Implementing Lagrangean Decomposition Technique to Acquire an Adequate Lower Bound on the Facility Location Problem Solution

Eiman Jadaan Alenezy, Rehab F. Khalaf
Department of Mathematics, College of Basic Education Public Authority for Applied Education and Training, Kuwait City, Kuwait
Email: *rihabfadil@gmail.com

Received November 10, 2012; revised January 2, 2013; accepted January 9, 2013

ABSTRACT

In this work, the Lagrangean Relaxation method has been discussed to solve different sizes of capacitated facility location problem (CFLP). A good lower bound has been achieved on the solution of the CFLP considered in this paper. This lower bound has been improved by using the Volume algorithm. The methods of setting two important parameters in heuristic have been given. The approaches used to gain the lower bound have been explained. The results of this work have been compared with the known results given by Beasley.

Keywords: Capacitated Facility Location Problem; Lagrangean Relaxation Technique; Volume Algorithm; Randomised Rounding Technique; Unit Cost Technique

1. Introduction

Due to its wide applications, capacitated facility location problem (CFLP) has been studied by several authors including Holmberg et al. [1], Melkote & Daskin [2], Ahuja et al. [3], Cortinhal & Captivo [4], Contreras & Diaz [5], and Leitner, M., Raidl, G.R. [6]. A much known goal in CFLP research area is to minimize the overall cost associated to a specific way of opening up facilities and serving customers.

The Lagrangean Decomposition technique has been used widely since Held and Karp [7] introduced it. The basic idea of Lagrangean Relaxation or Lagrangean Decomposition is relaxing some constrains in order to eliminate their effect. The motivation for the relaxing of these constrains is that many combinatorial optimization problems consist of an easy problem with some additional complicating constraints. So, relaxing these complicating constrains makes the problem much easier to solve.

This work implements the Lagrangean Decomposition technique to obtain a good lower bound, then the Volume algorithm used to improve this lower bound. Also the Unite Cost Technique is used, then after 50 passes, we switch to use the Randomised Rounding Technique. Moreover, the Randomised Rounding Technique and the Unit Cost Technique have been implemented to solve the facility location problem considered in this paper.

The rest of this paper is organized as the following: the notations, definitions and abbreviation used in this paper are listed in Section 2. In Section 3, the mathematical formulation of location problem is described. A clear explanation of the Lagrangean Relaxation and Lagrangean Decomposition are given in Sections 4 and 5 respectively. The steps of Volume algorithm have been given in Section 6. The approaches of computing the lower bound and the upper bound are given in Section 7. In Section 8, clear descriptions for the Randomised Rounding Technique and the Unit cost technique are given. In Section 9, the tables of the results and comparison are presented. The conclusions are given in Section 10.

2. Notations and Definitions

In the rest of this paper, the following list of notations, definitions and abbreviation is considered:
Notations | Definitions
--- | ---
\(F\) | A set of potential facility locations.
\(D\) | A set of customers or demand points.
\(d_j\) | The demand of customer \(j \in D\) where \((d_j \geq 0)\).
\(c_i\) | The unit cost of supplying the demand of customer \(j \in D\) from facility \(i \in F\) where \((c_i \geq 0)\).
\(S_i\) | The capacity of facility \(i\) (the upper limit on the total demand that can be supplied from facility \(i\) where \((S_i \geq 0)\)).
\(P\) | The desired number of open facilities.
\(f_i\) | The fixed cost associated with opening facility \(i\) where \((f_i \geq 0)\).
\(x_{ij}\) | The fraction of the demand of customer \(j\) supplied from facility \(i\) where \((0 \leq x_{ij} \leq 1)\).
\(y_i\) | \(y_i = 1\) if facility \(i\) is open, \(0\) otherwise.
\(IP\) | mixed-integer programming problem.

