The p-Median Problem with Concave Costs

Chuan Xu[1], Abdel Lisser[1], Janny Leung[2] and Marc Letournel[1]
[1] *Laboratoire de Recherche en Informatique(LRI),Bât 650, Université Paris-Sud, 91405, Orsay Cedex, France*
[2] *Department of Systems Engineering and Engineering Management, The Chinese University of Hong Kong, Hong Kong, Hong Kong*

Keywords: Nonlinear Integer Programming, p-Median, Concave Cost, Linearization, VNS.

Abstract: In this paper, we propose a capacitated p-median problem with concave costs, in which the global cost incurred for each established facility is a concave function of the quantity q delivered by this facility. We use DICOPT to solve this concave model. And then we transform this model into a linear programming problem and solve it using the commercial solver CPLEX. We also use the metaheuristic Variable Neighbourhood Search (VNS) to solve this problem. Computational results show that our linearization method helps to improve the calculations of the concave model. With VNS, we solve large size instances with up to 1500 facilities within a reasonable CPU time.

1 INTRODUCTION

The p-median problem has been widely studied in the literature during the last decades especially its linear version. In this paper, we study and solve large size instances using particular linearization of the concave function in order to solve the problem with LP software packages. We also study an efficient implementation of VNS to solve large size instances.

Given a set of n facility sites and m demand customers, the p-median problem (PMP) selects exactly p sites at minimum distribution costs between the customers and their open facilities. The PMP was formulated as a zero-one programming problem in 1970 (ReVelle and Swain, 1970). Later, it was proven to be a NP-hard (Kariv and Hakimi, 1979). The capacitated p-median problem (CPMP) is an extension of the PMP which considers capacities for the service to be given by each site, i.e., the total service demanded by customers cannot exceed its service capacities. It also arises in other contexts like those of vehicle routing, network design, political districting.

To solve large *PMP*, Avella et al. (2007) presented a Branch-Cut-Price algorithm. In (Beltran et al., 2006), authors introduced a semi-Lagrangian relaxation to generate lower bounds. Several approximation methods were proposed (Charikar and Guha, 1999; Jain and Vazirani, 2001). Considering to solve *CPMP*, heuristics and metaheuristics are the predominant techniques. Ceselli and Righini (2005) proposed a branch and price algorithm. Fleszar and Hindi (2008) applied an effective VNS to CPMP. Others like scatter search heuristic (Xu et al., 2010), genetic algorithm (Stanimirović, 2008) were also applied to solve this problem. Recently, some hybrid heuristics approaches were developed to enhance the performance like (Landa-Torres et al., 2012; Yaghini et al., 2013).

Both the CPMP and the PMP aforementioned can be formulated as linear integer programming problems. In our model, the CPMP problem with concave cost means that the distribution cost of each facility site depends on the total quantity delivered by the site. The unit distribution cost decreases with the increased quantity of output or demand generating economy of scales. This variant is in the class of nonlinear location problem. The objective function of our model is the same as the model proposed by Dupont (2008). In his work, he studied a concave facility location problem without capacity constraint and solved it with a branch and bound algorithm. There are also many other models using concave functions, see for instances (Nagy and Salhi, 2007; Ghoseiri and Ghannadpour, 2007). Sun and Gu (2002) proposed a network design problem where the function of delivery cost is concave. Their method consists in omitting the nonlinear factor and adjusting the solution to obtain an overall approximated optimal solution. Verter and Dasci (2002) solved the uncapacitated plant location and flexible technology acquisition problem with the monotone increasing concave function. They

Xu C., Lisser A., Leung J. and Letournel M..
The p-Median Problem with Concave Costs.
DOI: 10.5220/0004832502050212
In *Proceedings of the 3rd International Conference on Operations Research and Enterprise Systems* (ICORES-2014), pages 205-212
ISBN: 978-989-758-017-8

adopted the progressive piecewise linear underestimation (PPLU) technique (Verter and Dincer, 1995) to simplify the model.

In our case, we adopt PPLU method to linearize the objective function. Then, since the model is converted into an integer program, we apply the VNS to solve it.

VNS was first proposed by Hansen and Mladenović (1997) and rapidly developed since then. It has been applied to the design problems in communication, location problems, data mining, knapsack and packing problems. For more details on VNS and its main applications to p-median problem, see (Hansen et al., 2010; Fathali and Kakhki, 2006; Osman and Ahmadi, 2006; Hansen et al., 2009).

This paper is structured as follows. Section 2 introduces a mathematical formulation of the problem. Section 3 presents solution methods. Section 4 shows and discusses the computational results. We conclude the paper in Section 5.

