulate the mathematical program which is to be minimized if the optimum location scheme is sought.

(12) $\min Z = \sum_{i \in I} \sum_{j \in J} c_{ij} x_{ij} + \sum_{j \in J} F_j y_j$

(13) s.t. $\sum_{j \in J} x_{ij} = 1$ for all $i \in I$

(14) $\sum_{i \in I} x_{ij} - my_j \leq 0$ for all $j \in J$

x_{ij} and y_j are defined as in (8) and (9) respectively.

Let m = number of potential lock-box locations and n = number of customer centers, problem (12) has (n+m) constraints and m(n+1) binary variables. A zero-one integer programming code using the above formulation may prove to be very expensive if a real life application is involved. In the following section we describe an efficient algorithm for solving problem (8)-(9), (12)-(14).

II A GENERAL DESCRIPTION OF THE ALGORITHM

In order to facilitate the presentation, expressions (8)-(9), (12)-(14) will be referred to as problem P*. Our initial step consists of a relaxation of P* due to the deletion of constraint set (14) while letting $y_j = 0$ for all $j \in J$. Feasibility is then restored by the introduction of the constraints of (14) through the branch and bound procedure.

Denote P_o as the relaxed problem (8)-(9), (12)-(13). Then a solution to P_o is given by (15)-(16). Denoting $t \subset J$ as a location index

$$(15)\quad x_{it} = \left\{\begin{array}{ll} 1 & \text{if } c_{it} = \min\limits_{j \in J} (c_{ij}) \\ 0 & \text{otherwise} \end{array}\right\} \text{ for all } i \in I$$

$$(16)\quad Z = \sum_{i \in I} c_{it} + \sum_{j \in J} F_j\, y_j$$

Definition 1: A solution $X = \{x_{ij}\}$ to problem P_o is defined as a "better than optimal" solution, since any other bounded solution to P* yields a larger or equal value for Z in its corresponding subset.

Definition 2: A solution $X = \{x_{ij}\}$ to P_o is "interval infeasible" if $y_j = 0$ and $\sum\limits_{i \in I} x_{ij} \geq 1$ for at least one $j \in J$.

We are now ready to summarize the steps involved in reaching an optimum solution.

Step 1: Let $y_j = 0$ for all $j \in J$ and solve the relaxed problem (8)-(9), (12)-(13) - the solution is as indicated in (15-(16).

Step 2: If the solution is interval infeasible, i.e. $y_j = 0$ and $\sum\limits_{i \in I} x_{ij} \geq 1$ for at least one $j \in J$, select one potential lock-box location $t \in \bar{J}$ where $\bar{J}$ is the set of all potential locations for which interval feasibility is violated, and $F_t = \max\limits_{j \in \bar{J}} F_j$.

Step 3: Branch from the variable selected in step 2 into two new problems: in branch 1 eliminate location t from further consideration by deleting it from the list of potential candidates for lock-box location, and use (15)-(16) to obtain a new solution. In branch 2 set $y_t=1$ and increase the current value of the objective function by F_t.

Step 4: Select the one branch with the smallest objective function value among all active branches.

Step 5: If solution associated with the branch selected in step 4 is interval feasible it is the optimum of (8)-(9), (12)-(14). Otherwise go to 2.

Certain discussions of the above procedure are now relevant. a) The algorithm converges in a finite number of steps since the total number of bases for the constraint set given by (13)-(14) is finite and each search routine excludes at least one basis. b) The branching procedure results in a partition of the interval feasible basic feasible solutions in that subset and thus the algorithm is expected to be efficient. c) The solution of each subproblem is obtained directly throughout a short search routine as described in (15)-(16). Thus, reducing total computational effort considerably.

III. AN ILLUSTRATION

In this section we present a numerical example which demonstrates the operational steps of the algorithm. In order to facilitate the presentation we assume that c_{ij} and F_j have already been approximated according to (1)-(11) and the data is presented in a matrix form as in table 1.

