A LAGRANGIAN HEURISTIC FOR THE CAPACITATED PLANT LOCATION PROBLEM WITH SIDE CONSTRAINTS

By

R. Sridharan

W P No. 824
September 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 paper we present a Lagrangian relaxation approach for solving the capacitated plant location problem with side constraints. The side constraints are upper bound constraints on disjoint subsets of the (0-1) variables. We also provide an application where this procedure can be used to solve a particular Vehicle Routing Problem. Computational results are provided for some problems both on the main frame compuuter as well as the personal computer.

A Lagrangian Heuristic for the Capacitated Plant Location Problem with side constraints

Introduction

The location of plants, such as factories or warehouses, is an important strategic decision for organisations. Transportation costs which often form a major portion of the cost of goods supplied are a function of the location of plants. The fixed costs of opening and operating a plant may also vary from one location to another. Such problems have been widely studied in the literature under the names of plant, warehouse, or facility location problems. When each potential location has a capacity, that is, an upper bound on the demand it can service, the problem is known as the Capacitated Plant Location Problem (CPLP). Cornuejols, Sridharan and Thizy[1], Magnanti and Wong[2], Wong[3], Salkin[4], and Francis and Goldstein[5], provide excellent bibliographies on CPLP.

In this paper we present a Lagrangian relaxation approach for solving the capacitated plant location problem with side constraints. The side constraints are upper bound constraints on disjoint subsets of the (0-1) variables.

This extention can be used to solve the following "Capacitated Plant Location Problem". Suppose we have a number of choices on the size of the plant that can be considered at a given location. Then we can consider this as a CPLP with number of "potential locations" computed as follows. Each choice of the size of a plant at a given location is considered as a potential location.

That is, if there are three choices of plant sizes at a given location, then the number of potential locations is equal to three. But, we can not open more than one plant at a given location. Therefore, for each location we can have an upper bound constraint that restricts the number of plants that can be opened there to one. Then the solution to this CPLP with the side constraints will give the solution to the CPLP where we make the choice of the size of the plant to be opened at any location.

The CPLP with side constraints as described above has not been studied in the literature. In this paper we present an application for this extension and a solution method for this problem based on the Lagrangian relaxation approach. This solution procedure is an extension of the approach used by Christofides and Beasley[6] for the CPLP.

This paper is organized as follows. In section 2, we give an application where this extension of CPLP can be used. In section 3, we give a mathematical formulation of the problem. In section 4, we define a Lagrangian relaxation of the problem. We describe the procedure to obtain the upper bound in section 5. Then, in section 6 we provide a problem reduction procedure. In section 7 we describe the subgradient approach involved in the lagrangian procedure. Finally, in section 8, we provide some computational results and concluding remarks.

An Application

The motivation for including side constraints arises from the application of CPLP to the following variation of the Vehicle Routing Problem (VRP). VRP involves a choice of vehicles that will be assigned to serve a given set of customer locations. There are different vehicle types that can be used. All these vehicles start from a central depot to serve the customers. The objective of the VRP is to minimize the cost of operating the vehicles and serving customers. The demands of all the customers must be met and the capacities of the vehicles being used can not be exceeded. An application of CPLP to solve VRP can be considered as follows.

We were introduced to this problem when we visited DIAGMA, a consulting firm in Paris. At DIAGMA, they have a heuristic procedure to solve this problem. We will describe a part of the solution procedure here. Their solution procedure starts with a greedy type heuristic that assigns customers to vehicles. Then an interchange procedure is used to improve the solution. From their experience, they found that the interchange procedure works very well whenever all the capacities of the vehicles are equal. When the capacities of the vehicles are not all equal their interchange procedure does not perform well. This motivated us to conider an improvement procedure that uses the CPLP formulation as described below.