2 MATHEMATICAL FORMULATION

Our model is a capacitated p-median problem with concave distribution costs. We assume that our distribution costs are concave functions of the quantity q delivered by the facility site which is a situation in economies of scale. The main difference between our model and the standard model of *CPMP* is that, instead of minimizing the distribution costs related to the distances between sites and customers, we studied an objective function of distribution costs related to q and introduced a distribution range R for each facility site to constrain distances between sites and customers.

In practice, distribution costs are not always dependent on the distance between sites and customers but on the total quantity delivered by the sites (Dupont, 2008). For instance, if we outsource our transportation service to a third party logistics ($3PL$), the cost will be related to the quantity that we ask $3PL$ to deliver. That's the linear part $B_i \cdot q_i$ included in our objective function. And the more we delivered, the less we pay for the unit distribution cost. The square root of q_i in our objective function follows this rule.

The aim of our model is to select a subset of potential sites to satisfy all the customers requirements at minimum distribution costs.

We use the following notation in our model:
Let $I = \{1,...,n\}$ be a set of sites, $J = \{1,...,m\}$ be a set of customers. $I(j)$ shows the set of sites that can deliver customer j. $J(i)$ shows the set of customers that can be delivered by site i. $I(j), J(i)$ can be obtained by comparing the distance between site i and customer j with their distribution range R_i, R_j. d_j is the demand of customer j. p is the median parameter. δ_i presents the maximum capacity of site i. $F_i(q_i)$ presents the concave function for each site i which has A_i, B_i, C_i as coefficients.

We consider three decision variables:

z_{ij} : the quantity delivered by site i to customer j,

q_i : the quantity delivered by site i,

y_i : if site i is open $y_i = 1$ otherwise $y_i = 0$

The p-median concave costs problem can be written as follows:

$$\min \sum_{i=0}^{i=n} F_i(q_i) = \sum_{i=0}^{i=n} (A_i + B_i \cdot q_i + C_i \sqrt{q_i}) \cdot y_i \qquad (1)$$

$$\sum_{j \in J(i)} z_{ij} = q_i \qquad \forall i \in I, \qquad (2)$$

$$\sum_{i \in I(j)} z_{ij} = d_j \qquad \forall j \in J, \qquad (3)$$

$$\sum_{i \in I} y_i = p, \qquad (4)$$

$$q_i \leq \delta_i y_i \quad \forall i \in I, \qquad (5)$$

$$y_i \in \{0,1\} \qquad \forall i \in I, \qquad (6)$$

$$q_i \geq 0 \qquad \forall i \in I,$$

$$z_{ij} \geq 0 \qquad \forall i \in I, j \in J.$$

The objective function (1) minimizes the concave distribution costs. Constraint (2) shows that the quantity delivered by site i is the sum of all the customers' demands for this site. Constraint (3) shows that every customer's request must be satisfied. Constraint (4) presents the number of sites to open. Constraint (5) gives the capacity of each site. Constraint (6) defines variables domain.

3 SOLUTION METHODS

We use two different methods to solve the p-median problem (1-6) i.e., a linear program obtained by linearizing the concave function and the metaheuristic VNS.

3.1 Linearization Method

We linearize the function $F_i(q_i) = A_i + B_i \cdot q_i + C_i \sqrt{q_i}$ by dividing the range of each site capacity into small intervals. We assume that the range of the site i is $[\gamma_i, \delta_i]$.

Then, we obtain a set of small intervals i.e., $\{[\gamma_{i_0}, \gamma_{i_1}], [\gamma_{i_1}, \gamma_{i_2}], \ldots, [\gamma_{i_{K-1}}, \gamma_{i_K}]\}$ with $\gamma_{i_0} = \gamma_i, \gamma_{i_K} =$

δi. Therefore, the linear function in the kth interval $[\gamma_{i_{k-1}}, \gamma_{ik}]$ is:

$$f_{i_k}(q_i) = \lambda_{i_k} q_i + \beta_{i_k} \quad . \tag{7}$$

$$\lambda_{i_k} = \frac{F_i(\gamma_{i_k}) - F_i(\gamma_{i_{k-1}})}{\gamma_{i_k} - \gamma_{i_{k-1}}} \quad . \tag{8}$$

$$\beta_{i_k} = \frac{\gamma_{i_k} F_i(\gamma_{i_{k-1}}) - \gamma_{i_k-1} F_i(\gamma_{i_k})}{\gamma_{i_k} - \gamma_{i_{k-1}}} \quad . \tag{9}$$

where K is the number of intervals. If K is large, the approximation of the concave function is sharp.