Table 1

Customer location \ Lock-box location	1	2	3	4
1	5	3	8	9
2	6	1	2	3
3	8	2	1	3
4	3	4	5	2
5	7	5	4	2
6	6	7	4	5
F_j	10	15	6	9

There are 4 possible locations and 6 customer centers. The variable costs c_{ij} and the fixed cost F_j are given in table 1.

Step 0: The initial assignment is made based on the minimum variable cost c_{ij}. The basic variables are: $X^o = \{x_{12},x_{22},x_{33},x_{44},x_{54},x_{63}\}$ $Z^o = c_{12}+c_{22}+c_{33}+c_{44}+c_{54}+c_{63} = 3 + 1 + 1 + 2 + 2 + 4 = \underline{\underline{13}}$. This solution is interval infeasible for j=2,3,4, and since $F_2 = \max_{j \in J} F_j$ we branch from this variable into the next stage.

Step 1: Branch 1 denoted by $Z^1 = Z^o + F_2 = 13 + 15 = \underline{\underline{28}}$. The solution is interval infeasible for j=3,4, $X^1 = \{x_{12},x_{22},x_{33},x_{44},x_{54},x_{63},y_2\}$. In branch 2 we delete location 2 from further considerations and compute a new allocation. $X^2 = \{x_{11},x_{23},x_{33},x_{44},x_{54},x_{63}\}$, $Z^2 = Z^o + (c_{11} - c_{12}) + (c_{23} - c_{22}) = 13 + 2 + 1 = \underline{\underline{16}}$. This solution is interval infeasible for j = 1,3,4.

Step 2: $\min [Z^1,Z^2] = Z^2$

$\max [F_1,F_3,F_4] = F_1 = 10$

$Z^{21} = Z^2 + F_1 = 16 + 10 = \underline{\underline{26}}$

$X^{21} = \{x_{11},x_{23},x_{33},x_{44},x_{54},x_{63},y_1\} \Rightarrow$ interval infeasible for j=3,4

$Z^{22} = Z^2 + (c_{13} - c_{11}) = 16 + 3 = \underline{\underline{19}}$

$X^{22} = \{x_{13},x_{23},x_{33},x_{44},x_{54},x_{63}\} \Rightarrow$ interval infeasible for j=3,4.

Step 3: $\min [Z^1,Z^{21},Z^{22}] = Z^{22}$

$\max [F_3,F_4] = F_4 = 9$

$Z^{221} = Z^{22} + F_4 = 19 + 9 = \underline{\underline{28}}$ interval infeasible for j=3

$X^{221} = \{x_{13},x_{23},x_{33},x_{44},x_{54},x_{63},y_4\}$

$Z^{222} = Z^{22} + (c_{44} - c_{43}) + (c_{54} - c_{53}) = 19 + 3 + 2 = \underline{\underline{24}}$

$X^{222} = \{x_{13},x_{23},x_{33},x_{43},x_{53},x_{63}\} =$ interval infeasible for j=3.

Step 4: $\min [Z^1,Z^{21},Z^{221},Z^{222}] = Z^{222}$

$\max [F_3] = F_3$

$Z^{2221} = Z^{222} + F_3 = 24 + 6 = \underline{\underline{30}}$

$X^{2221} = \{x_{13},x_{23},x_{33},x_{43},x_{53},x_{63},y_3\}$ = interval feasible

X^{2222} = infeasible - deleted from list of active branches.

<u>Step 5</u>: min $[Z^1, Z^{21}, Z^{221}, Z^{2221}] = Z^{21}$

max $F_3, F_4] = F_4 = 9$

$X^{211} = \{x_{11}, x_{23}, x_{33}, x_{44}, x_{54}, x_{63}, y_1, y_4\}$

$Z^{211} = Z^{21} + F_4 = 26 + 9 = \underline{\underline{34}}$ interval infeasible for j=3 but since $Z^{211} > Z^{2221}$ this branch is deleted from the list of active branches

$Z^{212} = Z^{21} + (c_{41} - c_{44}) + (c_{53} - c_{54}) = 26 + 1 + 2 = \underline{\underline{29}}$

$X^{212} = \{x_{11}, x_{23}, x_{33}, x_{41}, x_{53}, x_{63}, y_1\}$ = interval infeasible for j=3.

