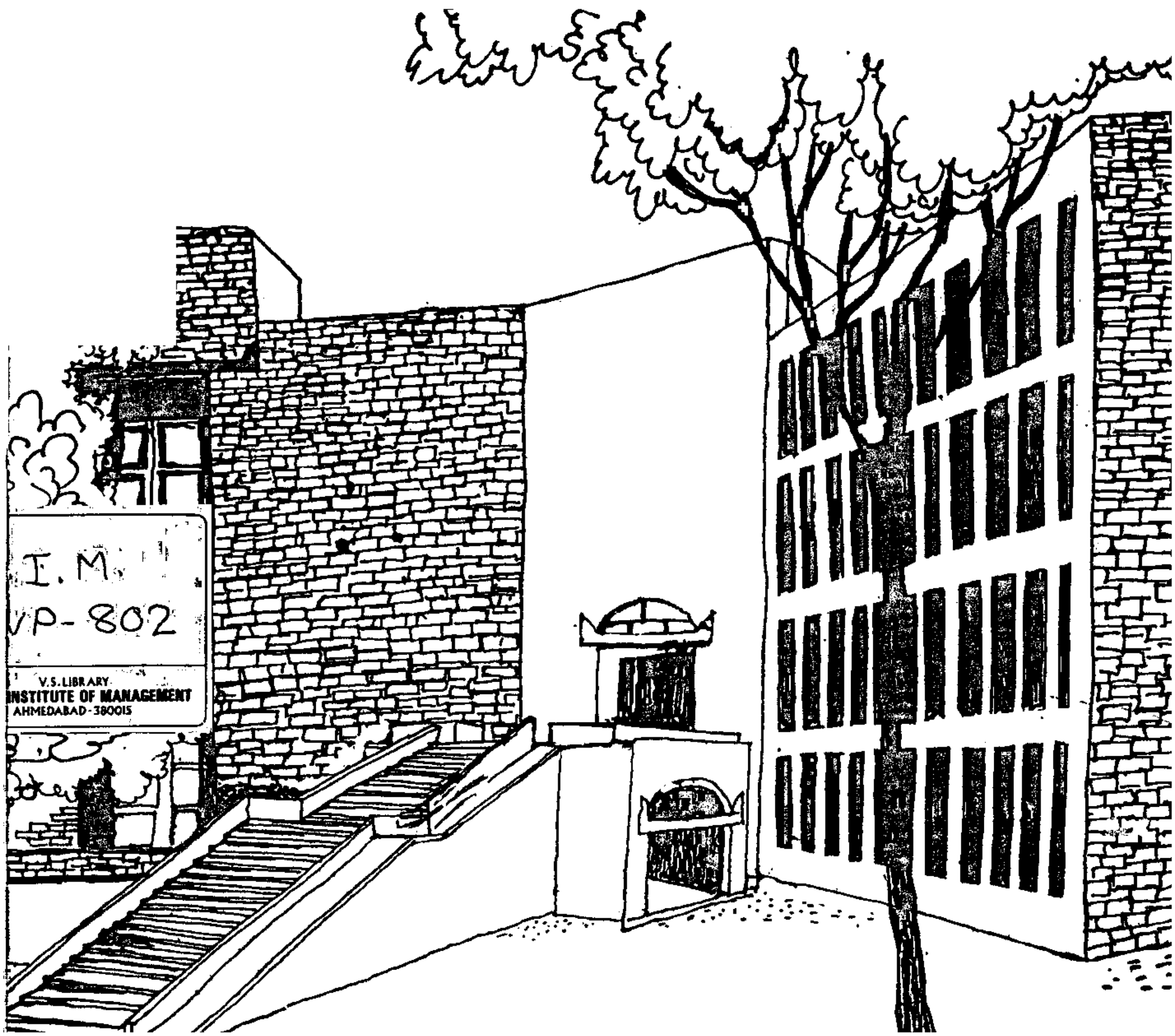
विद्याविनियोगाद्विकासः

IIM

AHMEDABAD

Working Paper



THE CAPACITATED PLANT LOCATION PROBLEM
- SOME WORST CASE ANALYSES

By

R. Sridharan

W P No. 802
May 1989

The main objective of the working paper series of the IIMA is to help faculty members to test out their research findings at the pre-publication stage.

INDIAN INSTITUTE OF MANAGEMENT
AHMEDABAD-380 056
INDIA

Abstract

In this note we show that the worst case solutions of the weak linear programming relaxation, the DROP heuristic and the ADD [illegible]uristic for the Capacitated Plant Location Problem are not [illegible]d.