From the formulas aforementioned, we can obtain the values of λ_{i_k} and β_{i_k} with the boundary of the kth interval and the coefficients A_i, B_i, C_i in the concave function $F_i(q_i)$. In order to figure out the boundary values $\gamma_{i_{k-1}}$ and γ_{i_k}, we need to compute the number of intervals K.

$$q_{i_k} = \left(\frac{C_i}{2(\lambda_{i_k} - B_i)}\right)^2 \quad . \tag{10}$$

$$d_{i_k} = F_i(q_{i_k}) - f_i(q_{i_k}) \quad . \tag{11}$$

$$\varepsilon_{i_k} = \frac{d_{i_k}}{F_i(q_{i_k})} \quad . \tag{12}$$

where q_{i_k} is the point with the greatest gradient in the concave function which also means the farthest point from the linear function; d_{i_k} represents the farthest distance and ε_{i_k} shows the percentage of deviation between the concave function and the linear function. We divide the range of capacity of each site continuously until the value of ε_{i_k} reaches the reference value that we have set before. At the end, we can figure out the value of K. After the linearization, we use binary variable y_{i_k} to check whether the total request for site i is in the k^{th} interval. If so, $y_{i_k} = 1$ and 0 o.w. Once the interval is located, the cost function is built based on suitable values of λ and β. Then, we add binary variables y_{1i_k},y_{2i_k} such that $y_{i_k} = y_{1i_k} \cdot y_{2i_k}$ and $Q_{i_k} = q_i \cdot y_{i_k}$ to perform the linearization of the terms $q_i y_{i_k}$. Those binary variables are defined as follows: $y_{i_k} = 1$ if $q_i \in (\gamma_{i_{k-1}}, \gamma_{i_k}]$, $y_{1i_k} = 1$ if $q_i > \gamma_{i_k-1}$ and $y_{2i_k} = 1$ if $q_i \le \gamma_{i_k}$.

Hence, $F_i(q_i)$ can be written as:

$$F_i(q_i) \approx \sum_{k=1}^{k=K} f_{i_k}(q_i) y_{i_k} = \sum_{k=1}^{k=K} (\lambda_{i_k} q_i + \beta_{i_k}) y_{i_k}, \forall i \in I. \tag{13}$$

Based on the aforementioned linearization, the equivalent linear programming problem is as follows:

$$\min \quad \sum_{i=0}^{i=n} \sum_{k=1}^{k=K} \lambda_{i_k} Q_{i_k} + \beta_{i_k} y_{i_k} \tag{14}$$

$$s.t. \quad \sum_{i \in I(j)} z_{ij} = d_j \qquad \forall j \in J, \tag{15}$$

$$\sum_{j \in J(i)} z_{ij} = q_i \qquad \forall i \in I, \tag{16}$$

$$\sum_{i \in I} y_i = p \qquad \forall i \in I, \tag{17}$$

$$\sum_{k=1}^{k=K} y_{i_k} = y_i \qquad \forall i \in I, \tag{18}$$

$$\gamma_i y_i \le q_i \le \delta_i y_i \qquad \forall i \in I, \tag{19}$$

$$q_i \ge (\gamma_{i_{k-1}} + 1) y_{1i_k} \qquad \forall i \in I, \forall k \in [1,K], \tag{20}$$

$$q_i \le \gamma_{i_{k-1}}(1 - y_{1i_k}) + \delta_i y_{1i_k} \ \forall i \in I, \forall k \in [1,K], \tag{21}$$

$$q_i \ge (\gamma_{i_k} + 1)(1 - y_{2i_k}) \qquad \forall i \in I, \forall k \in [1,K], \tag{22}$$

$$q_i \le \gamma_{i_k} y_{2i_k} + \delta_i (1 - y_{2i_k}) \quad \forall i \in I, \forall k \in [1,K], \tag{23}$$

$$y_{i_k} \le y_{1i_k}, y_{i_k} \le y_{2i_k} \qquad \forall i \in I, \forall k \in [1,K], \tag{24}$$

$$y_{i_k} \ge y_{1i_k} + y_{2i_k} - 1 \qquad \forall i \in I, \forall k \in [1,K], \tag{25}$$

$$Q_{i_k} \le q_i, Q_{i_k} \le \delta_i y_{i_k} \qquad \forall i \in I, \forall k \in [1,K], \tag{26}$$

$$Q_{i_k} \ge q_i - \delta_i (1 - y_{i_k}) \qquad \forall i \in I, \forall k \in [1,K], \tag{27}$$

$$y_i, y_{i_k}, y_{1i_k}, y_{2i_k} \in \{0,1\} \quad \forall i \in I, \forall k \in [1,K], \tag{28}$$

$$z_{ij}, Q_{i_k}, q_i \ge 0 \qquad \forall i \in I, \forall j \in J, \forall k \in [1,K]. \tag{29}$$

There are $(14nK + 5n + nm + m + 1)$ constraints. Constraints (20-27) are linearization constraints.