<u>Step 6</u>: min $[Z^1, Z^{212}, Z^{221}, Z^{2221}] = Z^1 = Z^{221} = 28$, select Z^1 arbitrarily

max $[F_3, F_4] = F_4 = 9$

$Z^{11} = Z^1 + F_4 = 28 + 9 = \underline{\underline{37}} > Z^{2221} = 30$ delete this solution from list of active branches

$X^{11} = \{x_{12}, x_{22}, x_{33}, x_{44}, x_{54}, x_{63}, y_2, y_4\}$ infeasible for j=3

$Z^{12} = Z^1 + (c_{41} - c_{44}) + (c_{53} - c_{54}) = 28 + 3 = \underline{\underline{31}} > Z^{2221}$, delete this branch from the list of active branches

$X^{12} = \{x_{12}, x_{22}, x_{33}, x_{41}, x_{53}, x_{63}, y_2\}$ infeasible vor j=1,3.

<u>Step 7</u>: min $[Z^{212}, Z^{221}, Z^{2221}] = Z^{221} = 28$

max $[F_3] = F_3 = 6$

$Z^{2211} = Z^{221} + F_3 = 28 + 6 = \underline{\underline{34}} > 30$ delete this branch

$X^{2211} = \{x_{13}, x_{23}, x_{33}, x_{44}, x_{54}, x_{63}, y_3, y_4\}$ = interval feasible

$Z^{2212} = Z^{221} + (c_{14} - c_{13}) + (c_{24} - c_{23}) + (c_{34} - c_{33}) + (c_{64} - c_{63}) =$
$28 + 1 + 1 + 2 + 1 = \underline{\underline{33}} > 30$ delete this branch.

$X^{2212} = \{x_{14}, x_{24}, x_{34}, x_{44}, x_{54}, x_{64}, y_4\}$ = interval feasible

Step 8: min $[Z^{212}, Z^{2221}] = Z^{212} = 29$

$\max [F_3] = F_3 = 6$

$Z^{2121} = Z^{212} + F_3 = 29 + 6 = 34 > 30$ delete this branch

$X^{2121} = \{x_{11}, x_{23}, x_{33}, x_{41}, x_{53}, x_{63}, y_1, y_3\}$ = interval feasible

$Z^{2122} = Z^{212} + (c_{21} - c_{23}) + (c_{31} - c_{33}) + (c_{51} - c_{53}) + (c_{61} - c_{63}) =$

$29 + 4 + 7 + 3 + 2 = \underline{\underline{45}}$ delete this branch.