The Capacitated Plant Location Problem
– Some Worst Case Analyses

Introduction

The location of plants, such as factories or warehouses, is an inevitable strategic decision for most organisations as it has a direct bearing on the cost of supplying commodities to customers. Transportation costs often form major portion of the price (or cost) of goods. Equally important are the fixed costs involved in opening and operating a plant at any given location. Such problems have been widely studied in the literature under the names of plant, warehouse, or facility location problems. When each potential plant has a capacity, that is, an upper bound, on the amount of demand that it can service, the problem is known as the capacitated plant location problem (CPLP). The capacitated plant location problem, with n potential plants and m customers, can be formulated as a mixed integer program, as follows.

$$Z = \min \sum_{i,j} c_{ij} x_{ij} + \sum_{j} f_j y_j \quad (1)$$

subject to

$$\sum_{j} x_{ij} = 1, \; i = 1, \dots, m; \quad (2)$$

$$\sum_{i} d_i x_{ij} <= s_j y_j, \; j = 1, \dots, n; \quad (3)$$

$$x_{ij} <= y_j \text{ for every } i,j. \quad (4)$$

$$x_{ij} >= 0 \text{ for every } i,j. \quad (5)$$

$$y_j = \{0,1\} \quad j = 1, \dots, n. \quad (6)$$

The constraints (2) guarantee that the demand of every client is satisfied, and constraints (3) guarantee that each open plant does not supply more than its capacity, and that the clients are supplied only from open plants.

The literature on CPLP is very rich; see Magnanti and Wong[9], Francis and Goldstein[6], Salkin[11], and Wong[12] for bibliographies on CPLP. Researchers have worked on both heuristic solution methods and exact algorithms to solve CPLP. The heuristic solutions primarily belong to the category of ADD heuristics, see Kuehn and Hamburger[8] and the DROP heuritics, see Feldman, Lehrer and Ray [4]. The exact algorithms for CPLP work with various relaxations of the problem. The relaxations considered in the literature were Linear Programming relaxations or Lagrangian relaxations. A linear programming relaxation of CPLP without constraints (4) is called the Weak Linear Programming relaxation. Sa[10], Ellwein and Gray[3], and Akinc and Khumawala[1] work with this relaxation. When constraints (4) are also included in the formulation, the relaxation is known as the Strong Linear Programming relaxation. In this note we will study the worst case behaviours of the Weak Linear Programming relaxation, the ADD heuristic and the DROP heuristic.

Worst Case Analysis

We give some worst case examples for the Weak Linear Programming relaxation, the DROP and the ADD heuristics for CPLP. The Weak Linear Programming relaxation has been explained in the previous section. The DROP heuristic starts with all plants open and, in each iteration of the application of the procedure, closes a plant that gives the maximum savings (decrease in the objective function) and stops when no more savings can be obtained. The ADD heuristic starts with no plants open and, at each iteration, adds a plant that gives the maximum savings and stops when no more savings can be obtained. See Jacobsen[7] for a complete description of these methods.

The performance of the relaxations or the heuristics can be analysed using different measures. Before we look at these measures let us define Z_H to be the objective value of any heuristic H; Z^* the optimal solution; Z_L the objective value of a relaxation; and Z_R a reference value. The value Z_R is an upper bound on the maximum value, see Fisher[5]. A popular performance measure for a heuristic value is $(Z_H - Z^*)/Z^*$. A major drawback with such a measure is that it is subject to the "scaling" problem. That is, we could add or subtract a positive constant to each of the costs in every element of row in the c_{ij} matrix without affecting the execution of the heuristic or the optimal solution. Although this does not affect the solutions, it does affect the performance measure $(Z_H - Z^*)/Z^*$ so that it could be made smaller or larger by scaling the data. Another problem with this measure is that we need to impose some restrictions on c_{ij}'s and f_j's so that Z^* does not become zero. In order to circumvent these problems we will use the following measures. For any greedy heuristic H, (ADD and DROP heuristics are greedy heuristics) the measure G is given by

$$G = \frac{Z_H - Z^*}{Z_R - Z^*}$$

and for the relaxation L the measure R is given by

$$G = \frac{Z^* - Z_L}{Z_R - Z_L}$$

When $Z^* = Z_R$ in G (or $Z_R = Z_L$ in R) the heuristic (or the relaxation) gives the optimal solution and therefore we define G = 0 (R = 0). In general $0 <= G <= 1$, and $0 <= R <= 1$.

A heuristic is "good" if sup G is smaller than 1 where the sup is taken over all possible data. A relaxation is "good" if sup R is smaller than 1, for all possible data.

Now we show that the weak linear program, the DROP heuristic and the ADD heuristic are not good.

<u>Weak</u> <u>Linear</u> <u>Program</u>

<u>Proposition</u> The weak linear program is not good.

<u>Proof</u> Consider the subfamily of problems with $f_j = 1$, $s_j = m$, $j = 1, ..., n$; and $d_i = 1, ..., m$; $m >= 2$ and

$$c = \begin{pmatrix} 0 & 1 & 1 & \cdots & 1 \\ 1 & 0 & 1 & \cdots & 1 \\ \cdot & & \cdot & & \cdot \\ \cdot & & & \cdot & \cdot \\ \cdot & & & \cdot & 1 \\ 1 & \cdots & & 1 & 0 \end{pmatrix}$$

The weak LP solution is $y_j = 1/n$, $j = 1, ..., n$ and $x_{ij} = 1$ giving $Z_L = 1$. Clearly $Z_R = m$ (the upper bound on all solutions). We can see that the optimal solution is $(m - 1)$.

$$R = \frac{(m-1)-1}{m-1} = \frac{m-2}{m-1}$$

Thus sup R = 1.