Proposition 3.1. *The farthest distance between the concave function F_i and the linear function f_i is located in the first interval $[\gamma_{i0}, \gamma_{i1}]$ and its value is d_{i1}.*

Proof. We substitute the part $F_i(q_i)$ with the concave function $F_i(q_i) = A_i + B_i \cdot q_i + C_i \sqrt{q_i}$ in (8) and (9). Then we obtain the values of λ_{i_k} and β_{i_k} as follows:

$$\lambda_{i_k} = B_i + \frac{C_i}{\sqrt{\gamma_{i_k}} + \sqrt{\gamma_{i_{k-1}}}} \quad . \tag{30}$$

$$\beta_{i_k} = A_i + \frac{C_i \sqrt{\gamma_{i_k} \gamma_{i_{k-1}}}}{(\sqrt{\gamma_{i_k}} + \sqrt{\gamma_{i_{k-1}}})} \quad . \tag{31}$$

By using (10), we get another function of λ_{i_k} in terms of q_{i_k}:

$$\lambda_{i_k} = B_i + \frac{C_i}{2} \sqrt{q_{i_k}} \quad . \tag{32}$$

Combining (30) and (32), we obtain

$$q_{i_k} = \frac{(\sqrt{\gamma_{i_k}} + \sqrt{\gamma_{i_{k-1}}})^2}{4} \quad . \tag{33}$$

Equation (33) reveals that the value of q_{i_k} has no relationship with the coefficients A_i, B_i and C_i of the concave function. It is related to the boundary of the interval.

Putting all the formulas aforementioned into (11) which gives the definition of d_{i_k}, we obtain a function of d_{i_k} related to the value of γ_{i_k}. Then, we set $h = \gamma_{i_k} - \gamma_{i_{k-1}}$ where h is a constant if we know the number of considered intervals.

$$d_{i_k}(\gamma_{i_k}) = \frac{C_i h^2}{4(\sqrt{\gamma_{i_k}} + \sqrt{\gamma_{i_k} - h})^3} \quad . \tag{34}$$

The value of distance d_{i_k} is only related to the interval $\left[\gamma_{i_{k-1}}, \gamma_{i_k}\right]$ and the coefficient C_i in the concave function. This formula of d_{i_k} is a decreasing function of the value of γ_{i_k}. Because the part of the function in the numerator is a decreasing function and the part in the denominator is an increasing function with the value of γ_{i_k}. As γ_{i_k} represents the value of boundary for the k^{th} interval, the smallest value of boundary γ_{i_k} is located in the first interval $k = 1$, thus the distance d_{i_1} in this interval is the farthest. In other words, when k increases, the value of d_{i_k} decreases.

If we set $\gamma_i = 0$, which means the minimum capacity for site i is zero, then

$$\gamma_{i_1} = h \tag{35}$$

$$d_{i_1} = \frac{C_i}{4}\sqrt{h} \tag{36}$$

□

Proposition 3.2. *The linear function in the first interval* $[\gamma_{i_0}, \gamma_{i_1}]$ *has the maximum percentage of deviation* ε *between the concave function and the linear function.*

Proof. In the proposition (1), we proved that d_{i_k} is a decreasing function of the value of k. For the part $F_i(q_{i_k})$ in the denominator, we know that with the increase of k, q_{i_k} is larger. And $F_i(q_{i_k})$ is obviously an increasing function with the value of q_{i_k}, because it is a concave one. Thus, the function in the denominator is an increasing function with the value of k. So we can say that ε_{i_k} is a decreasing function with the value of k. When we choose the first interval which means that k is the smallest ($k = 1$), the percentage of deviation is the largest.