Suppose we start with a heuristic solution that provides the subset of the vehicles to be used and the corresponding tours. Then, we can view the problem of optimizing these tours as a CPLP with "plants" being the combination of tours and vehicles. That is, each tour-vehicle combination is considered as a potential location for a plant. The number of potential locations is then equal to the product of number of vehicles and the number of tours. Let us use the subscript i for customers, j for vehicles and k for tours. Then the subscript jk refers a potential location, i.e., a tour-vehicle combination. The "transportation cost", c_{ijk} is the insertion cost of customer i to "location" jk. The insertion cost can be computed as a function of the marginal increase in the length of the tour due to the addition of customer i. The fixed cost, f_{jk} is the cost of using "location" jk.

Let the demand of customer i be d_i and the capacity of vehicle j be s_j. The capacity of location jk is computed as follows. For each tour k, we compute the total demand D_k of all the customers in that tour. When a vehicle j is considered in combination with tour K, we can compute the capacity of location jk, s_{jk}, as $s_j - D_k$. If $s_{jk} < 0$, then we set $f_{jk} = \infty$ as it is infeasible to assign vehicle j to tour k. During the solution procedure, whenever a customer is reassinged from one tour to another we need to update the corresponding capacities of the locations and the insertion costs.

The variables x_{ijk} represent the amount of demand of customer i satisfied by location jk. The variables y_{jk} will have a value 1 if a particular tour-vehicle combination is chosen and will have a value 0 otherwise. It is obvious that a vehicle can not be assigned to more than one tour. This, in turn, leads to the restriction in the number of "plants" that can be opened at a given "location" resulting in the above mentioned side constraints for the CPLP. Then, the solution to this "CPLP" with side constraints gives a solution to the VRP.

Problem Formulation

The capacitated plant location problem with side constraints can be formulated as a mixed-integer program as follows.

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

subject to

$$\sum_j x_{ij} = 1, \; i = 1, \ldots, m; \quad (2)$$

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

$$0 <= x_{ij} <= y_j \leq 1, \text{ for every } i,j; \quad (4)$$

$$y_j = 0,1 \quad (5)$$

$$P_L <= \sum_j y_j <= P_U \quad (6)$$

$$\sum_{j \in N_k} y_j <= M_k, \; N_k \cap N_l = \phi, \; k,l = 1, \ldots, K. \quad (7)$$

where P_L is a lower limit on the number of open plants, P_U is an upper limit on the number of open plants, M_k is an upper bound on the number of plants that can be opened in the set $N_k \subseteq J$, and all the other parameters have the usual interpretation associated with the capacitated plant location problem.

Constraint 6 is a surrogate constraint that strengthens the formulation of CPLP. This constraint improves the bounds obtained for the problem in the solution procedure. We also improve the bounds P_L and P_U during the solution procedure to provide tighter limits on the open plants.

Lower Bounds

The lower bound for (1) is obtained by solving a Lagrangian relaxation of the problem. Let u_i, $i = 1, \ldots, m$, be the Lagrange multipliers associated with the demand constraints (2). Then the Lagrangian dual program is obtained as

$$Z_D = \max_u Z_D(u)$$

where,

$$Z_D(u) = \min \sum_i \sum_j (c_{ij} - u_i) x_{ij} + \sum_j f_j y_j + \sum_i u_i \quad (8)$$

subject to (3), (4), (5), (6) and (7).

For a given u, the Lagrangian dual program can be solved very easily. This problem $Z_D(u)$ without the constraints (6) and (7) for a given u, breaks up into continuous knapsack problems for each j. We can then write,

$$Z_D(u) = \quad Z_D(u;j)$$

where,

$$Z_D(u;j) = \min \quad \sum_i (c_{ij} - u_i) x_{ij} + f_j \quad (9)$$

$$\sum_i d_i x_{ij} <= s_j \quad (10)$$

$$x_{ij} >= 0. \quad (11)$$

Let the objective of $Z_D(u;j)$ be f_j' and let x^*_{ij} be the solution to (9) to (11). The value f_j' can be thought of as a "penalty cost" of keeping plant j open.

We can now include the constraints (6) and (7) to obtain a (0,1) problem on the y_j variables. This problem KP_2 is

$$KP_2 = \min \sum_j f_j' y_j + \sum_i u_i \quad (12)$$

subject to

(5), (6) and (7)

The problem KP_2 is solved as follows:

Step 1: Sort f_j' in ascending order.