3. Mathematical Formulation of Location Problem

The formulation of the CFLP as a mixed-integer programming problem called \((IP)\) is given in Equations/Inequalities (1) to (7)

\[
\text{Minimise } \sum_{j \in D} \sum_{i \in F} d_j c_{ij} x_{ij} + \sum_{i \in F} f_i y_i \quad (1)
\]

where this is the objective equation

Subject to \(\sum_{i \in F} x_{ij} = 1, \forall j \in D\) \(\quad (2)\)

This equation ensure that the demand of each customer is satisfied

\[
\sum_{j \in D} d_j x_{ij} \leq S_i y_i, \forall i \in F \quad (3)
\]

\[
x_{ij} \leq y_i, \forall i \in F, j \in D \quad (4)
\]

These two inequalities ensure that the closed facility does not supply any customer and that the demand supplied from facility does not exceed the capacity of the facility.

\[
y_i \in \{0,1\}, \forall i \in F, \quad (5)
\]

This is the integrality constraint.

\[
p \leq \sum_{i \in F} y_i \leq p + 2 \quad (6)
\]

This inequality ensure that the number of open facilities lies between \(p\) and \(p + 2\), where these constrains were discussed in previous work of the first author, see Alenezy E. [8].

\[
x_{ij} \geq 0, \forall i \in F, j \in D \quad (7)
\]

The last inequality provide bounds on the allocation variables \(x_{ij}\).

4. Lagrangean Relaxation Technique (LR)

In order to develop a Lagrangean heuristic for the CFLP mentioned in the previous section, a linear programming relaxation for the \((IP)\) problem should be considered, so the same formulation \((IP)\) has to be used except replacing inequality (5) by:

\[
y_i \leq 1, \forall i \in F, \quad (8)
\]

The LP-relaxation will be denoted by \((P)\).

In this section, the Lagrangean relaxation is considered for the problem \((P)\). Then it is described how to use the Volume algorithm, which is an extension to the subgradient Optimisation, Held et al. [9].

Investigating the solution of large sized problems in order to solve the CFLP required decomposing it into \(m\) independent problems, which are easier to solve. So we need to relax the most complicated constraint for decomposition, then we relax Equation (2).

Let \(u_j\) be the dual multiplier from Inequation (2), and let \(\overline{c}_{ij} = d_j c_{ij} - u_j\). Then a lower bound \(L(u)\) is given by solving the following problem:

\[
\text{Minimise } \sum_{j \in D} \sum_{i \in F} \overline{c}_{ij} x_{ij} + \sum_{i \in F} f_i y_i \quad (9)
\]

subject to \(\sum_{j \in D} d_j x_{ij} \leq S_i y_i, \forall i \in F, \quad (10)\)

\[
x_{ij} \leq y_i, \forall i \in F, j \in D, \quad (11)
\]

\[
y_i \leq 1, \forall i \in F, \quad (12)
\]

\[
p \leq \sum_{i \in F} y_i \leq p + 2, \quad (13)
\]

\[
x_{ij} \geq 0, \forall i \in F, j \in D, \quad (14)
\]

5. Lagrangean Decomposition Technique (LD)

From the literature review, it is reported that solving the above \(L(u)\), provide an adequate lower bound on the integer optimum solution. This is improved by using the Volume algorithm.

Also from the literature review it is a fact that solving \(L(u)\) for large size problems is difficult. To compute a lower bound (LB), we relax constraint (13) and decompose the \(L(u)\) problem into \(m\) independent problems for each \(i \in F\). The generic form of each subproblem is:
Minimise \( f(y) + \sum c_j x_j \)
subject to \( \sum_j d_j x_j \leq s t \),
\( x_j \leq y, \ j \in D \),
\( 0 \leq y \leq 1 \),
\( x \geq 0 \).

Solving this to get a LB is easy. First, set any variable \( x_j \) with \( c_j > 0 \), to 0. Then we assume that the rest of the variables are ordered such that:

\[ \frac{c_1}{d_1} \leq \frac{c_2}{d_2} \leq \cdots \leq \frac{c_n}{d_n} \]

where \( n' \subseteq n \) and \( n = |D| \)

Now, let \( k \) be the largest index such that

\[ \sum_{j=1}^k d_j \leq S, \]

and let

\[ b(k) = \sum_{j=1}^k d_j \text{ and } r = \frac{S - b(k)}{d_{k+1}}. \]

If \( f + \sum_{j=1}^k c_j + c_{k+1}r \geq 0 \), then we set \( y = 0 \) and \( x_j = 0 \) for all \( j \), otherwise, we set

\[ y = 1, \ x = 1, \ 1 \leq j \leq k, \text{ and } x_{k+1} = r. \]

Having solved these \( m \) independent subproblems, next, consider the minimum number of facilities that are needed to supply all the demand. In this way the LB enhanced by comparing the number of open facilities denoted by \( h \), after having solved the \( m \) subproblems, to the minimum needed \( p \). If \( h < p \) then we sort the unopened facilities by their fixed costs and open the cheapest facilities until we have \( p \) opened. The LB is suitably increased to account for theses extra fixed costs.