We just need to calculate the percentage of deviation in the first interval to decide whether our linearization meets to the reference gap we set before. The function of ε_{i_k} is under the form:

$$\varepsilon_{i_k} = \frac{C_i h^2}{[4A_i + 2C_i(\sqrt{\gamma_{i_k}} + \sqrt{\gamma_{i_k} - h}) + B_i(\sqrt{\gamma_{i_k}} + \sqrt{\gamma_{i_k} - h})^2]} \times \frac{1}{(\sqrt{\gamma_{i_k}} + \sqrt{\gamma_{i_k} - h})^3} \tag{37}$$

If we set $\gamma_i = 0$ and $k = 1$, we get

$$\varepsilon_{i_1} = \frac{C_i\sqrt{h}}{4A_i + 2C_i\sqrt{h} + B_i h} \tag{38}$$

□

3.2 Variable Neighbourhood Search Method

VNS is a recent metaheuristic whose basic idea is to proceed a systematic change of neighbourhood within a local search algorithm (Hansen and Mladenović, 1997). In our case, we initialised our solution by distributing the customers to their nearest sites with respect to their distribution ranges. We suppose at first that each site is responsible for the same number of customers. The detail is shown in the procedure *InitialSolution* below. After this procedure, we obtain our decision vector $y = (y_1, y_2, ..., y_n)$. It follows that the solution space S for the problem contains $\binom{n}{p}$ solutions representing all the combinations of p sites from n candidates. We consider two solutions $x_1, x_2 \in S$, the distance between them is the number of y_i whose values are different. The neighbourhood k called $(N_k(x))$ is a set that contains all solutions at a distance k from a solution x. The largest distance (k_{max}) of our model is set to $min(p, n-p)$.

The main idea in our local search is to switch a customer c from a facility site p_{out} to another facility site p_{in}, while reducing the distribution costs. At the same time, the movement should respect the capacity constraint. In addition, after the movement, we should consider one median problem for the serving area of p_{in}, p_{out} and figure out the most suitable facility sites to be opened in these two areas. The local search stops when there exists no more better movements which improve the value of the objective function.

For each selected site, we calculate the total demand of its customers, called (DN). The algorithm for choosing c, p_{out}, p_{in} at each iteration consists in picking a $move-out$ site p_{out} with a smallest value of DN and a $move-in$ site p_{in} with a largest DN value from all the selected sites. Then we choose a customer c with the largest demand of all the customers delivered by p_{out}. The idea of the algorithm consists in maximizing the quantity delivered by certain sites. As a result, the unit distribution cost decreases. The details of this algorithm are shown in the procedure *OneTimeLocalSearch*.

4 NUMERICAL EXPERIMENTS

We test our models on the instances introduced by Osman and Christofides (1994) which can be down-

Procedure 1: VNS(k_{max}, r).

Input: k_{max}: maximum neighbourhood size; r: repeated times
Output: The best solution S_{best} that we found
$S \leftarrow InitialSolution()$;
for $k = 1$ **to** k_{max} *do repeat r times* **do**
  /* Generate a random solution from $N_k(S)$ */
  $S1 \leftarrow JumpToNeighbor(S, k)$;
  /* Local search */
  $S1 \leftarrow LocalSearch(S1)$;
  if $S1.cost < S.cost$ **then**
    $S \leftarrow S1, k \leftarrow 0$;
  end
end

Procedure 2: InitialSolution.

Output: The initial solution S
P /* P presents the set of selected median sites,N presents the set of all the sites */
$Range \leftarrow n/p$;
For each site i, calculate $DS[i]$, the sum of distances of its *Range* nearest customers;
for $k = 1$ **to** $k = p$ **do**
  $s \leftarrow$ the site with smallest *DS* value in N;
  if $Q[s]^1 \leq Cap[s]^2$ **then**
    /* $Q[s]$ gives the quantity delivered by site s to its customers, *Cap* gives the capacity value for each site */
    $P \cap \{s\}, N \setminus \{s\}$;
    Recalculate *DS* value for all the sites in N ;
  end
  else
    $DS[s] \leftarrow \infty$
  end
end

loaded from OR-library[3]. Set A contains 10 instances of size $n = 50, p = 5$ and set B contains 10 instances of size $n = 100, p = 10$. The parameters A_i, B_i, C_i are generated by *Matlab* with normal distributions $A_i \sim \mathcal{N}(500, 100), B_i \sim \mathcal{N}(0.04, 0.01), C_i \sim \mathcal{N}(10, 3)$ respectively. The distribution range R varies from 35 to 115. The number of intervals k for linearized model is 30.

Two models are coded on the General Algebraic

[3]http://people.brunel.ac.uk/∼mastjjb/jeb/orlib/pmedcapinfo.html

Procedure 3: LocalSearch(S).

Input: The initial Solution S
Output: The local minimum solution S'
$flag \leftarrow true$;
while $flag$ **do**
  $flag \leftarrow OneTimeLocalSearch(S)$;
end
$S' \leftarrow S$;
return S';

Procedure 4: OneTimeLocalSearch(S).