Step 2: Set y_j to one for the first P_L plants in the list taking into consideration the side constraints (7).

Step 3: If $P_L = P_U$, go to step 5.

Step 4: Keep setting y_j to one with due consideration to constraints (7) until either (a) P_U plants have been opened or (b) the f_j' values have become >= 0.

Step 5: Let the set of values y_j chosen as above be y_j^*. The y_j^* that are not set at one are set at zero.

The optimal solution to the Lagrangian dual program is y_j^*, and x_{ij}^* found in (9) to (11) and KP_2 is equal to

$$KP_2 = \sum_j f_j' y_j^* + \sum_i u_i . \quad (13)$$

A lower bound for (1) is given by the objective value of KP_2. We compute an upper bound for (1) by solving a transportation problem which uses the open plants as given by KP_2. The determination of the initial and subsequent upper bounds is described in the next section.

Upper Bounds

The solution to the lagrangian problem does not provide a feasible solution in general. However, we can easily find a feasible solution to (1) by solving a transportation problem with the set of open plants given by $\bar{y}_j^*$. The advantage of finding an improved feasible solution need not be over-emphasised. An improved feasible solution accelerates the chances of terminating the Lagrangian heuristic, and also in a branch and bound procedure it helps to prune the tree. We now give the procedure for finding the initial feasible solution and the improved feasible solutions.

The initial feasible solution is found by simply picking up the P_L plants with due consideration to constraints (7). The initial value of P_L is found by taking the first (subject to satisfying (7)) P_L plants with largest capacities such that the sum of their capacities just exceeds the total demand of all the customers. Then, with this set of plants being set open, we solve the transportation problem to give a feasible solution to (1). The objective of (1), the initial upper bound, is found by simply adding the transportation cost found above and the fixed costs of the P_L plants.

The improved feasible solutions are found by solving the transportation problem with the set of open plants being given by $y_j^* = 1$ in the Lagrangian procedure. The objective of this feasible solution is found by adding the transportation cost and the fixed costs of plants with $\bar{y}_j^* = 1$. Then, the upper bound

Z^{UB} is updated if necessary. If the set y_j^* has already been found in an earlier Lagrangian iteration, then it will be a waste of effort to compute the upper bound again. In order to eliminate such repetitions, we have a procedure that stores upto five different sets of y_j^* for which we have found a feasible solution already and if the newly found y_j^* is unique (compared to the five sets), then we resort to the upper bound procedure. The set is now updated with the newly found y_j^* by eliminating the oldest from the existing list. We have found that this procedure has significantly saved the total computation time.

A Reduction Test

In this section, we give a reduction test to close the gap between P_L and P_U. This test is given in Christofides and Beasley[6]. We can see that closing the gap between P_L and P_U will strenghthen the lower bound. The dual solution value obtained when we enforce the condition that there be exactly K open plants in the optimal solution, is given by

$$\sum_i u_i + \text{the K smallest } f_j', \ j \quad J. \qquad (14)$$

If the value of (14) exceeds the upper bound Z^{UB}, when $K = P_L$ ($K = P_U$), we can increase P_L (decrease P_U) by one without affecting the optimal solution to the original problem.

The Subgradient Procedure

The subgradient procedure for solving the problem (1) is described below.

Step 1: Solve the transportation problem with P_L open plants, (as found in the section on upper bounds), and all the customers. Adding their fixed costs to the transportation cost, we get the initial value of the upper bound Z^{UB}. Initialize the Lagrange multipliers $u_i = \min_j c_{ij}$. Go to step 2.

Step 2: Solve the continuous knapsack $Z_D(u;j)$ for a given u for each j, and compute f_j'. Then, solve the knapsack problem KP_2 to obtain the set of plants to be opened to meet all the demand. The objective of KP_2 gives us a lower bound Z_{LB}. Update the lower bound if necessary. Go to step 3.