6. The Volume Algorithm

The Volume algorithm has been used to improve the lower bound obtained from solving the decomposition of the problem in the previous section. The volume algorithm developed by Barhommna and Anbil [10], and it can be described by the following steps:

Step 1: Start with a vector \( \nu \) and solve Equations/Inequalities (9) to (14) to obtain \((\bar{x}, \bar{y})\) and \( L(\nu) \) set \( t = 1 \).

Step 2: Compute \( v' \), where \( v'_j = 1 - \sum_i \bar{y}_j \), and \( u' = \nu + sv' \).

The formula of the step size \( s \) given by

\[ s = \lambda \frac{UB - L(\nu)}{\|v'\|} \]

where this formula is used in the Subgradient method, see (Serra D. [11]).

\( \lambda \) is a number between 0 and 2. In order to set its value, we will define three types of iterations:

1) Interation E: which is the iteration with no improvement on the lower bound. A sequence of E iterations requires the need for a smaller step size. Therefore, after a sequence of 20 E iterations, we multiply \( \lambda \) by 0.66.

2) Interation Y: if \( L(u') > L(\nu) \), we compute \( w_j = 1 - \sum_i x'_j \) for all \( j \), and \( d = v' \cdot w \). If \( d < 0 \) this means that a larger step size would have a smaller value for \( L(u') \).

3) Interation T: if \( d \geq 0 \), this iteration suggests the need for a large step size, so we multiply \( \lambda \) by 1.1.

Now solve (9) to (14) with \( u' \). Let \((x', y')\) be the solution obtained. Then \((\bar{x}, \bar{y})\) is updated as

\[ (\bar{x}, \bar{y}) = \alpha (x', y') + (1-\alpha)(\bar{x}, \bar{y}) \]

where \( \alpha \) is a number between 0 and 1. In order to find the value of \( \alpha \), we solve the following one dimensional problem:

Minimise \( \|\alpha w + (1-\alpha)v'\| \)
Subject to \( \frac{b}{10} \leq \alpha \leq b. \)

Originally \( b \) is set to 0.1, and then every 100 iterations we check if \( L(u') \) had not increased by at least 1%, in which case we divide \( b \) by 2, otherwise we keep it as it is. When \( b \) becomes less than \( 10^{-5} \), we keep it constant at this value.

Step 3: Update \( \nu \) as \( \nu = u' \) only if \( L(u') > L(\nu) \).

Step 4: Stopping criteria
1) \( v'_j < 0.02 \).
2) \( \frac{UB - L(u')}{UB} \leq 0.02. \)

3) The number of passes = 200.
4) If stopping rules are not satisfied, then set \( t = t + 1 \) and go to Step 2.

7. Computing the Lower Bound (LB) and the Upper Bound (UB)

7.1. Computing the Lower Bound (LB)

There are two approaches to solve the \( L(u) \) formulation of the problem given in Section 4. Following are the summaries of these two approaches:

The first approach summarized by decomposing \( L(u) \) in to \( m \) independent subproblems. Solve them as explained in Section 5 and update the obtained LB by using the Volume algorithm explained in Section 6, we set the vector \( \nu \) to values that is denoted by “Warm Start
Duals”. The Warm Start Duals are the values of the duals of the relaxed constraints (1) and (2) obtained from solving the greedy Weak Representation of the CFLP as LP solution see Alenezy E., et al. [8].

The second approach summarized by removing constraint (11) from the $L(u)$ formulation to reduce the size of the problem to make it possible to solve. Then solve it without the decomposition technique above, to obtain a LB. To improve this LB again, we apply Volume algorithm with the same two ways discussed in the first approach.