Input: The intial solution S
Output: The boolean value which siginifies if we could find a better solution.
$P \leftarrow S.p\_sites$, P is a set of p-median sites.;
for *each site* $p_{in} \in P$ *in a descending order by* $DN[p_{in}]$*, the total demand of its customers* **do**
  if $DN[p_{in}] < Cap[p_{in}]$ **then**
    for *each* $p_{out} \in P$ *in an ascending order by* $DN[p_{out}]$ *such that* $p_{out}! = p_{in} \cap DN[p_{out}] < DN[p_{in}]$ **do**
      Suppose C_{out} is a set of all customers delivered by the site p_{out};
      for *each customer* $c \in C_{out}$*, in a descending order by its demand* $D[c]$ **do**
        if $D[c] + DN[p_{in}] \leq Cap[p_{in}] \cap Dis(c, pin) \leq R$ **then**
          /* *Dis*:distance matrix,*R*:distribution range */
          if *Move*(c, p_{in}, p_{out}, S) **then**
            /* In the procedure of *Move*, we give the better solution to S. */
            return true;
          end
        end
      end
    end
  end
end
return false;

Modeling System (*GAMS*). The computational experiments are carried out on a INTEL I7 2GHZ computer with 4G RAM.

We solve our concave model (1-6) using a com-

Procedure 5: Move(c, p_{in}, p_{out}, S).

Input: c: the customer to move; p_{in}, p_{out}: the delivery base site; S: the original solution

Output: The boolean value which siginifes if we put the customer c in the delivery area of p_{in} and the changed solution S

$cost \leftarrow S.cost()$;
$\{C_{in}\} \leftarrow c \cup \{C_{in}\}$;
$\{C_{out}\} \leftarrow \{C_{out}\}$;
/* C_{out} is a set of customers delivered by site p_{out}, we move customer c out of the delivery space of p_{in} */
$S' \leftarrow$ One_Median_Problem (p_{in}, C_{in}, S);
/* Choose the site i from $\{p_{in}\} \cap C_{in}$ to deliver its customers $\{p_{in}\} \cap C_{in} \setminus i$ with a minimum distribution cost */
$S' \leftarrow$One_Median_Problem (p_{out}, C_{out}, S');
$cost' \leftarrow S'.cost()$;
if $cost' \leq cost$ **then**
    $S \leftarrow S'$;
    return true;
end
else
    return false;
end

mercial solver Discrete and Continuous Optimizer (*DICOPT*) which was designed to solve mixed integer non-linear programming problems (*MINLP*) with an outer-approximation algorithm (Kocis and Grossmann, 1989). This solver can guarantee the global optimum if the nonlinear function is convex. However, since our objective function is concave, it gives an upper bound for the problem. Inside the configuration of *DICOPT*, we choose *CONOPT* (Drud, 1996) as the NLP solver and *CPLEX* as the MIP solver.

Then, we solve the linearized model (14-29) which is a mixed integer linear programming using *CPLEX* solver.

Firstly, in our experiments, we test our models with different distribution range R to see the influence of R. When R is small, each facility site could only serves certain number of customers. If R is too small, we can't even find a feasible solution. After several trials on R, we choose to run our tests with R from 35 ensuring at least one feasible solution to 115 so that each facility site could serve every customers. Fig.1 shows the average values obtained by *DICOPT* and *CPLEX* for set A instances as the distribution range R increases. Fig.2 presents the average execution times of *DICOPT* and *CPLEX* in seconds for solving set A instances. Fig.3 shows the average values obtained by *DICOPT* and *CPLEX* for set B instances.

We can see from Fig.1 that as R increases, we have more possible choices for sites and customers, the value of objective function tends to decrease. As solvers do not always guarantee a global optimal solution, the curves do not decrease smoothly. There exists some small fluctuations. The values of *CPLEX* curve are obtained by solving the linearized model first with *CPELX* and then recalculating concave objective function. Comparing the curve of *DICOPT* with the curve of *CPLEX*, we observed that our linearized model performs better than concave model in getting global optimal solution using the existing commercial solvers. Fig.3 shows that our advantage in quality of solution when solving linearized model holds also for larger instances of set B.

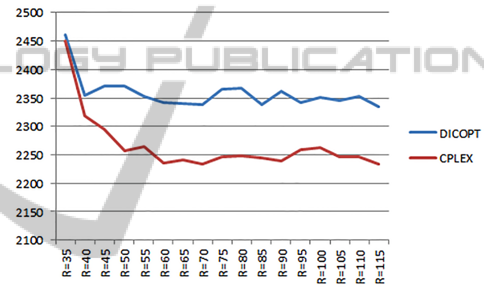

Figure 1: Average value of solutions of set A instances.