The DROP Heuristic

Proposition The DROP heuristic is not good.

Proof Consider the subfamily of problems with $n = m + 1 >= 2$, $s_j = 1$, $f_j = 1$, $j = 1, \ldots, n-1$ and $f_n = 2$, $s_n = m$, $d_i = 1$, $i = 1, \ldots, m$; and

$$c = \begin{pmatrix} 0 & 1 & 1 & \ldots & 1 & 0 \\ 1 & 0 & 1 & \ldots & 1 & 0 \\ . & & . & & . & 0 \\ . & & & . & . & 0 \\ . & & & & .\,1 & 0 \\ 1 & \ldots & & 1 & 0 & 0 \end{pmatrix}$$

We have $Z_R = m$ and $Z^* = 2$. At the first iteration, DROP heuristic will eliminate plant n, as savings for plant n is 2, and for all other plants 1 to n-1, the savings is 1. Then, no other plant is deleted since deleting any other plant will lead to infeasibility. Therfore, the heuristic value $Z_H = m$. Now,

$$G = \frac{m-2}{m-2} = 1.$$

The ADD Heuristic

Proposition The ADD heuristic is not good.

Proof. Take a subfamily of problems with $n = m + 1 >= 2$, $s_j = m$, $j = 1, \ldots, n$; $f_j = 1$, $j = 1, \ldots, n-1$; $f_n = K$, $d_i = 1$,

$i = 1, \ldots, m$; and

$$c = \begin{pmatrix} K & 0 & 0 & \cdots & 0 & 0 \\ 0 & K & 0 & \cdots & 0 & 0 \\ \cdot & & \cdot & & \cdot & 0 \\ \cdot & & & \cdot & \cdot & 0 \\ \cdot & & & \cdot & 0 & 0 \\ 0 & & & 0 & K & 0 \end{pmatrix}$$

Clearly, $Z_R = K$, and $Z^* = 2$. Now, at the first iteration, the ADD heuristic will "add" plant n, as the savings for adding plant n is M − K, and the savings for adding any of the plants 1, ..., n-1. is M − K − 1 where M is a very large number (since, when no plants are open, we do not have a feasible solution and we start with the objective function being M). We cannot "add" any more plants as we do not get any further savings. $Z_H = K$. Hence

$$G = \frac{K - 2}{K - 2} = 1.$$

Conclusion

As a consequence of these results, we can expect to have instances of the problem where the DROP and ADD heuristics, and the linear programming relaxation can perform very poorly. Domschke and Drexl[2] suggest starting procedures for ADD type heuristics showing that the performance of the heuristics improve with the starting procedures. We feel that such starting procedures will also improve the performance of the DROP heuristics.

REFERENCES

1. Akinc, U. and Khumawala, M. (1977), "An Efficient Branch and Bound Algorithm for the Capacitated Warehouse Location Problem". _Management Science_, 23, 585-94.

2. Domschke, W. and Drexl, A. (1985), "ADD-heuristics' starting procedures for capacitated plant location models", _European J. of Operational Research_, 21, 47-53.

3. Ellwein, L. B. and Gray, P. (1971), "Solving Fixed Charge Location-Allocation Problem with Capacity and Configuration Constraints", _AIIE Trans._, 4, 290-98.

4. Feldman, E., Lehrer, F. A. and Ray, T. L. (1966), "Warehouse Locations under Continuous Economies of Scale", _Management Science_, 2.

5. Fisher, M. L. (1980), "Worst-case Analysis of heuristic algorithms", _Management Science_, 26, 1-17.

6. Francis, R. L. and Goldstein, J. M. (1974), "Location Theory: A Selective Bibliography", _Operations Research_, 22, 400-410.

7. Jacobsen, S. K. (1983), "Heuristics for the Capacitated Plant Location Model", _European J. of Operational Research_, 12, 253-261.

8. Kuehn, A. A. and Hamburger, M. J. (1963), "A Heuristic Program for Locating Warehouses", _Management Science_, 9, 643-666.

9. Magnanti, T. L. and Wong, R. T. (1984), "Network Design and Transportation Planning", _Transportation Science_, 18.

10. Sa, G. (1969), "Branch and Bound and Approximate Solutions to the Capacitated Plant Location Problem", Operations Research, 1005-1016.

11. Salkin, H. M. (1975), Integer Programming, Addision-Wesley Publishing Co., Reading, Mass.

12. Wong, T. L. (1983), "Recent Research on Location and Network Design Problems: An Annotated Bibliography", Proceedings of the summer school on Combinatorial Optimisation, National Institiute for Higher Education, Dublin, 4-15.