7.2. Computing the Upper Bound (UB)

To compute the upper bound, first remove the constraints (4). Solve the (P) formulation for a LP solution. Then use a technique called randomises rounding with new technique called the Unit Cost Technique, these new techniques are explained in Section 8. To treat the fractional solutions of $y$ as a probability distribution, and keep opening them randomly to get enough capacity, we keep updating this UB using the Volume algorithm every 50 passes, until we meet one of the stopping criteria. We choose the passes to be 50, because the UB changes slowly, and from our experiments 50 passes usually shows some improvement in the UB.

8. The Randomised Rounding Technique and the Unit Cost Technique

To describe the Randomised Rounding Technique, let $(x^\ast, y^\ast)$ be the optimal solution to the linear programming relaxation (P) after removing constraints (4). We open facility at location $i \in F$ with probability $y_i^\ast$ independently. It is possible that there is not enough total capacity to service all the demand. If this is the case, then we repeat the random experiment and try again. Once we determine which facilities to open, the cheapest of clients can be easily found by solving the resulting transportation problem.

The process of this technique can be summaries as the following:

1) Check if opening all fractional $y^\ast$’s leads to enough capacity. If not, then open all the $y$’s, and then open the other $y^\ast$’s randomly until enough capacity is obtained.

2) If the fractional $y^\ast$’s providing enough capacity, we normalise the fractional $y^\ast$, so that the sum to 1. Then using this as a probability distribution of the $y^\ast$’s. We use the inverse transformation method to select facilities to open until we have enough capacity.

In the Unit Cost Technique, all the $y$’s sorted according to their unit cost, which is the value of (fixed cost/capacity) for each unopened facility. We start opening the facility with the cheapest cost until we reach the point where the total capacity exceeds the total demand.

In this situation, we sort the remaining facilities in ascending order of the fixed costs and open the first facility in this order list that provides enough capacity to cover the outstanding demand.

9. Problems and Their Tables

The Lagrangean Relaxation implemented in FortRAN, with FortMP as a solver when applicable. The performance of the Lagrangean Relaxation and Volume algorithm have been tested on different problems by using alternative benchmark models.

The results of our experiments are given in the following tables. The column “Passes” presented the number of time we go through the algorithm looking for better Lower bound and upper bound. The last row corresponds to the error percentage value which shows the percentage gap between the lower and the upper bound where

$$\text{Error\%} = \frac{\text{UB} - \text{LB}}{\text{UB}}$$

In Table 1, we use eight instances generated as in the work of Barahona and Anbil [10], and the work of Cornwall [12].

In Table 2, Beasly’s test problem has been considered and solved using the method of this work. Comparing our solutions with Beasly’s, we notice that we were able to obtain a very close solution of Beasley’s where the reason is using different machines to get the solutions see (Beasley [13]). The gap between our solutions and Beasly’s is less than 1%.

10. Conclusion

A good lower bound has been achieved on the solution of the Capacitated Facility Location Problem considered in

Table 1. The results of solving different size of CFLB problem using Lagrangean Relaxation Technique and Volume algorithm.

<table>
<thead>
<tr>
<th>Problem size $m \times n$</th>
<th>LB</th>
<th>UB</th>
<th>Time</th>
<th>Passes</th>
<th>Error %</th>
</tr>
</thead>
<tbody>
<tr>
<td>100 $\times$ 100</td>
<td>11088.854</td>
<td>11433.22</td>
<td>4.61</td>
<td>482</td>
<td>3.03</td>
</tr>
<tr>
<td>200 $\times$ 200</td>
<td>19542.08</td>
<td>19919.05</td>
<td>116.26</td>
<td>1400</td>
<td>1.89</td>
</tr>
<tr>
<td>300 $\times$ 300</td>
<td>28351.01</td>
<td>28933.05</td>
<td>935.91</td>
<td>1150</td>
<td>2.01</td>
</tr>
<tr>
<td>400 $\times$ 400</td>
<td>36935.04</td>
<td>37822.05</td>
<td>487.29</td>
<td>625</td>
<td>2.34</td>
</tr>
<tr>
<td>500 $\times$ 500</td>
<td>45409.29</td>
<td>45644.58</td>
<td>1902.88</td>
<td>1000</td>
<td>0.50</td>
</tr>
<tr>
<td>600 $\times$ 600</td>
<td>53756.414</td>
<td>54285.848</td>
<td>4133.19</td>
<td>601</td>
<td>0.96</td>
</tr>
<tr>
<td>700 $\times$ 700</td>
<td>61156.194</td>
<td>62048.793</td>
<td>5058.79</td>
<td>601</td>
<td>1.4</td>
</tr>
<tr>
<td>800 $\times$ 800</td>
<td>68764.846</td>
<td>69049.99</td>
<td>8601.67</td>
<td>602</td>
<td>0.413</td>
</tr>
<tr>
<td>900 $\times$ 900</td>
<td>76511.347</td>
<td>77028.705</td>
<td>9715.03</td>
<td>601</td>
<td>0.67</td>
</tr>
</tbody>
</table>
Table 2. Comparing the results obtain using Lagrangean Relaxation Technique and Volume algorithm with Beasley results.