Step 3: If needed, solve the transportation problem with the set of open plants identified in step 2. This along with the fixed costs of the open plants, gives us an upper bound Z^{UB}. Update the upper bound if necessary. If $Z^{UB} = Z_{LB}$ stop; we have an optimal solution. If the iteration count is exceeded, stop. If the lower bound converges to a particular value, stop. Else, go to step 4.

Step 4: Update the Lagrange multipliers u_i using the subgradient approach. If all subgradients are zero, stop. Else, go to step 2.

The subgradients for u_i are computed as follows. Let x^*_{ij} and y^*_j be the optimal solutions to the Lagrangian problem. Then the subgradients NU(i) for u_i are

$$NU(i) = \sum_j x^*_{ij} - 1$$

where j belongs to the set of open plants.

We then update the Lagrange multipliers as follows

$$u_i^{k+1} = u_i^k + t_k NU(i)$$

where

$$t_k = (Z^{UB} - Z_{LB})/Norm.$$

We start with an initial value of 0.6 and halve the value every twelve iterations. We terminate the procedure after 200 iterations. The transportation problems are solved by using the code developed by Srinivasan and Thompson[7]. Some computational results for this approach are provided in the next section.

Results

The procedure was coded in FORTRAN77 and run on a DEC2060 time-sharing system at C-MU. We tested this procedure on two different problem sets. The first problem set contained test problems described in Kim and Guignard[8] with some additional side constraints. The second problem set consisted of randomly generated problems. The parameters for the randomly generated problems were fixed as follows. The demands were generated from a uniform distribution in the interval [5, 35). The capacities were generated from a uniform distribution in the interval [10, 160). The fixed costs were generated using

$$f_j = U[0, 90) + U[100, 110)\ s_j$$

where f_j is the fixed cost of plant j, s_j is the capacity of plant j, and U[a, b) stands for a value from the uniform distribution picked from the interval [a, b). The transportation costs were computed by generating points in a unit square, computing the Euclidean distance between them and multiplying them by 10. The way we compute the transportation costs ensures that we are solving geometric problems and therefore the duality gaps we compute as given in this section are not subject to the well known scaling effects. The tightness of the capacity constraints was such that $\sum_j s_j = 2 * \sum_i d_i$. In addition to those basic parameters, we also included some side constraints representing (7).

We provide a summary of results that we obtained on those problem sets. In Table 1 we present the results for problem set 1. In Table 2 we present the results for the randomly generated problems which were solved on the DEC 2060 system at C-MU.

The duality gap referred to in the following tables was computed as described below.

$$\text{Duality gap} = (Z^{UB} - Z_{LB}) / Z^{UB} * 100.$$

As we see from Tables 1 and 2 this Lagrangian heuristic for the problem is very good. In problem set 1, none of the problems had a duality gap of more than 1%. We found 0% duality gaps in three of the seven problems. Also, as we had shown using a branch and bound algorithm for the CPLP, the upper bounds found in these problems are infact optimal. The optimality of problem one, which has a different optimal value compared to CPLP, was verified by hand computation. We have a duality gap of less than 1% in all but one problem in the problem set that contained randomly generated problems.

Results on PC/AT

A code has been written in Microsoft FORTRAN to run on the PC. Table 3 provides the result for some randomly generated problems that were solved on an IBM PC/AT with a 80286 coprocessor at the Indian Institute of Management, Ahmedabad. We see that even with a 80286 coprocessor the computation times are very small. These times can very easily be improved upon by using some of the latest coprocessors which are much faster.

Conclusion

We conclude this paper by observing that we have a very efficient heuristic based on a Lagrangian relaxation for solving CPLP with side constraints. Beasley[9] also shows that Lagrangian heuristics are efficient for different types of location problems. As we pointed out earlier in this paper, this extension of CPLP is useful in solving Vehicle Routing Problems, and also in solving a decision problem that involves the choice of picking a plant of a particular size, from a given set of alternatives, at a location.