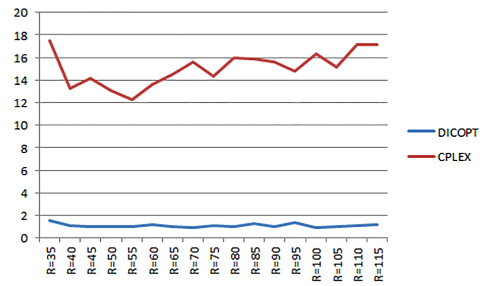

Figure 2: Average execution time of set A instances.

Secondly, we apply our approach of *VNS* to solve *CPMP* with large size instances. The number of sites is ranging from 100 to 1500. These large instances are generated randomly following normal distributions. *CPLEX* and *DICOPT* were very time-consuming. In this case, we only report the lower bounds by solving linearized model with *CPLEX* and continue our calculations with *VNS*. The latter could solve the problem with up to 1500 sites. The results are presented in the Tab.1. Columns $1, 2, 3$ give the number of sites,

Table 1: Results for the linearized model in the large data; the best solution found in 10 trials of VNS is reported.

			CPLEX		VNS		GAP
N	M	P	Value	time	Value	time	%
					K_{max}=P		
50	50	2	26873	132.30	26873	0.11	0.00
60	60	3	31969.2	446.96	31969.2	0.14	0.00
70	70	3	36249.4	289.46	36600.1	0.25	0.01
80	80	3	37362.4	248.90	37362.4	0.24	0.00
90	90	3	43059.8	3341.56	43060.3	0.44	0.00
100	100	3	49056.4	1121.96	49056.4	0.28	0.00
150	150	4	13950*	39.51	71413.5	0.18	#
200	200	6	19063.9*	67.75	97105.4	3.36	#
250	250	7	24981.6*	339.63	118861	10.27	#
300	300	8	30842.6*	495.57	137737	22.26	#
350	350	9	80002.1*	998.22	162297	50.92	#
400	400	11	47433.5*	1637.71	187413	141.40	#
450	450	12	49056.4*	1121.96	206303	217.15	#
					K_{max}=P/3		
500	500	13	57710.4*	1134.86	227456	195.77	#
600	600	16	83009.0*	1535.36	281224	592.29	#
700	700	18	102030.1*	1900.24	330264	1018.02	#
800	800	20	142003.5*	2503.31	367511	2096.62	#
900	900	23	184030.1*	2709.72	430663	1978.83	#
1000	1000	25	240248.2*	3501.29	463628	2796.42	#
1200	1200	30			537799	3891.14	#
1300	1300	32			562032	4029.12	#
1400	1400	33			591024	5129.20	#
1500	1500	34			677229	7428.21	#

* CPLEX failed to solve the integer problem, only lower bounds values are reported.

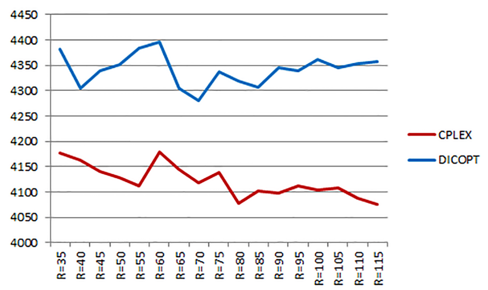

Figure 3: Average value of solutions of set B instances.

the number of customers and the median value respectively. Columns 4, 5 present the numerical results and the execution time in seconds for *CPLEX* and *VNS* respectively. Column "GAP" gives the relative gap of *VNS* to the optimal solution.

5 CONCLUSIONS

In this paper, we investigate a capacitated p-median model with concave cost. The cost is a concave function of the quantity handled by site i. We show that the problem can be approximated to a mixed linear programming problem. An efficient VNS for this model has been proposed.

We present computational results comparing the exact formulation and linearized formulation, using *DICOPT*, *CPLEX* respectively. Computational results show that our linearized method enables to reach optimal solution of our problem. When using VNS, we can solve instances with up to 1500 sites. Our VNS performances show that our adaptation of the algorithm to this difficult problem is very efficient as illustrated by the small gaps,i.e. less than 2%.

REFERENCES

Avella, P., Sassano, A., and Vasil'ev, I. (2007). Computational study of large-scale p-median problems. *Mathematical Programming*, 109(1):89–114.