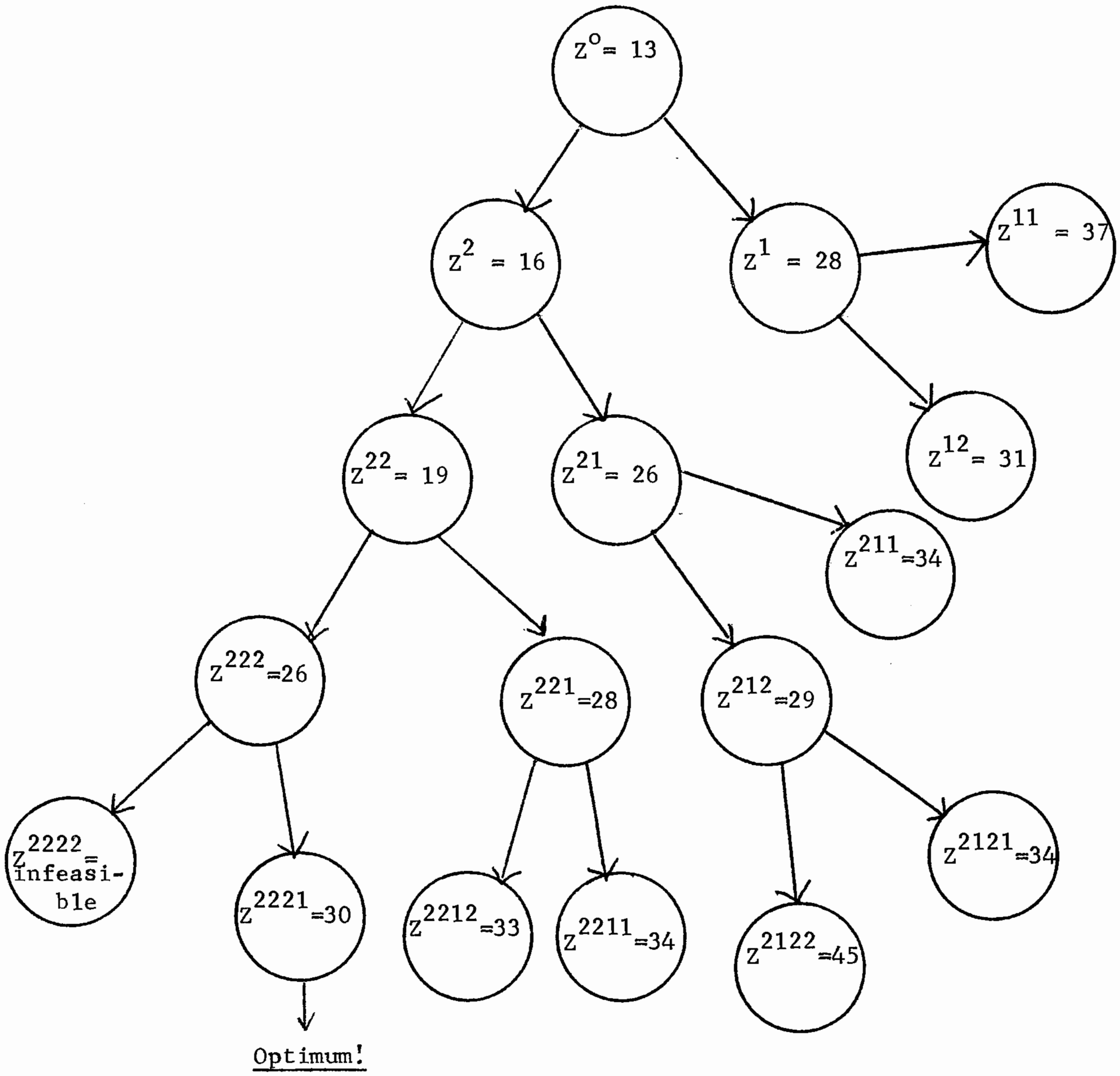
$X^{2122} = \{x_{11}, x_{21}, x_{31}, x_{41}, x_{51}, x_{61}, y_1\}$ interval feasible.

The optimum solution is $X^{2221} = \{x_{13}, x_{23}, x_{33}, x_{43}, x_{53}, x_{63}, y_3\}$ and the minimum cost is $Z^{2221} = \underline{\underline{30}}$.

The solution procedure involved 8 steps but the computational effort in each step was of no significance, a fact which contributed greatly to the overall efficiency of the algorithm.

A summary of the branching procedure is illustrated in Fig. 1.

Fig. 1



Note that Z^{11}, Z^{12}, and Z^{211} are interval infeasible but their current "better than optimal solution" is inferior to an existing interval feasible optimal solution, therefore further enumeration is not necessary.

REFERENCES

1. Balachandran, V. and Avinoam Perry, "Transportation Type Problems with Quantity Discounts," Discussion Paper No. 98, The Center for Mathematical Studies in Economics and Management Science, Northwestern University, Evanston, Illinois, July 1974.

2. Levy, Ferdinand K., "An Application of Heuristic Problem Solving to Accounts Receivable Management," Management Science, Vol. 12, No. 6, February 1966.