Problem	1	2	3	4	5	6	7
Size cust.X plts.	4x5	6x5	8x5	10x10	10x10	15x10	35x20
Optimal solution	434	2622	47600	3108	3425	6127	30484
Lower bound	431	2611	47600	3108	3425	6121	30216
Duality gap (%)	0.69	0.42	0.00	0.00	0.00	0.10	0.08
CPU time (secs)	0.342	0.706	0.053	0.319	0.379	2.779	7.314

Note:All times in Dec-2060 timesharing system at Carnegie-Mellon.

Table 1: Results for problem set 1

Problem	1	2	3	4	5	6
Size cust. X plts.	25x25	25x25	25x25	50x50	50x50	50x50
Lower bound	7028	7482	7587	14741	14515	14461
Upper bound	7129	7501	7632	14794	14625	14536
Duality gap (%)	1.42	0.25	0.59	0.36	0.68	0.52
CPU time (secs)	24.60	22.68	21.51	96.07	94.09	96.65

Note:All times in DEC-2060 timesharing system at Carnegie-Mellon.

Table 2: Results for Random Problems

Problem No	1	2	3	4	5
Size: 25 X 8					
Lower bound	5345	4528	5741	5658	5410
Upper bound	5539	4584	6064	5967	5589
Duality gap (%)	3.52	1.22	5.33	5.18	3.20
CPU time (seconds)	94	92	92	99	98
Size: 25 X 16					
Lower bound	6550	6645	7598	6270	6549
Upper bound	6688	6681	7874	6363	6608
Duality gap	2.06	0.54	3.51	1.46	0.89
CPU time	177	169	170	173	173
Size: 25 X 25					
Lower bound	7847	7291	7256	7730	7426
Upper bound	7934	7367	7310	7856	7579
Duality gap	1.10	1.03	0.74	1.60	2.02
CPU time	253	260	248	261	260
Size: 50 X 16					
Lower bound	9649	9388	8994	10121	9403
Upper bound	9822	9621	9053	10236	9679
Duality gap	1.76	2.42	0.65	1.12	2.85
CPU time	337	330	323	332	334
Size: 50 X 33					
Lower bound	11534	11730	11446	11382	12682
Upper bound	11697	11860	11601	11571	12724
Duality gap	1.39	1.10	1.34	1.63	0.33
CPU time	670	682	678	679	689
Size: 50 X 50					
Lower bound	14117	14076	16355	14943	13924
Upper bound	14301	15276	16938	15115	14035
Duality gap	1.29	7.86	3.44	1.14	0.79
CPU time	1024	1011	1011	1026	1031

Note: All times are in seconds on an IBM PC/AT with processing done with a 80286 coprocessor.

Table 3: Results for randomly generated problems.

REFERENCES

1. Cornuejols, G., R. Sridharan and J. M. Thizy. (1989), "A comparison of heuristics and relaxations for the capacitated plant location problem", under revision with European Journal of Operational Research.

2. Magnanti, T. L. and Wong, R. T. (1984), "Network design and transportation planning", Transportation Science, 18.

3. Wong, T. L. (1983), "Recent research on location and network design problems: an annotated bibliography", Proceedings of the summer school on combinatorial optimisation, National Institute of Higher Education, Dublin, 4-15.

4. Salkin, H. M. (1975), **Integer Programming**, Addison-Wesley Publishing Co., Reading, MA, USA.

5. Francis, R. L. and Goldstein, J. M. (1974), "Location Theory: A Selective Bibliography", Operations Research, 22, 400-410.

6. Christofides, N. and Beasley, J. E. (1983), "An algorithm for the capacitated plant location problem", European Journal of Operational Research, 12, 19-28.

7. Srinivasan, V. and Thompson, G. L. (1972), "An operator theory of parametric programming for the transportation problem-I & II", Naval Research Logistics Quarterly, 19, 205-252.

8. Kim, S. and Guignard, M. (1983), A strong Lagrangian relaxation for capacitated plant location problems. W. P. No. 56, Wharton School, University of Pennsylvania.

9. Beasley, J. E. (1988), Lagrangean heuristics for location problems. Working Paper. Imperial College, London.