Beltran, C., Tadonki, C., and Vial, J. P. (2006). Solving the p-median problem with a semi-lagrangian relaxation. *Computational Optimization and Applications*, 35(2):239–260.

Ceselli, A. and Righini, G. (2005). A branch-and-price algorithm for the capacitated p-median problem. *Networks*, 45(3):125–142.

Charikar, M. and Guha, S. (1999). Improved combinatorial algorithms for the facility location and k-median problems. In *Foundations of Computer Science, 1999. 40th Annual Symposium on*, pages 378–388. IEEE.

Drud, A. (1996). Conopt - a system for large scale nonlinear optimization. *REFERENCE MANUAL for CONOPT Subroutine Library, ARKI Consulting and Development A/S*.

Dupont, L. (2008). Branch and bound algorithm for a facility location problem with concave site dependent costs. *International Journal of Production Economics*, 112(1):245–254.

Fathali, J. and Kakhki, H. (2006). Solving the p-median problem with pos/neg weights by variable neighborhood search and some results for special cases. *European journal of operational research*, 170(2):440–462.

Fleszar, K. and Hindi, K. (2008). An effective vns for the capacitated p-median problem. *European Journal of Operational Research*, 191(3):612–622.

Ghoseiri, K. and Ghannadpour, S. (2007). Solving capacitated p-median problem using genetic algorithm. In *Industrial Engineering and Engineering Management, 2007 IEEE International Conference on*, pages 885–889. IEEE.

Hansen, P., Brimberg, J., Urošević, D., and Mladenović, N. (2009). Solving large p-median clustering problems by primal-dual variable neighborhood search. *Data mining and knowledge discovery*, 19(3):351–375.

Hansen, P. and Mladenović, N. (1997). Variable neighborhood search for the p-median. *Location Science*, 5(4):207–226.

Hansen, P., Mladenović, N., and Moreno Pérez, J. (2010). Variable neighbourhood search: methods and applications. *Annals of Operations Research*, 175(1):367–407.

Jain, K. and Vazirani, V. V. (2001). Approximation algorithms for metric facility location and k-median problems using the primal-dual schema and lagrangian relaxation. *Journal of the ACM (JACM)*, 48(2):274–296.

Kariv, O. and Hakimi, S. (1979). An algorithmic approach to network location problems. ii: The p-medians. *SIAM Journal on Applied Mathematics*, pages 539–560.

Kocis, G. and Grossmann, I. (1989). Computational experience with dicopt solving {MINLP} problems in process systems engineering. *Computers & Chemical Engineering*, 13(3):307 – 315.

Landa-Torres, I., Ser, J. D., Salcedo-Sanz, S., Gil-Lopez, S., Portilla-Figueras, J., and Alonso-Garrido, O. (2012). A comparative study of two hybrid grouping evolutionary techniques for the capacitated p-median problem. *Computers & Operations Research*, 39(9):2214 – 2222.

Nagy, G. and Salhi, S. (2007). Location-routing: Issues, models and methods. *European Journal of Operational Research*, 177(2):649–672.

Osman, I. and Ahmadi, S. (2006). Guided construction search metaheuristics for the capacitated p-median problem with single source constraint. *Journal of the Operational Research Society*, 58(1):100–114.

Osman, I. H. and Christofides, N. (1994). Capacitated clustering problems by hybrid simulated annealing and tabu search. *International Transactions in Operational Research*, 1(3):317–336.

ReVelle, C. and Swain, R. (1970). Central facilities location. *Geographical Analysis*, 2(1):30–42.

Stanimirović, Z. (2008). A genetic algorithm approach for the capacitated single allocation p-hub median problem. *Computing and Informatics*, 27.

Sun, J. and Gu, Y. (2002). A parametric approach for a nonlinear discrete location problem. *Journal of combinatorial optimization*, 6(2):119–132.

Verter, V. and Dasci, A. (2002). The plant location and flexible technology acquisition problem. *European Journal of Operational Research*, 136(2):366–382.

Verter, V. and Dincer, M. (1995). Facility location and capacity acquisition: an integrated approach. *Naval Research Logistics (NRL)*, 42(8):1141–1160.

Xu, X., Li, X., Li, X., and Lin, H. (2010). An improved scatter search algorithm for capacitated p-median problem. In *Computer Engineering and Technology (ICCET), 2010 2nd International Conference on*, volume 2, pages V2–316. IEEE.

Yaghini, M., Karimi, M., and Rahbar, M. (2013). A hybrid metaheuristic approach for the capacitated p-median problem. *Applied Soft Computing*, 13(9):3922 – 3930.