<table>
<thead>
<tr>
<th>Beasley data file</th>
<th>LB</th>
<th>UB</th>
<th>time</th>
<th>passes</th>
<th>Error %</th>
</tr>
</thead>
<tbody>
<tr>
<td>Capacity 5000</td>
<td>LD&amp;AV</td>
<td>13656379.575</td>
<td>13656379.578</td>
<td>130.17</td>
<td>350</td>
</tr>
<tr>
<td></td>
<td>Beasley sol.</td>
<td>-</td>
<td>13656379.578</td>
<td>74.2</td>
<td>961</td>
</tr>
<tr>
<td>Capacity 6000</td>
<td>LD&amp;AV</td>
<td>13323103.902</td>
<td>13434245.092</td>
<td>289.63</td>
<td>993</td>
</tr>
<tr>
<td></td>
<td>Beasley sol.</td>
<td>-</td>
<td>13361927.449</td>
<td>321.6</td>
<td>695</td>
</tr>
<tr>
<td>Capacity 7000</td>
<td>LD&amp;AV</td>
<td>13156282.077</td>
<td>13340947.302</td>
<td>319.98</td>
<td>1234</td>
</tr>
<tr>
<td></td>
<td>Beasley sol.</td>
<td>-</td>
<td>13198556.434</td>
<td>244.9</td>
<td>613</td>
</tr>
<tr>
<td>Capacity 8000</td>
<td>LD&amp;AV</td>
<td>13068650.565</td>
<td>13082516.496</td>
<td>393.48</td>
<td>1528</td>
</tr>
<tr>
<td></td>
<td>Beasley sol.</td>
<td>-</td>
<td>13082516.496</td>
<td>89.5</td>
<td>1546</td>
</tr>
<tr>
<td>Capacity 5000</td>
<td>LD&amp;AV</td>
<td>11628013.452</td>
<td>11646596.974</td>
<td>682.90</td>
<td>1551</td>
</tr>
<tr>
<td></td>
<td>Beasley sol.</td>
<td>-</td>
<td>11646596.974</td>
<td>150.6</td>
<td>737</td>
</tr>
<tr>
<td>Capacity 5750</td>
<td>LD&amp;AV</td>
<td>11521183.12</td>
<td>11603865.648</td>
<td>791.33</td>
<td>1500</td>
</tr>
<tr>
<td></td>
<td>Beasley sol.</td>
<td>-</td>
<td>11570340.289</td>
<td>294.7</td>
<td>669</td>
</tr>
<tr>
<td>Capacity 6500</td>
<td>LD&amp;AV</td>
<td>11507989.519</td>
<td>11604762.0</td>
<td>564.41</td>
<td>1070</td>
</tr>
<tr>
<td></td>
<td>Beasley sol.</td>
<td>-</td>
<td>11518743.744</td>
<td>108.7</td>
<td>7.1</td>
</tr>
<tr>
<td>Capacity 7250</td>
<td>LD&amp;AV</td>
<td>11500107.09</td>
<td>11509365.405</td>
<td>536.94</td>
<td>1154</td>
</tr>
<tr>
<td></td>
<td>Beasley sol.</td>
<td>-</td>
<td>11505767.394</td>
<td>87.70</td>
<td>795</td>
</tr>
</tbody>
</table>

this work. The Volume algorithm has been used to improve the lower bound.

REFERENCES